Universität Rostock

Fachbereich Elektrotechnik

Institut für Nachrichtentechnik und Informationselektronik

Studienarbeit

Vergleich von

H.323 und SIP

beim Aufbau von VoIP Netzen

Bearbeitet durch: Alexander Noack

Betreuer: Dr.-Ing. H.-D. Melzer

Dipl.-Ing. T. Kessler

Rostock, September 2003

1. Einführung 4
2. H.323-Protokollstapel 5
2.1 Protokolle in H.323 6
2.1.1 Audio Codec 6
2.1.2 Video Codec 6
2.1.3 H.225.0 Framing, Sequencing, Error Detection & Correction 7
2.1.4 H.225 Registration, Admission and Status (RAS) 7
2.1.5 H.225 Call Signalling (Q.931) 8
2.1.6 H.245 Control Signalling 9
2.1.7 H.450.x Supplementary Services 9
2.1.8 Weitere Protokolle 11
2.2 Komponenten von H.323 11
2.2.1 H.323-Terminal 11
2.2.2 H.323-Gatekeeper 12
2.2.3 H.323-Gateway 13
2.2.4 Multipoint Control Unit (MCU) 13
3. SIP – Session Initiation Protocol 13
3.1 SIP-Nachrichten 14
3.2 SIP Komponenten 15
3.2.1 User Agents 16
3.2.2 Proxy Server 16
3.2.3 Location Server 17
3.2.4 Redirect Server 17
3.2.5 Registrar Server 17
4. Praktischer Einsatz 18
4.1 H.323 Protocol Stack 18
4.1.1 Gateway Software 19
4.1.2 Gatekeeper Software 20
4.1.3 H.323 Client Software (Softphones) 20

4.2 SIP 21
4.2.1 SIP-Server 21
4.2.2 SIP-Clients 23
4.3 VoIP-Telefonanlage 24
4.4 Der Business Fall – H.323 im Geschäftseinsatz 26
5. Vergleich H.323 – SIP 28
5.1 Verbreitung, Akzeptanz und Verfügbarkeit 28
5.2 Leistungsfähigkeit und Funktionsumfang 28
5.3 Migrationsfähigkeit und Anpassbarkeit 30
5.4 Skalierbarkeit 30
5.5 Sicherheit 31
5.6 Quality of Service Unterstützung 32
5.7 Management und Accounting 34
5.8 IPv6 Fähigkeit 35
5.9 Zusammenfassung 36
6. Schlussfolgerungen 37
7. Abkürzungsverzeichnis 39
8. Abbildungsverzeichnis 42
9. Literaturverzeichnis 43
10. Anhang 45
10.1 Vergleich: Freie Gateways 45
10.2 Vergleich: Freie Gatekeeper 47
10.3 Konfiguration: isdn2h323 Gateway 49
10.4 Konfiguration: GNUGatekeeper 50
10.5 Konfiguration: ASTERISK 51

1. Einführung

Nach Erfindung des Telefons fokussierte sich die Entwicklung in den folgenden 100 Jahren vorrangig auf eine Verbesserung des verbindungsbasierten Telefonsystems. Neue Verfahren wurden entwickelt, um mehrere, gleichzeitige Telefonate über die selbe Leitung zu übertragen. So entstand z.B. das N-ISDN, aber auch die PDH- und SDH-Infrastrukturen auf Verteilungsebene – Grundlagen für das B-ISDN (ATM).

Parallel dazu entwickelte sich eine weitere Art von Kommunikationsnetzen, die zunächst eine untergeordnete Rolle spielte. Die Arbeiten Zuses und v. Neumanns brachten bereits zu Beginn des 20. Jahrhunderts die ersten Rechenmaschinen hervor, die sich langsam zu Computern entwickelten. Computer, die zumeist ein eigenes Rechenzentrum mit speziellem Stromanschluss und Klimaanlagen benötigten. Die Bedienung erfolgte zunächst über Terminals, die über eine einfache serielle Schnittstelle miteinander verbunden waren. Die Kommunikation der Rechenzentren untereinander erfolgte über das öffentliche Telefonnetz. Erst als die Intelligenz zunehmend in die Endgeräte verlagert wurde – hin zum Personal Computer – entstand die Notwendigkeit von Inter-PC Kommunikationsnetzen. Der Bedarf, verschiedenste Endgerätetechnik miteinander zu verbinden, führte Ende der 80er Jahre zu einer Bereinigung des sich bildenden Protokolldschungels. Es etablierten sich schließlich nur die Protokolle mit der größten Verbreitung, da sie eine gemeinsame Kommunikationsbasis boten. Dabei mag es sich auch um politische Entscheidungen der führenden Hardware- und/oder Softwarefirmen gehandelt haben, es bleibt jedoch festzuhalten, dass bereits Ende der 90er Jahre dem Internetprotokoll (TCP/IP) die größte Bedeutung in der Rechnerkommunikation zufiel.

Seit der Entstehung der Computernetze bestand für Unternehmen die Notwendigkeit, neben dem Telefonnetz auch ein Datennetz zu pflegen und zu unterhalten. Um die Kosten dafür zu minimieren, gab und gibt es Bestrebungen, die jeweils anderen Dienste in *ein* Netz zu integrieren. Auf Seiten des PSTN führen ISDN und ATM diese Bestrebungen an, während die Konvergenzanstrengungen auf der Netzwerkseite sich zunehmend auf die Anpassung in Software konzentriert – namentlich dem TCP/IP-Stack. Dies führt zu der uns heute bekannten Situation, dass die Weitverkehrskommunikation bis OSI-Layer 2 auf bestehenden, gut ausgebauten Telekommunikationsinfrastrukturen abläuft, wohingegen darüberliegende Schichten sich des IP-Protokolls bedienen.

Aktuell gibt es zwei Ansätze Sprachverbindungen über IP-Netzwerke zu signalisieren. Das H.323-Protokoll sowie SIP, das Session Initiation Protocol. Ziel dieser Arbeit soll es sein, beide Protokolle unter verschiedenen Gesichtspunkten zu vergleichen (siehe Abschnitt 5.) und deren Praxistauglichkeit anhand freier Softwareprodukte zu untersuchen (siehe Abschnitt 4.1 und 4.2) um im Ergebnis über die Theorie hinausgehende Aussagen über die weitere Entwicklung von Voice over IP treffen zu können (siehe Abschnitt 6.). In den Abschnitten 2.) und 3.) soll zunächst die Funktionsweise und der Aufbau von H.323 bzw. SIP erläutert werden.

2. H.323-Protokollstapel

Neben proprietären Ansätzen, multimediale Daten über TCP/IP zu übertragen, wurde H.323 bereits 1996 als Empfehlung von der ITU-T spezifiziert. Es handelt sich hierbei um einen sogenannten „Umbrella“ Standard, d.h. H.323 ist eigentlich nur der Überbegriff für eine Reihe von Protokollstandards, die der Audio- und Videoübertragung über IP-Netze dienen. Im Gegensatz zu seinen Vorgängern wurde die H.323-Empfehlung speziell für LANs ohne das Vorhandensein von QoS (Quality of Service) Garantien entworfen.

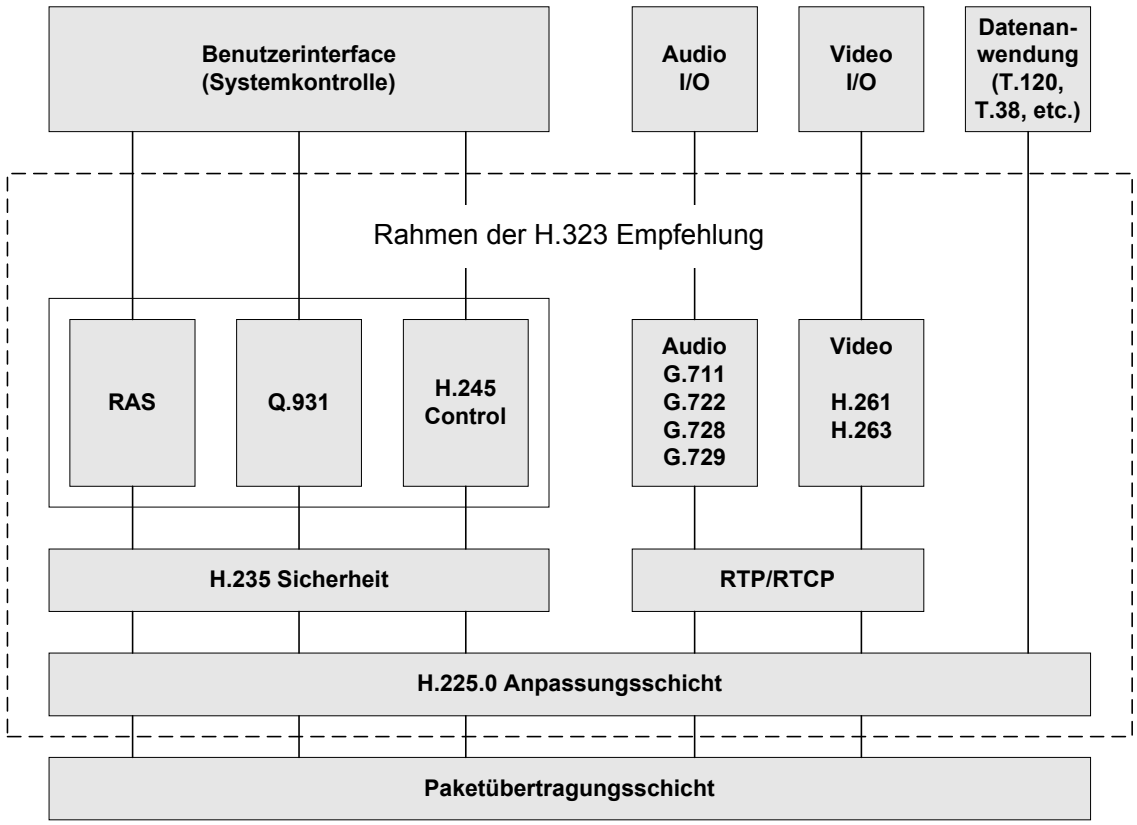

Abbildung 2-1 Architektur des H.323 Protocol Stack

2.1 Protokolle in H.323

Da H.323 unabhängig von dem darunterliegenden paketorientierten Netzwerk arbeitet, spezifiziert der Standard eigene Protokolle zur Datenübertragung in höheren Schichten.

2.1.1 Audio Codec

Ein Audio Codec kodiert ein aufgezeichnetes Tonsignal (z.B. vom Mikrofon) zum Zweck der Übertragung. Die empfangende Station gibt das dekodierte Signal z.B. über die Lautsprecher wieder.
Da die Audiofähigkeit die Minimalanforderung an einen H.323-Dienst ist, muss jeder Endpunkt mindestens einen Audio Codec unterstützen, z.B. den ITU-T G.711-Codec (Audiodaten mit 64 kbps). Mögliche Codecs sind:

G.711	*PCM Audio mit 56/64 kbps (A-Law, µ-Law)*
G.722	*Audio bis 7 kHz mit 48/56/64 kbps*
G.723.1	*Sprach Codec mit 5.3/6.3 kbps (Teil von H.324)*
G.728	*Sprach Codec mit 16 kbps (low delay)*
G.729	*Sprach Codec mit 8/13 kbps (low delay)*
GSM	*Sprach Codec mit 13 kbps (GSM Mobilfunk)*
LPC10e	*Sprach Codec mit 2,4 kbps (linear predictive)*

2.1.2 Video Codec

Ein Video Codec kodiert ein aufgezeichnetes Videosignal (z.B. von einer Kamera) zum Zweck der Übertragung. Die empfangende Station gibt das dekodierte Signal z.B. über den PC-Monitor wieder.
Die Videofunktionalität ist als optionaler Bestandteil im H.323 spezifiziert. Stellt eine Station Videofunktionalitäten zur Verfügung, so muss sie Videodaten nach dem ITU-T Standard H.261 kodieren können. Der abwärtskompatible H.263 verbessert die Bildqualität, ist jedoch nur optional.

2.1.3 H.225.0 Framing, Sequencing, Error Detection & Correction

Die H.225.0-Schicht definiert unter anderem die Anpassung der höheren Schichten auf die Transportschicht. H.323 bedient sich für die Echtzeitübertragung von Audio und Video dem Realtime Transport Protocol (RTP). Obwohl RTP über beliebige Transportschichtprotokolle übertragen werden könnte (z.B. Novell IPX), findet es meist zusammen mit dem User Datagram Protocol (UDP) Anwendung.
Zuverlässige Übertragung realisiert das Realtime Transport Control Protocol (RTCP). In der Regel wird es per TCP übertragen (auch SPX ist denkbar). Eingebaute Kontrollmechanismen erlauben es, Aussagen über die Qualität der Übertragung zu treffen. Mittels eines kanonischen Namens, der einen RTP-Strom kennzeichnet, kann RTCP die Synchronisierung von Audio und Video auf Transportebene sicherstellen

2.1.4 H.225 Registration, Admission and Status (RAS)

Befindet sich in einer H.323-Zone ein Gatekeeper, so erweitert dieser die Zone um diverse Verwaltungsfunktionen (siehe Abschnitt 2.2.2). H.225 spezifiziert einen zusätzlichen Kommunikationskanal, über den Nachrichten zur Gatekeeper-Erkennung und Endpunktregistrierung übertragen werden (siehe auch Abb. 2-2).
Diese Nachrichten umfassen [6]:

Gatekeeper Request (GRQ)	*Terminal sucht Gatekeeper*
Gatekeeper Confirm (GCF)	*Gatekeeper Antwort auf GRQ*
Register Request (RRQ)	*Terminal registriert sich bei Gatekeeper*
Register Confirm (RCF)	*Gatekeeper bestätigt RRQ*
Register Reject (RRJ)	*Gatekeeper weist RRQ ab*
Admission Request (ARQ)	*registriertes Terminal leitet Anruf ein*
Admission Confirm (ACF)	*Gatekeeper Antwort auf ARQ*
Admission Reject (ARJ)	*Gatekeeper weist ARQ ab*

Bandwidth Request (BRQ)	*Terminal fordert eine bestimmte Bandbreite an*
Bandwidth Confirm (BCF)	*Gatekeeper weist Bandbreite zu*
Bandwidth Reject (BRJ)	*Gatekeeper weist Bandbreitenanforderung ab*
Disengage Request (DRQ)	*Terminal informiert GK über Verbindungsende*
Disengage Confirm (DCF)	*Gatekeeper Antwort auf DRQ*
Disengage Reject (DRJ)	*Gatekeeper weist DRQ ab*

Alternativ existieren auch:

Unregister Request (URQ)	*Terminal meldet sich vom Gatekeeper ab*
Unregister Confirm (UCF)	*Gatekeeper Antwort auf URQ*
Unregister Reject (URJ)	*Gatekeeper weist URQ ab*

2.1.5 H.225 Call Signalling (Q.931)

Für die Ende-zu-Ende Signalisierung wird in H.225 ein weiterer Kanal definiert. Dabei bedient man sich des bereits für ISDN (Integrated Services Digital Network) spezifizierten Signalisierungsprotokolls Q.931.
Folgende Nachrichten sind definiert (analog zum ISDN):

Setup	*Anruf wird eingeleitet*
Call Proceeding	*Nachricht wurde empfangen*
Alerting	*Endpunkt "klingelt"*
Connect	*Verbindung wurde aufgebaut*
Release Complete	*Verbindungsabbruch wird eingeleitet*
Close	*Verbindung abgebaut*

(siehe auch Abb. 2-2).

2.1.6 H.245 Control Signalling

Zusätzlich stellt H.323 einen Kontrollkanal bereit, über den die Endpunkte nach erfolgtem Verbindungsaufbau ihre Funktionalitäten austauschen (Terminal Capabilities Exchange, TCS). Diese Notwendigkeit ergibt sich z.B. bei der Aushandlung eines gemeinsamen Audio- oder Video Codecs.
Weiterhin besteht zwischen zwei H.323-Terminals eine Master/Slave Beziehung, um Konferenzen zu verwalten. Entsprechende Mechanismen stellt H.245 zur Verfügung. Nachdem alle Parameter verhandelt wurden, fällt diesem Kontrollkanal die Aufgabe zu, die logischen Kanäle zu öffnen (Open Logical Channel, OLC).

2.1.7 H.450.x Supplementary Services

Um weitere Dienstmerkmale – wie aus dem PSTN bekannt – zu unterstützen, definiert H.450.1 einen Rahmen für zusätzliche Dienste. Somit ist H.323 auch für zukünftige Funktionen erweiterbar. Bereits bestehende Funktionen umfassen:

H.450.2	*Call Transfer (Anruf Übergabe)*
H.450.3	*Call Diversion (Anruf Umleitung)*
H.450.4	*Call Hold (Anruf Halten)*
H.450.5	*Call Park & Pickup (Anruf Parken mit Umstecken)*
H.450.6	*Call Waiting (Signal. eines gehaltenen Anruf)*
H.450.7	*Message Waiting Indication (Nachrichtenlampe)*
H.450.8	*Name Identification (alternative CallerID Anzeige)*
H.450.9	*Call Completion (Vervollständigung der Rufnr.)*
H.450.10	*Call Offer (Signalisierung eines Anrufes innerhalb einer Anrufergruppe)*
H.450.11	*Call Intrusion (Übernahme eines Anrufes innerhalb einer Anrufergruppe)*

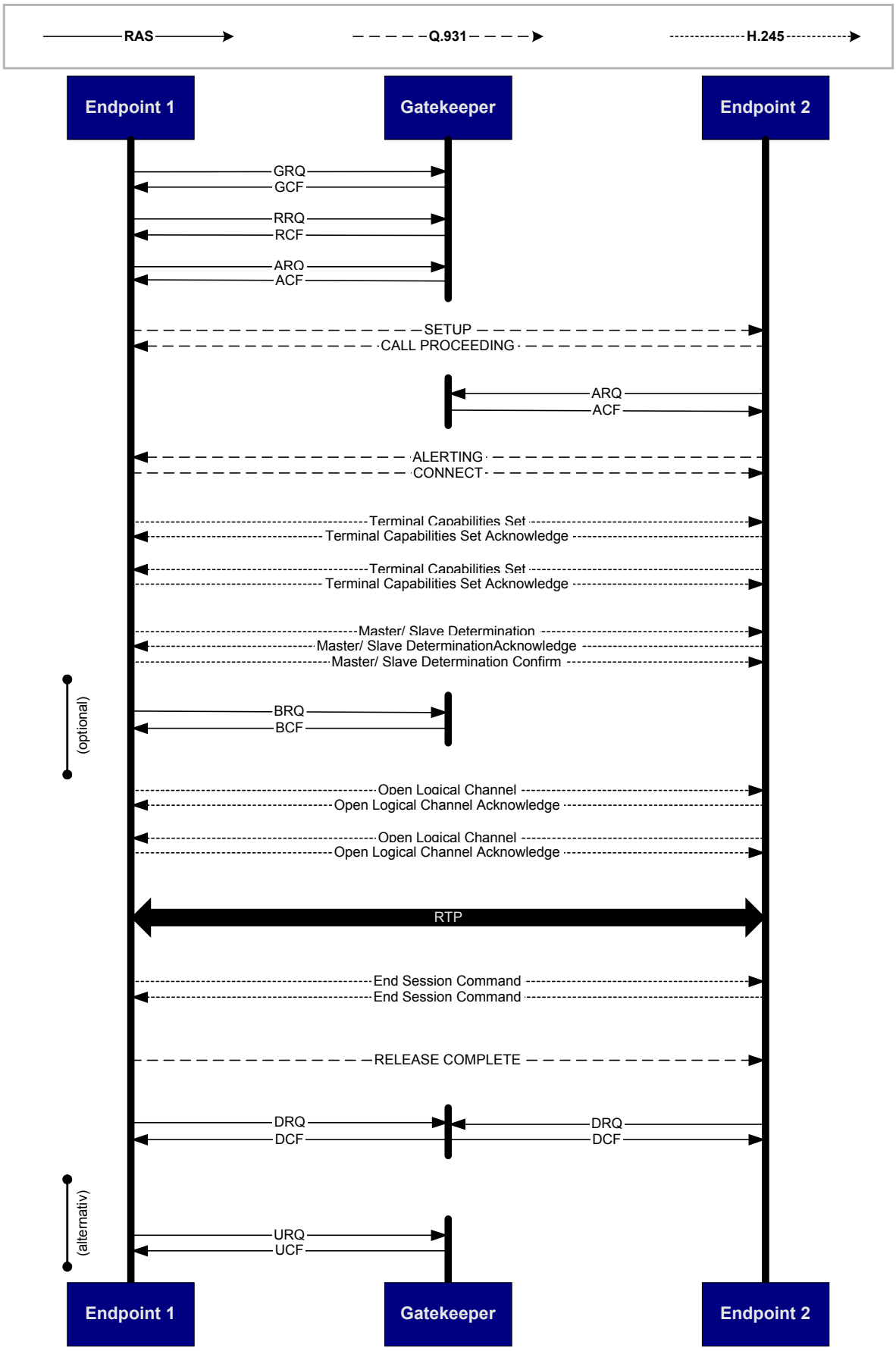

Abbildung 2-2 Signalisierungsablauf in H.323

2.1.8 Weitere Protokolle

Außerdem spezifiziert H.323 eine Reihe zusätzlicher Protokolle z.B. T.120 zum Austausch von binären Daten in einer Konferenz (Whiteboard, Chat, Programme), T.38 zum Faxversand über IP, H.235 als Sicherheitslayer in H.323 sowie die H.500-Gruppe zur Unterstützung mobiler Endgeräte.

2.2 Komponenten von H.323

Im Standard sind sowohl H.323-Endpunkte als auch verschiedene H.323-Instanzen definiert. Die Endpunkte sind anrufbar bzw. können angerufen werden; während die übrigen Instanzen für ihre spezifischen Aufgaben adressiert werden können. Kontrollnachrichten und Kontrollprozeduren werden in H.323 definiert, um die Kommunikation zwischen den Komponenten zu gewährleisten.

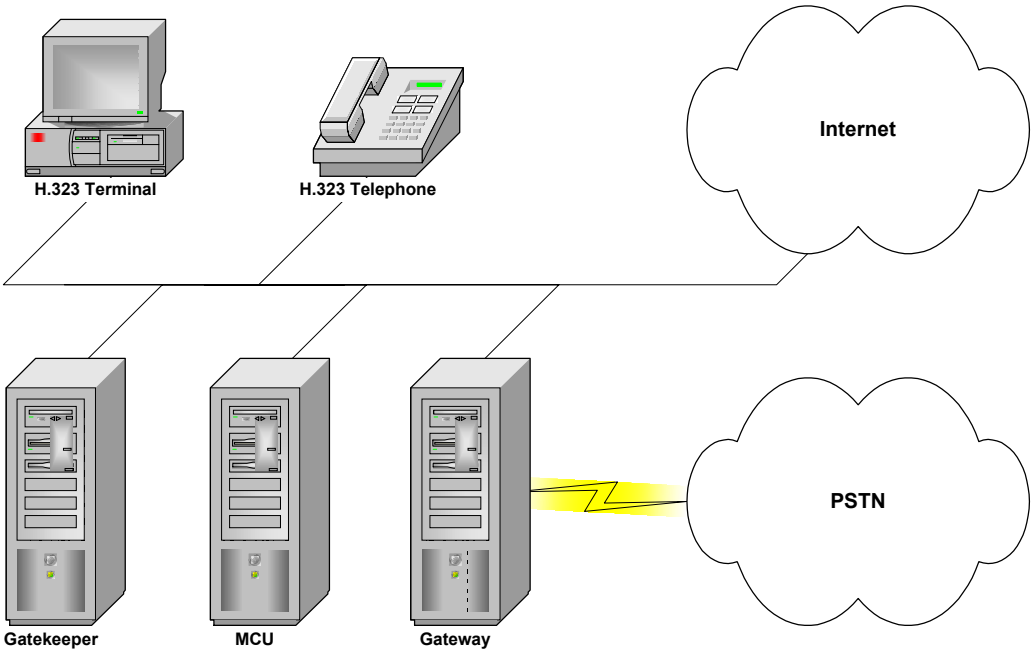

Abbildung 2-3 Komponenten eines H.323 Netzes

2.2.1 H.323-Terminal

Ein H.323-Terminal bildet einen Endpunkt in einer H.323-Zone. Es unterstützt die Echtzeitduplexkommunikation mit anderen Endpunkten. Dabei kann die Kommunikation aus Kontrollnachrichten, Audio, Video und/ oder reinen Datenströmen zwischen zwei Terminals bestehen. Die Kombination der zu übertragenden Medien ist dabei – bis auf die Kontrollnachrichten – dem Anwender überlassen (siehe auch Abschnitte 2.1.1 und 2.1.2).

2.2.2 H.323-Gatekeeper

Ein H.323-Gatekeeper übernimmt die Verwaltungsaufgaben in einer H.323-Zone. Dazu zählen Adressumsetzung, Autorisierung und Authentifizierung von Endpunkten, Bandbreitenkontrolle sowie Kostenerfassung und -abrechnung. Außerdem kann ein Gatekeeper Routing Funktionalitäten bereitstellen. Folgende Aufgaben muss ein Gatekeeper erfüllen:

Adressumsetzung:	*Innerhalb einer H.323-Zone können Aliasnamen für die einzelnen Endpunkte verwendet werden. Anrufe über einen Gateway hinweg (z.B. ins PSTN) benutzen E.164-Adressen (kanonische Form, ISDN-Nummern). Ein Gatekeeper übersetzt Aliasnamen oder E.164-Adressen in Netzwerkadressen (z.B. IP-Adressen für ein darunterliegendes TCP/IP-Netzwerk). Anhand dieser Netzwerkadresse kann ein Endpunkt erreicht werden.*
Zugangskontrolle:	*Mittels H.225-RAS verhandelt der Gatekeeper den Beitritt eines Endpunkts zum H.323 Netzwerk.*
Bandbreitenkontrolle:	*H.225 spezifiziert zusätzlich RAS Nachrichten zur Bandbreitenregulierung (BRQ, BCF, BRJ). So kann ein Gatekeeper weitere Verbindungen abweisen, sobald eine bestimmte Anzahl von Verbindungen besteht, um das Netz nicht zu überlasten.*

Optionale Funktionen eines Gatekeepers sind:

Call Control Signaling:	*Ein Gatekeeper kann die H.225-Nachrichten für die Endpunkte routen. Das erhöht die Kontrolle über die Anrufe und ermöglicht z.B. eine gleichmäßige Lastverteilung auf die Gateways.* *Alternativ kann ein Gatekeeper den Terminals auch den direkten Austausch der H.225-Nachrichten untereinander erlauben.*

Call Authorization:	*Abhängig von Zugangskontrolllisten oder tageszeitabhängigen Filtern kann ein Gateway Anrufe erlauben oder verbieten.*
Call Management:	*Werden alle Anrufe über einen Gatekeeper eingeleitet, so kann dieser Protokoll führen. Dies ist wichtig für das Bandbreitenmanagement, für erweiterte Dienstmerkmale aber auch für die Abrechnung der Anrufleistungen.*

2.2.3 H.323-Gateway

Ein H.323-Gateway stellt die Verbindung zwischen einem H.323-Netzwerk und anderen Switched Circuit Networks (SCN), z.B. dem öffentlichen Telefonnetz (PSTN) her. Es realisiert dazu eine Umsetzung der beteiligten Kommunikationsprotokolle für den Verbindungsaufbau und -abbau. Gegebenenfalls beherrscht ein Gateway auch die Transkodierung der beteiligten Medienströme. Aus diesem Grund muss ein Gateway ähnlich wie ein Terminal die Funktionen der H.225-Schicht implementieren, sowie die Möglichkeiten von H.245 nutzen, um einen unterstützten Datenstrom auszuwählen. Nach dem Einschalten registriert sich ein Gateway mittels RAS-Nachrichten beim Gatekeeper.

2.2.4 Multipoint Control Unit (MCU)

Multipoint Control Units ermöglichen eine Konferenzschaltung von drei oder mehr Terminals. Alle Konferenzteilnehmer verbinden sich mit der MCU, die die beteiligten Ressourcen verwaltet und zwischen den Terminals den zu verwendenden Audio/Video Datenstrom verhandelt. Der Datenstrom selbst kann ebenfalls von der MCU abgewickelt werden.

3. SIP – Session Initiation Protocol

Im März 1999 wurde mit dem RFC 2543 die erste SIP-Spezifikation publiziert (inzwischen abgelöst durch RFC 3261). SIP steht für Session Initiation Protocol, wobei der Name bereits auf das Einsatzgebiet hinweist. SIP dient dem Aufbau und Management von Multimediaverbindungen aber auch dem Auffinden von Benutzern.

3.1 SIP-Nachrichten

Die Kommunikation zwischen SIP-Geräten erfolgt über Signalisierungsnachrichten. Diese erfüllen unterschiedliche Zwecke:

Benutzerregistrierung:	*Der Registrierungsdienst stellt eine Beziehung zwischen einem Benutzer und dessen Standort (im Netzwerk) sowie dem angeforderten Dienst her. Dazu definiert SIP eine „REGISTER“ Nachricht.*
Beitrittsaufforderung:	*Die Beitrittsaufforderung besteht aus einer „INVITE“ Nachricht an das angerufene Gerät. Interne Adressumsetzungsmechansimen erlauben dabei die Verwendung von vielen verschiedenen Formaten, z.B. E.164-Telefonnummern, IP-Adressen oder Uniform Resource Identifiers (URI).*
Verbindungsparameter:	*Die Parameter einer Verbindung werden über das Session Description Protocol (SDP) ausgehandelt, nachdem SIP den Kommunikationspartner lokalisiert hat. Der SDP-Payload kann dazu z.B. als Anhang im Textkörper einer „INVITE“ Nachricht transportiert werden. So können, bevor eine Verbindung erstellt wird, die Fähigkeiten der Kommunikationspartner abgeglichen werden.*
Verbindungsaufbau:	*Wurde eine „INVITE“ Nachricht akzeptiert („ACK“ Nachricht), können auch die Medienströme verbunden werden. Für Multimediadaten wird RTP als Übertragungsprotokoll genutzt. Diese RTP-Ströme können unabhängig vom Weg der SIP-Nachrichten übertragen werden.*

Verbindungsabbau:	*Der auflegende Endpunkt sendet zum ordnungsgemäßen Verbindungsabbau eine „BYE“ Nachricht.*

Es existieren weitere SIP-Nachrichten, die hier nicht genannt werden.
Zusätzlich sind eine Reihe von SIP-Rückmeldungen definiert, die sich in ihrer Form an andere IETF-Protokolle anlehnen [6]:

1xx Responses:	*Informational Responses* *Example: 180 Ringing*
2xx Responses:	*Successful Responses* *Example: 200 OK*
3xx Responses:	*Redirection Responses* *Example: 302 Moved Temporarily*
4xx Responses:	*Request Failure Responses* *Example: 404 Not Found*
5xx Responses:	*Server Failure Responses* *Example: 503 Service Unavailable*
6xx Responses:	*Global Failure Responses* *Example: 600 Busy Everywhere*

3.2 SIP Komponenten

Die SIP-Architektur besteht aus den SIP-User Agents, die die Schnittstelle zum Benutzer bilden, also IP-Telefone oder Softphones sein können und den SIP-Servern, die das Anrufmanagement- und -kontrollsystem bilden. In der Praxis werden verschiedene Serverkomponenten in der SIP-Serversoftware integriert, dabei wird nicht mehr nach den einzelnen SIP-Servern unterschieden, wie sie in diversen RFCs definiert sind. Diese sind jedoch hilfreich für das Verständnis des Konzepts, welches hinter SIP steht.

3.2.1 User Agents

Ein SIP-User Agent besteht genau genommen aus einem User Agent Client (UAC) und einem User Agent Server (UAS). Generell wird aber nicht zwischen der Client- und der Server-Komponente unterschieden, so dass man allgemein von User Agents (UAs) spricht.
Diese können in Hardware als Telefongeräte oder in Software als Softwaretelefone realisiert sein. Sie bilden die Endpunkte in einem SIP-System.

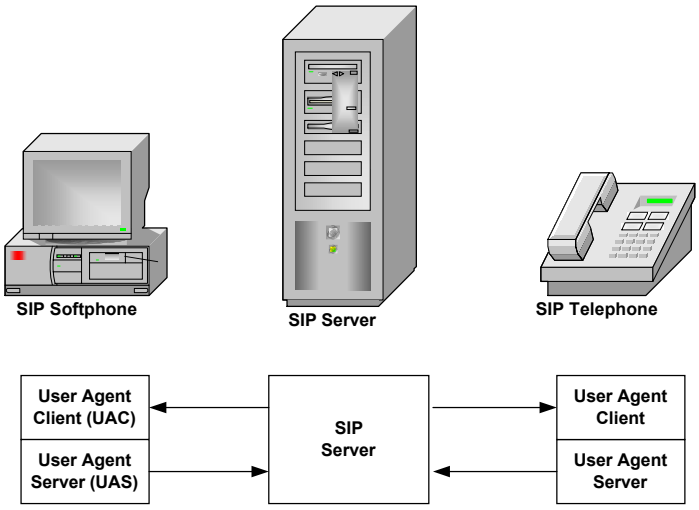

Abbildung 3-1 SIP User Agents (Client/Server Komponente)

3.2.2 Proxy Server

Der Proxy Server vermittelt zwischen Anrufer und Angerufenem. Er agiert dazu sowohl als Client als auch als Server. Eingehende Anfragen können intern verarbeitet oder stellvertretend an andere Server weitergeleitet werden. Eine der Hauptaufgaben ist die Weiterleitung von Anfragen an den Empfänger. Dabei verläuft die Lokalisierung des Kommunikationspartners transparent für den Anrufer. Die Adresse, mit der letztendlich eine Kommunikation zustande kommt und die eventuell etwas über den Aufenthaltsort bekannt geben könnte, bleibt verborgen. Unterstützt ein Proxy Server die Aufspaltung von Anforderungen, so kann er mehrere Teilnehmer zugleich rufen, bis einer der Teilnehmer das Gespräch annimmt.
Des weiteren kann ein Proxy als Zugangskontrolle in einem SIP-Netzwerk eingesetzt werden. Üblicherweise wird er dazu mit einem Registrar Server kombiniert.

3.2.3 Location Server

Ein Proxy oder Redirect Server kann einen Location Server kontaktieren, um Informationen über den möglichen Aufenthaltsort des Angerufenen zu erhalten.

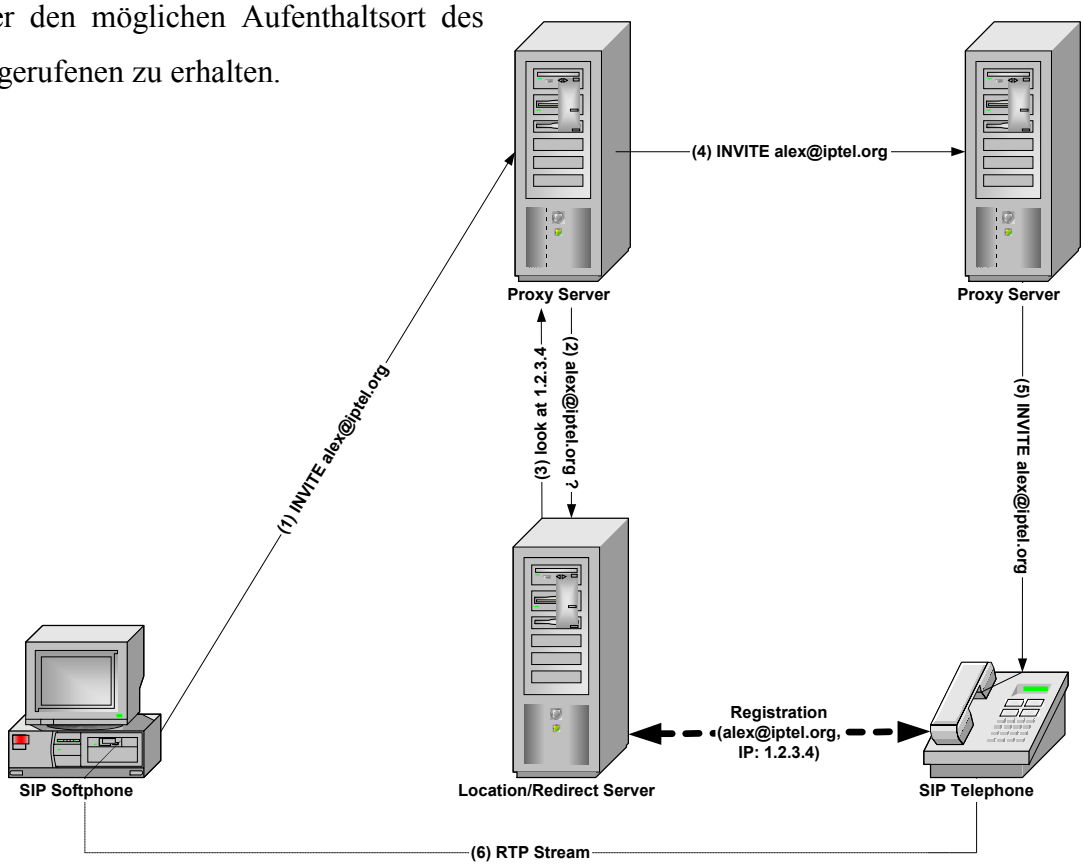

Abbildung 3-2 SIP Verbindungsaufbau

3.2.4 Redirect Server

Im Gegensatz zu einem Proxy Server kann ein Redirect Server keine Anrufe annehmen oder selbst initiieren. Ihm obliegt es lediglich einkommende SIP Anforderungen in eine oder mehrere Zieladressen aufzulösen (ggf. auch keine), die er dem Client (UAC) in einer SIP-Response mitteilt.

3.2.5 Registrar Server

Der Registrar Server empfängt die „REGISTER“ Nachrichten eines Clients. Er ist daher Teil der Authentifizierungsmechanismen in SIP. Weitaus wichtiger ist jedoch seine Aufgabe zur Lokalisierungsunterstützung. Da er während der Registrierung die Client SIP-URL mitsamt der Client IP-Adresse erhält kann er diese Informationen später bei eintreffenden „INVITE“ Nachrichten den anderen SIP-Servern zur Verfügung stellen (Proxy, Redirect), die die Nachricht dann entsprechend weiterleiten können.

4. Praktischer Einsatz

Um H.323 und SIP auch im praktischen Umfeld miteinander vergleichen zu können, war es nötig aus den existierenden Implementierungen die jeweils geeigneten herauszustellen. Dabei werden die Server Komponenten (Abschnitte 4.1.1 und 4.1.2 bzw. 4.2.1) getrennt von den Klienten (Abschnitte 4.1.3 bzw. 4.2.2) betrachtet.
Abschnitt 4.3 zeigt eine mögliche Lösung einer Implementierung, die sowohl H.323 als auch SIP unterstützt. Schließlich soll Abschnitt 4.4 verdeutlichen, dass für den Einsatz in geschäftskritischen Bereichen durchaus andere Faktoren berücksichtigt werden müssen.

4.1 H.323 Protocol Stack

Seit Ende 1999 ist der Protocol Stack „openh323“ von Equivalence Pty Ltd. als freie Implementierung des H.323 Protokolls verfügbar. Der Stack steht unter der Mozilla Public License (MPL) und steht somit auch für die Integration in kommerzielle Anwendungen kostenlos zur Verfügung. Ziel des Projekts ist es, die Bemühungen, einen H.323-Stack zu programmieren, zu bündeln, so dass sich nicht jeder Programmierer erneut Gedanken machen muss, wie die Bits und Bytes kodiert werden müssen, sondern dass sich auf das Entwickeln von Anwendungen konzentriert werden kann [15].
Neben dem eigentlichen Protokollstapel programmiert das openh323-Entwicklungsteam auch Beispielapplikationen. So gibt es neben der Telefonanwendung (OpenPhone, bzw. OhPhone als Kommandozeilenanwendung) auch einen Anrufbeantworter (OpenAM) und eine Sprachmenüsteuerung (OpenIVR). Die typischen Komponenten in einem H.323-Netz stellt das Entwicklerteam ebenfalls bereit: einen Gatekeeper (OpenGK), einen Gateway (PSTNGw) und einen Konferenzkontroller (OpenMCU).

Ob seiner Komplexität stellte der openh323-Stack im praktischen Einsatz das größte Problem dar, da sämtliche H.323-Software auf ihm aufbaut. Durch die Verwendung einer Abstraktionsschicht (PWLib, entwickelt von Equivalence), kann der generierte Code sowohl unter Linux als auch unter Windows Betriebssystemen genutzt werden. Was auf der einen Seite ohne Zweifel ein großer Vorteil (Plattformunabhängigkeit, kürzere Entwicklungszeit, etc.) sein kann, erweist sich im praktischen Einsatz als problematisch. Die Benutzung der Plattformbibliothek vergrößert den übersetzten Code, da viele unnötige Aufrufe einfach mitübersetzt werden. Hinzu kommt die Tatsache, dass das gesamte Projekt in C++

programmiert wurde – dies ermöglicht wiederverwendbaren Code, aber bringt auch umfangreicheren Binärcode hervor, da C++ Compiler nicht so effizient arbeiten wie ihre C-Pendants. Wie in vielen Open Source Projekten ist auch „openh323“ in ständiger Weiterentwicklung begriffen. Das erschwert den Einsatz des Stacks mit Software anderer Hersteller, da diese in der Regel nach einem Update des Stacks neu kompiliert werden muss. Ändern sich sogar Schnittstellen in den Bibliotheken des Stacks, muss die Software anderer Hersteller zunächst angepasst werden, was sich für nicht-C/C++ Programmierer ausschließt, so dass nur das (mitunter lange) Warten auf den angepassten Code verbleibt. Schließlich ist die Installation der zur Entwicklung nötigen Bibliotheken (insbesondere „include“ Dateien) in einer Multiuser-Umgebung (z.B. unter Linux LSB-konform) mit einem erheblichen Aufwand verbunden, da die Entwickler dafür keine Mechanismen vorsehen.
Sind die Installationshürden genommen, stellt sich der openh323-Stack jedoch als ausgereifte Bibliothek dar, die eine Vielzahl der bisher spezifizierten Funktionen des H.323 bereitstellt.

4.1.1 Gateway Software

Die bereits angesprochene Software „PSTNGw” (Abschnitt 4.1) eignet sich im europäischen Raum nicht als Gateway in das PSTN, da die dafür notwenigen analogen Einsteckkarten („Quicknet“) in europäischen Telefonnetzen nicht unterstützt werden. Das weitverbreitete ISDN ist hierzulande maßgebend, so dass eine Gateway Software mit ISDN-Funktionalität erforderlich ist. Für einen Vergleich siehe Anhang 10.1 .
Beide untersuchten Gateways stellen die erforderlichen Funktionen jeweils mit und ohne zusätzlichen Gatekeeper bereit. Die Verzögerung der Sprachdaten ist jedoch ein großer Kritikpunkt, der größtenteils auf das Modem-Interface von „ISDN4Linux“ zurückzuführen ist. Die Verwendung (teurer) aktiver ISDN-Hardware mit eigenem DSP kann hier Abhilfe schaffen. Eine Vereinfachung der Schnittstellen wird, mit Einführung der CAPI-Schnittstelle für HiSax-basierte passive Karten im Kernel v2.6, ebenfalls eine Verbesserung der Sprachverzögerung zur Folge haben.
Gelingt es, die Sprachverzögerung zu vermindern und die rechenintensiven Algorithmen zur Echo Unterdrückung konsequent einzusetzen (wie im isdn2h323-Gateway), so lassen sich Gateway Dienste mit Telefonnetzqualitäten realisieren. Insgesamt spricht der Fakt, dass das isdn2h323-Gateway weitergepflegt wird (in aktuellen Ausgaben von SuSE-Linux an die entsprechenden openh323-Versionen angepasst) neben der Echo Unterdrückung für sich. Es kam daher im Test vorrangig zum Einsatz. Konfigurationen finden sich in Anhang 10.3 .

4.1.2 Gatekeeper Software

Wie schon unter Abschnitt 2.2 herausgestellt wude, kommt dem Gatekeeper in einem H.323-Netzwerk die tragende Rolle zu. Seine Aufgabe ist neben der Anrufverwaltung auch die Adressumsetzung. Weiterhin regelt er den Zugang zum Netzwerk und überwacht die zur Verfügung stehende Bandbreite. Neben dem erwähnten Gatekeeper von „openh323.org“ (OpenGK), der nur grundlegende Funktionen (Registration, Address Translation) beherrscht, gibt es drei weitere Entwicklungen, die zusätzliche Funktionen mitbringen (siehe Anhang 10.2).
Grundlegende Funktionen werden von allen untersuchten freien Gatekeepern unterstützt. Allerdings schränkt die Verfügbarkeit der entsprechenden openh323-Bibliotheken die Auswahl stark ein, so dass eigentlich nur der GNUGatekeeper für einen stabilen Einsatz in Frage kommt. So gibt es z.B. Installationen bei der Telefonica Deutschland, und über 150 GNUGatekeeper sind bei einem Community Projekt in Indonesien im Einsatz [16]. Im Verlauf der praktischen Tests wurde maßgeblich der GNUGatekeeper verwendet. Eine Beispielkonfiguration ist in Anhang 10.4 dargetstellt.

4.1.3 H.323 Client Software (Softphones)

Im Test der Softphones (für H.323 und SIP) kristallisierte sich gleich zu Beginn ein großer Kritikpunkt heraus. Intuitive und übersichtliche Funktion stehen für den Anwender eines Telefones im Vordergrund. Leider beschränken sich viele Softphones (Ausnahmen sind OhPhone und OpenPhone) nicht auf reine Funktionalität, sondern nutzen die zusätzliche Flexibilität gegenüber einem Hardware Telefon manchmal zum Nachteil des Benutzers. Komplizierte Menüs mit detaillierten Einstellmöglichkeiten erschweren das rasche Deployment einer solchen Lösung. Die Möglichkeit, verschiedene Skins (Oberflächen) zu wählen, erschwert dem Support mitunter zusätzlich die Fehlersuche. Problematisch sind weiterhin proprietäre Lösungen für Komfortmerkmale, die nur zwischen gleichen Softphones funktionieren.

Die H.323-Software mit der größten Verbreitung ist das Programm Netmeeting von Microsoft, bei jeder Windows9X Installation ist es dabei - seit dem MS Internet Explorer 5.0. Allerdings stellt sich Netmeeting nicht immer ganz problemlos dar, da der Hersteller an einigen Stellen nicht dem Standard folgt. Reine H.323-Funktionen wie FastStart, H.245-Tunneling werden nicht unterstützt. Die Codec Auswahl lässt nicht spezifizierte Codecs zu,

was beim Verbindungsaufbau zu Problemen führt. Zur Registrierung an einem Gatekeeper muss dieser manuell eingetragen werden. Ein Gatekeeper Discovery wird nicht unterstützt. Obwohl Netmeeting eine LDAP-Anbindung mitbringt (zur Registrierung am ILS-Directory), kann diese nur über Zusatzsoftware mit bestehenden LDAP-Verzeichnissen kommunizieren. Dennoch muss jede H.323-Umgebung auch mit Netmeeting zusammenarbeiten, insbesondere wegen dessen großer Verbreitung.

Die Softphones von „openh323.org“ haben sich im Test bewährt. OhPhone ist wegen seiner Beschränkung auf die Kommandozeile nur bedingt für eine grafische Oberfläche geeignet, für sehbehinderte Menschen stellt es jedoch bisher die einzige Alternative dar. Im Funktionsumfang unterscheidet sich die grafische Variante OpenPhone nicht. Beide verhalten sich standardkonform und unterstützen Gatekeeper Discovery. Auch Komfortmerkmale wie „Anruf Halten“ (Call Hold), „Anruf Weiterverbinden“ (Call Transfer), „Anruf Weiterschaltung“ (CFU, CFBS, CFNR) funktionieren. Das Heranholen eines Anrufes (Intrude Call) konnte bisher nicht zum Funktionieren gebracht werden. Die Möglichkeit einen STUN-Server (NAT Traversal über UDP) anzugeben, sowie die Verwendung von H.245-Tunneling macht OpenPhone auch hinter Firewalls flexibel einsatzfähig. Das OpenPhone war aus diesen Gründen die Wahl für weitere Tests.

4.2 SIP

4.2.1 SIP-Server

Im Gegensatz zur H.323-Fraktion gibt es eine Reihe freier SIP-Implementierungen. Die Gründe dafür sind im einfacheren Aufbau des Protokolls zu suchen.
Eine der ersten SIP-Implementierungen stammt von der Fraunhofer Gesellschaft Fokus. Inzwischen bietet „iptel.org“ als Spin-off eigene SIP-Dienste an. Der SIP Express Router (SER) ist eine unter der GPL stehende Entwicklung, mit derer sich SIP-Infrastrukturen realisieren lassen. Er arbeitet als Proxy/Redirect Server, kann Benutzer registrieren und Nachrichten weiterleiten. Die Webseite bietet eine Reihe (kostenloser) Dienste an, die auf einem SER-System bei „iptel.org“ laufen. Besonders populär ist der Location/Redirection Service, der eine weltweite Erreichbarkeit unter einer einheitlichen SIP-URL ermöglicht (z.B. sip:alexander.noack@iptel.org) Ergänzt wird der Express Router durch Module für Anwendungsfälle, wie z.B. Voice Mail, NAT-Unterstützung, Instant Messaging, usw. Ein PSTN-Gateway existiert nicht. Zu Testzwecken wurde der SER-Server auf

„iptel.org“ genutzt. Dabei funktionierte die Registrierung mit allen Klienten problemlos. Zusätzlich stand die netzinterne Voice Mail Funktion zur Verfügung. Hervorzuheben sind die Interoperabilitätsbestrebungen von SER mit NAT-Firewalls. Ein spezielles Modul (nathelper) schreibt dazu den SDP-Inhalt entsprechend um.

Ein weiterer SIP-Stack stammt von der Free Software Foundation – die GNU oSIP Bibliothek. Es handelt sich hierbei um einen besonders kleinen, in C geschriebenen Stack. Er kann in vielen Softwareprojekten zum Einsatz kommen, da er nur Abhängigkeiten zur C-Bibliothek besitzt. So nutzt z.B. der User Agent LinPhone diese Bibliothek. GNU oSIP ist zudem nicht auf Endpunktfunktionalität beschränkt und findet z.B. in PartySIP auch Verwendung als Proxy.

Vorwiegend für den Carrier Markt ausgelegt ist das SIP-Tooklit „VOCAL“ von „vovida.org“. Der VOCAL-Server enthält alle Bestandteile eines SIP-Netzwerkes. Dies wird über einzelne Module realisiert, die auch getrennt (unabhängig von „VOCAL“) genutzt werden können. Das Hauptaugenmerk bei der Entwicklung des VOCAL-Systems liegt ganz klar auf Robustheit und Skalierbarkeit. So kann jeder Prozess (SIP-Registrar, -Proxy, -Redirect) auf getrennten Rechnern installiert werden. Zusätzlich verschafft das Konzept eines Feature Servers zusätzliche Flexibilität bei der Implementierung kundenspezifischer Wünsche. Aktuell unterstützt der Feature Server Komfortmerkmale wie Anrufweiterleitung, Anrufweiterschaltung, Parken/Heranholen von Anrufen usw. Die Abrechnung (Billing) und Konfiguration (Provisioning) von Nutzern findet ebenfalls große Bedeutung. Dazu nutzt „VOCAL“ offene Standards und Empfehlungen wie z.B. OSP und LDAP. Die Entwickler stehen auf dem Standpunkt, dass die herkömmliche Telephonie nur durch VoIP-Lösungen abgelöst werden kann, wenn diese mindestens die gleiche Qualität der Verbindungen gewährleistet – bezogen auf Sprache und Verbindungsaufbau. Aus diesem Grund unterstützen die VOCAL-Einheiten Quality of Service Merkmale, wie z.B. das RSVP. Leider sind viele interessante Teile des Systems nur über den aktuell äußerst instabilen CVS-Zweig zu erhalten. Dazu gehören grundlegende Module wie der SIP-H.323 Translator (bedient sich der openh323-Bibliotheken), der MGCP-Translator, der Policy Server aber auch nützliche Dinge wie das JTAPI-Feature, die Konferenzerweiterung oder das neue Provisioning System „Mascarpone“. Selbst ohne diese Features gelang das Übersetzen des Sourcecodes erst nach umfangrecher Anpassung der Makefiles. Ein schneller Rechner (>1GHz) mit mind. 512MB RAM und reichlich freiem Platz auf der „/var“-Partition (>1Gbyte) ist leider keine Garantie für ein

sauberes Build. Mit der Version 1.3.0 von „VOCAL“ ist es gelungen, zumindest das Proxy Modul in Betrieb zu nehmen. Die aktuelle Version 1.5.0 konnte nicht zum laufen gebracht werden. Es besteht weiterhin die Möglichkeit, vorkompilierte Binaries zu benutzen, doch sind dort leider wichtige Module nicht enthalten (MGCP, H.323), so dass das Übersetzen der aktuellen CVS-Quellen den einzigen Weg darstellt. Mit den geeigneten Ressourcen (Manpower, Rechentechnik, etc.) ist das Deployment eines VOCAL-basierten VoIP-Systems durchführbar. Geeignet ist eine derartige Installation eher für große Carrier, selbst Firmen müssten mehrere tausend Anrufe pro Sekunden aufbringen, um das Potential dieses Systems nutzen zu können. Abschnitt 4.3 beschreibt eine Lösung, die anstelle eines VOCAL-Systems für die praktische Umsetzung gewählt wurde.

4.2.2 SIP-Clients

Wiederum kommt einer der verbreitetesten SIP-Clients von Microsoft. Der MS Messenger (nicht zu verwechseln mit dem MSN Messenger) versteht seit der Version 4.6 das SIP Protokoll und nutzt die SIMPLE-Erweiterung (SIP for Instant Messaging) für seine Nachrichtendienste. Aber auch normale Telefoniedienste sind möglich. Zu diesem Zweck muss allerdings der Registrierschlüssel *HKCU\Software\Microsoft\MessengerService -> CorpPC2Phone* auf „1“ gesetzt werden, so dass man das Wähl Panel benutzen kann. Da der Messenger die Digest Authentifizierung nicht unterstützt, muss man auf die HTTP-Basic Authentification zurückgreifen. Das Passwort geht in diesem Fall im Klartext über das Internet. Der SIP-Proxy muss ebenfalls diese Art der Authentifizierung unterstützen. Im Test verlief das normale Telefonieren ohne Probleme. Gehende und ankommende Gespräche funktionierten, allerdings sucht man Komfortmerkmale beim Messenger vergebens.

Die Software „X-Lite“ von Xten Networks Inc. stellt ein voll funktionsfähiges SIP-Softphone dar. Die kostenlose „Lite“ Variante unterstützt etwas weniger Features als die ebenfalls erhältliche „Pro“ Version, die auch für WindowsCE verfügbar ist. Für den Testbetrieb erwies sich „X-Lite“ als ausreichend. Die Konfigurationsmenüs sind teilweise redundant und missverständlich. Im Internet finden sich jedoch ausreichend Hinweise für die Konfiguration mit verschiedenen Diensten.

Im praktischen Einsatz wurde dem einfacher zu bedienenen „SJPhone“ von SJ Labs den Vorzug gegeben. Es existiert eine kostenfreie Evaluierungsversion mit unbekannten

Beschränkungen. Es nimmt eine Sonderrolle unter den Softphones ein, da es sowohl H.323 als auch SIP beherrscht. Das „SJPhone“ bietet zusätzlich eine hervorragende Einbindung in Lokalisierungsdienste für beide Protokolle („Free World Dialup“ und ILS). Die H.323-Konformität ist etwas eingeschränkt (kein Gatekeeper Discovery), aber auf SIP-Seite findet man dafür Features wie die Angabe eines separaten Registrierungsservers oder die Weiterleitung eingehender Anrufe auf beliebige SIP-URLs.

4.3 VoIP-Telefonanlage

Reine H.323-Lösungen bzw. reine SIP-Lösungen sind nicht immer der einzige mögliche Weg. Deshalb wurde die „ASTERISK PBX“, eine vollständige Linux Telefonanlage unter der GPL in diesem Zusammenhang näher untersucht.

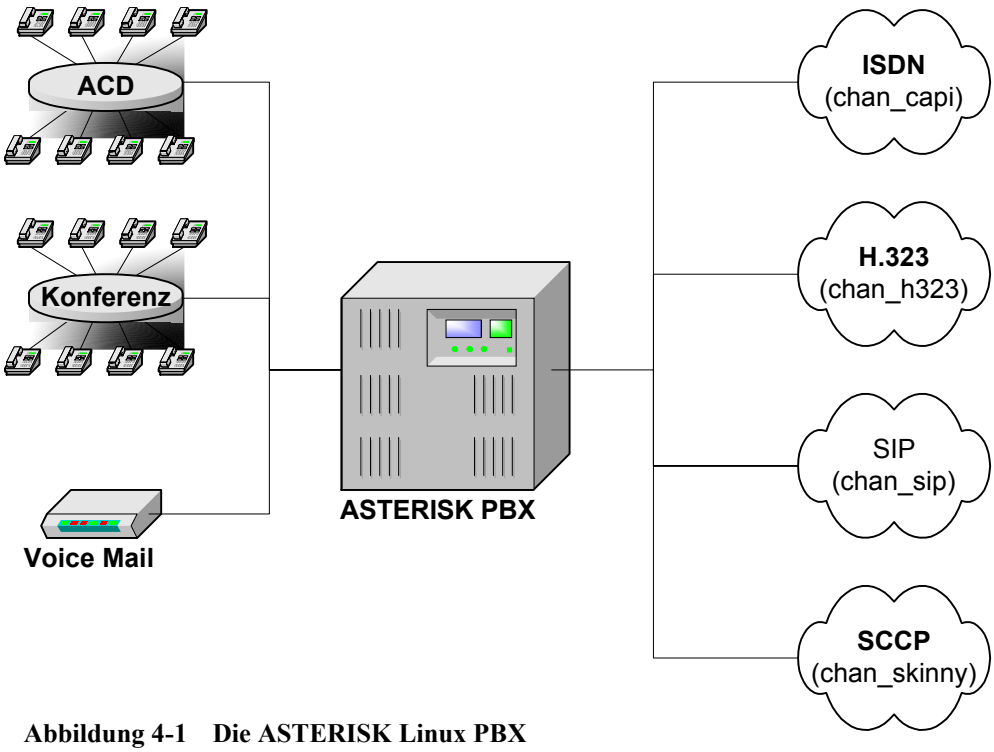

Abbildung 4-1 Die ASTERISK Linux PBX

„ASTERISK“ wurde ursprünglich von Mark Spencer bei der Firma Digium entwickelt. Digium stellt FXO und FXS Hardware her und fördert mit dieser Entwicklung den Absatz ihrer Hardwareprodukte. Unabhängig davon arbeitet „ASTERISK“ auch mit Hardware anderer Hersteller zusammen (alle durch ISDN4Linux bzw. CAPI4Linux unterstützten ISDN-Karten). Für jeden möglichen Kommunikationskanal (Channel) existieren eigene Treiber, mitunter stehen sogar verschiedene zur Auswahl. So gibt es z.B. hardwareseitig mit „chan_oss“ oder „chan_alsa“ Linux Soundkartenunterstützung, aber auch PSTN-

Konnektivität mittels „chan_modem“ bzw. „chan_modem_i4l“. Besonders hervorzuheben ist die im europäischen Raum bedeutungsvolle ISDN-Integration, wobei der „chan_capi“ Treiber dem „chan_modem_i4l“ Treiber, der auf das ISDN4Linux Interface zurückgreift, in Bezug auf Funktionsumfang und Qualität weit überlegen ist (native ISDN-Funktionen, weniger Protokoll Overhead). Auf der Softwareseite werden sowohl SIP als auch H.323 (mittels openh323-Bibliothek) über Kanaltreiber unterstützt. Aber auch ein spezielles Inter-ASTERISK Protokoll (IAX) findet zunehmend Anhänger, da es relativ unproblematisch in NAT-Umgebungen einzusetzen ist. Auch das proprietäre Protokoll SCCP (aka „skinny“) der Cisco IP-Telefone findet Treiberunterstützung.
Neben den Kommunikationskanälen bringt die „ASTERISK“ Telefonanlage eine Reihe von eingebauten Anwendungen mit, von einfachen Komfortmerkmalen bis hin zu Music-on-Hold, Voice Mail, Conferencing und Call Queueing. „ASTERISK“ ist in der Lage zwischen den verschiedenen Medien Verbindungen zu schalten, ganz wie man es von einer Telefonanlage erwartet.
Ursprünglich war neben dem IAX-Protokoll SIP die Wahl der Entwickler, so dass die Unterstützung bereits sehr ausgereift ist. Die Implementierung tritt dem Anwender gegenüber als Proxy auf, kann aber auch gleichzeitig gegenüber einem anderen Proxy/Redirect Server als User Agent in Erscheinung treten (z.B. um einen Anlagenbenutzer über zentrale Registrierungsinstanzen wie „iptel.org“ oder „Free World Dialup“ zu erreichen). In besonders großen Einsatzszenarien kann ein zusätzlicher Dienst wie „VOCAL“ zum Einsatz kommen- dies war in diesem Zusammenhang nicht erforderlich. Etwas jünger ist die H.323-Unterstützung; hier wurde die auf „openh323“ basierende Anpassung getestet. Leider sind die Funktionen bisher auf Gateway Dienste beschränkt, so dass sich für H.323-Klienten zusätzlich ein Gatekeeper empfiehlt. Hierfür wurde der in Abschnitt 4.1.2 beschriebene „gnuGK“ verwendet.

Anrufe waren von beliebigen Clients (Openphone, SJPhone, Netmeeting, MS Messenger, Cisco7960 mit SCCP und SIP Images, POTS) auf beliebige Telefone möglich, je nach verwendetem Rufnummernplan. Die Qualität der Verbindungen sowie die Signalisierungsdauer können absolut überzeugen. Selbst ein PSTN-PSTN Anruf über „ASTERISK“ erfolgt ohne wahrnehmbare Verzögerung. Zu diesem Zweck wurden zwei AVM Fritz!PCI Karten installiert [21] und die Anrufe entsprechend über die „ASTERISK“ Anlage gerouted. Auszüge aus den Konfigurationsdateien sind unter Anhang 10.5 nachzulesen.

Die vielfältigen Anwendungen bis hin zur Einbindung von Sprachsynthese ermöglichen jeden nur denkbaren Einsatz. Momentan beschränkt sich das Management und die Abrechnung auf sehr rudimentäre Methoden (Administration über eine Shell; Call Detail Records in einer MySQL Datenbank), was dem kommerziellen Einsatz noch im Wege stehen dürfte. Allerdings existiert eine CGI-ähnliche Programmierschnittstelle die beliebige Erweiterungen denkbar werden lässt. Zusätzlich gibt es eine Perl Implementierung für das Management Interface.

4.4 Der Business Fall – H.323 im Geschäftseinsatz

Während eines Fachpraktikums bei der KomSa AG im Sommer 2000 entstand die Aufgabe, eine neue Außenstelle per VoIP anzubinden. Die Umsetzung soll im folgenden kurz im Hinblick auf die Kriterien für die Auswahl und die Akzeptanz von VoIP in geschäftskritischen Bereichen erläutert werden.

Konkret bestand der Fall, dass ein Programmierteam von 30-40 Mitarbeitern ein neues, ca. zwei Kilometer entferntes Gebäude beziehen sollten. Die erforderliche Datenanbindung wurde über eine 11Mbit-Richtfunkstrecke gewährleistet. Der klassischen Argumentation über die Konvergenz der Telefon- und Datennetze folgend, war der Kostenfaktor für die Entscheidung VoIP einzusetzen, maßgebend – die Infrastrukturanbindung der Außenstelle war stark eingeschränkt.

Die Herausforderung bestand zunächst darin, eine geeignete Software-/ Hardwareplattform zu evaluieren. SIP spielte zu diesem Zeitpunkt noch keine Rolle, so dass sich die Auswahl auf H.323-Produkte beschränkte, die durch Faktoren wie Verfügbarkeit, Stabilität, Skalierbarkeit und insbesondere Kosten bestimmt wurde. Dabei wurde im ersten Schritt die PSTN-Interoperabilität (Gatewaying) getrennt von der Endegerätesituation untersucht und im Anschluss die beste Kombination beider ausgewählt.

Neben der obligatorischen S_0-Schnittstelle (Punkt-zu-Punkt als auch im Punkt-zu-Mehrpunkt Konfiguration) musste auch die Erweiterung durch eine S_{2M}-Schnittstelle möglich sein. Als PSTN-Gateways bestanden Angebote von Cisco und Radvision in Form von Hardware Gateways sowie diverser Software Gateways anderer Hersteller. Alle Produkte kombinierten Gateway und Gatekeeper in Hardware oder brachten zumindest eine zugeschnittene Gatekeepersoftware mit. Ein wichtiger Punkt war die Anbindung an die unternehmensinterne TK-Anlage (Siemens HiCOM300E), deshalb wurde auch die Siemens Lösung „HighPath

3000“ in Betracht gezogen, allerdings sogleich wieder verworfen als sich herausstellte, dass es sich hardwareseitig um einen Zukauf von Radvision handelte (zu einem ungleich höheren Preis). Bei näherer Untersuchung stellte sich heraus, dass für die beabsichtigte Funktionalität (VoIP-Teilnehmer ruft Teilnehmer der Unternehmens-TK-Anlage und vice versa) zusätzliche (teure) Hardware für die TK-Infrastruktur beschafft werden musste – unabhängig von der VoIP-Gateway Lösung.

Endgeräteseitig stellte sich grundsätzlich die Frage, ob eventuell eine reine Softwarelösung (enorm kostengünstig) für das Programmierteam ausreichen würde. Erfolg und Misserfolg darüber waren unmittelbar von der Akzeptanz der Anwender abhängig. Deshalb wurden infrage kommende Anwendungen zunächst eingehend im Testlabor auf Praxistauglichkeit, Anwenderfreundlichkeit und Stabilität hin untersucht. Als Zugeständnis an die klassische Telephonie wurden Handsets beschafft, die über eine Off-Hook Funktion (Hörer abnehmen/auflegen) verfügten. Letztendlich war eine Benutzerschulung unablässlich, die neben der Funktionsweise der Endgeräte auch insbesondere die Vorteile der neuen Technik in den Vordergrund stellte. Zusatzfeatures wie Voice Mail und Konferenzfunktionen, telefonieren mit Handset und/oder Headset, ortsunabhängige Anrufannahme waren Argumente die gegen skeptische Einwände wie Rechnerstabilität, Netzzuverlässlichkeit, etc. aufgewogen wurden. Die Anforderung ein Fax-Endgerät bereitzustellen konnte aufgrund fehlender Standardisierung (T.38 für Fax over IP wird erst in der Version 2 des H.323 Anhang D hinreichend spezifiziert) nicht stabil implementiert werden (FastStart Probleme des Siemens Terminal-Adapters).

Technologisch gab es keine überlegene Lösung, da an vielen Stellen Probleme auftraten die durch das sich ständig verkürzende Time-To-Market der Hersteller bedingt waren. Einige Produkte gehörten eher zurück in die Entwicklungslabors denn in eine Installation beim Endkunden. Daher entschied man sich zum einen aus Kostengründen, zum anderen aus unternehmenspolitischen Gründen für eine Softwarelösung der Firma SWYX, die u.A. durch die KomSa AG vertrieben wird. Zusätzlich erleichterte eine Testinstallation eventuelle zukünftige Supportleistungen für Kunden dieses Produkts.

Zusammenfassend bleibt festzuhalten, dass der Einsatz von VoIP im Unternehmen maßgeblich von der Akzeptanz der Mitarbeiter abhängig ist. Hinzu kommen technische Aspekte wie Verkehrspriorisierung von VoIP (durch DSCP, ToS, oder VLAN-Tagging). Ein häufig unterschätzter Bereich ist die Integration in bestehende TK-Infrastrukturen – mit einem einfachen PSTN-Gateway ist es oft nicht getan.

5. Vergleich H.323 – SIP

5.1 Verbreitung, Akzeptanz und Verfügbarkeit

Aufgrund der Tatsache, dass die H.323-Spezifikation bereits 1996 veröffentlicht wurde (erstes SIP RFC: 1999), findet diese Technologie bisher eine größere Verbreitung. Die Alternative SIP begeistert viele VoIP-Anhänger wegen des zunächst einfacheren Aufbaus, z.B. werden Signalisierungsinformationen als normale Textnachrichten verschickt, im Gegensatz zu den binärkodierten (ASN.1) Nachrichten im H.323. Nicht zuletzt setzt SIP auf erprobte Internetprotokolle wie das Simple Mail Transport Protocol (SMTP), das Hypertext Transport Protocol (HTTP) sowie den Domain Name Service (DNS).
Entscheidern gibt die Existenz zweier Protokolle für die VoIP-Kommunikation die Möglichkeit zu wählen, denn lange Zeit war es kaum möglich die Funktionalität und Zweckmäßigkeit von H.323 zu bewerten – mangels Alternativen. Vor diesem Hintergrund ist es nachvollziehbar, dass SIP als Newcomer im VoIP-Geschäft bereits nach vergleichsweise kurzer Zeit eine weite Verbreitung gefunden hat.
Zumindest im Produktportfolio findet sich bei jedem Telekommunikationsausrüster mindestens eine SIP-Lösung gegenüber mehreren H.323-Produkten. Gemessen an bestehenden Installationen im kommerziellen Umfeld findet H.323 jedoch eine um ein Vielfaches höhere Verbreitung. Die Integration von SIP in den mit Microsoft WindowsXP ausgelieferten MS Messenger dürfte die Installationsbasis von SIP-fähigen Produkten maßgeblich erhöht haben. Doch erst die Integration von SIP in UMTS Release5 wird auch hier den kommerziellen Durchbruch bringen.

5.2 Leistungsfähigkeit und Funktionsumfang

Das Session Initiation Protocol war angetreten, Medienströme durch einfache bereits existierende Mechanismen zu verwalten, im Gegensatz zu dem komplizierten, umfangreichen aber sehr explizit definierten H.323. Gemessen am benötigten Funktionsumfang eines Telefonienutzers unterscheiden sich beide Protokolle heute kaum noch voneinander. Einzig die Nachrichtenlampe (Message Waiting Indication) ist als Feature bei SIP bisher über eine Standardisierungsaufforderung (Call for consideration as a Proposed Standard) nicht hinausgekommen. H.323 unterstützt dieses seit H.450.7, verfügt jedoch nicht – von

proprietären Lösungen abgesehen – über eine Sekretärinnenfunktionalität (Third Party Call Control), die nur von SIP angeboten wird. Beide Protokolle unterstützen verschiedene Konferenzfunktionen, allerdings nicht immer in gleichem Umfang. Dieser Bereich unterliegt ständig vielen Veränderungen und Verbesserungen, so dass sich über Unterschiede keine allgemeine Aussage treffen lässt. Insbesondere die jüngsten Veröffentlichungen in der H.450-Protokollreihe (siehe Abschnitt 2.1.7) bieten Funktionen für H.323, die bei SIP (noch) nicht definiert sind. Die „vovida.org“ Implementierung eines SIP-Systems („VOCAL“) definiert für diese Aufgabe einen Feature Server, der über eine Anrufverarbeitungssprache (Call Processing Language) verfügt, mit deren Hilfe sich nahezu beliebige Funktionen nachrüsten lassen.

Dank des umfangreichen H.245 (Control Signalling, siehe Abschnitt 2.1.6) besitzt H.323 einen etwas robusteren Mechanismus zum Austausch der Dienstemerkmale (Capabilities Exchange) als dies das Session Description Protocol bei SIP ermöglicht. Zusätzlich hat sich H.323 in der Version 3 in Bezug auf die Performance an das Session Initiation Protocol angenährt. Dazu wurde das seit Version 2 bekannte „Fast Connect“, dass einen Call Setup in 1,5 Round-trips durchführen kann um die Möglichkeit erweitert, über ein verbindungsloses Transportschichtprotokoll (z.B. UDP) übertragen zu werden. Zunächst war der schnellere Call Setup ein wichtiger Vorteil von SIP gewesen, doch in diesem Punkt gibt es keine signifikanten Unterschiede mehr.

Die Kommunikation mit nicht IP-Netzwerken beherrscht H.323 problemlos, da es in vielen Bereichen an Protokolle aus der alten Telephoniewelt angelehnt ist. Insbesondere die Anrufsignalisierung (Q.931, siehe Abschnitt 2.1.5) steht dafür als Beispiel. Deshalb gibt es auch klar definierte Mechanismen für die PSTN-Interaktion (siehe Abschnitt 4.1.1), wo H.323 diese Vorteile intensiv nutzen kann. Die Implementierung eines SIP-Gateways bleibt dagegen vollständig dem Entwickler überlassen, eine Trennung zwischen Media Gateway und Media Gateway Controller ist nicht definiert. Erst die gemeinsamen Bemühungen von ITU und IETF brachten aus einer Reihe konkurrierender Protokolle (darunter das Media Gateway Control Protocol, MGCP) den Standard H.248 (auch als „Megaco“ bezeichnet) hervor. Dieses Protokoll bezeichnet die Kommunikation zwischen dem Media Gateway (MG) und der steuernden Einheit (Media Gateway Control, MGC). Somit beschränkt sich die PSTN-Anbindung auf die Implementierung eines MGC-Servers. Implementierungen existieren sowohl für H.323 (z.B. von Radvision oder Cisco) als auch für SIP (z.B. als Zusatzmodul von „VOCAL“).

Die Möglichkeit der Datenanbindung von SIP an DNS und LDAP zur Benutzerlokalisierung

geht über die H.323-Fähigkeit Facility Nachrichten mit alternativen Kontaktadressen zu versenden weit hinaus. Neben der Anrufsignalisierung ist SIP deshalb insbesondere ein Lokalisierungshilfsmittel. Ein Anrufer kann an verschiedene Adressen weitergeleitet werden, wobei für jede Kontaktadresse weitere Informationen übermittelt werden können – ohne jedoch dem Anrufer Informationen über den Aufenthaltsort des Angerufenen preiszugeben. Auch eine Integration von SIP in das ENUM-Projekt (tElephone Number Mapping) ist realisierbar, da dieses ebenfalls über DNS-Einträge abgewickelt werden wird [26].

5.3 Migrationsfähigkeit und Anpassbarkeit

Im Gegensatz zur vertikalen Architektur von H.323 ist SIP modular aufgebaut. Das erhöht die Anpassbarkeit und die Flexibilität von SIP. Weiterhin können zusätzliche Funktionalitäten in H.323 nur über herstellerspezifische Felder im ASN.1 realisiert werden. Die Komplexität einer derartigen Anpassung verhindert eine schnelle Verbreitung und erschwert somit die Interoperabilität unter Geräten verschiedener Hersteller. SIP bietet dagegen mehr Freiheiten, so können z.B. zusätzliche Rückmeldungen einfach in bestehende Freiräume der hierarchischen Response Codes eingefügt werden. Um herstellerunabhängig zu bleiben, können diese Codes bei der Internet Assigned Number Authority (IANA) registriert werden.
Neben dem Modularen Aufbau beschleunigt insbesondere der textbasierte Ansatz von SIP die Entwicklung und Anpassung an neue Gegebenheiten. Während für H.323 immer zunächst ein komplexer ASN.1-Parser implementiert werden muss, können die textuellen SIP-Nachrichten mit wenigen Handgriffen sogar in Scriptsprachen wie Perl, Python oder TCL verarbeitet werden. Die Textrepräsentation erleichtert zudem das Debugging.
Die zwingende Abwärtskompatibilität von H.323 erhöht bei Programmierprojekten zusätzlich die Komplexität und damit die Größe des Codes.

5.4 Skalierbarkeit

H.323 wurde ursprünglich für die LAN-Kommunikation entworfen. Mit der fortschreitenden Verbreitung des Internets wurde das Konzept der H.323-Zone eingeführt, um WAN-Adressierung zu gewährleisten. Basierend auf den E-Mail Adressen der Benutzer kann so eine Lokalisierung über Zonengrenzen hinweg stattfinden. Im Anhang G der H.323-Empfehlung sind Verfahren definiert, die Adressauflösung, Authentifizierung und Abrechnung über administrative Grenzen hinweg ermöglichen.
Bedingt durch seinen Hintergrund als Internetprotokoll, unterstützt SIP Adressierung über

LAN-Grenzen hinweg. Seine Lokalisierungs- und Adressierungsdienste bauen auf bestehende Internetdienste wie z.B. DNS oder LDAP (Lightweight Directory Access Protocol).
Das Problem der Schleifenerkennung (Loop Detection), das entsteht sobald eine Vermaschung verschiedener Einheiten (H.323 oder SIP) angestrebt wird, lässt sich in H.323 nur schwer über das „PathValue“ Feld oder über den „CallIdentifier“ lösen. Dazu ist allerdings ein verbindungsorientiertes Verhalten der beteiligten Einheiten erforderlich und vermindert dadurch die Skalierbarkeit. Der von SIP genutzte Algorithmus (ähnlich dem Border Gateway Protocol, BGP) erfolgt verbindungslos und erhöht dadurch die Skalierbarkeit. Der „Via:“-Header erlaubt es einem Proxy, vor der Weiterleitung zu überprüfen, ob sein Name bereits gelistet ist. In diesem Fall würde es sich um eine Schleife handeln.
Da sowohl SIP als auch H.323 Anrufkontrollnachrichten prinzipiell verbindungslos übertragen können, unterscheiden sie sich in diesem Punkt in Bezug auf Skalierbarkeit nicht voneinander.
Pro Zone darf im H.323 nur ein Gatekeeper existieren, daher kann eine Lastverteilung nur an den Gateways erfolgen. Dafür erforderliche Nachrichten sind in H.225 spezifiziert. Im SIP Ansatz kann jedem Useragent ein SIP-Server zur Seite gestellt werden. Das ermöglicht eine n-zu-n Skalierung. Eine Lastverteilung kann bei SIP mittels eines DNS SRV-Records erfolgen (erfordert natürlich auch einen redundant ausgelegten DNS-Server). Prinzipbedingt benötigt SIP zur Verarbeitung der textbasierten Kontrollnachrichten weniger Prozessorleistung. Ein SIP-Server könnte theoretisch dadurch mehr Anrufe bearbeiten als ein H.323-Server (Gatekeeper, Gateway). Für die Skalierbarkeit und Stabilität von SIP spricht, dass die Intelligenz in den Endgeräten steckt und der Nachrichtenaustausch über ein vergleichsweise primitives Protokoll erfolgt.

5.5 Sicherheit

Bei der Betrachtung der Sicherheit müssen grundsätzlich die Integrität und die Vertraulichkeit der übertragenen Informationen unterschieden werden.
Zur Sicherstellung der Integrität unterstützen beide Protokolle auf Transportebene das Transmittion Control Protocol. Fehlererkennungsmaßnahmen in höheren Schichten ziehen in jedem Fall Performanceeinbußen nach sich, da sich der Umfang der Kommunikation erhöht (größere Nachrichten bzw. höhere Anzahl Nachrichten).
Die Vertraulichkeit einer übertragenen Nachricht setzt die Authentizität der Nachricht voraus,

d.h. eine Überprüfung des Absenders bzw. des Empfängers muss möglich sein. SIP schlägt dafür eine S/MIME oder PGP/MIME Verschlüsselung des Nachrichtenkopfes vor, ähnlich der DIGEST-Authentifizierung im HTTP. Für H.323 gibt es die Sicherheitsempfehlung H.235 (Securing realtime communication over insecure networks), wobei für die Authentifizierung ein dem Challenge Response Protocol (CHAP) ähnlicher Mechanismus verwendet werden könnte (Cisco Implementierung). Andere Mechanismen, z.B. zertifikatsbasiert mit Public Key Infrastrukturen sind denkbar.

Um die Vertraulichkeit auch für den Inhalt der Nachrichten sicherzustellen, müssen diese Verschlüsselt werden. Es sind dabei die Signalisierungskanäle von den Medienkanälen zu unterscheiden. Letztere müssen bei beiden Protokollen über Mechanismen der Transportschicht (SRTP) oder darunterliegender Schichten (IPSec) realisiert werden und unterscheiden sich deshalb nicht voneinander. Die Signalisierungskanäle werden bei SIP ebenso wie die Authentizität über eine Verschlüsselung der Header sichergestellt. Hierfür existieren bisher nur asynchrone Verfahren (S/MIME) die eine vorherige Authentifizierung erfordern. H.235 sieht für H.323-Kommunikation zwar eine Verschlüsselung des RTP-Payloads vor und definiert Verfahren zum sicheren Schlüsselaustausch im RAS-Kanal, aber eine Sicherung des RTCP-Stroms ist nicht definiert [7].

So bleibt bei beiden Protokollen für die vertrauliche Kommunikation, von proprietären Implementierungen abgesehen, nur eine Sicherung auf darunterliegenden Schichten.

5.6 Quality of Service Unterstützung

Mindestübertragungskapazität (Bandwidth), Paketverlust (Paket Loss), Paketlaufzeiten (max. Delay) sowie Laufzeitunterschiede (Jitter) sind Merkmale von Quality of Service (QoS). Da es sich bei IP aber um ein verbindungsloses Protokoll mit variablen Paketgrößen und Netwerkkomponenten die nach dem FIFO-Prinzip arbeiten handelt, ist eine garantierte Dienstgüte nicht ohne weiteres realisierbar. Zur Lösung des Problems existieren inzwischen drei verschiedene Architekturen.

Die explizite Variante wird DiffServ (Differentiated Services) genannt – explizit deshalb, weil für QoS äußere Maßnahmen getroffen werden müssen, die nicht durch das Protokoll abgedeckt sind. Die Idee des DiffServ ist es, das FIFO-Verhalten der Router durch priorisierte Warteschlangen zu ersetzen. Im Internet Protocol sind drei Bit in einem Feld namens Type of Service (ToS) vorgesehen (RFC 791), mit deren Hilfe sich Serviceklassen (Class of Service, CoS) realisieren lassen. Erweiterungen nutzen ein sechs Bit langes ToS-Feld (RFC 2474)

welches fünf Serviceklassen anhand von Differentiated Services Code Points (DSCP) definiert. RFC 2597 und RFC 2598 spezifizieren eine Klasse für „Beschleunigte Weiterleitung“ (Expedited Forwarding) und vier feiner granulierte Klassen für „Gesicherte Weiterleitung“ (Assured Forwarding). Die Router zwischen zwei Kommunikationspartnern priorisieren den Paketstrom so anhand der im ToS-Feld markierten Serviceklasse (korrekte Konfiguration vorausgesetzt).
Ein Protokoll, dass QoS über IP bietet ist das Resource Reservation Setup Protocol (RSVP). Router auf dem gesamten Kommunikationsweg müssen dieses beherrschen, um einen durchgehenden Kanal mit QoS zu erhalten. Dieser implizite Ansatz heißt IntServ (Integrated Services). Er erlaubt Anwendungen, mit Hilfe des RSVP eine Ende-zu-Ende Verbindung mit fester Bandbreite und garantierter Verzögerung zu reservieren. Da die Anforderung und Überwachung des RSVP-Tunnels Ressourcen bindet, kann IntServ zu Problemen bei Performance und Skalierbarkeit führen.
Eine Lösung, zwischen der zweiten und dritten Schicht des OSI-Modells angesiedelt, ist das Multi Protocol Label Switching (MPLS). Über sogenannte Grenzrouter (Border Router) erfolgt der Eintritt in das MPLS-geschaltete Netzwerk – bei Bedarf unter Aushandlung von Dienstgütemerkmalen. Das Prinzip von MPLS lehnt sich an das Switching im ATM an (Switching anhand von VPI/VCI). Router/Switches leiten Pakete anhand von Tags weiter, die pro Verbindung vergeben werden (Tagged Switching).

Insbesondere an eine Übertragung von Sprachdaten werden Anforderungen in Bezug auf Laufzeit und Verzögerung gestellt, die mit Dienstgütegarantien erfüllt werden können. Deshalb ist VoIP auch eines der Hauptanwendungsfelder für QoS-Mechanismen. Die Sicherung des Medienstroms über geschilderte Mechanismen ist dabei nur ein Aspekt. Die Signalisierungsprotokolle müssen eine Aushandlung der Dienstgüteparameter unterstützen. Zusätzlich bringen sie die Anrufverzögerung als Dienstgütemerkmal ein.
Im H.323-Umfeld stellen Gatekeeper die nötige QoS-Unterstützung bereit. Sie entscheiden bei der Eintrittskontrolle (Admission), ob das Netzwerk über die für einen Anruf benötigten Ressourcen verfügt und damit die Dienstgütemerkmale erfüllt. Das Bandbreitenmanagement (im einfachsten Fall die Kontrolle der zur Verfügung stehenden Bandbreite) ist ebenfalls Aufgabe der Gatekeeper. Mechanismen für das Setzen von DiffServ Parametern (ToS Bits in Layer3 und CoS Bits in Layer2) werden ebenso unterstützt wie die Aushandlung von RSVP-Kanälen. Aufgrund fehlender Standardisierung (z.B. Einordnung von VoIP in Serviceklassen) existieren stark herstellerabhängige Implementierungen.

Die SIP-Implementierungen favorisieren eine Dienstgütesicherung nach dem IntServ-Verfahren, sichergestellt werden sie durch den Common Open Policy Service (COPS, RFC 2748), ebenfalls ein Standardisierungsvorschlag der IETF. Die einheitliche Unterstützung von DiffServ für den Medienstrom ist momentan in der Diskussion (Februar 2003). Als SIP-Einheiten unterstützen die Proxy und Registrar Server Dienstgüteanforderungen. Der Standard bedient sich dafür zusätzlicher Header wie z.B. das „Proxy-Require“ Feld, das die Zusatzanforderungen an einen Proxy definiert.
Das Problem der Anrufverzögerung (Call Setup Delay) wurde bereits unter 5.2) diskutiert. Hier besitzt H.323 einen kleinen Vorteil, da es Anrufe parallel per TCP und UDP aufbauen kann, so dass, sollte der Rufaufbau per UDP erfolglos sein, die TCP Verbindung übernehmen kann. SIP müsste diesen Vorgang nacheinander ausführen. Erfolgt die Signalisierung jedoch ausschließlich über UDP, so gibt es keine Unterschiede in der Anrufverzögerung. Zusätzlich unterstützen beide Protokolle das Setzen der DiffServ Parameter für den Anrufkontrollkanal. Herstellerabhängige Unterschiede existieren jedoch auch hier.

5.7 Management und Accounting

In einem ungemangeten Netzwerk wie dem Internet, wo Pakete viele verschiedene Wege zum Ziel nehmen können, müssen bei Bedarf zusätzliche Mechanismen installiert werden, um ein Management und Accounting zu gewährleisten. Grundsätzlich besteht die Möglichkeit auf Netzwerkebene oder Transportebene einen Tunnel aufzusetzen wie es in Virtual Private Networks (VPNs) erfolgt. Allerdings führt dieser Ansatz zusätzliche Komplexität ein, die den Verwaltungsaufwand mitunter erhöht. Eine transparente Möglichkeit des Managements ist daher wünschenswert.
Naturgemäß unterstützt das aus dem Telefonieumfeld stammende H.323 Möglichkeiten zur Abrechnung (Billing) eines Telefongesprächs. Erfolgt ein Anruf über den Gatekeeper (Gatekeeper Routed) so erhält der Gatekeeper ständig Informationen über bestehende Gespräche, also auch über Anrufbeginn und -ende. Selbst im Direct Call Modell (Client ruft Client direkt) erfolgt eine Rückmeldung an den Gatekeeper über das RAS-Protokoll. Management eines aktiven Anrufes wird über die erweiterten Funktionen der H.450.x Standards ermöglicht.
Accounting in SIP erfolgt nativ nur dann, wenn ein SIP-Proxy für die Dauer eines Gespräches in der Signalisierungskette verbleibt (Performance- und Skalierungsproblem). In diesem Fall Protokolliert der Proxy den Zeitpunkt der „INVITE“ und „BYE“ Nachrichten. Allerdings

kann es im Gegensatz zum RAS-Protokoll zu zeitlichen Ungenauigkeiten kommen, da das Timing der SIP-Nachrichten nicht hinreichend spezifiziert ist. Weitaus flexibler ist das Open Settlement Protocol (OSP), ein ETSI Standardisierungsvorschlag, der Prozeduren für Authentifizierung, Zugangskontrolle und Abrechnung für Multimediadienste über Domaingrenzen hinweg definiert. Zusammen mit COPS (siehe 5.6) werden beide Verfahren von verschiedenen SIP-Implementierungen verwendet. Eine relativ neue Möglichkeit ist die Nutzung eines Back-to-Back User Agents (B2BUA), der anders als ein Proxy die gesamte Zeit eines Anrufs im Kommunikationskanal verbleibt und so eine detaillierte Abrechnung ermöglicht. Einige Implementierungen nutzen dieses Verhalten für Third Party Control Services. Für Accounting Zwecke ließe sich dieses Verfahren z.B. für die Implementierung von Prepaid Diensten nutzen. Wenn das Guthaben aufgebraucht ist, werden einfach beide Seiten des Anrufs terminiert. Ein B2BUA benötigt ähnliche Performance- und Skalierungsbetrachtungen, wie sie auch für einen SIP User Agent bestehen.
Es sind Szenarien denkbar, in denen Benutzer zweier nicht vertrauenswürdiger VoIP-Einheiten (SIP UA oder H.323 Terminal) eine Anrufendesignalisierung durchführen, die Medienströme aber aufrecht erhalten, um eine Abrechnung zu umgehen (Toll Fraud). In diesem Fall sollten die Managementsysteme (COPS' Policy Enforcement Points oder Gatekeeper) die Dienstgütegarantie der Verbindung entfernen, so dass diese auf „Best Effort" umgestellt wird. Einige Netzwerke, insbesondere UMTS R5, sind dabei so konfiguriert, dass der Pakettransport ohne QoS-Vertrag gestoppt wird und somit eine derartige Umgehung unterbindet.

5.8 IPv6 Fähigkeit

Prinzipiell sind beide Protokolle IPv6 fähig, da sie unabhängig von der Netzwerkschicht arbeiten können. D.h. der reine Datenaustausch erfolgt auch in IPv6.
Es ändern sich allerdings die Adressierungsmechanismen (größerer Adressraum) sowie Funktionalitäten die direkt auf die IP-Paketstruktur zugreifen (QoS). Die starke Anlehnung an bestehende Internetprotokolle kommt SIP dabei zugute. Sobald diese eine Anpassung erfahren, wird auch die von SIP genutzte Funktionalität IPv6 fähig.
Es bestehen Gateway Implementierungen für beide Protokolle. Im Falle von H.323 übersetzt ein spezieller Gatekeeper die Signalisierungsinformationen, bei SIP erledigt das ein B2BUA. Ein entsprechender Media Gateway sorgt bei H.323 zusätzlich für die Anpassung der Medienströme.

5.9 Zusammenfassung

	H.323	SIP
Akzeptanz, Verbreitung, Verfügbarkeit	länger auf dem Markt; hoher Reifegrad; verschiedene kommerzielle und nicht-kommerzielle Einsatzgebiete;	baut auf bestehende Internetprotokolle auf; attraktiv für Entwickler, da textbasiert
Leistungsfähigkeit und Funktionsumfang	robuste Mechanismen zum Austausch der Dienstmerkmale über H.245; gleichzeitiger Verbindungsaufbau über TCP und UDP; sehr gute PSTN-Integration	vielfältige Möglichkeiten zur Anbindung an bestehende Verzeichnisdienste zwecks Benutzerlokalisierung
Migrationsfähigkeit und Anpassbarkeit	Abwärtskompatibel zu den einzelnen Versionen	modularer Aufbau; textbasiert
Skalierbarkeit	H.225 definiert Gateway Nachrichten, die die Gatekeeper zum Load-Balancing über mehrere Gateways nutzen	Adressierung über LAN-Grenzen hinweg durch Nutzung verschiedener Internetprotokolle; einfache Schleifenerkennung mittels „Via:“-Header oder Hop-Count; evtl. weniger Prozessorlast durch textbasierten Nachrichtenaustausch
Sicherheit	H.235 definiert Authentifizierungs-mechansismen; SSL-Verschlüsselung der Transportschicht per SRTP	Authentifizierung über HTTP-Auth; Hop-zu-Hop Sicherung über SSL/TLS, SSH oder SHTTP; Ende-zu-Ende Sicherung über S/MIME oder PGP; Authentizität über S/MIME oder PGP

Quality of Service Unterstützung	Bandbreitenkontrolle über Beitrittsbeschränkungen; Unterstützung von DiffServ und IntServ	SIP-Proxy und Registrar regeln Dienstgüteanforderungen über zusätzliche Header-Informationen; standardisierte Unterstützung von IntServ
Management und Accounting	umfangreiche Möglichkeiten zur Abrechnung am Gatekeeper über das RAS-Protokoll; H.450.x Erweiterungen bieten verschiedene Managagement-ansätze bestehender Verbindungen	zusätzliche Protokolle wie OSP und COPS sind erforderlich
IPv6 Fähigkeit	ein spezieller Gatekeeper übersetzt Signalisierungsinformationen ein Media Gateway übersetzt Medienströme	ein B2BUA übersetzt Signalisierungsinformationen und Medienströme

6. Schlussfolgerungen

Nach eingehender praktischer Untersuchung sowie dem Vergleich der Hintergründe beider Protokolle in der Theorie, ist zusammenfassend zu bemerken:

- SIP und H.323 verfügen (inzwischen) über einen Funktionsumfang, der über den aus dem PSTN bekannten hinausgeht, der sich aber zwischen den beiden Protokollen kaum unterscheidet.

- Das Session Initiation Protocol ist einfach strukturiert und gut dokumentiert. Seine Einfachheit ist ein Faktor, der eventuell zu den zahlreichen verschiedenen Implementierungen beiträgt. Für Projekte, die über begrenzte Ressourcen verfügen, ist es daher das Protokoll der Wahl, zumal es durch seine offene Architektur leicht an bestimmte Vorgaben anzupassen ist. Die

Fehlersuche wird durch den Textcharakter des Protokolls erheblich erleichtert.

- Für das H.323-Protokoll spricht seine Robustheit, bestimmt durch eine Reihe von Standardisierungswerken. Dem Anspruch seiner Schöpfer, ein PSTN-Equivalent für Best Effort Netzwerke zu schaffen, wird es gerecht. Die klare Definition der einzelnen Funktionen kommt der Interoperabilität zugute und unterstreicht den ausgereiften Charakter von H.323.

- Während SIP sich hervorragend für den Einsatz in Internetstrukturen eignet, bringt H.323 bessere Vorraussetzungen für die PSTN-Integration mit. Zum einen orientiert sich H.323 an PSTN-Signalisierungsmechanismen und zum anderen bringt es eine eigene Komponente (H.323-Gateway) mit, die diese Funktion vorsieht. Geht die Realisierung über einen lokalen Rahmen hinaus ist eindeutig SIP mit seiner URL-basierten Adressierung im Vorteil. H.323 muss für derlei Einsatzszenarien umfangreich konfiguriert werden.

- Im Hinblick auf das Management und Accounting von Verbindungen offenbart SIP noch Schwächen aufgrund fehlender Spezifikationen. H.323 besitzt entsprechende Definitionen bereits im Protokoll.

Beide Protokolle haben voneinander gelernt und tun es immer noch. Herstellern bleibt nur die Unterstützung beider Protokolle oder zumindest die Bereitstellung von entsprechenden Schnittstellen. Im praktischen Teil ist sehr klar zu sehen, dass die Variante der Kombination beider Protokolle, den einzelnen Implementierungen stark überlegen ist, denn durch eine derartige Kombination können die Vorteile beider Protokolle optimal genutzt werden.
Solange der Fokus auf PSTN-Integration liegt, kann nach Meinung des Autors das H.323-Protokoll eine minimale Führungsrolle beanspruchen. Es leistet in Verbindung mit PSTN-Interconnects zuverlässige Dienste, die man im Telefonnetz erwartet. Bewegt man sich in reinen IP-Umgebungen, sind die Karten gleichgut verteilt. Dem Argument, eine binäre Kodierung der Daten hätte kleinere Pakete zur Folge steht die Simplizität und die ständig wachsende zur Verfügung stehende Bandbreite entgegen. Schließlich, seit spätestens 10-12 Jahren hat sich schon einmal ein offenes Protokoll gegenüber einer Vielzahl von komplizierten (teilweise proprietären) Ansätzen durchgesetzt.

7. Abkürzungsverzeichnis

ACD	Active Call Distribution
ASN.1	Abstract Syntax Notation (One)
ATM	Asynchron Transfer Mode
BGP	Border Gateway Protocol
B2BUA	Back to Back User Agent
CAPI	Common ISDN Application Programmino Interface
CFBS	Call Forward on Busy
CFNR	Call Forward on Not Reachable
CFU	Call Forward, Unconditional
CGI	Common Gateway Interface
CHAP	Challenge Response Authentification Protocol
CODEC	Coder/ Decoder
COPS	Common Open Policy Service
CoS	Class of Service
CTI	Computer Telephony Integration
CVS	Concurrent Versioning System
DiffServ	Differential Services
DNS	Domain Name Service
DSCP	Differentiated Services Code Points
DSP	Digitaler Signalprozessor
ENUM	tElephone Number Mapping
ETSI	European Telecommunications Standards Institute
FIFO	First In – First Out
FXO	Foreign Exchange Office
FXS	Foreign Exchange Service
GK	Gatekeeper
GNU	GNU is Not Unix
GPL	General Public License
GW	Gateway
HTTP	Hypertext Transfer Protocol

I4L	ISDN for Linux
IANA	Internet Assigned Numbers Authority
IAX	Inter Asterisk Exchange Protocol
IETF	Internet Engineering Taskforce
ILS	Internet Locator Service
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPSec	Internet Protocol Secure
IPX	Internetwork Packet Exchange
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
IVR	Interactive Voice Response System
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LSB	Linux Standard Base
MCU	Multipoint Control Unit
MGCP	Media Gateway Control Protocol
MIME	Multipurpose Internet Mail Extension
MPL	Mozilla Public License
MPLS	Multi Protocol Label Switching
NAT	Network Address Translation
OLC	Open Logical Channel
OSI	Open Systems Interchange
OSP	Open Settlement Protocol
PABX	Private Analogue Branch Exchange
PBX	Private Branch Exchange
PDH	Plesiochrone Digitale Hierarchie
PGP	Pretty Good Privacy
POTS	Plain Old Telephony System
PSTN	Public Switched Telephony Network
QoS	Quality of Service
RAS	Registration, Admission, Status

RFC	Request for Comments
RPM	Redhat Package Manager
RSVP	Resource Reservation Setup Protocol
RTP	Realtime Transport Protocol
RTCP	Realtime Control Protocol
SCCP	Skinny Client Control Protocol
SCN	Switched Circuit Network
SDH	Synchrone Digitale Hierarchie
SDP	Service Description Protocol
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
S/MIME	Secure MIME
SMTP	Simple Mail Transfer Protocol
SPX	Sequenced Packet Exchange
SRTP	Secure Realtime Transport Protocol
STUN	Simple Traversal of UDP through NAT
TCP	Transport Control Protocol
TCS	Terminal Capabilities Exchange
TK	Telekommunikation
ToS	Type of Service
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UMTS	Universal Mobile Telephony Service
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
VCI	Virtual Channel Identifier
VLAN	Virtual Local Area Network
VPI	Virtual Path Identifier
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network

8. Abbildungsverzeichnis

Abbildung 2-1 Architektur des H.323 Protocol Stack.....5
Abbildung 2-2 Signalisierungsablauf in H.323.....10
Abbildung 2-3 Komponenten eines H.323 Netzes 11
Abbildung 3-1 SIP User Agents (Client/Server Komponente).....16
Abbildung 3-2 SIP Verbindungsaufbau 17
Abbildung 4-1 Die ASTERISK Linux PBX 24

9. Literaturverzeichnis

[1] A. Badach & E. Hoffmann, Technik der IP-Netze, 2001, Hanser Verlag, München

[2] U. Black, Internet-Technologien der Zukunft, 1999, Addison-Wesley-Longman, München

[3] R. Schaphorst, Videoconferencing and Videotelephony, 1996, Artech house Inc., Boston

[4] D. & E. Minolli, Delivering Voice over IP Networks, 1998, Wiley Computer Publishing, New York

[5] L. Dang, C. Jennings & D. Kelly, Practical VoIP Using Vocal, 2002, O'Reilly, Sebastopol

[6] K.-O. Detken, Kampf der Normen, NetworkWorld Germany, November 2001

[7] L.L. Iacono, Rote Telefone – Abhören von IP-Telefonaten, iX Magazin für professionelle Informationstechnik, Mai 2002

[8] Microtronix Systems Ltd, SIP vs. H.323 – A Comparison, http://microtronix.ca/sip_vs_h323.htm, 2003

[9] H. Schulzrinne, SIP Design Choices, http://www.cs.columbia.edu/sip/h323-comparison.html, 2003

[10] Packetizer, Inc., H.323 versus SIP: A Comparison, http://www.packetizer.com/iptel/h323_vs_sip/, 2003

[11] P. Carden, Network Design Manual – Building Voice over IP, http://www.networkcomputing.com/netdesign/1109voip2.html, 2000

[12] E.Doron, SIP And H.323 For Voice/Video Over IP – Complement, Don't Compete!, http://www.tmcnet.com/it/0801/0801radv.htm, 2001

[13] Intel Corp., H.323 Tutorial, http://www.iec.org/online/tutorials/h323/topic01.html, 2000

[14] S. Yang, Seminararbeit: VoIP-H.323, 2001, Technikum Wien

[15] M. Bertozzi, VoIP: le problematiche di integrazione voce/dati, http://bsing.ing.unibs.it/~pier/seminari/Bertozzi2.pdf, 1999

[16] Craig Southeren, Co-Initiator des openh323-Projekts auf http://www.openh323.org

[17] GNUGatekeeper Success Stories, http://www.gnugk.org/gnugk-success.html

[18] The Open Source Linux PBX, http://www.asterisk.org

[19] Chan_capi für ASTERISK, http://www.junghanns.net/asterisk/

[20] VoIP an der Universität des Saarlandes Saarbrücken, http://graphics.cs.uni-sb.de/VoIP/

[21] Two Fritz!Card Howto, http://www.quiss.org/caiviar/Two-Fritzcards-HOWTO

[22] X-Lite Homepage, http://www.xten.com/lite.php

[23] SJPhone Homepage, http://www.sjlabs.com/products.html

[24] iptel.org Homepage, http://www.iptel.org

[25] Free World Dialup Homepage, http://www.pulver.com/fwd/index.html

[26] M. Haberler, The Asterisk IP PBX and ENUM, http://enum.nic.at/documents/AETP/Presentations/Austria/0021-2003-08-The_Asterisk_IP_PBX_and_ENUM.ppt

10. Anhang

10.1 Vergleich: Freie Gateways

OpenISDNGw:

Entwickelt von Carlos Sevilla

http://www.gae.ucm.es/~openisdngw/home_en.php,

Stand:	*06.02.2002 (Version 1.2.2);* *openh323 v1.7.8, pwlib v1.2.9;* *kompiliert z.B. mit den openh323-Paketen von SuSE8.0*
Features:	*Gatekeeper Discovery; H.323-Alias Mapping über Konfigurationsdatei oder über DTMF als Prompt; unterstützt A-LAW, µ-Law, GSM und LPC10 Codecs, Kanalmonitoring; unterstützt beliebige HiSax kompatible ISDN-Karte in „isdn4linux“*
Funktion:	*Netmeeting 3.01(Win98SE) => OpenISDNGw => Telefon;* *Verzögerung deutlich spürbar (500-800ms);* *Echo auf beiden Seiten*

ISDN2h323:

Entwickelt von Marco Budde telos EDV Systementwicklung GmbH

http://www.telos.de/linux/H323/

Stand:	*26.04.2002 (Version 0.4a1), 15.08.02 (CVS); openh323 v1.8.5, pwlib v1.2.15; (openh323 und pwlib selbst komiliert, SuSE7.3 RPMs unter http://gauss.dyndns.org/~alex/h323/); ab SuSE8.1 in der aktuellen Version enthalten (nicht getestet)*
Features:	*Gatekeeper Discovery; H.323-Alias-Mapping mit E-Mail Notification; unterstützt A-LAW, µ-Law, GSM und LPC10 Codecs; Echokompensation; Aufnahmepegelanpassung; Webstatusinterface; Webtelefonbuch; Least Cost Routing*
Funktion:	*Netmeeting 3.01(Win98SE) => isdn2h323 => Telefon; Verzögerung spürbar (<400ms); Echo auf Netmeeting Seite*

(Eine weitere Software „isdngw“ existiert als Fork von „isdn2h323“. Die Entwicklung wurde aber im März 2002 gestoppt, und Verbesserungen wurden in „isdn2h323“ eingearbeitet.)

10.2 Vergleich: Freie Gatekeeper

OpenGatekeeper:

Entwickelt von Egoboo Ltd.

http://opengatekeeper.sf.net

Stand:	*26.08.2001 (Version 0.9alpha4); openh323 v1.6.0, pwlib v1.1.36; kompiliert z.B. mit den openh323-Paketen von SuSE7.3*
Features:	*GKrouted calls; Gateway Prefix Registration; H.323v2 Aliases*
Funktion:	*Gateway Registrierung funktioniert (isdn2h323 v0.3a1); Direct Call mit Party Number Alias funktioniert; GKrouted Calls produzieren Abstürze in Kombination mit Netmeeting (OpenPhone ⇔ Netmeeting; isdn2h323 ⇔ Netmeeting)*

OpenH323Proxy:

Entwickelt von Roman Skvirsky (Weiterentwicklung von opengatekeeper)

http://openh323proxy.sf.net

Stand:	*13.05.2002 (Version 0.9.10); openh323 v1.8.5, pwlib v1.2.15; (openh323 und pwlib selbst komiliert, SuSE7.3 RPMs unter http://gauss.dyndns.org/~alex/h323/);*
Features:	*basiert auf opengatekeeper, daher ähnlicher Funktionsumfang; zusätzlich Proxy-Funktion für H.323-Calls*

Funktion: *Gateway Registrierung funktioniert; GKrouted Calls funktionieren auch mit Netmeeting; Proxy funktioniert nur auf dem NAT-Router*

Openh323Gatekeeper (gnuGK):

Entwickelt von Jan Willamowius (und anderen)

http://www.gnugk.org

Stand: *29.06.2003 (Version 2.0.5); 28.08.2003 (CVS); openh323 v1.12.2, pwlib v1.5.2 (19.08.2003); aktuell noch keine RPMs verfügbar*

Features: *unterstützt zusätzlich Gatekeeper Hierarchien (Redundanzen); Authentifizierung gegen MySQL und/ oder LDAP (Version im CVS unterstützt Standardkonform nur noch RADIUS); QoS Unterstützung (ToS Feld im RTP Stream); Proxy-Funktion; NAT-Unterstützung; Call-Queueing (ACD Funktionalität); Umfangreiches Monitoring; SoftPBX; Multi-threaded (skalierbar)*

Funktion: *GKrouted Calls und ProxyRouted Calls funktionieren mit Netmeeting und OpenPhone; SoftPBX Funktionalität erlaubt das trennen einer bestehenden Verbindung über ein Management Interface; CTI funktioniert z.B. mittels einer einfachen ACD Implementierung in Java*

10.3 Konfiguration: isdn2h323 Gateway

```
#
# /etc/isdn2h323.config
#

# ISDN devices
device=/dev/ttyI5
device=/dev/ttyI6

# our default ISDN MSN
default_msn=5193519

# Use automatic gain control?
use_agc=yes

# Use echo compensation?
use_echo_comp=yes

# H.323 name of the gateway
gateway_name=COMLAB_GW

# H.323 ALaw/muLaw codecs high network bandwidth (intranet)
use_law_codec=yes

# H.323 GSM codec  low network bandwidth (modem)
use_gsm_codec=yes

# H.323 LPC10 codec very low network bandwidth (modem)
use_lpc10_codec=yes

# Name of the gatekeeper to be used
gatekeeper=automatic

# Tells the gateway which IP address to listen to.
network_address=139.30.208.29

# Tells the gateway which TCP/IP port to listen to (default H.323 port: 1720)
network_port=1720

# Who is allowed to dial out (H.323 -> ISDN)?
dialout_ip=139.30.208.

# Prefixes of forbitten ISDN numbers
denied_prefix=0190

# Prefix ISDN numbers
prefix=0

# The e-mail address of the administrator
admin=root@lingua.comlab.uni-rostock.de

# name of the HTML status file
status_filename=/var/www/lingua.comlab.uni-rostock.de/isdn2h323.html

# name of the HTML phone directory file
directory_filename==/var/www/lingua.comlab.uni-rostock.de /isdn2h323_directory.html

# routing
number=5193519
user=alex
```

10.4 Konfiguration: GNUGatekeeper

```
#
# /etc/gnugk.ini
#

[Gatekeeper::Main]
# 'config is present' indicator. Has to be 42.
Fourtytwo=42
Name=COMLAB_GK

[RoutedMode]
GKRouted=1
H245Routed=0
CallSignalPort=1721
CallSignalHandlerNumber=1
RemoveH245AddressOnTunneling=0
AcceptNeighborsCalls=0
AcceptUnregisteredCalls=0
DropCallsByReleaseComplete=1

[Proxy]
Enable=0

[RasSrv::GWPrefixes]
COMLAB_GW=0

[RasSrv::RRQFeatures]
AcceptEndpointIdentifier=1
AcceptGatewayPrefixes=1

[GkStatus::Auth]
rule=allow

# settings for inbound call distribution with virtual queue
[CTI::Agents]
VirtualQueue=CC
CTI_Timeout=120
```

10.5 Konfiguration: ASTERISK

Beispiel für Asterisks SIP-Konfiguration:

```
;
; /etc/asterisk/sip.conf
;
[general]
port = 5060
bindaddr = 0.0.0.0
context = default
tos=lowdelay

[1234]
type=friend
insecure=yes
username=1234
secret=Hansi123
host=dynamic
dtmfmode=inband
mailbox=1234
context=sip-intern-intern

[1001]
type=friend
username=1001
secret=c0wed67he2
host=dynamic
dtmfmode=rfc2833
mailbox=1001
context=sip-intern-intern
callerid=Comlab <1001>

[1002]
type=friend
username=1002
secret= a4hj24khdf3
host=dynamic
dtmfmode=rfc2833
mailbox=1002
context=sip-intern-extern
callerid=Comlab <1002>
```

Beispiel für Asterisks H.323-Konfiguration:

```
;
;/etc/asterisk/h323.conf
;
[general]
port = 1720
bindaddr = 0.0.0.0
context=default
tos=lowdelay
gatekeeper = DISCOVER
AllowGKRouted = yes

[1235]
type=user
dtmf=inband
context=h323-intern
```

Beispiel für Asterisks Skinny-Konfiguration:

```
;
;/etc/asterisk/skinny.conf
;

[general]
port = 2001
bindaddr = 0.0.0.0
context = default
dateFormat = M-D-Y
keepAlive = 120

[7960]
device=SEP000B46A7E557
callerid="Cisco 7960 Comlab" <7960>
linelabel="Comlab"
context=skinny-intern
mailbox=7960
line => 7960
```

Beispiel für Asterisks CAPI-Konfiguration:

```
;
;/etc/asterisk/capi.conf
;

[general]
nationalprefix=0
internationalprefix=00
rxgain=0.8
txgain=0.8

[interfaces]
; external lines Controller I
msn=5193519,5193520,5193521
incomingmsn=5193519,5193520,5193521
controller=1
softdtmf=1
context=dialin-extern
echosquelch=1
echocancel=yes
echotail=64
devices=2

; internal lines Controller II
msn=401
incomingmsn=*
controller=2
softdtmf=1
context=dialin-intern
echosquelch=1
echocancel=yes
echotail=64
devices=2
```

Beispiel für Asterisks Extensions:

```
;
;/etc/asterisk/extensions.conf
;

[general]
static=yes
writeprotect=no

[globals]

;
; mit "0" gelangt man ins Uni-TK-Netz
;
[dialout-intern]
ignorepat => 0
exten => _0.,1,Dial(CAPI/401:b114${EXTEN:1}|20|tr)

;
; mit "9" gelangt man ins externe (Telekom) Netz
;
[dialout-extern]
ignorepat => 9
exten => _9.,1,Dial(CAPI/5193520:b${EXTEN:1}|20|tr)

;
; "40X" sind die MSNs des internen (Uni) S0
;
[dialin-intern]
exten => 401,1,Dial(SIP/1001)
exten => 402,1,Dial(SIP/1002)
exten => 409,1,Dial(Skinny/7960@7960)
exten => _40[3-9],1,Answer()
exten => _40[3-9],2,SayDigits(${EXTEN})
exten => _40[3-9],3,Hangup()

;
; "51935XX" sind die MSNs des externen (Telekom) S0
;
[dialin-extern]
exten => 5193519,1,Dial(Skinny/7960@7960))
exten => 5193521,1,Dial(SIP/1002)
exten => 5193520,1,Answer()
exten => 5193520,2,SayDigits(${EXTEN})
exten => 5193520,3,Hangup()

;
; der Kontext für H323 Verbindungen, die aus dem
; internen Netz (Uni) initiert werden
;
[h323-intern]
include => parkedcalls

;
; der Kontext für SIP Verbindungen, die aus dem
; internen Netz (Uni) initiert werden und in das Uni-TK-Netz gehen
;
[sip-intern-intern]
include => parkedcalls
```

```
exten => _XXXX,1,Dial(CAPI/401:b114${EXTEN}|20|tr)
exten => _796X,1,Dial(SKINNY/${EXTEN}@${EXTEN}|20|tr)

;
; der Kontext für SIP Verbindungen, die aus dem
; internen Netz (Uni) initiiert werden und in das Telekom-TK-Netz gehen
;
[sip-intern-extern]
include => parkedcalls
exten => _.,1,Dial(CAPI/5193521:b${EXTEN}|20|tr)

;
; der Kontext für Skinny Verbindungen, die aus dem
; internen Netz (Uni) initiiert werden
;
[skinny-intern]
include => dialout-intern
include => dialout-extern
include => parkedcalls
exten => _100X,1,Dial(SIP/${EXTEN}|20|tr)
exten => _796X,1,Dial(SKINNY/${EXTEN}@${EXTEN}|20|tr)

;
; der Default-Kontext falls kein anderer passt
;
[default]
```

Konfiguration für Cisco7960 mit SCCP-Image „*XMLDefault.cnf.xml*“:

```
<Default>
 <callManagerGroup>
  <members>
   <member  priority="0">
    <callManager>
     <ports>
      <ethernetPhonePort>2001</ethernetPhonePort>
     </ports>
     <processNodeName>139.30.208.29</processNodeName>
    </callManager>
   </member>
  </members>
 </callManagerGroup>
 <loadInformation7      model="IP Phone 7960">P00303020214</loadInformation7>
 <loadInformation124    model="Addon 7914">S00103020002.bin</loadInformation124>
</Default>
```

Konfiguration für Cisco7960 mit SIP-Image „*SIPDefault.cnf*“:

```
proxy1_address: "lingua.comlab.uni-rostock.de"
proxy1_port: 5060
proxy_register: "1"
timer_register_expires: "120"
preferred_codec: "none"
tos_media: "5"
dtmf_inband: "1"
dtmf_outofband: "avt"
dtmf_db_level: "3"
dtmf_avt_payload: "101"

sntp_server: "160.45.10.8"
sntp_mode: directedbroadcast
time_zone: CET
```

```
telnet_level: 2
remote_party_id: "1"
call_hold_ringback: "1"
phone_prompt: "SIP Phone"
phone_password: "cisco"
```

Gerätespezifische Konfiguration für Cisco7960 mit SIP-Image „*SIP000B46xxxxxx.cnf*“:

```
line1_name: 1001
line1_authname: "1001"
line1_shortname: "Intern"
line1_password: "secret"
line1_displayname: "1001"

line2_name: 1002
line2_authname: "1002"
line2_shortname: "Extern"
line2_password: "secret"
line2_displayname: "1002"

line3_name: "UNPROVISIONED"
line4_name: "UNPROVISIONED"
line5_name: "UNPROVISIONED"
line6_name: "UNPROVISIONED"

phone_label: "Comlab 1001"
user_info: none

logo_url: "http://139.30.208.29/cip/logo/test.bmp"
services_url: "http://139.30.208.29/cip/services/index.xml”
directory_url: " http://139.30.208.29/cip/directory/index.xml"
```