Universität Rostock

Fakultät für Ingenieurwissenschaften

Fachbereich Elektrotechnik und Informationstechnik

Institut für Nachrichtentechnik und Informationselektronik

Diplomarbeit

Einsatz von Firewallstrukturen in modernen Kommunikationssystemen

cand. ing. Andreas Olsson

30. Juni 2001

Betreuer

Dr. H.-D. Melzer (Universität Rostock)

Dipl.-Ing. B. Podey (Universität Rostock)

Dipl.-Ing. T. Krüger (Siemens AG Hamburg)

Vorwort

Diese Diplomarbeit wurde durch die Unterstützung einer Reihe von Personen möglich, bei denen ich mich bedanken möchte.

Besonderer Dank gilt den Betreuern meiner Diplomarbeit; Herrn Dr. H.-D. Melzer (Universität Rostock), Herrn B. Podey (Universität Rostock) und Herrn T. Krüger (Siemens AG Hamburg) für die Themenfindung sowie die konstruktive Zusammenarbeit und Betreuung während der Diplomarbeit.

Als Nichtmuttersprachler, danke ich besonders meinen Freunden für nützliche Anregungen und Hinweise. Sie haben mir geholfen, meine Niederschrift in die richtige Form zu bringen.

Abschließend danke ich allen Personen, die mich während der gesamten Zeit meines Studiums auf vielfältigste Art und Weise unterstützt haben und es so zu einem unvergeßlichen Lebensabschnitt werden ließen.

Rostock, Deutschland im Juni 2001

Andreas Olsson

Abstract

The growth of the Internet with new digital services like VoIP coming up, has gathered a lot of people around the world to communicate in a new way. With the growth and dependence of the new medium, it also makes the private networks into a valuable target in form of different aspects.

The goal of this final thesis is to show how VoIP can be used in a secure way from a LAN to the Internet. Special interest is on how a firewall manages the H.323 call setup and NAT without exposing the internal network to any threats.

Some tests were also made at the university of Rostock, institute of telecommunications and information electronics. They comprised an internal LAN, connected over a firewall gateway to the Internet. On the firewall, different programs were tested and configured to show their support for the H.323 protocol and at the same time fully protect the internal LAN from any security flaws.

Inhaltsverzeichnis

VORWORT I

ABSTRACT II

INHALTSVERZEICHNIS III

ABBILDUNGSVERZEICHNIS VI

TABELLENVERZEICHNIS VIII

ABKÜRZUNGVERZEICHNIS IX

1 EINLEITUNG 1

2 FIREWALL ALS EIN SCHUTZ VOR DEM INTERNET 3

2.1 Paketfilter 5

2.1.1 Statische Paketfilter 5

2.1.2 Dynamische Paketfilter 7

2.2 Applikationsfilter 8

2.3 Firewall-Architekturen 9

2.3.1 Einsatz eines Paketfilters 10

2.3.2 Einsatz eines Applikationsfilters 10

2.3.3 Screened Subnet mit Single Homed Application Gateway 11

2.3.4 Screened Subnet mit Dual Homed Application Gateway 12

3 VOICE OVER IP – VOIP 14

3.1 Qualitätsprobleme 16

3.2 Sicherheitsanordnungen für VoIP 17

4 ÜBERTRAGUNGSSTANDARDS FÜR VOIP 19

4.1 Die H.323 Empfehlung 19
4.1.1 Das Protokoll Stack 20
4.1.2 H.323 Architektur 21
4.1.2.1 H.323-Terminal 22
4.1.2.2 Gateway Eigenschaften 26
4.1.2.3 Gatekeeper Eigenschaften 26
4.1.2.4 MCU - Mulitpoint Control Unit 28
4.1.3 IP-Netzwerke und Multimediakonferenzen 29
4.1.3.1 UDP- und TCP-Ports 30
4.1.3.2 Q.931 und H.245 Verbindungsaufbau 30
4.1.4 Adressierungen 32
4.1.4.1 Netzwerkadressen 32
4.1.4.2 TSAP-Identifier 32
4.1.4.3 Aliasadressen 33
4.1.5 Call Signalling Prozeduren 33
4.1.5.1 Phase A - Call Setup 33
4.1.5.2 Phase B – Initial Communication and Capability Exchange 35
4.1.5.3 Phase C – Establishment of Audio Visual Communication 35
4.1.5.4 Phase D – Call Services 36
4.1.5.5 Phase E – Call Termination 36

4.2 H.323 und die verschiedenen Firewalltechnologien 36
4.2.1 Schwierigkeiten innerhalb des Zusammenwirkens von H.323 und Firewalls 36
4.2.2 H.323 und die verschiedenen Firewalltechnologien 38
4.2.3 Die Kontrollfunktionen eines H.323-Proxy 40
4.2.3.1 H.225 Call Control Channel und H.323-Proxy 40
4.2.3.2 H.245-Kontrollkanal und H.323-Proxy 40

4.3 Die SIP Empfehlung 42
4.3.1 SIP Einleitung 42
4.3.2 SIP-Architektur 43
4.3.3 SIP Call Setup 45

4.4 SIP und die verschiedenen Firewalltechnologien 47
4.4.1 Implementierung auf einem ALG 48
4.4.2 SIP über SSL/TLS 49

4.5 Vergleich von H.323 und SIP 50

5 FIREWALLPRODUKTE 54

5.1 Checkpoint Firewall-1 54

5.2 Axent Raptor 6.5 58

5.3 NAI Gauntlet 58

5.4 Linux Netfilter 59

5.5 Cisco Systems 59

5.5.1 Cisco Secure PIX Firewall 59

5.5.2 Cisco Adaptive Security Algorithm 60

5.5.3 Cisco Multimedia Conference Manager 60

5.5.4 Cisco Routern und IOS Betriebsystem 61

5.6 Zusammenfassung der Firewallprodukte 62

6 BEISPIELIMPLEMENTATION 64

6.1 Das Comlab-Netzwerk 64

6.2 Linux Firewall Testnetzwerk 65

6.3 Phonepatch 67

6.4 Linux Netfilter iptables 70

6.5 Checkpoint Firewall-1 Testnetzwerk 71

6.5.1 Konfiguration Firewall-1 74

6.5.2 Scanning der Firewall 76

6.6 Schlußfolgerung 77

7 ERGEBNISSE UND AUSBLICK 79

LITERATURVERZEICHNIS 82

ANHANG 1 86

Abbildungsverzeichnis

Abbildung 2.1: Firewall als Schutz für ein lokales Rechnernetzwerk 3
Abbildung 2.2: Einsatz der beiden Filtertypen im ISO/OSI Referenzmodell [POD00] 4
Abbildung 2.3: Typischer Firewall-Router mit Paketfilter [CB94] 5
Abbildung 2.4: Aufbau des IP-Headers [FUH98] 6
Abbildung 2.5(a): Aufbau des UDP-Headers [FUH98] 6
Abbildung 2.5(b): Aufbau des TCP-Headers [FUH98] 6
Abbildung 2.6: Funktionsweise eines Proxies [CER99] 9
Abbildung 2.7: Firewall mit einem Paketfilter 10
Abbildung 2.8(a): Single Homed Application Gateway 10
Abbildung 2.8(b): Dual Homed Application Gateway 11
Abbildung 2.9: Screened Subnet mit Single Homed Application Gateway 11
Abbildung 2.10: Screened Subnet mit Multi Homed Application Gateway 12
Abbildung 3.1: VoIP zusammen mit PSTN Netz 14
Abbildung 4.1: H.323-Protokoll Stack [DAT00] 20
Abbildung 4.2: H.323-Architektur [WAL00], [DAT00] 22
Abbildung 4.3: H.323-Terminal Ausstattung [H.323] 23
Abbildung 4.4: Direct Endpoint Call Signalling [H.323] 24
Abbildung 4.5: Gatekeeper Routed Call Signalling [H.323] 24
Abbildung 4.6: Zone aus Gatekeeper und Endpunkten [WAL00] 27
Abbildung 4.7: Verbindungsaufbau von Q.931 und H.245 [INT00] 31
Abbildung 4.8: Basic Call Setup zwischen Endpunkt 1 und Endpunkt 2 [H.323] 34
Abbildung 4.9: Call Setup über Gatekeeper [H.323] 35
Abbildung 4.10: Firewall als Gateway zum Internet 38
Abbildung 4.11: Q.931 Verbindungsaufbau über einen H.323-Proxy-Prozeß [INT00] 40
Abbildung 4.12: Verbindungsaufbau des H.245-Kontrollkanals über H.323-Proxy [INT00] 41
Abbildung 4.13: Call Setup zwischen UAC und UAS [CEN00] 45
Abbildung 4.14: Verbindungsaufbau zu einem UAS [CEN00] 46
Abbildung 4.15: SIP-Netzwerk mit einer SIP ALG [ROS00] 48
Abbildung 5.1: Struktur der Firewall-1 [STR99] 55
Abbildung 5.2: Verbindung zwischen internem Netz und Internet mit dem Security-Server 57

Abbildung 6.1: Netzaufbau im Labor für Kommunikationssysteme (April 2001) 64
Abbildung 6.2: Linux Testnetzwerk 65
Abbildung 6.3: Phonepatch in der Firewall 68
Abbildung 6.4: External Switchboard im Phonepatch 69
Abbildung 6.5: Testnetzwerk mit Checkpoint Firewall-1 72
Abbildung 6.6: Checkpoint Firewall-1 Policy Editor 75

Tabellenverzeichnis

Tabelle 3.1: Vergleich von Qualität und Komplexität von Audio Codecs [HAR00] 17
Tabelle 4.1: Funktionen eines Gatekeepers [H.323] 27
Tabelle 4.2: Optionale Funktionen eines Gatekeepers [H.323] 28
Tabelle 4.3: Von H.323 benutzte Ports [SHO00] 30
Tabelle 4.4: Vergleich von H.323 und SIP [DAL99] 53
Tabelle 5.1: Firewallprodukte für synchrone IP-Dienste 63

Abkürzungverzeichnis

A

ACELP	Alegbraic Code Excited Linear Prediction
ACF	Admission Confirmation
ACK	Acknowledge
ADPCM	Adaptive Differential Pulse Code Modulation
ALG	Application Level Gateway
ARJ	Admission Reject
ARQ	Admission Request
ASA	Adaptive Security Algorithm
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode

B

B-ISDN	Broadband-ISDN
BCF	Bandwidth Confirmation
BRJ	Bandwidth Reject
BRQ	Bandwidth Request

C

CBAC	Context Based Access Control
CS-ACELP	Conjugate Structure-Algebraic Code Excited Linear Prediction

D

DMZ	Demilitarised Zone
DNS	Domain Name Server
DoS	Denial of Service
DSL	Digital Subscriber Line

E

ETSI	European Telecommunication Standardization Union

F

FTP	File Transfer Protocol
FW	Firewall
FWD	Firewall Daemon
FWM	Firewall Module

G

GEMS	Global Enterprise Management System
GUI	Graphical User Interface
GSTN	General Switched Telephone Network

H

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure socket layer

I

I/O	Input/Output
ILS	Internet Locator Service
IP	Internet Protocol
iPBX	internet Private Branche Exchange
IPsec	Internet Protocol security
IPX	Internetwork Protocol Exchange
ISDN	Integrated Services Digital Network
ISO	International Standardization Organisation
ISP	Internet Service Provider
ITU	International Telecommunication Union

L

LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LD-CELP	Low Delay-Code Excited Linear Prediction

M

MC	Multipoint Controler

MCM	Multimedia Control Manager
MCU	Multipoint Controler Unit
MOS	Mean Opinion Score
MIPS	Millions Instructions Per Second
MP	Multipoint Prozessor
MP-MLQ	Multipulse-Maximum Likelihood Quantization

N

N-ISDN	Narrowband-ISDN
NAI	Network Associates
NAT	Network Address Translation

O

OSI	Open System Interconnection

P

PBX	Private Branch Exchange
PCM	Pulse Code Modulation
PER	Packet Encoding Rule
PIX	Private Internet Exchange
PLC	Power Line Communication
PSTN	Public Switched Telephone Network

Q

QoS	Quality of Service

R

RAM	Random Access Memory
RAS	Registration Admission Status
RMS	Raptor Management Console
RSVP	Resource Reservation Protocol
RT	Round Trip
RTCP	Remote Transfer Control Protocol
RTP	Remote Transfer Protocol

S

SCN	Switched Circuit Networks
SDH	Synchronous Digital Hierarchy
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SNAT	Source Network Address Translation
SNMPD	Simple Network Management Protocol Daemon
SSL	Secure Socket Layer
SYN	Synchronize

T

TCP	Transport Control Protocol
TIPHON	Telecommunications and Internet Protocol Harmonization Over Networks
TLS	Transport Layer Security
TSAP	Transport Layer Service Access Point

U

UAC	User Agent Control
UAS	User Agent Server
UDP	User Datagram Protocol
UTF-8	UCS Transformation Formats-8

V

VoIP	Voice over IP
VPN	Virtual Private Network

1 Einleitung

Das Wachstum des Internets und die Entwicklung neuer Dienstleistungen bieten viele neue Möglichkeiten, bergen aber auch neue Schwierigkeiten, insbesondere für Unternehmen.

VoIP stellt mit der richtigen Ausstattung ein neues Kommunikationsmedium dar, das es ermöglicht, Sprach- und Datenübertragung im IP-Netz zu integrieren. Das paketvermittelte VoIP empfiehlt sich besonders für die privaten Netze eines Unternehmens, das die hohen Anforderungen an die Bandbreite gewährleisten kann. In diesem Fall ist VoIP eine kostengünstige und vielseitige Kommunikationslösung.

Der Nachteil von VoIP ist das sehr komplexe Protokoll, und die Verwendung von vielen dynamischen und hohen UDP-Portnummern, die erst beim Verbindungsaufbau bestimmt werden. Aus Sicht vieler Experten ist die einzige Lösung dieses Problems, alle hohen UDP-Portnummern in der Firewall zu öffnen. Dies kann allerdings aus Sicherheitsaspekten des internen LANs nicht als optimal angesehen werden.

Gegenstand dieser Diplomarbeit ist es, herauszufinden, welche Probleme besonders beim Verbindungsaufbau mit VoIP auftreten können und welche Firewalltypen und Firewallprodukte sich besser für einen Einsatz im internen LAN eignen. Sie werden von dem H.323-Protokoll unterstützt und bieten gleichzeitig völligen Schutz.

Um die theoretische Betrachtung auch praktisch erproben zu können, wurden im Labor für Kommunikationssysteme (Comlab) der Universität Rostock zwei Testnetzwerke aufgebaut. Diese bestanden aus einem Client in einem internen LAN, einem externen Client im Comlab und einem Firewallrechner mit Vermittlungsfunktion für die durchgehenden Datenpakete zwischen dem internen LAN und dem externen Comlab.

Auf dem Firewallrechner mit Intel x86-Plattform, wurde die Linux Netfilter Firewall konfiguriert und getestet und außerdem die Firewall-1 Software der Firma Checkpoint auf einer SUN Sparc-Plattform implementiert.

Um die Möglichkeit der Kommunikation nachzuweisen, wurde mit der Microsoft Netmeeting Applikation über das H.323-Protokoll eine Verbindung zwischen dem internen LAN und dem Comlab aufgebaut.

Bezüglich solcher Sicherheitsfaktoren wie z.B. Schutz vor internen Angriffe gibt es bei den vorgestellten Lösungen Einschränkungen. Dies gilt besonders für den vollständigen Schutz des internen LANs, denn in der Realität würde nur eine Firewall nicht ausreichen.

2 Firewall als ein Schutz vor dem Internet

Als das Internet 1969 startete, war das primäre Ziel ein elektronischer Informationsaustausch, der Sicherheitsaspekt wurde vernachlässigt. Heute ist das Internet ein weltumspannendes, elektronisches Netzwerk mit Millionen von Anwendern, das man zu Beginn noch nicht absehen konnte. Mit so vielen Benutzern und verschiedenen Anwendungsschwerpunkten, wird auch die Frage der Sicherheit von Datenübertragung als ein wichtiger Aspekt angesehen.

Mit einer Firewall oder einem sogenannten Gateway, das normalerweise als letztes Gerät in ein lokales Netz eingesetzt wird, wird das lokale Netz vom Internet getrennt und zum Teil geschützt. Es ist sowohl mit Software, als auch mit Hardware möglich, eine Firewall zu realisieren. Firewalls können und werden meistens von Unternehmen und ISP benutzt. Dabei sollten aber möglichst viele Kommunikationsprotokolle mit dem Internet kompatibel sein. Abbildung 2.1 zeigt, wo eine Firewall eingesetzt werden kann.

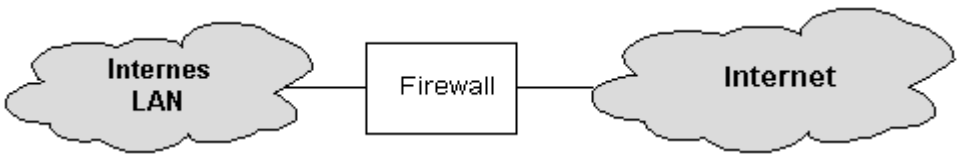

Abbildung 2.1: Firewall als Schutz für ein lokales Rechnernetzwerk

Wie Abbildung. 2.1 zeigt, soll die Firewall das letzte Gerät eines lokales Netzwerkes sein, das mit dem Internet in Kontakt steht. Die Firewall wird auch häufig als **Bastion Host** [CB94] bezeichnet. Bastion Host kann entweder ein **Paketfilter** oder ein **Applikationsfilter** sein. Die beiden unterschiedlichen Filtertypen einer Firewall, werden in den Kapiteln 2.1 und 2.2 näher beschrieben. Die Firewall stellt eine Form des Internetschutzes dar. Damit ist aber kein vollständiger Schutz vor dem Internet, mit seinen Angriffen, wie beispielsweise einer Attacke von Viren und vom internen, eigenen Netzwerk gegeben. Eine Firewall muß sich zusätzlich selbst gegen Angriffe schützen.

Für ein lokales Netzwerk bedeutet eine Firewall Schutz vor Unbefugten und vor Angriffen aus dem Internet. Weiter besitzt die Firewall eine Steuer- und Überwachungsfunktion über das Passieren von Daten, die erlaubt oder nicht erlaubt sind. Bei einer Firewall unterscheidet man zwei Filtertypen:

- Paketfilter
- Applikationsfilter (Proxy)

Der Paketfilter kann in zwei Untergruppen eingeteilt werden, dem **statischen** und dem **dynamischen** Paketfilter. Den Applikationsfilter unterscheidet man in **Application Level Proxies** und **Circuit Level Proxies.**

Abbildung 2.2 zeigt die Position der beiden Filtertypen in dem international standardisierten ISO/OSI 7- Schichten-Referenzmodell [ISO00].

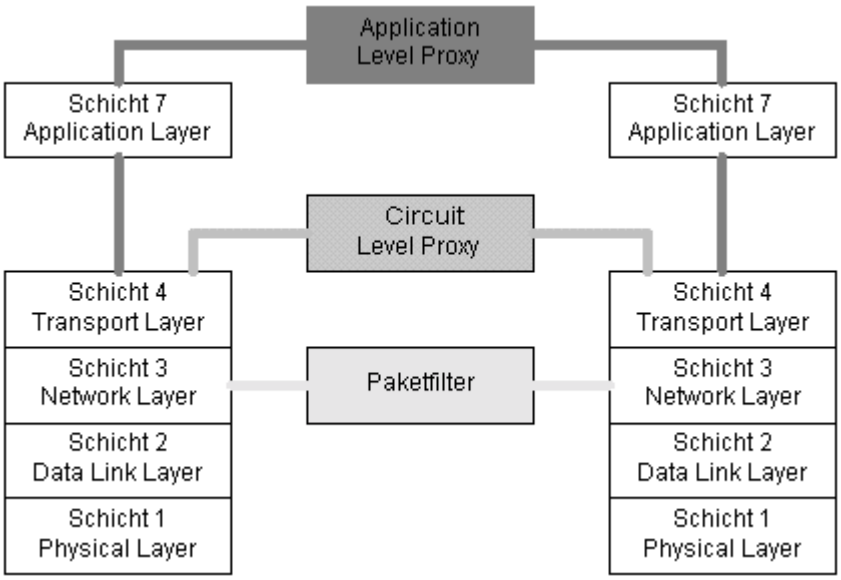

Abbildung 2.2: Einsatz der beiden Filtertypen im ISO/OSI Referenzmodell [POD00]

2.1 Paketfilter

2.1.1 Statische Paketfilter

Ein Paketfilter ist meistens in einem Router eingebaut, kann aber auch als eigenständiges Gerät funktionieren. Seine Hauptaufgabe besteht darin, die IP-Pakete zu kontrollieren.

Ein Router besteht mindestens aus zwei Netzwerkkarten oder Anschlüssen wie in Abbildung 2.3 gezeigt wird. Der Paketfilter kann sowohl an Punkt (a) oder Punkt (b), als auch an beiden Punkten installiert werden. Hier werden die Pakete als Allow- oder Deny Daten nach bestimmten Regeln im Paketfilter eingeteilt. Das heißt, daß die bei einem Paketfilter angewendeten Regeln als Erlaubnis- oder Verbotsregel definiert werden können.

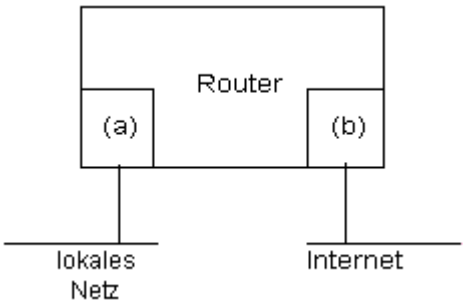

Abbildung 2.3: Typischer Firewall-Router mit Paketfilter [CB94]

Am häufigsten werden IP-Addresse und Portnummer zwischen Sender und Empfänger geprüft. Ob die IP-Pakete durch den Paketfilter passieren dürfen oder nicht, ist von den bestimmten Allow/Deny Regeln abhängig. Im Normalfall braucht man nur zwei Regeln für eine Verbindung in den Paketfilter eintragen, eine für Datenpakete von der Quelle eines Dienstes zum Ziel und eine zweite für dessen Antwortpakete.

Die Information, die ein Paketfilter bei seiner Entscheidung über die Art der Behandlung eines IP-Paket Headers benötigt, ist in Abbildung 2.4 auf grauem Untergrund dargestellt.

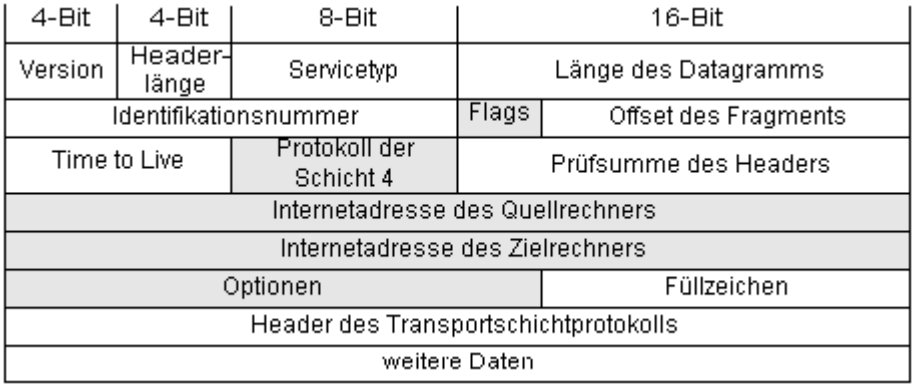

Abbildung 2.4: Aufbau des IP-Headers [FUH98]

In der Netzwerkschicht des ISO/OSI Referenzmodells, werden aus dem IP-Header von den Allow/Deny Listen die Adresse des Quell-Rechners und die des Zielrechners ausgewertet. Das Feld „Protokoll der Schicht 4“ in Abbildung 2.4 sagt aus, welches Protokoll der Transportschicht das IP-Paket erzeugt hat (meistens TCP oder UDP). Die Flags geben unter anderem Auskunft über eine mögliche Fragmentierung des Datenpakets.

In der Transportschicht, in der die UDP- und TCP-Protokolle liegen, werden die Portnummern von Quelle und Ziel kontrolliert. Hiermit kann man Dienste und Services kontrollieren und überwachen. Die Abbildungen 2.5(a) und 2.5(b) zeigen die beiden Header in der Transportschicht und deren Entscheidung über die Behandlung des IP-Paket Headers, die auf grauem Grund dargestellt sind.

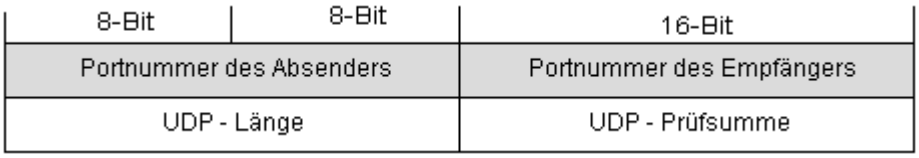

Abbildung 2.5(a): Aufbau des UDP-Headers [FUH98]

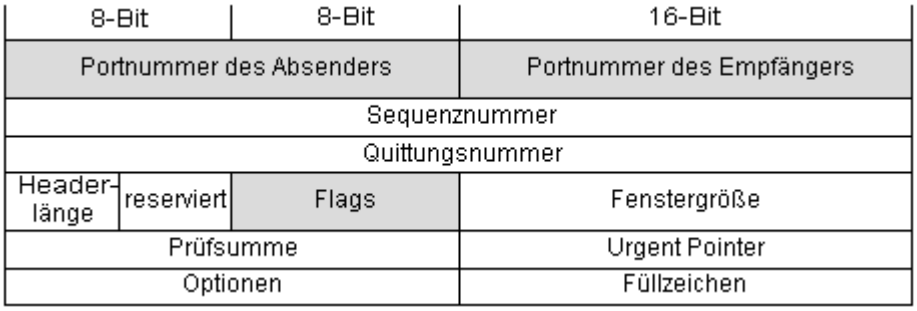

Abbildung 2.5(b): Aufbau des TCP-Headers [FUH98]

Mit dem TCP-Protokoll werden die FLAG-Bits geprüft. Dort informiert sich der Paketfilter, was für ein Typ von Verbindung es ist, beispielsweise ob das SYN-Bit, oder das ACK-Bit gesetzt ist, oder ob beide Bits gesetzt sind. Von diesen FLAG-Bits erhält man Information, ob es sich um einen Verbindungsaufbau oder eine Antwort handelt.

Für Anwendungen, die auf UDP basieren, sind statische Paketfilter nicht geeignet, weil ein UDP-Header keine FLAGS im Header wie das TCP-Header besitzt. Deswegen kann ein Paketfilter nicht analysieren, um welche Datenform es sich handelt. „Der Paketfilter kann beispielsweise nicht erkennen, ob es sich bei einem DNS-Paket um die Antwort auf eine Abfrage eines DNS-Servers im Internet handelt, was erlaubt wäre, oder um eine Abfrage eines DNS-Servers im zu schützenden Netz, was nicht erlaubt sein sollte.“ [FUH98] Dies bedeutet, daß die UDP-Protokolle schlecht mit dem Paketfilter zusammenarbeiten. Hier steht zur Wahl, entweder das UDP ganz zu eliminieren oder alle Ports über 1023 öffnen zu lassen.

2.1.2 Dynamische Paketfilter

Ein Beitrag zur Reduzierung des UDP-Problems, bietet der sogenannte dynamische Paketfilter. Dieser Filter besitzt die gleichen Eigenschaften wie ein statischer Paketfilter, nur daß dieser den Vorteil hat, daß er die Quell/Ziel-Adressen und die Quell/Ziel-Portnummern eines UDP-Pakets im Kernel für eine gewisse Zeitspanne speichern kann. Wenn ein passendes UDP-Paket mit der richtigen Quell/Ziel-Adresse in das Zeitfenster gelangt, bedeutet das, daß es zugelassen wird.

Eine Weiterentwicklung des dynamischen Paketfilters ist der **zustandsabhängige Paketfilter**. Dieser Paketfilter verwendet nicht nur die Netzwerk- und die Transportschicht zum Filtern, sondern betrachtet die Pakete auch in Abhängigkeit vom Zustand, den die FLAG-Bits darstellen, welche Verbindungsaufbau, Übertragung und Verbindungsabbau beschreiben.

Der zustandsabhängige Paketfilter (auch **Stateful Inspection Filter**) wurde von der Firma Checkpoint [Che00] entwickelt. Checkpoint ist ein marktführender Hersteller von Firewallprodukten, wie beispielsweise dem Firewall-1 Modell, dessen Testergebnisse in Kapitel 6 aufgeführt sind. Die Firewall-1, die UDP unterstützt, ist mit dem H.323-Protokoll für IP-Telefonie kompatibel.

Zustandsabhängige Paketfilter verwalten die Verbindungen in Form einer Tabelle, in der folgende Einträge gespeichert sind:

- Nummer zur Identifizierung der Verbindung
- Zustand der Verbindung, (z.B. Aufbau, Datenübertragung, Abbau)
- Quell-Adresse des ersten Pakets
- Ziel-Adresse des ersten Pakets
- Interface, durch welches das erste Paket gekommen und durch welches es verschickt worden ist

Anhand dieser Daten kann der Paketfilter entscheiden, welche nachfolgenden Pakete zu welcher Verbindung gehören.

[CB94], [CER99], [CHE00], [FUH98], [POD00]

2.2 Applikationsfilter

Der Applikationsfilter arbeitet zwischen Schicht 4 und Schicht 7, wie in Abbildung 2.2 dargestellt ist. Hierzu gehören z.B. Authentisierungsinformationen oder spezifische Befehle der Anwendungsschicht. Ein Applikationsfilter verwendet zur Filterung in der Regel sogenannte Proxy-Prozesse.

Hierbei unterscheidet man zwischen **Application Level Proxy**, das ein bestimmtes Protokoll kontrolliert, (z.B. von einem HTTP-Proxy) und **Circuit Level Proxy**, das als ein generisches Proxy für alle Protokolle eingesetzt werden kann, aber keine Möglichkeit hat, protokollspezifisch zu filtern oder zu protokollieren.

Wie ein typischer Applikationsfilter funktioniert, ist in Abbildung 2.6 dargestellt. Der Applikationsfilter wird häufig **Proxy-Server** genannt, wenn mehrere Proxy-Prozesse am gleichen Gerät installiert werden können.

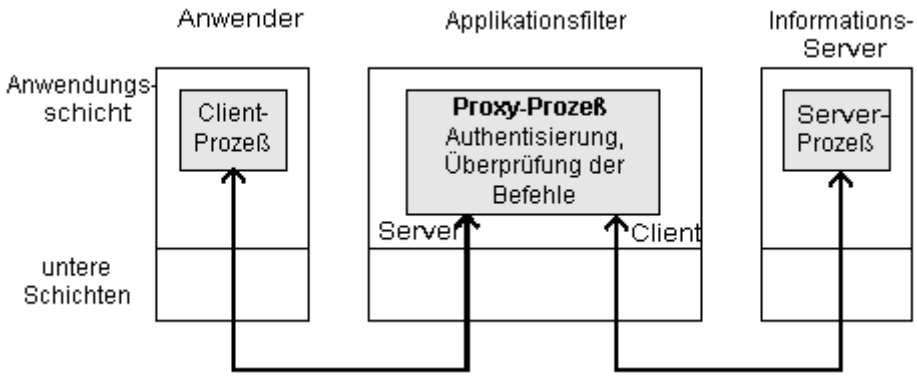

Abbildung 2.6: Funktionsweise eines Proxies [CER99]

Der Anwender muß zunächst eine Verbindung zum Proxy-Prozeß aufbauen und diesem die notwendigen Informationen für eine korrekte Authentisierung und die Adresse des eigentlichen Ziels mitteilen.

Häufig wird in Routern und Applikationsfiltern, eine NAT (Network Address Translation) in Applikationsfiltern eingesetzt. Die IP-Adresse des Anwenders wird im Applikationsfilter zwischen Server und Client ausgetauscht, also von einer privaten zu einer öffentlichen, vermittelbaren IP-Adresse. Einkommende Datenpakete werden von öffentlichen IP-Adressen zu privaten IP-Adressen übersetzt. Dieses wird zum Schutz des eigenen internen LANs durchgeführt und um die öffentlichen IP-Addressen effizienter zu nutzen.

[CB94], [CER99]

2.3 Firewall-Architekturen

Aus der Kombination, Paketfilter und Applikationsfilter entstehen verschiedene Möglichkeiten, eine komplexe Firewall aufzubauen. In den folgenden Kapiteln werden einige Konfigurationsmöglichkeiten der Firewallkomponeneten vorgestellt.

2.3.1 Einsatz eines Paketfilters

Die Abbildung 2.7 zeigt die einfachste Form eines Firewall-Systems, das meistens schon im Router eingebaut ist. Der Paketfilter trennt das lokale Netz vom Internet, der Schutz des Paketfilters ist dabei gering.

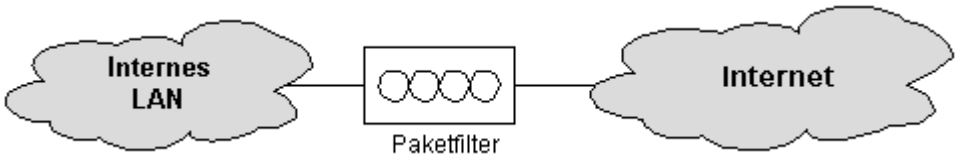

Abbildung 2.7: Firewall mit einem Paketfilter

2.3.2 Einsatz eines Applikationsfilters

Es gibt zwei Typen von Applikationsfiltern (Application Level Proxy & Circuit Level Proxy), die auf einem Gateway (Bastion Host) eingesetzt werden. Realisierungsmöglichkeiten sind:

a) Single Homed Application Gateway
b) Dual Homed Application Gateway

Sie unterscheiden sich durch die Anzahl der verwendeten Netzschnittstellen. (Abbildung 2.8(a) und 2.8(b)).

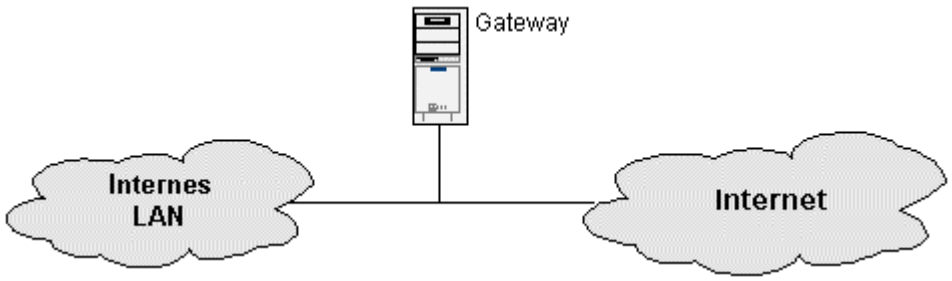

Abbildung 2.8(a): Single Homed Application Gateway

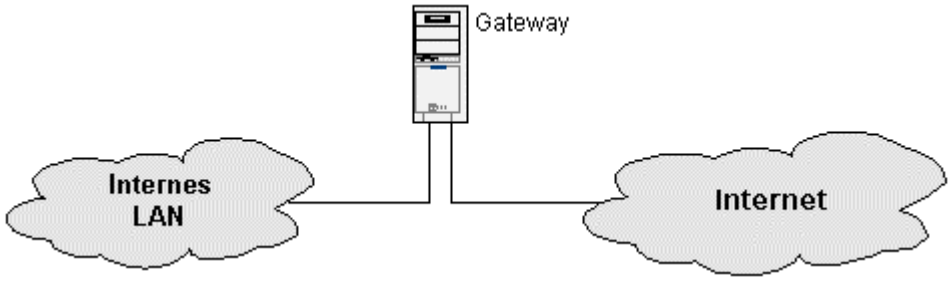

Abbildung 2.8(b): Dual Homed Application Gateway

Im Single Homed Application Gateway mit einer Netzschnittstelle werden alle eingehenden und ausgehenden Datenpakete über eine identische Netzschnittstelle transportiert. Mit dem Dual Homed Application Gateway, welcher zwei Netzschnittstellen besitzt, wird sichergestellt, daß eine physikalische Trennung des lokales Netzes vom Internet gegeben ist.

2.3.3 Screened Subnet mit Single Homed Application Gateway

Ein Screened Subnet ist ein geschütztes Teilnetz zwischen dem internen Netz und dem Internet. Dieses zusätzliche Netz wird häufig als DMZ oder DeMilitarised Zone benannt. Ein Screened Subnet mit Single Homed Application Gateway ist in Abbildung 2.9 dargestellt.

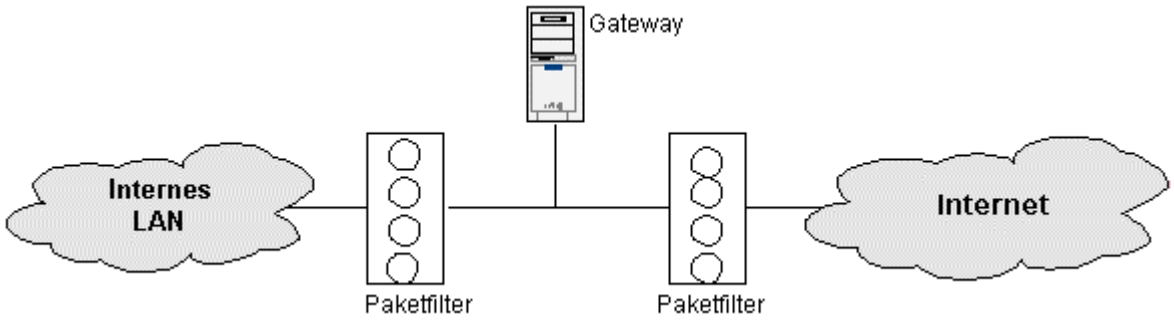

Abbildung 2.9: Screened Subnet mit Single Homed Application Gateway

Der äußere Paketfilter hat die Aufgabe, eine Verbindung zum Internet herzustellen. Er schützt das Screened Subnet und das lokale Netz vor Angriffen aus dem Internet und muß so eingestellt sein, daß er die Datenpakete zum Applikationsfilter durchläßt.

Durch die Proxy-Prozesse in dem Applikationsfilter wird die Filterung auf der Anwendungsschicht durchgeführt. Der zweite Paketfilter schützt das lokale Netz auch vor Angriffen aus dem Internet und dem Screened Subnet. Hier wird der Paketfilter so eingestellt, daß die Datenpakete vom lokalen Netz zum Proxy-Prozeß im Applikationsfilter durchgelassen werden.

Dienste, bei denen keine Proxies eingesetzt werden, können direkt über den äußeren Paketfilter gesendet werden. Diese Möglichkeit vereinfacht die Implementierung neuer Protokolle, die keinen Proxy besitzen oder nicht vom Proxy unterstützt werden.

Der Vorteil der Anordnung mit Screened Subnet gegenüber dem Einsatz von nur einem Paketfilter oder Applikationsfilter ist, daß alle Möglichkeiten zum Filtern angewendet werden können, aber nicht müssen.

2.3.4 Screened Subnet mit Dual Homed Application Gateway

Diese Kombination schützt ein lokales Netz gegen Angriffe jeglicher Art am besten. Es unterscheidet sich von Single Homed Screened Subnet dadurch, daß der Applikationsfilter zwei oder mehrere Netzwerkschnittstellen hat.

Eine dieser Schnittstellen bildet zusammen mit dem äußeren Paketfilter ein äußeres Screened Subnet und die zweite Schnittstelle zusammen mit dem inneren Paketfilter ein inneres Screened Subnet. Diese Anordnung ist in Abbildung 2.10 dargestellt.

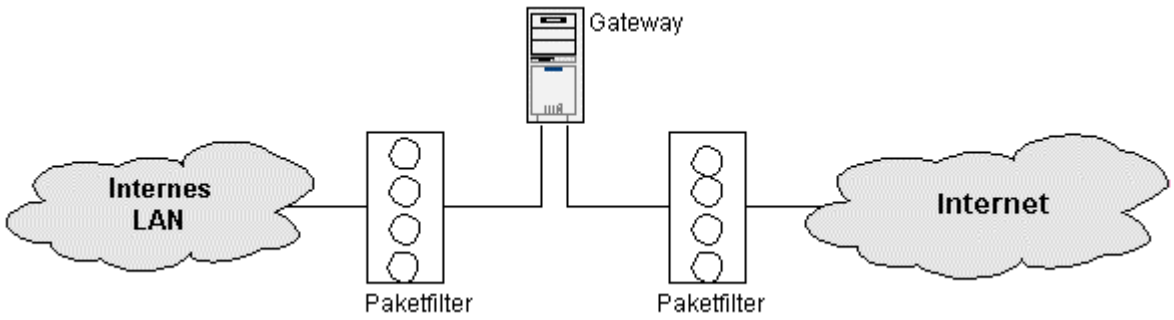

Abbildung 2.10: Screened Subnet mit Multi Homed Application Gateway

Möglich wäre eine dritte Schnittstelle, die ein weiteres Screened Subnet darstellt, z.B. für Informationsserver. Hier werden alle Verbindungen durch einen Proxy-Prozeß auf dem Applikationsfilter durchgeführt, so daß alle Vorteile der Filterung auf der Anwendungsschicht benutzt werden. Dabei kann es allerdings auch ein Nachteil sein, daß es für die Verbindung keine Möglichkeit gibt, um am Applikationsfilter vorbei in das Internet zu kommen. Diese Eigenschaft von Multi Homed Application Gateway verringert die Flexibilität gegenüber neuen Diensten, für die keine Application Level Proxies existieren.

[CER99], [FUH98]

3 Voice Over IP – VoIP

Bisher wurde und wird das Internet zum größten Teil für asynchronen Datenverkehr genutzt (z.B. Email, FTP, HTTP). Mit dem Wachstum des Internets und der Entwicklung neuer multimedialer Dienste wird der synchrone Datenverkehr wie VoIP im Internet immer häufiger.

Damit synchroner und asynchroner paketbasierter Datenverkehr über IP oder ATM zusammenpassen, hat die *International Telecommunicaton Union*, ITU [ITU00] den Vorschlag H.323 **Packet-Based Multimedia Communication Systems** entwickelt. Auf diese stützt sich heute zum größten Teil die Implementierung von VoIP. Das H.323-Protokoll wird in Kapitel 4.1 näher beschrieben.

In einem LAN ist es heute kein größeres Problem mehr, VoIP zu nutzen, weil die Router die Endpunkt IP-Adressen kennen und die Sicherheitsmechanismen wie jene der Firewalls nicht so hoch sind. Über das Internet ist es schwerer VoIP zu benutzen, weil mit einem höheren Sicherheitsgrad und in vielen Fällen mit einer Addressenübersetzung wie NAT gearbeitet wird. Für die implementierung des VoIP Netzes gibt es bereits mehrere Lösungen von verschiedenen Anbietern. In Abbildung 3.1 ist dargestellt, wie die Integration zwischen VoIP und PSTN aussehen könnte.

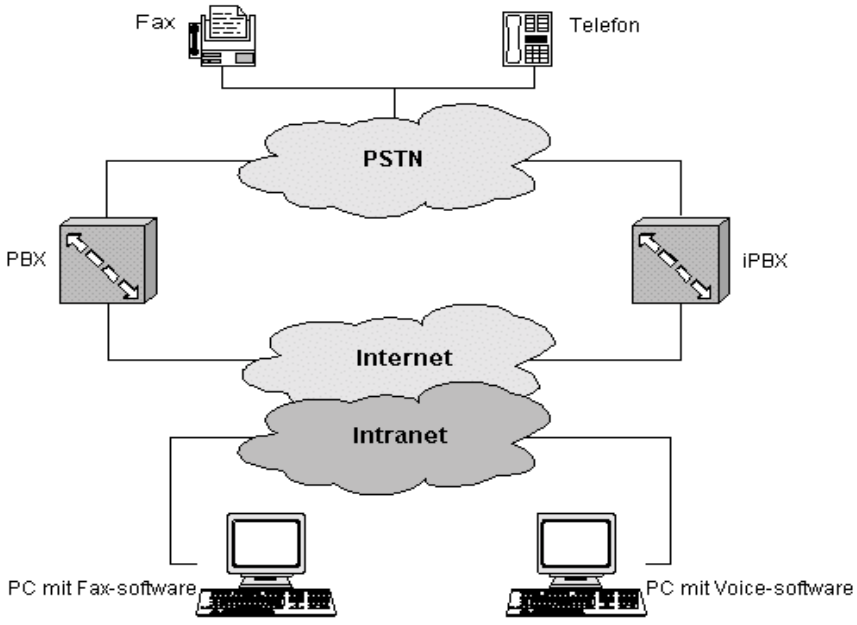

Abbildung 3.1: VoIP zusammen mit PSTN Netz

Als Komplement zu einem PBX wird ein PSTN-Gateway oder ein sogenannter IP-basierter PBX (iPBX) eingeführt. iPBX hat die gleichen Funktionen wie ein PBX, aber es kann zusätzlich Multimediarechner zu PSTN vermitteln.

Die Vorteile von paketbasiertem Telefonieren sind im besonderen die geringen Preise und die neuen Dienste für Kommunikation. Was im Einzelnen bedeutet, daß das Telefonieren über längere Strecken z.B. zwischen verschiedenen Ländern billiger geworden ist und die neuen Dienste der Kommunikation sind beispielsweise Video, Chat oder Whiteboard. Auch mit VoIP würde eine ähnliche Standardisierung gelten, die die Verbindung zwischen verschiedenen Ländern vereinfacht.

Um eine Kompatibilität von VoIP mit dem Telefonnetz zu erreichen, hat das *European Telecommunications Standards Institute* (ETSI) die Spezialgruppe *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) gebildet. Diese Spezialgruppe arbeitet an Standardisierungen der Sprachkommunikation über das Internet.

Das Ziel ist die Standardisierung von Gateways, die die Kommunikation zwischen IP-Netzen und Switched Circuit Networks (SCN) beispielsweise PSTN/ISDN möglich machen. TIPHON definiert vier Szenarien der Internet Telefonie [WAL00]:

Phase 1: Verbindungsaufbau von einem H.323-Terminal zum Telefonteilnehmer
Phase 2: Verbindungsaufbau von einem Telefonteilnehmer zum H.323-Terminal
Phase 3: Verbindung von Teilnehmern am Telefonnetz über IP-Netze
Phase 4: Verbindung zwischen H.323-Terminals über das Telefonnetz

VoIP wird am häufigsten von Unternehmen mit Intranet und Hochgeschwindigkeitsanschluß zum Internet genutzt. Die Übertragung von Sprach-Datenpaketen oder anderer synchroner, multimedialer Dienste über längere Strecken, stellt hohe Anforderungen an die Ausrüstung und ist noch nicht so einfach zu benutzen wie PSTN. VoIP muß benutzerfreundlicher werden und die Sprachqualität sollte der des PSTN gleichkommen, auch zwischen verschiedenen Netzwerken mit unterschiedlichen QoS.

[ITU00], [VOI98], [WAL00]

3.1 Qualitätsprobleme

Mean Opinion Score, MOS ist die benutzte Einheit zur Bewertung der Sprachqualität, die von 5 = excellent bis 1 = bad reicht. PSTN liegt ungefähr bei 4, bei VoIP ist die Note gerringer.

Typische Probleme der QoS Sprachqualität für synchrone Dienste über asynchrone Netze sind:

- **Paketverzögerung***:* Diese sollte so klein wie möglich sein. Sie beschreibt zwei verschiedene Typen der Verzögerung, der erste Typ stellt die Dauer der Übertragung von der Quelle zum Ziel dar. Die ITU-T G.114 Empfehlung besagt, daß die Zeitspanne von Quelle nach Ziel im Idealfall nicht 150 ms überschreiten darf. Der zweite Typ beschreibt die Zeitvariation in der Ankunft der Datenpakete beim Empfänger, was auch als Jitter bezeichnet wird. Hier muß eine Cache die Datenpakete beim Empfänger zwischenspeichern und danach diese in der richtigen Ordnung weiterleiten.

- **Bandbreite:** Eine garantierte Bandbreite ist in den IP-Netzen schwer zu bekommen. Diese ist von verschiedenen Umständen abhängig. Typisch für PCM-Systeme ist, daß man die G.711 Empfehlung mit 64Kbit/s Übertragungsrate benutzt. G711 ist weltweit standardisiert und wird für PSTN und ISDN benutzt. Für Sprachübertragung im Internet empfiehlt sich G.711 weniger, weil sie zu viel Bandbreite aufnimmt. Deswegen gibt es von der ITU andere Empfehlungen für Codecs synchroner Dienste im Internet, die sich besser eignen [ITU00]. Diese sind in Tabelle 3.1 dargestellt.

- **Packet loss:** In IP-Netzen gibt es keine Sicherheit, daß die Datenpakete wirklich übertragen und richtig geliefert werden. Im Falle eines Paketverlustes, muß der Sender die verlorenen Datenpakete noch einmal senden.

Um diese Probleme zu minimieren und die synchronen Dienste für das Internet zu verbessern, wird viel Engagement von Firmen und Organisationen benötigt. Typische Herangehensweisen für einen besseren QoS sind:

- **Höhere Übertragungsgeschwindigkeit:** Neue Übertragungsmedien, die das analoge Modem ersetzen, sind z.B. xDSL, PLC und Kabelmodem. Auch die Transportnetze wie ATM und SDH verringern die Paketverzögerung deutlich.

- **Schnellere Prozessoren:** Bei der Berechnung von Kodierung, Transport und Dekodierung für Audio- oder Video-Applikationen, spricht man von Millions of Instructions per Second (MIPS), die für die Kommunikation und den gebrauchten Speicher-RAM benötigt werden.

- **Effizientere Codecs:** Von der ITU werden die G-Serien entwickelt [ITU00], die unterschiedliche Sprachdigitalisierung und -kompression nutzen. Die Kompromisse, die es zwischen Sprachqualität, Prozessorleistung und Paketverzögerung gibt, sind in Tabelle 3.1 dargestellt.

Codec	Standard	Übertragungsrate Kbit/s	Sprachqualität (MOS)	Delay (ms)	Prozessorlast (MIPS)
PCM	G.711	64	4.3	0.125	0
ADPCM	G.721	32	4.1	0.125	6.5
ACELP/ MP-MLQ	G.723	5.3/6.4	4.1	70	25
LD-CELP	G.728	16	4.1	2	37.5
CS-ACELP	G.729	8	4.1	20	34
CS-ACELP	G.729a	8	3.4	20	17

Tabelle 3.1: Vergleich von Qualität und Komplexität von Audio Codecs [HAR00]

[HAR00], [ITU00], [WAL00]

3.2 Sicherheitsanordnungen für VoIP

Um eine Konferenz mit VoIP über das Internet abzuhalten, besteht die Forderung, daß die Konferenz vertraulich ist. Die Konferenz muß das gleiche Sicherheitsniveau wie im PSTN einhalten. Im PSTN weiß der Sender, wer der Empfänger ist und daß die gesendeten Informationen, die direkt zwischen zwei Endpunkten gesendet werden, nicht von ungehörigen Endpunkten zugänglich sind.

Bei der Datenübertragung über das ungeschützte Internet besteht die Gefahr, daß Unbefugte Daten abhören und verändern. Grundsätzlich gibt es keine abhörsicheren Leitungen, sondern lediglich Unterschiede im Aufwand des Abhörens.

Um die VoIP Datenpakete über das Internet mit Authentifikation, Schutz der Integrität und Verfügbarkeit allgemein sicher zu machen, ist das Transport Layer Security (TLS) und IPsec [POD00] entwickelt worden. IPsec ist als zusatz zu IPv4 implementiert. IPsec garantiert VPN und Tunneln der Datenpakete über das Internet. In IPv6 ist IPsec als Standard enthalten. Mit IPsec werden die Datenpakete und Header Dateien verschlüsselt, die nur von dem Empfänger mit dem richtigen Schlüssel aufgemacht werden können.

Um IPsec nutzen zu können, ist es erforderlich, daß Sender und Empfänger einigen, die gleichen Encryption/Decryption Applikationen benutzen.

Als Schutz für das LAN eines Unternehmens, wird häufig eine Firewall zwischen LAN und Internet eingesetzt, wie bereits in Kapitel 2 beschrieben wurde. Die Firewall schützt das LAN gegen ungehörigen Zugriff.

VoIP als ein synchroner Dienst ist für eine kooperation mit den "klassischen" Firewalls nicht geeignet. VoIP benutzt mehrere dynamische Ports, die während des Verbindungsaufbaus in einer Konferenz bestimmt werden. Zum Teil benutzt VoIP auch das UDP als verbindungsloses Übertragungsprotokoll, das keine FLAG-Bits beinhaltet. Deswegen ist die Bearbeitung der Datenpakete durch die Firewall erschwert.

Es gibt jedoch einige Firewalltechnologien, die die synchronen Dienste behandeln können. Diese öffnen die im Verbindungsaufbau besetzten Portnummern und können das UDP in einer Konferenz bearbeiten. Wenn die Konferenz beendet ist, werden die geöffneten Portnummern wieder geschlossen.

Auf die Problematik der Zusammenarbeit von Firewalls und VoIP wird in Kapitel 4.2 und 4.4 eingegangen.

[POD00]

4 Übertragungsstandards für VoIP

4.1 Die H.323 Empfehlung

H.323 ist der VoIP Standard für multimediale Dienste über ein paketbasiertes, lokales Netzwerk, wie z.B. TCP/IP, IPX, Fast Ethernet oder Token Ring-Topologien ohne die Garantie von QoS.

Voraussetzungen von H.323 sind [WAL00], [DAT00]:

- **Codec Standards:** Kompression- und Dekompression-Standards für Audio und Video garantieren Interoperabilität mit den Produkten anderer Hersteller.

- **Netzwerkunabhängigkeit:** H.323 unterstützt vielfältige Netzwerkarchitekturen und kann neu implementierte Netzwerkfunktionen z.B. Bandbreitenmanagement nutzen.

- **Plattform- und Applikationsunabhängigkeit:** H.323 ist nicht an eine Hardware bzw. ein Betriebssystem gebunden.

- **Multipointunterstützung** bildet eine flexible Grundlage für Multipointkonferenzen. Multipointunterstützung kann auch in anderen Komponenten enthalten sein.

- **Bandbreitenmanagement:** Der Network Manager kann die Anzahl der gleichzeitigen H.323-Verbindungen entsprechend der zugeteilten Bandbreite begrenzen.

- **Multicastunterstützung:** Multicastübertragung benutzt die Bandbreite effizienter, da alle Empfänger der Multicastgruppe, die gleichen Medienströme benutzen können.

- **Flexibilität:** Endpunkte mit unterschiedlichen Fähigkeiten (Audio, Video, Daten) können miteinander kommunizieren, z.B. kann ein Terminal, das nur Audio unterstützt, auch mit einem, das nur Video und/oder Daten unterstützt, kommunizieren.

- **Inter Network Conferencing** ermöglicht eine Verbindung von einem lokalen Netzwerk über verschiedene SCN.

4.1.1 Das Protokoll Stack

Das H.323-Protokoll ist die Bezeichnung für mehrere Unterprotokolle, die das VoIP möglich macht. Die H.323-Kommunikation kann als eine Mischung aus Audio-, Video-, Daten- und Kontrollsignalen betrachtet werden. Das H.323-Protokoll Stack, mit den verschiedenen, geltenden Protokollen, ist in Abbildung 4.1 dargestellt.

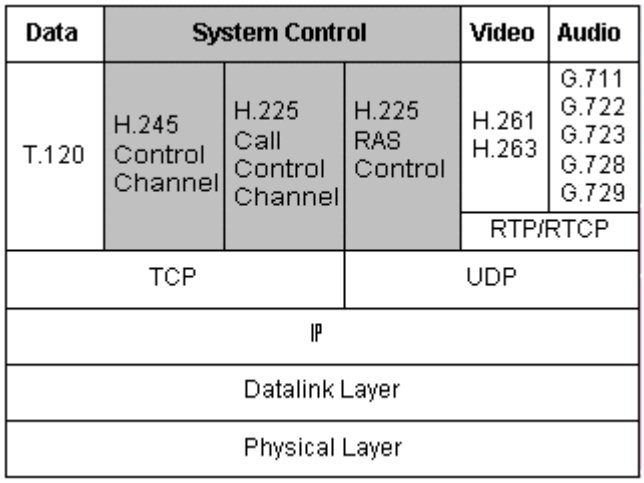

Abbildung 4.1: H.323-Protokoll Stack [DAT00]

Obligatorische Protokolle im H.323:

- **G.711** 64 Kbit/s PCM
- **H.225** Call Control Channel
- **H.245** Control Channel
- **RTP/RTCP** Real Time Protocol/Real Time Control Protocol

Der Audio Standard G.711 muß immer unterstützt werden. Andere Codecs sind optional. Wenn mehrere Codecs auszuwählen sind, wechseln die Endpunkte zwischen verschiedenen Codec Standards nach den geltenden Bedingungen, mit Unterstützung des H.245 Kontrollkanals.

Verschiedene Audio Codecs wurden in Kapitel 3 beschrieben, wo sich zeigte, daß G.711 mit 64 Kbit/s Übertragungsrate, im vergleich zu anderen Codecs viel Bandbreite benötigt, was sich im H.323 weniger empfiehlt.

G.723 mit niedrigerer Bitrate (Tabelle 3.1) hat sich als Empfehlenswert herausgestellt. Für Video sind H.261 und H.263 empfohlen. Datasupport wie z.B. Chat oder Whiteboard wird von T.120 unterstützt. Eine T.120-Verbindung wird von einem Teilnehmer mit dem H.245 Kontrolkanal initiiert. Sie ist in Kapitel 4.1.2.1 beschrieben.

Für die bidirektionale T.120-Verbindung wird die statische TCP-Portnummer 1503 oder eine dynamische Portnummer benutzt.

[DAT00], [H.323]

4.1.2 H.323 Architektur

Die H.323-Empfehlung setzt voraus, daß die Übertragung zwischen den Anwendern, mindestens über ein LAN stattfindet. Abbildung 4.2 stellt dar, daß das H.323-Netz in einem LAN mit verschiedenen SCN über den Gateway kommunizieren kann.

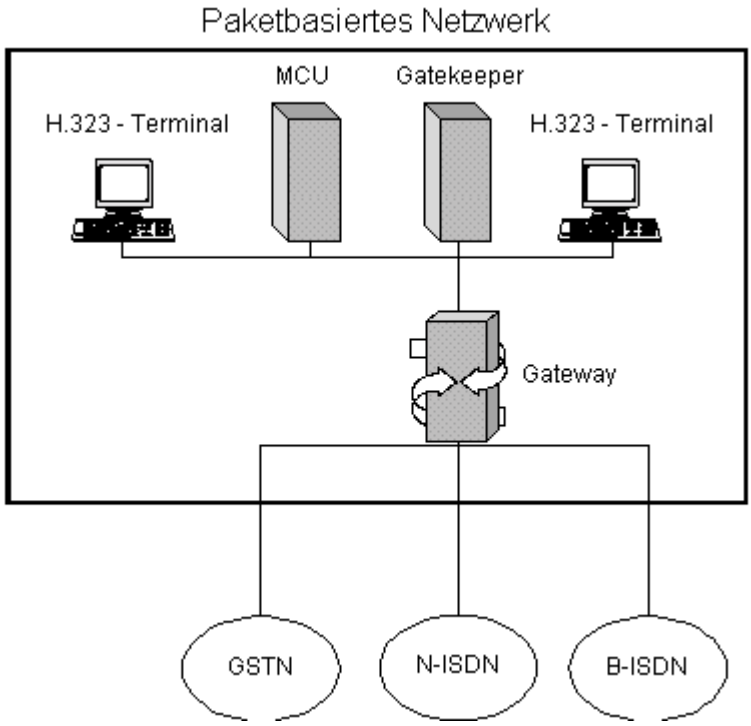

Abbildung 4.2: H.323-Architektur [WAL00], [DAT00]

Sowohl die Komponenten in Abbildung 4.2 als auch MC und MP, werden als **Entities** bezeichnet. Sie werden in den kommenden Unterkapiteln beschrieben. H.323-Terminal, Gateway oder Multi Control Unit (MCU), die für den Kommunikationsaufbau und -abbau zuständig sind, werden **Endpoints** genannt. Ein LAN das H.323 unterstützt, kann auch ohne Gateway funktionieren, dann aber nur zwischen H.323-Endpunkten.

4.1.2.1 H.323-Terminal

Ein H.323-Terminal stellt innerhalb eines LANs eine bidirektionale Echtzeitübertragung bereit, wie z.B. IP-Telefon oder Multimediarechner. Abbildung 4.3 stellt die H.323-Terminal Ausstattung dar.

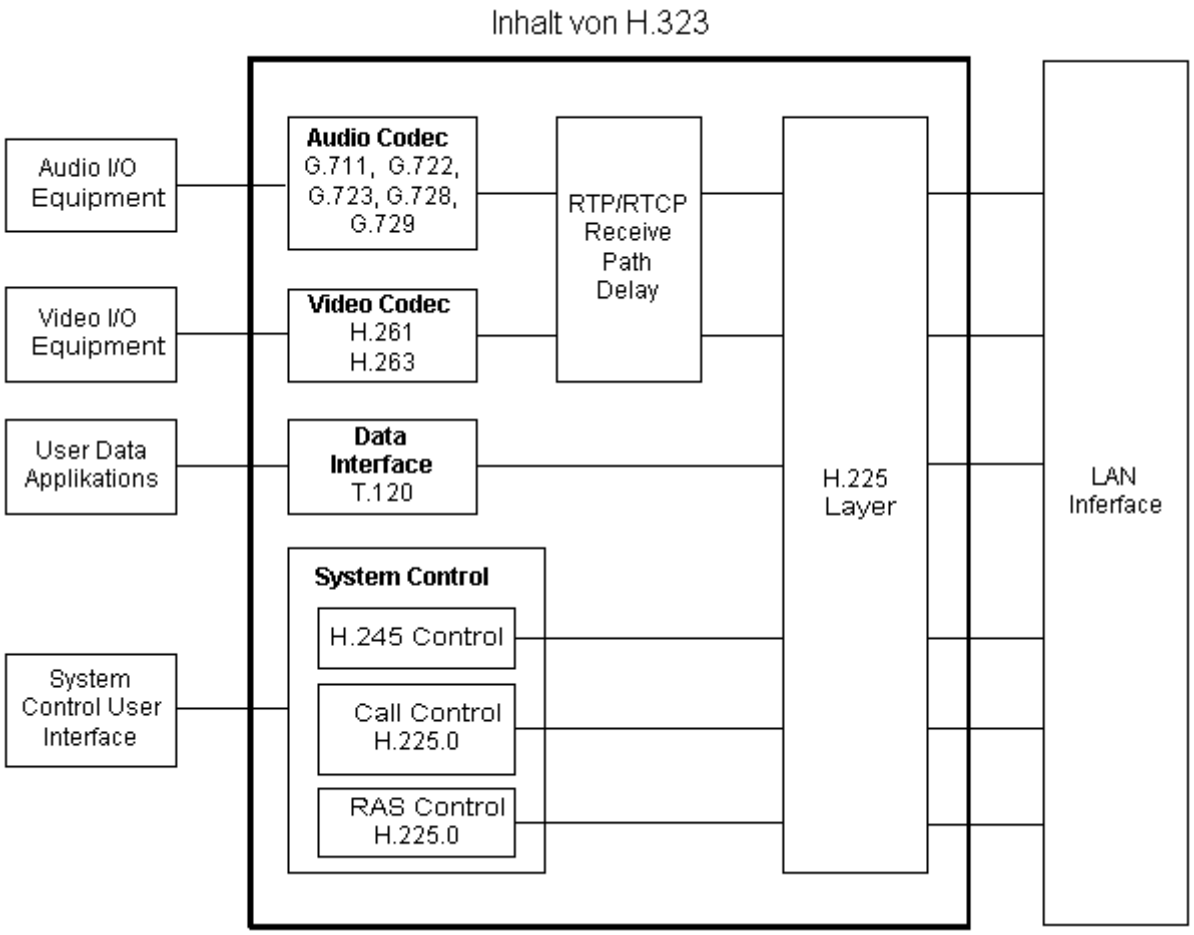

Abbildung 4.3: H.323-Terminal Ausstattung [H.323]

Jeder H.323-Terminal muß eine **System Control** Einheit, **H.225 Layer**, **LAN Interface, RTP/RTCP** und einen **Audio Codec** beinhalten.

Video, Data und die Unterstützung von MCU sind optional. Audio, Video, Data und Kontrollfunktionen werden durch die H.225-Schicht gesendet, die die Anpassung der Datenströme an die LAN-Schnittstelle übernimmt. Für die Unterstützung der Sprache im System Control eines H.323-Terminals dienen die folgenden Kontrollfunktionen:

- **H.225 Call Control Channel**
- **RAS Registration/Admission/Status**
- **H.245 Control Channel**

H.225 Call Control Channel: Der ISDN Q.931 Call Signalling Channel wird für die H.225 Call Control Nachrichten benutzt. Im LAN mit Gatekeeper wird der Nachrichtenaustausch zwischen Gatekeeper und Endpunkten mit dem H.225 RAS-Control ausgeführt. Im ersten Nachrichtenaustausch mit dem Q.931 Call Signalling Channel sendet der Gatekeeper die Set-up Nachricht entweder per Call Signalling direkt zum anderen Endpunkt wie in Abbildung 4.4, oder er läßt es durch den Gatekeeper vermitteln wie in Abbildung 4.5.

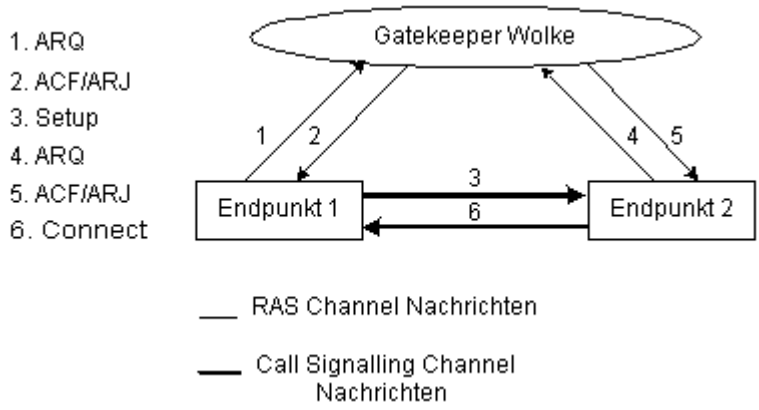

Abbildung 4.4: Direct Endpoint Call Signalling [H.323]

In der Gatekeeper Routed Call Signalling Methode sind die Q.931 Call Signalling Nachrichten zwischen Gatekeeper und Endpunkten vermittelt.

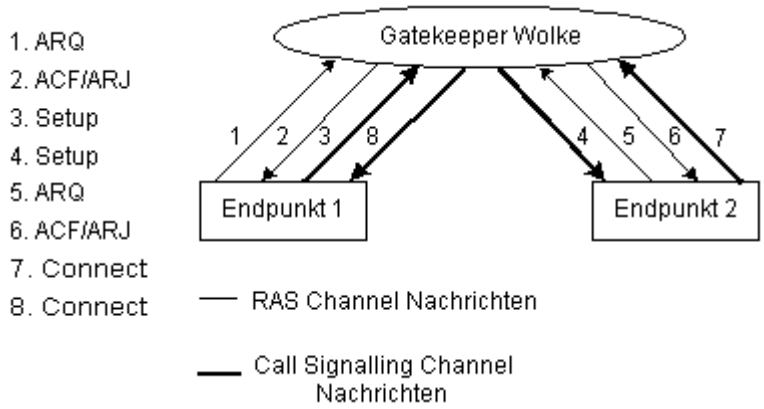

Abbildung 4.5: Gatekeeper Routed Call Signalling [H.323]

Die Gatekeeper Wolken in Abbildung 4.4 und Abbildung 4.5 bestehen aus einem oder mehreren Gatekeepern die, wenn nötig, miteinander kommunizieren können. Die Endpunkte sind entweder mit dem gleichen oder mit verschiedenen Gatekeepern verbunden.

Die Q.931 Call Signalling Nachrichten können auch direkt zwischen zwei Endpunkten mit der Call Signalling Transportadresse gesendet werden, ohne den Gatekeeper abzufragen. Voraussetzung für den Anrufer ist, daß die Transportadresse des Empfängers bekannt ist. Weitere Möglichkeiten zum Verbindungsaufbau sind in [H.323] beschrieben.

RAS Registration/Admission/Status: Wenn ein Gatekeeper vorhanden ist, wird die RAS Signalisierungsfunktion zwischen Endpunkt und Gatekeeper benutzt. In Netzwerken ohne Gatekeeper wird RAS nicht benutzt. Es benutzt die H.225-Nachrichten für Registrierung (Gatekeeper Discovery, Endpunktregistrierung), Zugangskontrolle anhand der Bandbreite, Bandbreitenänderung, Status und Verbindungsabbau zwischen Endpunkt und Gatekeeper.

H.245 Control Channel: Die Kontrollfunktionen benutzen den H.245-Kontrollkanal, der über eine TCP-Verbindung mit Ende zu Ende Steuermeldungen arbeitet. Ein H.245-Kontrollkanal kann zwischen zwei Terminals, Terminal und MCU bzw. Terminal und Gateway, aufgebaut werden. Für jede Verbindung die einen Endpunkt hat, existiert ein H.245-Kontrollkanal. Ein Endpunkt mit mehreren Verbindungen würde auch mehrere H.245-Kontrollkanäle haben. H.323-Endpunkte müssen die Prozeduren des folgenden H.245-Kontrollkanals unterstützen [H.323], [WAL00]:

- **Capability Exchange:** Die Empfehlung H.245 stellt für die Capability Exchange Prozeduren zur Aushandlung der verschiedenen Empfangs- und Sendemöglichkeiten bereit. Der Sender darf nur einen Modus wählen, den der Empfänger beim Capability Exchange signalisiert hat. Es besteht die Möglichkeit, daß ein Sender einem Empfänger eine Auswahl möglicher Betriebsmodi anbietet und dieser einen von ihm bevorzugten auswählt. Dem Sender bleibt es aber letztendlich vorbehalten, einen durch den Empfänger signalisierten Modus auszuwählen.

- **Master Slave Determination:** Konfliktlöser zwischen zwei Endpunkten, die mit einander in einer Konferenz stehen und bei der entschieden wird, wem der MC zugeteilt wird. Dieses gilt auch zwischen Endpunkten, die eine bidirektionale Verbindung aufbauen wollen.

- **Logical Channel Signalling:** Jeder logische Kanal trägt Information vom Sender zu einem oder mehreren Empfängern und ist durch eine logische Kanalnummer gekennzeichnet, welche eindeutig für jede Übertragung ist.

4.1.2.2 Gateway Eigenschaften

Ein Gateway muß die notwendigen, unterschiedlichen Übertragungsformate wie Kontrollsignale und Audio/Video Codecs zwischen dem H.323 LAN und SCN übersetzen.

Ein H.323-Endpunkt kann mit einem anderen Endpunkt im gleichen LAN ohne ein Gateway kommunizieren. Es ist auch möglich, für einen H.323-Terminal die ausgehenden Medienkanäle durch einen Gateway zu schicken und die einkommenden durch einen anderen Gateway kommen zu lassen. Es ist also möglich, an einem Router vorbeizugehen und den Bandbreitenflaschenhals zu vermeiden.

4.1.2.3 Gatekeeper Eigenschaften

Der Gatekeeper ist eine optionale Komponente innerhalb eines LANs, der mit H.323 unterstützt wird. Wenn der Gatekeeper vorhanden ist, müssen die Terminals diesen benutzen.

Ein Gatekeeper spielt eine zentrale Rolle für alle Anrufe innerhalb einer Zone und stellt den registrierten H.323-Endpunkten Call Control Services bereit. Eine wichtige Funktion eines Gatekeepers ist die Adressenübersetzung der IP/IPX-Addresse zu einer E.164 Adresse und umgekehrt.

Eine weitere wichtige Funktion ist das Bandbreitenmanagement. Ein Gatekeeper kann konfiguriert werden, nur eine bestimmte Anzahl von Konferenzen zuzulassen. Was die totale Bandbreitenverfügung für diesen Dienst begrenzt. Diese Funktionen werden mit dem RAS-Protokoll zwischen Gatekeeper und Endpunkt durchgeführt.

Die zu einem Gatekeeper gehörenden Endpunkte bilden eine Zone. Eine Zone ist unabhängig von der Netzwerktopologie und kann somit mehrere LANs, die durch Router verbunden sind, umfassen. In Abbildung 4.6 wird eine Zone definiert.

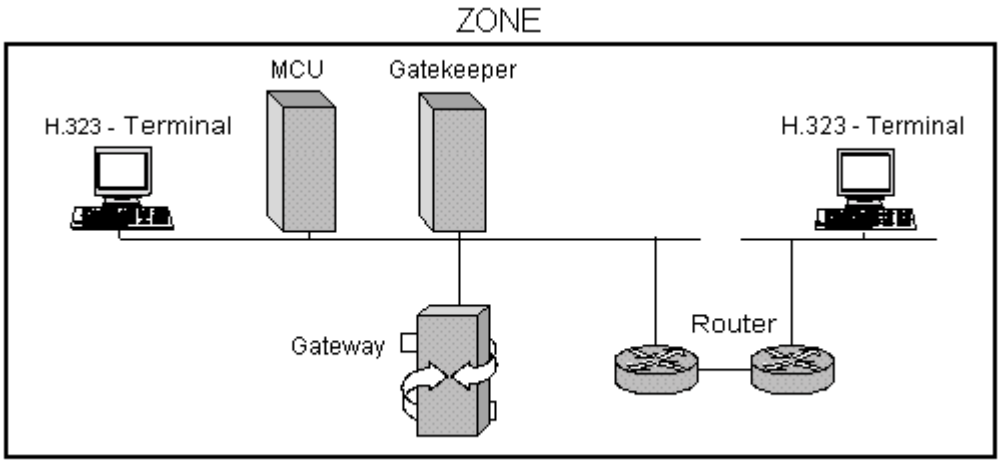

Abbildung 4.6: Zone aus Gatekeeper und Endpunkten [WAL00]

Der Gatekeeper ist von anderen Endpunkten logisch getrennt, kann aber physikalisch innerhalb eines Terminals, Gateways oder MCU implementiert sein. Falls ein Gatekeeper im System vorhanden ist, dient er den folgenden Funktionen (Tabelle 4.1 und 4.2):

Address Translation	- Abbildung von Aliasadressen auf Transportadressen mittels einer Registrierungsnachrichtentabelle
Admission Control	- Authentisierung auf dem LAN durch ARQ/ARC/ARJ-Nachrichten - Kriterien der Zugangskontrolle sind z.B.Call Authentisierung und Bandbreite
Bandwidth Control	- Unterstützung von BRQ/BCF/BRJ-Nachrichten - Kann auf optionalem Bandbreitenmanagement basieren
Zone Management	- Der Gatekeeper unterstützt obige Funktionen in seiner Zone für Terminals, MCU und Gateways, welche in der Zone registriert sind

Tabelle 4.1: Funktionen eines Gatekeepers [H.323]

Optional können folgende Funktionen vom Gatekeeper unterstützt werden:

Call Control Signalling	- In einer Punkt zu Punkt Verbindung, kann der Gatekeeper selbst die Q.931 Call Control Nachrichten bearbeiten, oder sie den Endpunkten zuweisen, die untereinander sich einen direkten Signalisierungskanal aufgebaut haben
Call Authorization	- Kann den Anruf eines Endpunkte die mittels der Q.931 Signalisierungsfunktion zurückweisen
Bandwidth Management	- Kann einen Anruf zurückweisen, wenn nicht genug Bandbreite zur Verfügung steht
Call Management	- Erzeugung einer Liste von aktiven Verbindungen zu deren Verwaltung
Dialed Digit Translation	- Übersetzt gewählte Ziffern zu E.164 Nummern oder Private Network Nummern

Tabelle 4.2: Optionale Funktionen eines Gatekeepers [H.323]

4.1.2.4 MCU - Mulitpoint Control Unit

Die MCU ist ein Endpunkt der Multipointkonferenzen unterstützt. Sie besteht aus Multipoint Controller, Multipoint Prozessor und Audio/Video Streams

Der Multipoint Controller (MC) hat die Kontrollfunktionen des H.245-Kontrollkanals zur Untersuchung einer Multipointkonferenz zwischen drei oder mehreren Endpunkten. Der MC stellt die Capabilities Exchange bereit, welche die verschiedenen Empfangs- und Sendemöglichkeiten aushandelt und mit jedem teilnehmenden Endpunkt in einer Konferenz steht. Dann sendet er ein Capability Set zu den Endpunkten, welches den Kommunikationsablauf eines jeden Endpunktes beschreibt. Das Capability Set kann sich auch ändern, wenn neue Teilnehmer hinzukommen und alte die Konferenz verlassen.

Der Multipoint Prozessor (MP) bekommt Audio-, Video- und Datenstreams von den Endpunkten gesendet, welche sich an einer Konferenz beteiligen. Der MP bearbeitet die Datenstreams und schickt sie weiter zu den Endpunkten.

[DAT00], [H.323], [WAL00]

4.1.3 IP-Netzwerke und Multimediakonferenzen

Für eine Kommunikation im LAN, benutzt H.323 sowohl zuverlässige als auch unzuverlässige Protokolle. Für die T.120-Empfehlung, H.245-Kontrollkanal und H.225 Call Signalling Channel wird das verbindungsorientierte TCP benutzt.

Audio, Video und RAS benutzen das verbindungslose UDP. Die Datenpakete werden hier mit einem kleineren Header schneller und effizienter übertragen, dafür besteht aber eine größere Wahrscheinlichkeit, daß sie verloren gehen. Datenpakete zu behandeln stellt hohe Anspruche an den Empfänger, besonders bei Jitter und Packet Loss, siehe Kapitel 3.1. Ähnlich den Multipointkonferenzen, die mit mehreren Audio- und Videostreams über das unzuverlässige UDP laufen, arbeitet das IP-Multicastprotokoll, das die Kommunikation zwischen Sender und multiplen Empfänger möglich macht.

Auf dem IP-Multicastprotokoll aufsetzend wird das Real Time Protokoll (RTP) benutzt, das von *Internet Engineering Task Force* (IETF) entwickelt wurde. Die Position des Protokolls ist in Abbildung 4.1 dargestellt.

RTP ist unidirektional und wurde zur Handhabung von Audio- und Videostreams über das Internet entwickelt. Vom RTP erhält jedes UDP-Paket einen neuen RTP-Header, der time stamp und sequence number enthält. Mit genügend Puffer beim Empfänger verhindern timing und sequence Informationen, daß die gleichen Pakete mehrmals kommen, da sie die Datenpakete in die richtige Reihenfolge bringen, sowie Audio- und Videostreams synchronisieren.

Das RTCP (Real Time Control Protokoll) ist bidirektional und wird zur Kontrolle des RTP benutzt. RTCP überwacht die QoS und sendet den H.323-Anwendern periodisch Information darüber, wie gut die Verbindung und die Qualitätsinformation ist.

4.1.3.1 UDP- und TCP-Ports

Für Firewalls, die das LAN schützen, ist die H.323-Empfehlung weniger geeignet. Der H.323-Standard benutzt die Ports wie in Tabelle 4.3 dargestellt, in dem mehrere dynamische Ports jeder Verbindung zugeteilt werden. Das hat zur Folge, daß man nicht vorraussagen kann, welche Ports auf der Firewall geöffnet werden sollen und welche nicht.

Port	Type	Description	H.323 Client	Microsoft ILS	H.323 MCU	H.323 Gatekeeper
80	Static TCP	HTTP Interface (Optional)		X		
389	Static TCP	ILS Registration (LDAP)	X	X		
1503	Static TCP	T.120	X		X	
1718	Static TCP	Gatekeeper Discovery	X		X	X
1719	Static TCP	Gatekeeper RAS	X		X	X
1720	Static TCP	H.225 Call Setup	X		X	
1731	Static TCP	Audio Call Control	X		X	
8080	Static TCP	HTTP Server Push (optional)		X		
1024-65536	Dynamic TCP	H.245 (Call Parameters)	X		X	
1024-65536	Dynamic UDP	RTP (Video Data Streams)	X		X	
1024-65536	Dynamic UDP	RTP (Audio Data Streams)	X		X	
1024-65536	Dynamic UDP	RTCP (Control Info)	X		X	

Tabelle 4.3: Von H.323 benutzte Ports [SHO00]

Für H.323 unterstützte, eingehende und ausgehende Anrufe ist es ein Problem, über Firewalls zu kommunizieren, wenn sie NAT für einen Adressenwechsel benutzen. Mit NAT ist es einem Anrufer möglich, das Ziel zu erreichen, aber der Empfänger kann nicht antworten, weil die IP-Adresse und Portnummer nicht stimmt.

In einem durchgeführten Test (siehe Kapitel 6) war es einem Empfänger auserhalb der Firewall, möglich, den Sender aus dem internen Netz zu hören und zu sehen. In gegenrichtung funktionierte dies nicht. Auf das Zusammenwirken und auf Probleme zwischen Firewalls und H.323 wird in Kapitel 4.2 eingegangen.

4.1.3.2 Q.931 und H.245 Verbindungsaufbau

Abbildung 4.7 stellt den relevanten Teil des Verbindungsaufbaus zwischen zwei Endpunkten dar. Endpunkt 1 ist in diesem Fall der Anrufer.

Er stellt mit dem Q.931 Call Signalling direkt (Kapitel 4.3.1) eine Verbindung zu Endpunkt 2 her. Diese Verbindung ist über die definierte TCP-Portnummer 1720 hergestellt worden.

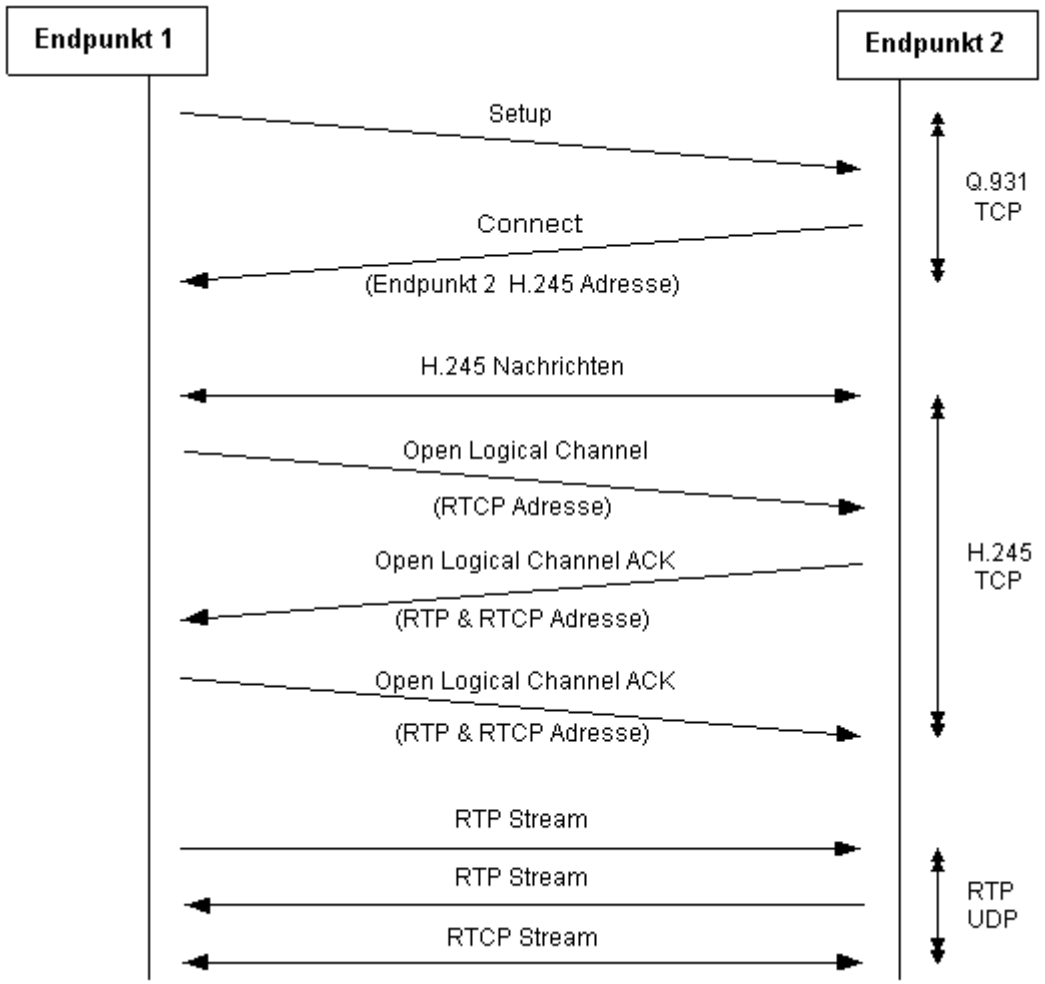

Abbildung 4.7: Verbindungsaufbau von Q.931 und H.245 [INT00]

Dem H.245-Kontrollkanal wird innerhalb der Q.931-Verbindung zwischen den beiden Endpunkten ein dynamisches TCP-Port über 1023 zugeteilt. Endpunkt 2 schickt die zugeteilte Portnummer, im Q.931 Call Signalling Channel eingepackt, zurück zum Endpunkt 1.

H.245 stellt Kontrollfunktionen wie Capability Exchange und zugängliche Codecs zur Anwendung bereit. Mit dem H.245-Kontrollkanal werden auch die Portnummern für die logischen UDP-Kanäle zwischen Endpunkt 1 und Endpunkt 2 bestimmt und geöffnet.

Zum Öffnen eines Medienkanals z.B. Audio oder Video wird die RTCP-Adresse und die Portnummer von Endpunkt 1 zu Endpunkt 2 geschickt, letzterer antwortet mit seiner RTP/RTCP-Adresse und der Portnummer.

Die RTP-Verbindungen in den logischen Kanälen sind Einwegübertragungen, deswegen werden für eine Audiokonferenz zwischen zwei Teilnehmern zwei RTP-Kanäle benötigt. RTP und RTCP benutzen das UDP, welches die Portnummern ab 1024 dynamisch zugeteilt bekommt. RTCP erhält die jeweils folgende, ungerade Nummer von RTP.

[DAT00], [INT00], [SHO00], [SOR00]

4.1.4 Adressierungen

4.1.4.1 Netzwerkadressen

Zur Identifizierung im Netzwerk, muß jede H.323-Entity wenigstens eine Netzwerkadresse besitzen. Für verschiedene Netzwerkumgebungen ist es möglich, daß die Netzwerkadressen unterschiedliche Formate haben. Für die Kommunikation zwischen verschiedenen Netzwerkumgebungen mit unterschiedlichem Adressenformat muß im Gatekeeper eine Übersetzung zur Anpassung bestehen.

4.1.4.2 TSAP-Identifier

Es ist möglich, daß für jede Netzwerkadresse einer H.323-Entity mehrere TSAP-Identifiers vorhanden sind. TSAP-Identifiers lassen mittels Multiplexing verschiedene "channels" einer Verbindung die gleiche Netzwerkadresse benutzen. Bekannte TSAP-Identifiers sind:

- **Endpunkte:** Call Signalling Channel TSAP-Identifier
- **Gatekeeper:** RAS-Channel TSAP-Identifier

4.1.4.3 Aliasadressen

Ein Endpunkt darf eine oder mehrere Aliasadressen haben. Die Aliasadressen ermöglichen eine alternative Methode zur Endpunktadressierung. Diese alternative Methode besteht aus einer privaten Telefonnummer oder aus H.323-IDs. H.323-IDs bestehen aus ASCII-Zeichen und sind in H.225.0 [H.225] definiert.
H.323-IDs sind z.B. Anwendername, Email oder eine andere Form der Identifikation. In einer Zone, wie in Kap 4.1.2.3 definiert, muß eine Aliasadresse eindeutig sein. Gatekeeper, MC und MP können kein Alias haben.

In einem Anruf zwischen zwei Endpunkten ohne Gatekeeper benutzt der Anrufer, um den Empfänger zu erreichen, direkt die Call Signalling Channel Transportadresse. Wenn ein Gatekeeper vorhanden ist, muß er die Aliasadresse, in eine Call Signalling Channel Transportadresse übersetzen.

[H.323], [WAL00]

4.1.5 Call Signalling Prozeduren

Der Ablauf einer Anrufprozedur verläuft in folgenden Phasen [H.323]:

- **Phase A -** Call Setup
- **Phase B -** Initial Communication an Capability Exchange
- **Phase C -** Establishment of Audiovisual Communication
- **Phase D -** Call Services
- **Phase E –** Call Termination

4.1.5.1 Phase A - Call Setup

Das Call Setup zwischen zwei Endpunkten wird mit Call Control Nachrichten, wie in Kapitel 4.1.2.1 beschrieben, in der H.225-Empfehlung durchgeführt. Als erstes muß der Endpunkt einen Antrag auf Bandbreite stellen, der vom Gatekeeper angenommen wird.

Die Aufbauphase ist entsprechend den vorhanden Komponenten (Gateway, Gatekeeper) und deren Arbeitsweise (Call Signalling Channel über einen Gatekeeper oder direkt) verschieden.

Abbildung 4.8 zeigt einen Basic Call Setup, ohne daß Endpunkt 1 oder Endpunkt 2 am Gatekeeper registriert sind.

Die RAS-Funktionen fallen hieraus. Endpunkt 1, der Endpunkt des Zurufes, startet mit der Setup (1) Nachricht zum bereits bekannten Call Signalling Channel TSAP-Identifier am Endpunkt 2. Endpunkt 2 antwortet dann mit Connect (4).

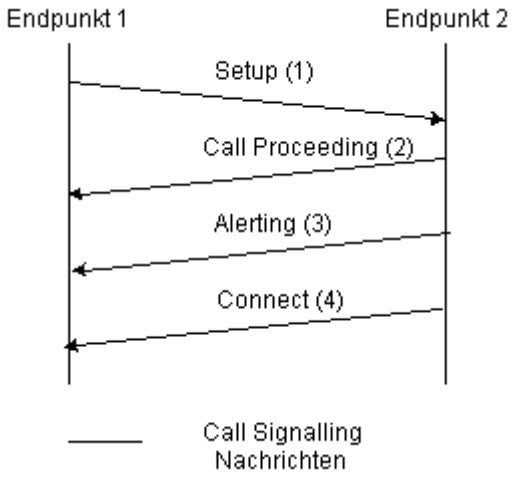

Abbildung 4.8: Basic Call Setup zwischen Endpunkt 1 und Endpunkt 2 [H.323]

Abbildung 4.9 zeigt einen Verbindungsaufbau zwischen den Endpunkten, die bei dem selben Gatekeeper registriert sind. Endpunkt 1 fängt hier mit ARQ (1) einen Austausch mit dem Gatekeeper an. Der Gatekeeper antwortet durch ACF/ARJ (2), mit der Call Signalling Channel Transportadresse des aufgerufenen Endpunktes 2. Endpunkt 1 sendet danach die Setup (3) Nachricht mit der gegebenen Transportadresse zum Endpunkt 2. Wenn Endpunkt 2 den Anruf akzeptiert, antwortet er mit ARQ (5) dem Gatekeeper. Endpunkt 2 antwortet auf Endpunkt 1 mit der Connect (8) Nachricht, welche die H.245-Kontrollkanal- transportadresse für H.245-Signalisierung beinhaltet. Weitere Möglichkeiten eines Verbindungsaufbaus sind in [H.323] aufgeführt.

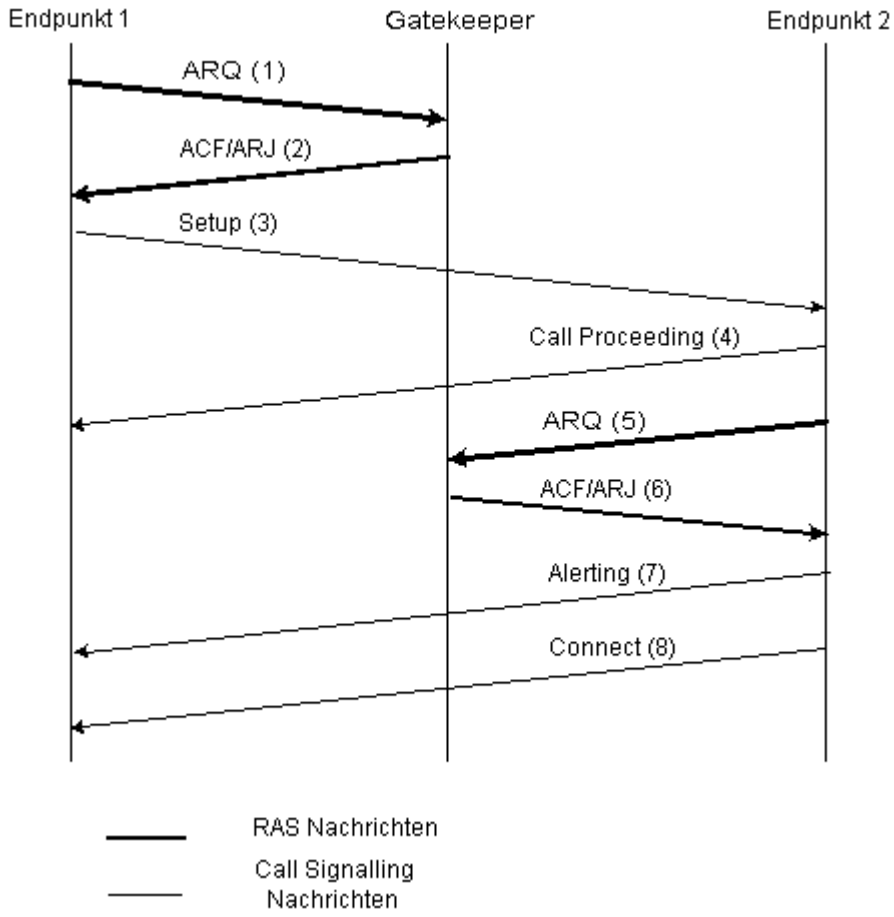

Abbildung 4.9: Call Setup über Gatekeeper [H.323]

4.1.5.2 Phase B – Initial Communication and Capability Exchange

Nach dem Austausch von Call Setup Nachrichten in Phase A müssen sich die beiden Endpunkte entscheiden, ob es notwendig ist, ein H.245-Kontrollkanal zu etablieren. Der H.245-Kontrollkanal ist für die Capability Exchange und das Öffnen des Medienkanals notwendig. Die Endpunkte müssen die in Kapitel 4.1 aufgeführten Aspekte in jedem Fall unterstützen.

4.1.5.3 Phase C – Establishment of Audio Visual Communication

Nach dem Austausch von H.245-Kontrollkanalnachrichten werden die logischen Kanäle für verschiedene Daten Ströme geöffnet. Audio- und Video Streams werden über dynamische TSAP-Identifiers und UDP übertragen. Die Kontrolldaten werden über TCP übertragen.

4.1.5.4 Phase D – Call Services

Die Call Services beziehen sich hauptsächlich auf den Bandbreitenwechsel und den Status der Endpunkte. Der Status der Endpunkte wird durch den Gatekeeper erfragt.

4.1.5.5 Phase E – Call Termination

Ein Endpunkt kann eine Verbindung abbrechen. Außerdem müssen nach einer Konferenz die aufgebauten Kanäle wieder geschlossen werden. Ist ein Gatekeeper vorhanden, beschreibt die H.323-Empfehlung verschiedene Abläufe zum Schließen der Verbindung.

[H.323], [WAL00]

4.2 H.323 und die verschiedenen Firewalltechnologien

4.2.1 Schwierigkeiten innerhalb des Zusammenwirkens von H.323 und Firewalls

Bei der H.323-Empfehlung über das Internet gibt es Probleme, wenn eine H.323-Verbindung durch eine Firewall geht. Das Problem stellt sich besonders, wenn H.323 sehr komplexe und viele verschiedene Portnummern benutzt. Die meisten Portnummern sind dynamisch besetzt und benutzen das UDP. Die größten Probleme folgen aufgelistet:

- Die zwei wichtigsten Protokolle von Verbindungsaufbau, Übertragung und Verbindungsabbau sind H.225 Call Control und H.245-Kontrollkanal. Im Setup wird der H.225 Call Control Channel zu der bereits gewählten Portnummer 1720 initiiert. H.225 Call Control Channel benutzt Q.931 Call Setup für den Verbindungsaufbau mit einem anderem Endpunkt. In H.225 Call Control Nachrichten wird entschieden, welcher dynamische TCP-Port den H.245-Kontrollkanal bekommt. Die Portnummer ist vorher nicht bekannt. Danach bekommen die logischen Kanäle RTP und RTCP dynamische UDP-Ports, die vom H.245-Kontrollkanal festgelegt und zwischen den Endpunkten übertragen werden.

Die Übertragung der Kontrollkanäle ist in den Datenpaketen der Applikationsebene eingebettet. Die Aufgabe, die Datenpakete auf Information zu untersuchen, ist für eine Firewall nicht einfach durchzuführen. Zur Übertragung von Sprache werden mindestens vier Ports geöffnet: H.225-Kanal, H.245-Kanal, RTCP-Kanal und RTP-Kanal, wobei alle Kanäle außer dem H.225-Kanal dynamisch besetzt sind. Mit so vielen dynamischen Ports ist es fast unmöglich, eine Firewall für das H.323-Protokoll zu konfigurieren, ohne, daß alle UDP-Ports über 1023 geöffnet werden, die für ein LAN nicht gangbar sind.

- Bei Firewalls die NAT benutzen besteht das Problem, daß im H.225 Call Control Channel und im H.245-Kontrollkanal die benutzten IP-Adressen und Portnummern in den eingebetteten Kanälen, wie zuvor erläutert übertragen werden. Bei der Benutzung von NAT sind es die privaten Adressen des internen Netzwerkes, die durch den H.225 Call Control Channel übertragen werden. Wenn die Datenpakete durch den eingebetteten H.225 Call Control Channel die öffentliche, vermittelbare IP-Adresse und Portnummer im IP-Header durch die NAT zugeteilt bekommen, dann befinden sich verschiedene Adressen im internen Netzwerk des H.225 Call Control Channels und im IP-Header. Wenn der Empfänger die Datenpakete erhält und den eingebetteten H.225 Call Control Channel liest, erhält der H.323-Empfänger die interne IP-Adresse und damit ist es ihm nicht möglich, darauf zu antworten.

- Verbindungen sollen auch von außerhalb aufgebaut werden können, so daß das wesentliche Grundprinzip nur abgeschickte Verbindungen zuzulassen verletzt werden muß.

- Der größte Teil der Kontrollinformationen ist in ASN.1 [ITU00] kodiert.

[INT00], [SHO00]

4.2.2 H.323 und die verschiedenen Firewalltechnologien

Abbildung 4.10 zeigt das LAN mit den H.323-Entities, wie in Kapitel 4.1.2 definiert und eine Firewall, die LAN und Internet miteinander verbindet.

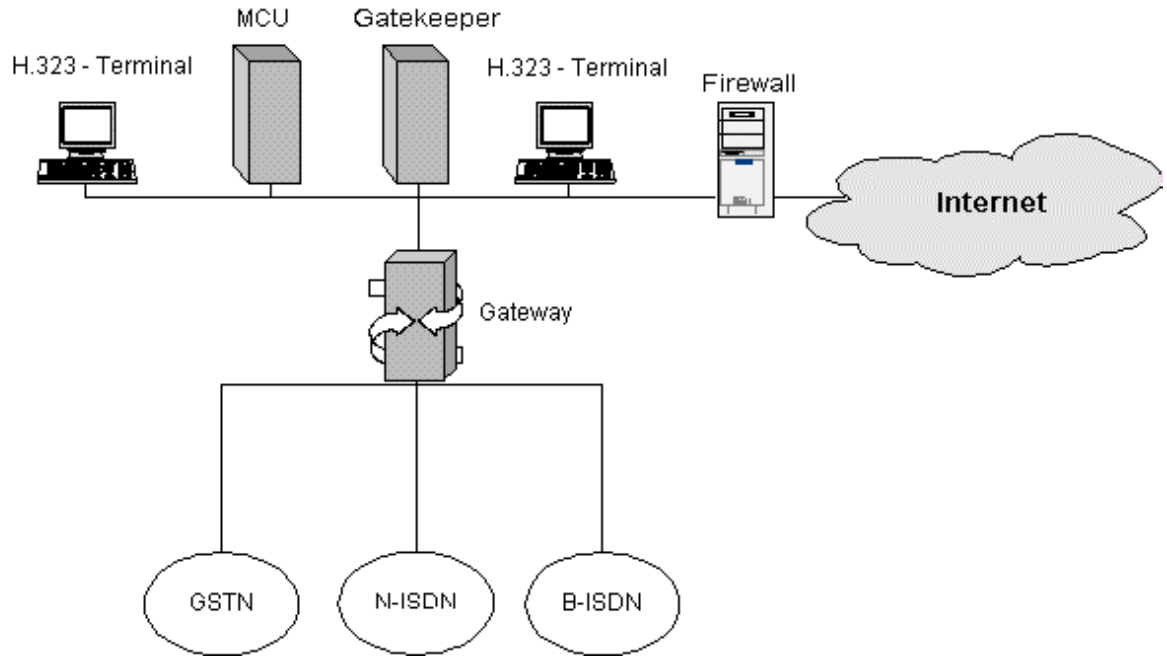

Abbildung 4.10: Firewall als Gateway zum Internet

Eine Beschreibung der verschiedenen Firewalltechnologien ist bereits in Kapitel 2. Erfolgt. Deren Zusammenwirken mit H.323 soll hier nur kurz erläutert werden.

- **Statischer Paketfilter:** Ist die einfachste Form einer Firewall, der eine Deny/Allow Liste mit IP-Adressen und Portnummern beschreibt und entscheidet, ob die Datenpakete passieren dürfen oder nicht. H.323 funktioniert nicht mit dem statischen Paketfilter, denn es enthält keine Unterstützung für UDP und kann nicht die in der Applikationsebene eingebetteten Kanäle, H.225 und H.245, bearbeiten.

- **Dynamischer Paketfilter:** Hat die gleiche Funktion wie der statische Paketfilter. Er unterscheidet sich nur dahingehend, daß er die benutzten IP-Adressen und Portnummern für eine gewisse Zeitspanne speichern kann. Diese Zeitspanne gibt dem UDP die Möglichkeit, mit diesem Filter arbeiten zu können. Die eingebetteten Kontrollkanäle enthalten die Portnummern und werden nicht ausgepackt und analysiert.

- **Stateful Inspection:** Ist die Erweiterung des dynamischen Paketfilters und ermöglicht eine intelligentere Analyse der Datenpakete, sowohl die Informationen über IP-Adressen und Portnummern, als auch zustand.

- **Application Level Proxy:** Wird oft auch als Proxy-Server bezeichnet, wenn mehrere Proxy-Prozesse auf dem gleichen Server arbeiten. Hier muß ein H.323-Proxy, der als ein Kontakt zwischen dem LAN und dem Internet funktioniert, eingesetzt werden. Der H.323-Proxy muß die Kontrolldaten analysieren können und die RTP/RTCP-Datenpakete mit Audio/Video weiterleiten. Zwischen dem LAN und dem Internet überwacht der H.323-Proxy alle H.323-Verbindungen und versichert, daß nur gültiger H.323-Verkehr durch die Firewall geht. Die Zugriffskontrollregeln am H.323-Proxy bestimmen und überwachen, welche Anwender eine Konferenz beginnen oder empfangen können und welche Endpunkte oder Ziele verwendet werden.

  Im Proxy-Server wird in den meisten Fällen eine Adressenumsetzung, NAT zwischen der privaten und öffentlichen vermittelbaren IP-Adresse vorgenommen. Proxy-Server muß der H.323-Proxy die eingebetteten Kontrollkanäle öffnen und analysieren, um sehen zu können, wer der Empfänger ist.

  Damit der Empfänger antworten kann, müssen die Datenpakete erst zum Proxy-Server zurückgesendet werden und von dort muß der Proxy-Server die Datenpakete zum Endpunkt weiterleiten. Um diese Zusammenarbeit zu ermöglichen, müssen die Endpunkte so entwickelt werden, daß sie mit H.323-Proxy kompatibel sind.

- **Circuit Level Proxy:** Funktioniert ähnlich wie Application Level Proxy, es unterscheidet sich nur darin, daß es als ein generisches Proxy für alle Protokolle eingesetzt werden kann. Das bekannteste Circuit Level Proxy Protokoll ist das von IETF entwickelte SOCKS [AFT01]. Der Endpunkt im internen Netzwerk untersucht die Datenpakete und fordert anschließend am Proxy-Server an die spezifischen Portnummern, die geöffnet werden sollen.

[INT00], [SHO00]

4.2.3 Die Kontrollfunktionen eines H.323-Proxy

4.2.3.1 H.225 Call Control Channel und H.323-Proxy

Wie der Q.931 Call Setup über den H.225 Call Control Channel mit einem H.323-Proxy zusammen-arbeitet, ist in Abbildung 4.11 dargestellt. Endpunkt 1 ist der Anrufer und Endpunkt 2 befindet sich im internen Netzwerk hinter einem Proxy-Server.

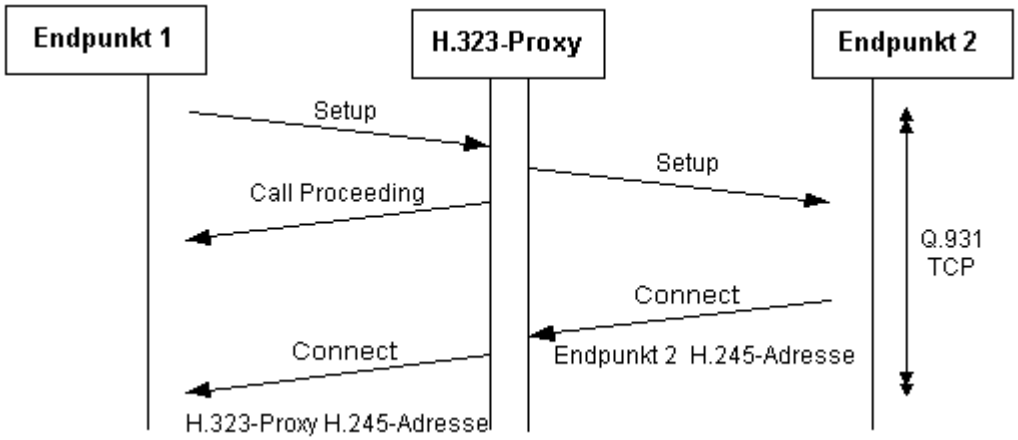

Abbildung 4.11: Q.931 Verbindungsaufbau über einen H.323-Proxy-Prozeß [INT00]

Wenn ein Antrag auf eine neue Konferenz bei einem definierten Q.931 Call Setup mit der Portnummer 1720 gestellt wird, antwortet der H.323-Proxy dem Endpunkt 1 mit einem Call Proceeding. Danach muß der H.323-Proxy die Zieladresse kontrollieren. Entspricht diese der Richtigkeit, wird eine Setup Nachricht zum Endpunkt 2 gesendet. Wenn die Connect Nachricht mit der Portnummer des Endpunktes 2 H.245-Kontrollkanals gesendet worden ist, wird diese im Proxy kontrolliert und gespeichert. Dann wird eine Portnummer vom H.323-Proxy mit der H.245-Portnummer verknüpft und die Portnummer des H.323- Proxy wird zum Endpunkt 1 gesendet.

4.2.3.2 H.245-Kontrollkanal und H.323-Proxy

Wenn es sich um eine Audio- oder Videokonferenz handelt, müssen den RTP- und RTCP-Medienkanälen vom Proxy Server an beiden Seiten Portnummern zugeteilt werden.

Die RTP- und RTCP-Medienkanäle werden über die logischen Kanäle übertragen und vom H.245-Kontrollkanal werden ihnen die Portnummern zugeteilt. In Abbildung 4.12 hat der Anrufer, Endpunkt 1, die Konferenz initiiert und sendet dann seine RTCP-Portnummer zum H.323-Proxy.

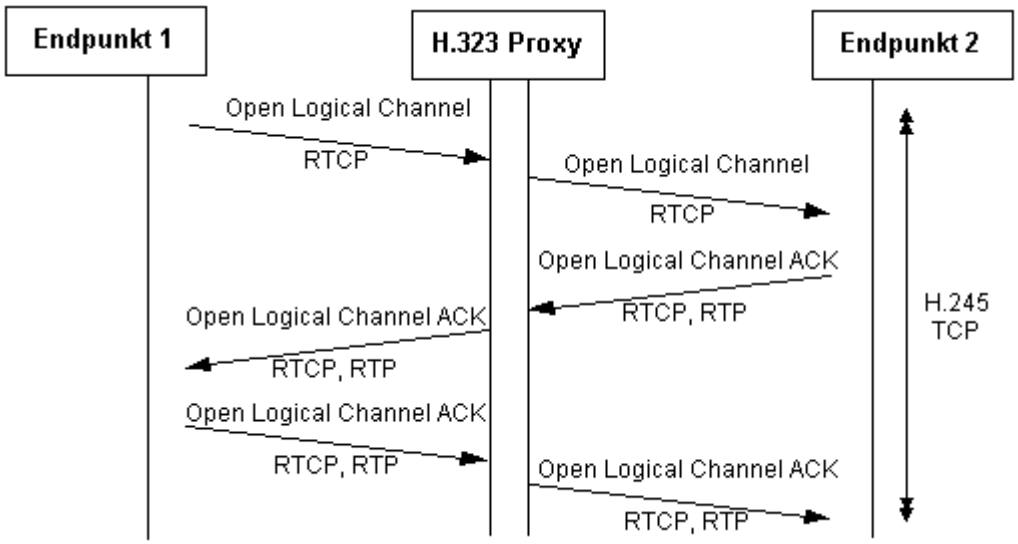

Abbildung 4.12: Verbindungsaufbau des H.245-Kontrollkanals über H.323-Proxy [INT00]

Der H.323-Proxy ersetzt die RTCP-Portnummer mit seiner eigenen im H.245-Kontrollkanal und sendet die Datenpakete weiter zum Endpunkt 2. Wenn Endpunkt 2 die RTCP-Portnummer erhalten hat, wird die unmittelbare, automatische Paketweiterleitung am RTP- und RTCP-Datenpaket aktiviert. Danach werden die RTP- und RTCP-Portnummern am Endpunkt 2 in die ACK-Datenpakete geschrieben und zurück zum H.323-Proxy gesendet. Der H.323-Proxy, der schon zuvor die RTCP-Portnummer von Endpunkt 1 erhalten hat, leitet die Datenpakete von Endpunkt 2 zurück zu Endpunkt 1. Endpunkt 1 antwortet dann noch einmal mit einer Eingangsbestätigung (ACK) aber diesmal mit der RTP- und RTCP-Portnummer.

[INT00], [SHO00]

4.3 Die SIP Empfehlung

4.3.1 SIP Einleitung

Das SIP (Session Initiation Protocol) ist ein von IETF entwickeltes, neues Multimediasignalisierungs-protokoll für LANs ohne QoS. SIP ist ein textbasiertes Protokoll, das in der Applikationsschicht arbeitet und mit der Funktionsweise von HTTP und SMTP vergleichbar ist.

SIP ist als ein einfaches Kontrollprotokoll gedacht und kann eine Verbindung zwischen zwei oder mehreren Endpunkten herstellen.

SIP etabliert ein Call Setup, das Änderungen in einer Konferenz vornehmen und Verbindungen schließen kann. Um andere Funktionen zu unterstützen, benutzt SIP schon getestete und zuverlässige Protokolle wie z.B. RTP/RTCP für den Medientransport, RSVP um Netzwerkressourcen zu erhalten und das später noch genauer behandelte Session Description Protocol (SDP) zum Initiieren des Call Setups.

SIP kann mit Unterstützung des Multicastprotokolls als peer to peer oder als Multipointkommunikation arbeiten. Unterstützte Medien sind Audio, Video und Daten wie z.B. File Transfer, Chat und Whiteboard. Zum größten Teil wird SIP für die Sprachübertragung benutzt. Die Ursache dafür ist die einfache Implementierung, und daß SIP als ein relativ junges Protokoll für die anderen Medien noch nicht so gut entwickelt ist. Bisher wird SIP oft als textbasiertes Protokoll benutzt, um neue Software zu implementieren. SIP wird häufig in Programmiersprachen wie Java und Perl geschrieben.

Ein SIP, das in einer Applikation implementiert ist, muß die folgenden Funktionen unterstützen:

- **Namensübersetzung und Anwenderlokalisierung:** Garantiert, daß der gesuchte Anwender erreichbar ist, egal ob er sich im Home, Office oder anderswo befindet.

- **Aushandlung von geltenden Bestimmungen:** Erlaubt den Endpunkten einer Konferenz z.B. zu entscheiden, welche Medien und Codecs unterstützt und in der Konferenz benutzt werden sollen.

- **Teilnehmeranrufmanagement:** An einer Konferenz können sich neue Teilnehmer beteiligen, und alte können die Konferenz verlassen.

- **Änderung von Kommunikationsmedien:** Ein Anwender in einer Konferenz muß die Möglichkeit haben, die Kommunikationsmedien zu ändern oder zu kombinieren z.B. Sprache und Video.

[CEN00], [RAD01]

4.3.2 SIP-Architektur

SIP ist als eine einfache Client/Server Architektur genau wie HTTP entwickelt worden. Die Client/Server- SIP-Komponenten sind **User Agent**, **SIP-Proxy-Server**, **SIP-Redirect-Server** und der **Registrar-Server**.

User Agent ist ein SIP-Endpunkt, der eine Verbindung initiiert oder von einem anderen aufgerufen wird.

Wenn ein SIP-Endpunkt eine Verbindung initiiert, wird er als Client betrachtet und User Agent Client (UAC) genannt. Wird ein SIP-Endpunkt aufgerufen, fungiert er als Server und wird dann User Agent Server genannt (UAS). User Agents können entweder direkt miteinander kommunizieren oder über einen zwischenliegenden Server, wenn das die Strecke zwischen den beiden Endpunkten erfordert.

Der zwischenpositionierte Server hat die Funktion, empfangene Datenpakete weiterzusenden und kann als Stateless oder Stateful Proxy-Server auftreten. Ein Stateful Proxy-Server hat den Vorteil, daß er die Information über eingehende und ausgehende Datenpakete für eine Konferenz mit Hilfe eines Cache speichert und daher eine bessere Datenvermittlung gewährleisten kann. Der Stateful Proxy-Betrieb kann aber auch ein Nachteil sein, wenn der Datenfluß zu groß wird und der Cache die anfallende Datenmenge nicht mehr bewältigen kann. Ein Stateless Proxy-Server speichert die Datenpakete nicht, sondern überprüft nur, wohin die Datenpakete weitergesendet werden müssen und er schreibt seine Adresse in die Adressenliste der Datenpakete.

In einem Call Setup kommuniziert der UAC zuerst mit dem Redirect-Server, der eine Adressenliste von Rufnummern wie z.B. sip:Email oder Kurznamen enthält, um den gesuchten Endpunkt, einen UAS, zu erreichen. Der SIP-Proxy-Server behandelt dann die Call Setup Datenpakete vom UAC, schreibt seine eigene Adresse in die Datenpakete und leitet die Anfrage vom UAC zum UAS weiter.

Um erreichbar zu sein, muß sich der Anwender in bestimmten Zeitintervallen registrieren. Der User Agent sendet dann eine Registrierungsnachricht mit seiner aktuellen Adresse zum SIP-Registrar Server.

Diese Nachricht wird von dem Registrar-Server weitergeleitet und auf dem External Location Server gespeichert.

Zum Verbindungsaufbau zwischen UAC und UAS wird das SDP benutzt, das dem H.225 Call Control Channel für H.323 in der Funktionsweise ähnlich ist. Ein UAC, der eine Verbindung aufbaut, kann zwischen TCP oder UDP als Übertragungsprotokoll wählen und auch die Portnummer für das Call Setup festlegen.

Das standardmäßige Übertragungsprotokoll im SDP Call Setup ist das UDP mit der Portnummer 5060. Das SIP benutzt für Medienübertragung das RTP als Übertragungsprotokoll. Die Portnummern über 1023 für das RTP werden dynamisch mit Hilfe von SDP reserviert. Weil RTP unidirektional ist, müssen an jedem Endpunkt zwei Ports benutzt werden. SDP eröffnet auch einen Port für das bidirektionale RTCP, da dieses das RTP überwachen soll.

Die SDP Call Setup Nachrichten enthalten:

- den Medientyp, welchen der Teilnehmer wählt z.B. Audio, Whiteboard und Video
- die in der Konferenz unterstützten Codecs
- die vom Teilnehmer gewünschten Portnummern und die IP-Adresse

Die oben genannten Informationen müssen zwischen den Endpunkten ausgetauscht werden, um eine beliebige Verbindung herzustellen.

Auch wenn sich die Verhältnisse in einer Konferenz verändern, z.B. neue Teilnehmer und neue Codecs hinzukommen, muß das SDP die Kommunikation zwischen den Endpunkten regeln. Dies geschieht, wie am Anfang einer Konferenz, mit einer neuen SDP Call Setup Anfrage.

[CEN00], [RAD01], [SIP99]

4.3.3 SIP Call Setup

Eine Verbindung kann über verschiedene Server und Netzwerke laufen. Die Server haben die Möglichkeit sogenannte External Location Server zu kontaktieren, die z.B. LDAP-Server sind und Service- und Anwenderinformationen vermitteln.

Ein von SDP initiiertes Call Setup besteht aus dem vom UAC gesendeten INVITE Request Befehl. Der aufgerufene UAS antwortet mit einem ACK-Befehl. Abbildung 4.13 zeigt einen möglichen Aufbau einer Verbindung zwischen zwei SIP-Endpunkten und die Komponenten einer SIP-Architektur.

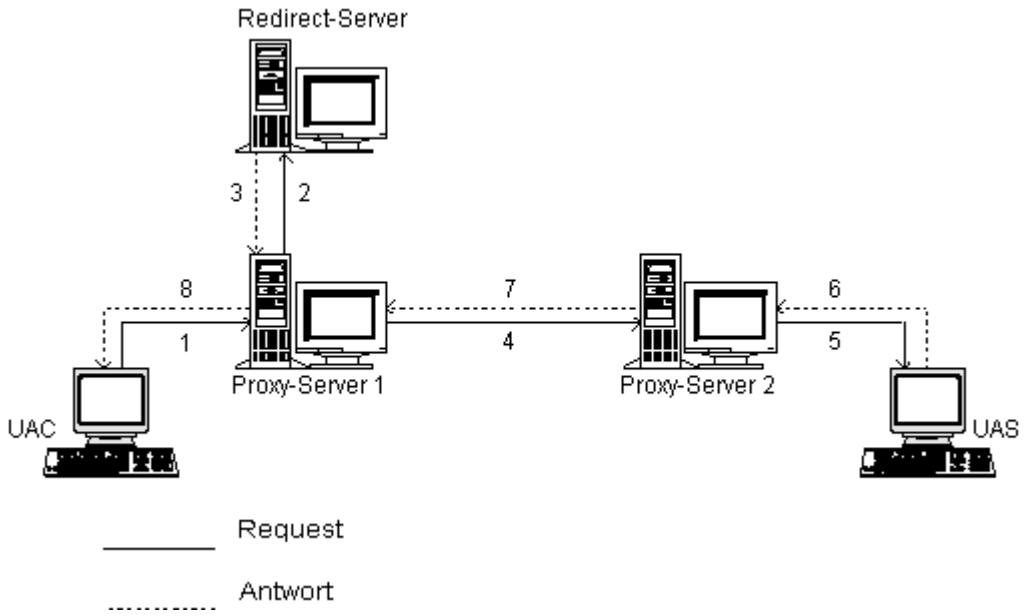

Abbildung 4.13: Call Setup zwischen UAC und UAS [CEN00]

Der UAC startet ein SDP Call Setup mit seiner Adresse und Portnummer direkt zur aufgerufenen UAS IP-Adresse und Portnummer, wenn keine Proxy-Server dazwischen liegen. Das ist z.B. möglich, wenn sich die Endpunkte im gleichen LAN befinden. Sonst kommuniziert UAC erst mit dem Redirect-Server und Proxy-Server 1, wie es in Abbildung 4.13 dargestellt ist. Vom Redirect-Server bekommt der UAC die Adresse des UAS.

Proxy-Server 1 schreibt seine Adresse in die SDP Call Setup Nachricht, so, daß die Datenpakete erfahren, welchen Weg die Antwortpakete gehen müssen, um zum Anrufer zurückzukommen.

Die SDP Call Setup Nachricht wird dann von Proxy-Server 1 zu Proxy-Server 2 weitergeleitet, der die Call Setup Nachricht zum Ziel des UAS sendet. Wenn der UAS dort erreichbar ist, sendet er das Antwortpaket zurück zu der vom UAC spezifizierten Adresse und Portnummer, die in der SDP Request Call Setup Nachricht angegeben sind. Im Antwortpaket schreibt UAS, welche Portnummer für die Medienübertragung mit UAC benutzt werden soll.

Wird der UAS nicht vom Proxy-Server 2 erreicht, kann der Proxy-Server 2 beim External Location Server nach dem gesuchten UAS fragen, siehe Abbildung 4.14. Die Anfrage an den External Location Server wird nicht mit einem SIP-Protokoll ausgeführt.

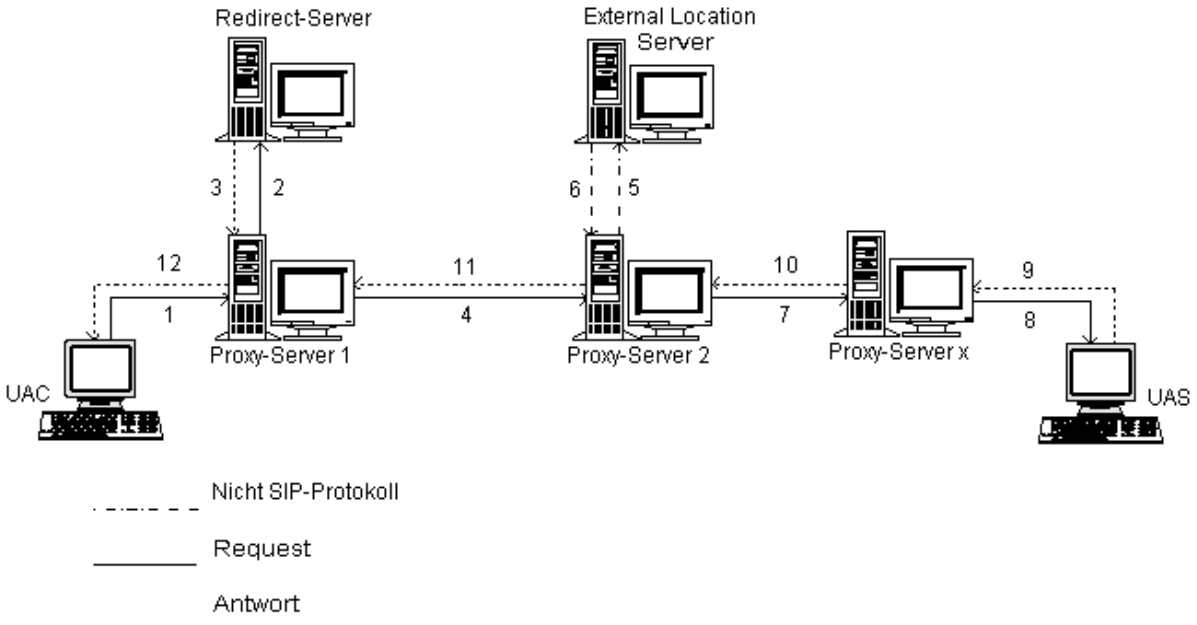

Abbildung 4.14: Verbindungsaufbau zu einem UAS [CEN00]

Ist der UAS wie in Abbildung 4.14 am ersten SIP-Redirect-Server nicht unter seiner Adresse, z.B. seiner Office-Adresse, erreichbar, fragt der Proxy-Server 2 beim External Location Service Server nach einer anderen Adresse des gesuchten UAS. Der External Location Server kann dann andere Protokolle benutzen, wie z.B. Finger und rwhois, um die Position des Anwenders zu bestimmen. Der Proxy-Server 2 bekommt daraufhin vom External Location Server eine Liste mit den Adressen bei welchen der Anwender angemeldet war bzw. ist. Die Call Setup Nachricht wird dann zum Proxy-Server x gesendet, an dem sich der UAS befindet. Dies kann beispielsweise seine Home-Adresse sein.

Wenn der UAS die Call Setup Anfrage erhalten hat, antwortet er dem UAC mit einer oder mehreren SDP Nachrichten.

[CEN00], [RAD01], [SIP99]

4.4 SIP und die verschiedenen Firewalltechnologien

SIP ist genau wie das H.323-Protokoll weniger für Firewalls geeignet, da es, wie schon erläutert, ein neues Protokoll ist und sich noch in der Aufbauphase befindet. Derzeit beschäftigt sich kein Firewall-Hersteller intensiver mit SIP. Trotzdem ist SIP ein einfacheres Protokoll mit Client/Server-Struktur und wenigen Ports für die Kommunikation über eine Firewall als z.B. H.323.

Bisher gibt es nur wenige, eher diffuse Vorschläge zur SIP-Unterstützung in Firewalls. Daher sollen diese nur kurz erläutert werden.

In Abbildung 4.15 wird erläutert, wie das SIP-Netzwerk mit einem am Firewall installierten SIP ALG aussieht.

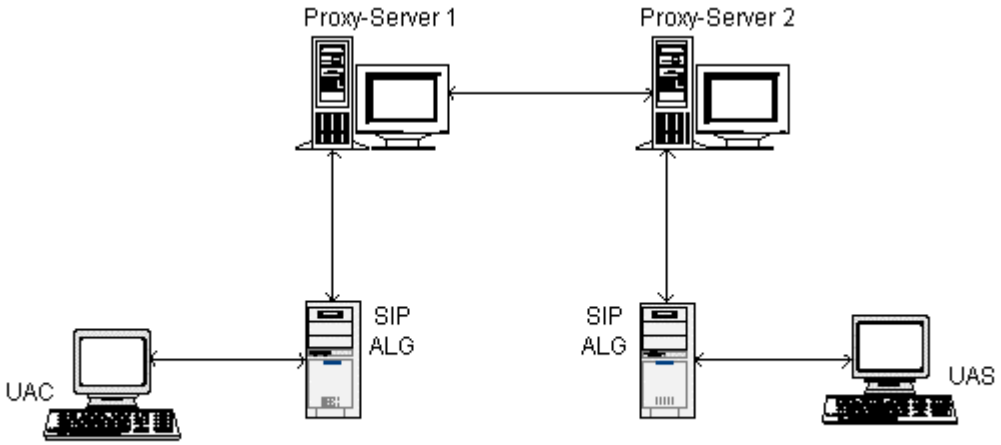

Abbildung 4.15: SIP-Netzwerk mit einer SIP ALG [ROS00]

4.4.1 Implementierung auf einem ALG

Eine Möglichkeit ist, ein SIP ALG an einem Firewall-Rechner wie in Abbildung 4.15 zu installieren. Jedes SIP IP-Datenpaket, sowohl mit der Default Portnummer 5060 als auch mit anderen Spezifikationen, muß zur Bearbeitung von NAT zu SIP ALG gegeben werden, egal ob die Datenpakete von einem externen oder internen Netzwerk kommen. Damit der ALG weiß, welche Anwender sich im internen LAN hinter der Firewall befinden, wird ein SIP-Registrar-Server gebraucht. Der SIP-Registrar-Server informiert ALG, welche Endpunkte bei welchen IP-Adressen angemeldet sind.

Damit die SIP-Kommunikation zwischen UAC im internen Netzwerk und UAS im externen Netzwerk über eine Firewall funktioniert, muß der ALG, der am Firewall installiert ist, zwei Bedingungen erfüllen:

- Zum einen muß der ALG bei ausgehenden SIP-Datenpaketen die interne IP-Adresse gegen eine von NAT generierte, externe IP-Adresse austauschen. Wenn die Antwortpakete zurückkommen, muß der ALG wieder die externe IP-Adresse durch die interne IP-Adresse ersetzen und zum UAC weiterleiten.

- Zum anderen sind die in die Datenpakete eingebetteten Adressen, die sogenannten SDP-Daten, im SIP zu behandeln. Diese enthalten die IP-Adresse des User Agent und die Portnummern, die er für die RTP-Medienübertragung benutzen möchte.

In den ausgehenden SIP-Datenpaketen muß ALG die IP-Adressen und Portnummern mit denen von der NAT tauschen. Für eingehende SIP-Datenpakete wird die Prozedur umgekehrt. Zur Lösung des Problems wird eine SIP ALG Implementierung in Java geschrieben, die die Umwandlungen durchführt.

[THE00]

4.4.2 SIP über SSL/TLS

Ausgehend von Abbildung 4.15 gibt es drei Probleme zu überwinden, diese sind:

- Übertragung der SIP-Anfragen vom UAC zum Proxy-Server 1 und der Antwortpakete von Proxy-Server 1 zurück zum UAC

- Übertragung der SIP-Anfragen von Proxy-Server 2 zum aufgerufenen UAS und der Antwortpakete vom UAS zurück zum Proxy-Server 2

- Medienübertragung zwischen den beiden User Agents

Bei der Übertragung vom UAC zum Proxy-Server 1 wird davon ausgegangen, daß die Firewall fast alle Portnummern blockiert, aber nicht die TLS/SSL Portnummer 443, die die Aufgabe hat, die Datenpakete zu verschlüsseln. TLS/SSL wird häufig zur Verschlüsselung von HTTP-Seiten benutzt und dann als HTTPS bezeichnet. TLS/SSL ist zum Verschlüsseln verschiedener Protokolle geeignet und wird einfach zum Tunneln der Datenpakete benutzt. Ein Vorschlag ist, daß der UAC ein SDP Call Setup über die TLS/SSL Portnummer 443 zum Proxy-Server 1 startet.

Dann wird empfohlen, daß der UAC die Verbindung mit dem Proxy-Server 1 als einen geöffneten Kommunikationskanal hält. Dann sendet der UAC die Registrierungsinformationen über diese Verbindung zum Registrar-Server, wie zuvor beschrieben.

Bei einkommenden SIP Call Setups würde die Firewall erst den Registrar-Server daraufhin kontrollieren, welche UAS bei ihm registriert sind. Stimmt ein solcher mit dem einkommenden SIP Call Setup überein, wird der SIP Call zum UAS weitergeleitet.

Das größte Problem stellt die Medienübertragung via RTP dar. Dabei benutzt RTP das UDP, wobei die Portnummern dynamisch besetzt werden. Um das Problem zu lösen, wird empfohlen, einen RTP-Forwarder über TCP oder TLS/SSL zu verwenden.

Mit dem RTP-Forwarder soll erreicht werden, daß an einer bestimmten Portnummer eingehende RTP-Medienpakete über eine andere Portnummer weitergeleitet werden, entweder über eine TCP-Verbindung oder über TLS/SSL.

[ROS00]

4.5 Vergleich von H.323 und SIP

Die Protokolle H.323 und SIP erfüllen den gleichen Zweck, die Unterstützung von VoIP. Zum einen gibt es die ITU-Organisation, die das H.323-Protokoll bereits seit 1996 entwickelt. Diese Organisation zielt darauf ab, in anderen ITU-Normen die synchronen Dienste wie VoIP über Internet mit dem bereits bestehenden Netzwerk für Telekommunikation zu integrieren.

Zum anderen gibt es das von der IETF entwickelte neue SIP, das nach dessen Intention VoIP in das schon existente Internetprotokoll integriert und zusammenarbeiten läßt.

Innerhalb der beiden Organisationen gibt es VoIP betreffend Gemeinsamkeiten und Unterschiede. Sowohl H.323 als auch SIP haben Nachteile und Vorteile. Deswegen kann man nicht sagen, welches Protokoll sich in welchem Zusammenhang besser eignet, sondern es muß jedes Protokoll auf seine eignung hin untersucht werden. Erst dann kann eine Entscheidung getroffen werden, welches von beiden im jeweiligen Kontext das richtige ist.

Eine Gemeinsamkeit beider Protokolle besteht z.B. beim Endpunkt des SIP User Agents, der mit dem H.323-Terminal vergleichbar ist, denn sie erfüllen das gleiche Prinzip.

Dies betrifft auch den SIP-Redirect/Registrar-Server und den H.323-Gatekeeper als auch die Medienübertragung, denn beide Protokolle benutzen das gleiche RTP/RTCP.

Unterschiede sind innerhalb der Kontrollfunktionen festzustellen, wie man in der folgende Tabelle entnehmen kann [DAL99]:

	H.323	SIP
Kontrollkanal	H.245	SDP
Signalisierungsfunktion	RAS, Q.931	SDP

Als 1996 die erste Version von H.323 erschien, war das H.323-Protokoll nur für ein Kommunikations-medium im LAN, über kürzere Strecken und ohne QoS, gedacht.

Seit Beginn der H.323-Entwicklung aber gab es eine rasante Entwicklung des Internets, was ein großes Interesse für H.323 zur Folge hatte und noch immer hat. Dieses Interesse gilt insbesondere synchronen Diensten über Internet und dessen Konsequenzen. H.323 ist für die Unterstützung einer Übertragung auf längeren Strecken und für die Kommunikation zwischen LAN und SCN entwickelt worden. Mit fortgeschrittener Entwicklung und Kompatibilität ist das H.323 ein sehr komplexes und umfangreiches Protokoll geworden, was sowohl von Vorteil als auch von Nachteil sein kann.

Mit der Entwicklung von ITU wächst auch das H.323-Protokoll in das Netzwerk der Telekommunikation hinein und wird damit integriert. Das H.323-Protokoll benutzt bereits implementierte Protokolle des SCN, was deren Zusammenarbeit vereinfacht.

H.323 ist mit dem standardisierten ISDN SS7 kompatibel, weil es unter anderem die gleiche ISDN E.164 Adressierung und die gleiche ISDN Q.931 Signalisierungsfunktion für Call Set-up benutzt.

SIP ist als ein jüngeres Protokoll bisher noch nicht so viel wie H.323 getestet worden, sondern befindet sich noch in einer Aufbauphase. SIP ist von der IETF als textbasiertes Client/Server-Protokoll entwickelt worden und vergleichbar mit Internetprotokollen wie HTTP und SMTP.

Damit SIP und PSTN zusammen arbeiten können, wird eine Übersetzungsfunktion der Nachrichten zwischen den beiden Kommunikationsnetzen benötigt. Bisher gibt es lediglich einen Vorschlag [ROA00]. Das Zusammenwirken der beiden Kommunikationsnetze ist in näherer Zukunft zu erwarten. Dies gilt auch für die Videoübertragung, die bei SIP noch nicht endgültig entwickelt ist.

Es ist schwerer die verschiedenen VoIP Dienste im H.323 als im SIP zu implementieren. Die Implementierung ist von der Kodierung abhängig, da H.323 Nachrichten binär in ASN.1 PER, Packet Encoding Rules kodiert sind. Das hat zur Folge, daß Implementierungen und Fehlersuche erschwert sind.

SIP-Nachrichten werden in textbasierten UTF-8 Kodierungen übertragen und vereinfachen wesentlich die Implementierung in Programmiersprachen wie z.B. Java und Perl. In Tabelle 4.4 werden einige Eigenschaften von H.323 und SIP miteinander verglichen.

FUNCTIONALITY	H.323	SIP
CALL CONTROL SERVICES		
Call Holding	Yes	Yes
Call Transfer	Yes	Yes
Call Forwarding	Yes	Yes
Call Waiting	Yes	Yes
ADVANCED FEATURES		
Third Party Control	Yes	Yes
Conference	Yes	Yes
Capability Exchange	Yes & Better	Yes
QUALITY OF SERVICE		
Call Setup Delay	2 ~ 3 RT (Version 3)	2 ~ 3 RT
MANAGEBILITY		
Admission Control	Yes	Yes
Policy Control	Yes	No
Resource Reservation	No	No
SCALABILITY		
Complexity	More	Less
Server Processing	Stateful or Stateless	Stateful or Stateless
Inter-Server-Communication	Yes	Yes
FLEXIBILITY		
Transport Protocol Neutrality	TCP/UDP	TCP/UDP
Ease of Customitzation	Harder	Easier
INTEROPERABILITY		
Version Compatibility	Yes	Unknown
SCN Signaling Interoperability	Better	Worse
EASE OF IMPLEMENTATION		
Protocol Encoding	Binary	Text

Tabelle 4.4: Vergleich von H.323 und SIP [DAL99]

[CEN00], [DAL99]

5 Firewallprodukte

In diesem Kapitel werden die auf dem Markt befindlichen Firewallprodukte beschrieben. Die Firewalls, von denen einige Implementierungen in Kapitel 6 getestet werden, sollen mindestens ein internes LAN schützen und das H.323-Protokoll unterstützen. In Bezug auf SIP stehen dem Markt noch keine Firewallprodukte zur Verfügung, da vorhandene SIP bisher keine Marktreife erreichten. Aus diesem Grund wird dieses Protokoll in den folgenden Kapiteln vernachlässigt.

Die hier beschriebenen Firewallprodukte kommen von Spezialherstellern und bieten somit einen hohen Schutz. Damit sind die meisten Firewallprodukte nicht als frei verfügbare Quellcode oder Freeware zu installieren. Abschließend werden die Firewallprodukte in einer Tabelle zusammengefaßt und miteinander verglichen.

5.1 Checkpoint Firewall-1

Checkpoint ist mit seinem Firewall-1 Softwareprogram weltweit der führende Hersteller von Firewalls. Firewall-1 unterstützt mehr Protokolle und Applikationen als vergleichbare Produkte anderer Anbieter und dabei auch das H.323-Protokoll.

Stateful Inspection, welches die erste Ergänzung von dynamischer Paketfilter war, wird zwischen der Schicht 2 (Data Link Schicht) und Schicht 3 (Netzwerk Schicht), eingesetzt. Im Unterschied zu dynamischen Paketfiltern, welche nur die Netzwerkebene untersuchen, analysiert Stateful Inspection alle sieben Schichten im ISO/OSI Referenzmodell. Das Analysieren alle sieben Schichten ist mit Hilfe von komplettierenden Zusatzprogrammen gemacht, auf die später in diesem Kapitel genauer eingegangen wird.

Checkpoint Firewall-1 ist aus Firewall Modulen aufgebaut. Sie implementieren die dynamische Filterung auf unterschiedlichen Ebenen. Die Firewall Module machen Firewall-1 sehr flexibel, was den Anforderungen des Netzes entgegenkommt.

Wenn Firewall-1 auf einem Unix-System installiert ist, können die Firewall Module im Kernel geladen werden. Diese virtuelle Maschine im Kernel wird von Checkpoint als INSPECT Engine bezeichnet.

Die Firewall Module kommunizieren über einen Firewall Daemon (FWD) mit einem Management-Server wie in Abbildung 5.1, in der auch die Komponenten der Firewall-1 zu sehen sind. Sie bestehen aus einem Graphical User Interface (GUI), einem Management-Server und Firewall Modulen. Von dem GUI werden die gewünschten Sicherheitsregeln aus der Database des Management-Servers ausgewählt. Diese Anforderungen werden dann mit dem Firewall Modul durchgesetzt.

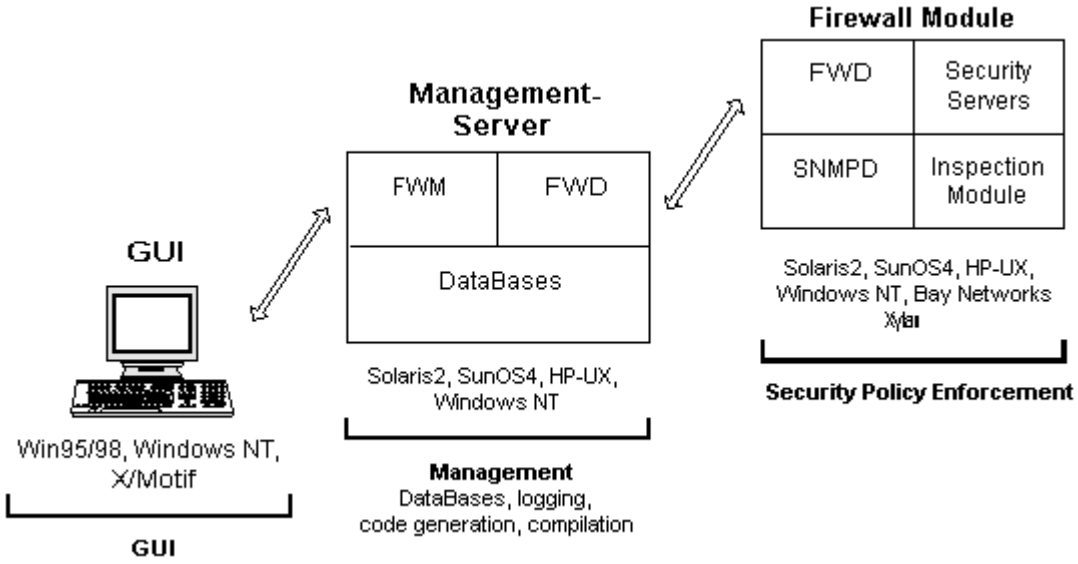

Abbildung 5.1: Struktur der Firewall-1 [STR99]

Auf dem Management-Server läuft zusätzlich ein Management Daemon (FWM), welcher die Schnittstelle zu der grafischen Oberfläche, GUI darstellt. Die Komponenten können entweder auf dem gleichen Rechner oder auf verschiedenen Rechnern installiert werden. Sind es verschiedene Rechner, können sie über die Tunnelfunktion zusammenarbeiten. Eine Bedieneroberfläche ist in dem Fall von Vorteil, wenn mehrere Firewall Module und Management-Server kontrolliert und überwacht werden müssen.

Um das interne Netz, sicherer zu machen, gibt es als Erweiterung für Firewall-1 verschiedene Zusatzprogramme. Viele dieser Programme werden von anderen Herstellern entwickelt und später durch eine Allianz mit Checkpoint certifiziert [OPS01].

Die Zusatzprogramme arbeiten in der Applikationsebene und analysieren dort die Datenpakete. Sie bilden eine gute Ergänzung zur Firewall-1, denn in der Grundausstattung ist Firewall-1 nur ein intelligenter, dynamischer Paketfilter, der nicht selbst die höheren Schichten vollständig analysiert.

Zusatzprogramme können z.B. sein:

- **Authentizierung, PKI**
- **Antivirus**
- **Content Security:** Programm, das Adressen und Inhalt von z.B. Homepages überwacht und kontrolliert
- **Intrusion Detection:** Programm, das Attacken, Einbrüche und ungewöhnliche Merkmale im eigenen Netz registriert
- **Management und Reporting**

Die meisten die Firewall-1 komplettierenden Programme, welche als Security-Servers bezeichnet werden, benötigen ein hohen Prozessor- Speicher- und RAM-Ausbau. Deswegen werden sie üblicherweise auf einem separaten Rechner im lokalen Netz oder in der DMZ installiert. Die Content Security wird z.B. von einem Security-Server ausgeführt, wie in Abbildung 5.2 dargestellt ist.

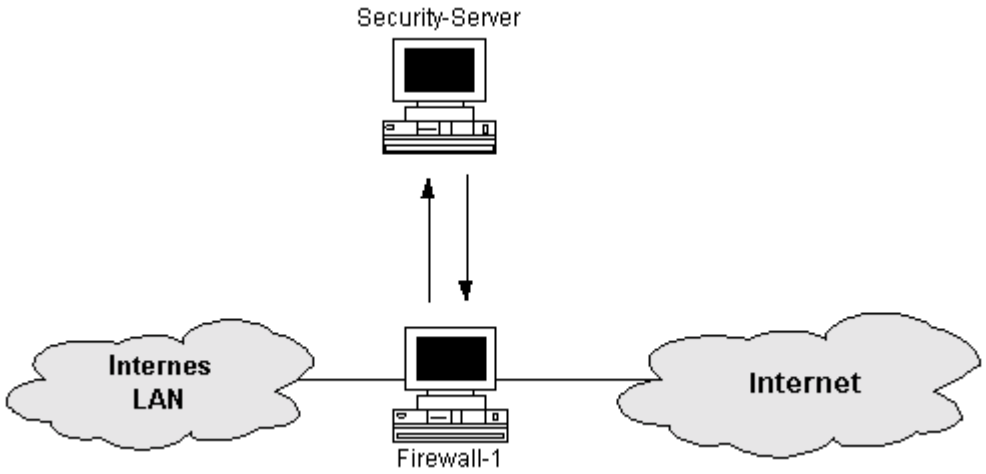

Abbildung 5.2: Verbindung zwischen internem Netz und Internet mit dem Security-Server

Die Content Security Analyse gibt es für die in der Applikationsebene installierten Telnet, Rlogin, FTP, HTTP und SMTP-Protokolle.

Die in Firewall-1 eingehenden Datenpakete werden erst mit dem Firewall Modul nach den definierten Regeln in der Bedieneroberfläche kontrolliert. Von dort aus, wenn so konfiguriert, werden auch die definierten Datenpakete zu dem Security-Server geschickt. Dies wird mit einer Proxy-Verbindung durchgeführt, deswegen muß jeder Rechner eines internen Netzes die Security-Server als Proxy-Server konfiguriert haben.

Danach werden die vorher definierten Protokolle in der Applikationsebene analysiert und ausgewertet. Es wird entschieden, ob, passend oder nicht passend, Information übertragen wird. Diese Entscheidung wird im Vergleich mit den Regeln des Security-Servers getroffen.

Proxy- und Applikationsfilter sind in Kapitel 2.2 beschrieben.

[CHE00], [POD00], [STR99]

5.2 Axent Raptor 6.5

Von der Firma Axent Technologies ist die Raptor Firewall entwickelt geworden. Raptor unterstützt eine Vielzahl von Protokollen, darunter auch das H.323-Protokoll. Raptor funktioniert im Grunde wie eine Proxy-Firewall (Kapitel 2.2), die auf der Applikationsebene die Datenpakete systematisch nach den gewünschten Anforderungen kontrolliert. Weiterhin überwacht Raptor alle durchgehenden Verbindungen und läßt die Datenpakete nur durch, wenn sie schon zu einer bekannten, etablierten Verbindung gehören.

Raptor als Proxyfilter unterstützt auch verschiedene Möglichkeiten der Manipulation von IP-Adressen. Die verschiedenen NAT Addressenumsetzungen können hier z.B Adressenversteck, Transparenz und Nachsendung sein.

Raptor ist ab Version 6.0 deutlich einfacher geworden. Dies wurde durch die Einführung einer grafischen Bedieneroberfläche erreicht, die von Axent als Raptor Management Console (RMC) bezeichnet wird.

Für ein größeres, lokales Netzwerk an verschiedenen Standorten, bietet Raptor eine höhere Sicherheit im Übertragungsverfahren zwischen den Standorten durch Unterstützung von IPsec und VPN.

[RFW99], [SYM01]

5.3 NAI Gauntlet

Die Gauntlet Firewall wurde ursprünglich von der Firma Trusted Information Systems entwickelt und später von Network Associates (NAI) übernommen. NAI ist der weltweite Hersteller der McAffe Antivirus Software, welche auch in der Firewall integriert ist.

Zur Implementierung eines privaten Netzes über Internet, wird IPsec und VPN von Gauntlet mitgeliefert. Konfiguration und Administration von Firewalls an verschiedenen Standorten lassen sich von einem Client, mit der sogenannten Global Enterprise Management System (GEMS) Software, einfach durchführen.

Die Untersuchungsmethoden von Gauntlet werden mit der Paket- und Proxyfilterung durchgeführt. Mit diesen beiden Filterfunktionen kann eine Firewall optimal konfiguriert werden.

Die Proxyfilterung mit einer hohen Sicherheit und/oder die Paketfilterung mit weniger Sicherheit, aber mit einer höheren Durchlassgeschwindigkeit, können kontrolliert und manipuliert werden.

[GAU99], [NAI01]

5.4 Linux Netfilter

Das Betriebsystem Linux mit frei verfügbarem Quellcode hat seit der Kernel Version 1.1 Unterstützung für Paketfilterung. Seit Kernel 2.3.15 wird die neueste Technik für Filter als Linux Netfilter iptables bezeichnet. Linux Netfilter bietet nicht nur Paketfilter an, wie den vorangegangenen Versionen ipfwadm und ipchains, sondern unterstützt auch Stateful Paket Filter.

5.5 Cisco Systems

5.5.1 Cisco Secure PIX Firewall

Die von Cisco selbst entwickelte Secure PIX Firewall Hardware/Software Lösung, früher als Cisco PIX Firewall bekannt, ist eine optimierte Firewall mit den besten Rezensionen in diesem Bereich, aber wahrscheinlich auch die teuerste Alternative zur Firewall.

Der Vorteil mit einem eigenen, besonders für die Firewall entwickelten, Betriebsystem ist, daß dafür kein überflüssiger Quellcode kompiliert wird. Das Betriebsystem, das am Client heterogen ist, verringert die Möglichkeiten verschiedener Attacken und Virenangriffe.

Die Secure PIX Firewall wird im 19-Zoll-Gehäuse angeboten und verwendet ähnlich der Checkpoint Firewall-1 eine dynamische Filterung, die von Cisco als Adaptive Security Algorithm (ASA) benannt wird.

5.5.2 Cisco Adaptive Security Algorithm

ASA ist der zentrale Punkt der Secure PIX Firewall und trägt die Verantwortung und Kontrolle für alle eingehenden und ausgehenden Datenpakete, welche durch die Firewall vermittelt werden. Die Datenpakete werden aufgezeichnet und aus den eingesammelten Information macht die Firewall eine dynamische Verbindungstabelle, die jede Verbindung überwacht und kontrolliert.

Die kontrollierten Daten beziehen sich auf:

- Sender/Empfänger IP-Adresse
- Randomized TCP Sequence Number
- Portnummern
- Zusätzliche TCP-Flags

5.5.3 Cisco Multimedia Conference Manager

Die multimedialen Dienste im Internet stellen ständig höhere Ansprüche und somit wird das H.323-Protokoll immer häufiger eingesetzt. Dafür gibt es eine sichere Lösung von Cisco, den sogenannten Multimedia Conference Manager (MCM). MCM ist eine erweiterte Cisco IOS Feature Set Software und wird zur Zeit für die Secure PIX Firewall und das Cisco Betriebsystem IOS Feature Set angeboten. Der MCM basiert hauptsächlich auf zwei Hauptfunktionen, dem H.323-Proxy und dem Gatekeeper.

Der Gatekeeper, der in Kapitel 4.1.2.3 ausführlich beschrieben worden ist, erfüllt Administration und Kontrollfunktionen der Endpunkte im internen Netz.

Die H.323-Proxy erstellt in Zusammenarbeit mit dem Gatekeeper die Sicherheitsanforderungen für das H.323-Protokoll. Um dieses zu ermöglichen, analysiert der H.323-Proxy die Datenpakete des H.225 Call Setups und des H.245-Kontrollkanals und öffnet dementsprechend die in den Datenpaketen lokalisierten Portnummern.

5.5.4 Cisco Routern und IOS Betriebsystem

Damit die Möglichkeit besteht MCM im IOS zu integrieren, empfiehlt sich, das H.323-Protokoll in Cisco Routern zu implementieren. Die MCM bietet Cisco für Router der Serien 2500, 2600, 3600, 7200 und MC3810 [MCM00] an.

Wie in Kapitel 2 beschrieben, werden die Firewallfunktionen häufig im Router eingebaut oder zusammen eingesetzt. Cisco unterstützt auch die Möglichkeit, in Routern eine Firewall zu konfigurieren. Sie wird dann als Cisco IOS Firewall Software Module im Betriebsystem integriert.

Funktionen einer Cisco IOS Firewall [IOS99]:

- **Standard Accessliste:** Enthält eine grundlegende statische und dynamische Paketfilterung, deren Möglichkeiten in Kapitel 2.1 beschrieben stehen. In der erweiterten Accessliste ist es möglich, die Transportschicht zu untersuchen.

- **Lock and Key:** Enthält eine dynamische Accessliste, die nur bestimmte Anwender nach Authentifizierung durch ein Paßwort durchläßt. Diese Möglichkeit öffnet spezifische Löcher in der Firewall, die nach der Session wider geschlossen werden.

- **Reflexive Accessliste:** TCP- und UDP-Paketen ist es nur erlaubt zu passieren, wenn die Verbindung vom internen Netz startet.

- **TCP Intercept:** Schützt gegen die TCP SYN-flooding Attacke, die eine Form der Denial of Service Attacke ist.

- **Network Address Translation:** Schützt die internen IP-Adressen davor, vom Internet aus gesehen zu werden.

- **IPsec:** Verschlüsselt bestimmte Datenpakete.

- **Event Logging:** Sammelt Information über Datenpakete und führt Statistik.

- **User Authentication:** Hilft vor Zugriff von ungehörigen Anwendern.

Um das H.323-Protokoll zu unterstützen, wird die MCM mit einem Gatekeeper und dem H.323-Proxy im IOS Betriebsystem eingesetzt und konfiguriert.

Für ein internes Netz, das H.323-fähig ist und durch die im Router eingebaute Firewall geschützt ist, existiert ein Firewall Software Module, das sogenannte Cisco IOS Firewall Feature Set, eine Erweiterung der IOS Firewall. Das erweiterte Firewall Software Module enthält die sogenannte CBAC (Context Based Access Control) Funktion [CBA99]. CBAC ist ein Applikationsfilter und empfiehlt sich deswegen besonders gut für die Unterstützung des H.323-Protokolls und anderer multimedialer Dienste.

CBAC stellt Connection Tracking der geführten Accesslisten her. Durch die Analyse des Applikationsfilters weiß CBAC auch, welche Portnummern für eine Verbindung temporär geöffnet werden müssen. Diese temporär geöffneten Portnummern werden direkt geschlossen, wenn die Verbindung abgebrochen ist.

Für private und nicht vermittelbare IP-Adressen hinter einem Cisco Router und mit der Cisco IOS NAT Funktion [NAT01] ausgestattet, gibt es mit einigen Einschränkungen trotzdem die Möglichkeit, das H.323-Protokoll zu benutzen. Für die Unterstützung von H.323 mit NAT wird auch das Cisco IOS Firewall Feature Set gebraucht.

[CBA99], [DEP00], [IOS99], [MCM00], [NAT01], [PIX00]

5.6 Zusammenfassung der Firewallprodukte

Checkpoint als marktführender Hersteller von Firewallprodukten ist ein bevorzugter Arbeitspartner vieler Datensicherheitssoftwarefirmen. Deswegen gibt es auch viele Zusatzprogramme mit unterschiedlichen Funktionen für Firewall-1. Mit diesen Zusatzprogrammen werden alle sieben Sichten in dem ISO/OSI Referenzmodell gesichert.

Die Unterschiede zwischen der Firewall-1 und dem Linux Netfilter sind gering, da beide in den Schichten drei und vier des ISO/OSI Referenzmodells arbeiten.

Ein Vorteil der Firewall-1 ist die grafische Bedieneroberfläche. Sie hilft wesentlich bei der Vereinfachung und Übersichtlichkeit, was für die Sicherheit sehr wichtig ist. Vorteile von Linux sind das Vorliegen des Quellcodes und die Kostenlose Verfügbarkeit.

NAI Gauntlet und Symmantec Raptor sind zwei Applikationsfilter, die deswegen mehr Zeit und Leistung in Anspruch nehmen, weil sie die Datenpakete analysieren.

Gleichzeitig bedeutet dies aber auch die beste Kontrollmethode, z.B. werden bei dem H.323-Protokoll die dynamischen UDP-Portnummern in der Applikationsebene ausgetauscht. Der Proxyfilter analysiert und öffnet nur die definierten Portnummern und schließt diese automatisch wieder nach dem Verbindungsabbau.

Cisco Systems hat mit PIX gezeigt, daß sie Firewalls mit hoher Durchlaßrate bauen können. Die meisten der von Cisco entwickelten Firewalls sind paketfilterbasiert und werden zusammen mit dem Router eingesetzt. Mit der steigenden Nachfrage, hat die Firma auch einige Lösungen für die Unterstützung von VoIP zeigen können, die im Router integriert sind.

Hersteller-Modelle	Betriebsystem	Filtermethode	H.323-Support	GUI Support	VPN IPsec	Antivirus Scanning	Log-Support
Checkpoint Firewall-1	Sun Solaris 2.6, 2.7 HP-UX IBM AIX Red Hat Linux NT, 2000	Stateful Inspection	Ja	Ja	Ja	Nein	Ja
Symmantec Raptor	Sun Solaris 2.6, 2.7 HP-UX TRUE 64 NT, 2000	Applikationsfilter	Ja	Ja	Ja	Nein	Ja
Cisco PIX	Cisco IOS	Paket und Applikationsfilter	Ja	Ja	Ja	Nein	Ja
NAI Gauntlet	Sun Solaris 2.6 HP-UX NT	Paket und Applikationsfilter	Ja	Ja	Ja	Ja	Ja
Linux Netfilter	Linux	Stateful Paketfilter	Ja	Nein	Nein	Nein	Ja

Tabelle 5.1: Firewallprodukte für synchrone IP-Dienste

6 Beispielimplementation

6.1 Das Comlab-Netzwerk

Für die Testversuche von VoIP über eine Firewall wurde ein Testnetzwerk im Labor für Kommunikationssysteme (Comlab) aufgebaut. Das Comlab hat ein IP-Netz mit Subnet Netmask 255.255.255.224 und kann damit maximal 32 IP-Adressen benutzen. Diese liegen im Bereich von 139.30.208.0 - 139.30.208.31. Im Grunde ist das Labor ein Fast Ethernet 10/100 Mbit/s LAN, das mit dem Cisco Catalyst 5000 Switch verbunden ist. Darüber hinaus ist es über einen Cisco Lightstream 1010 ATM Switch mit dem Universitätsnetz verbunden. Die Übertragungsrate des universitären Backbone-Netzwerkes beträgt 155 Mbit/s. Außerdem gibt es einen Cisco 4700 Router und eine Radvision L2-W323 [WAL00] für IP-Telefonie. Diese kann entweder direkt über das Internet oder über den eingebauten Gateway, der mit dem Siemens Hicom 300 PBX verbunden ist, betrieben werden.

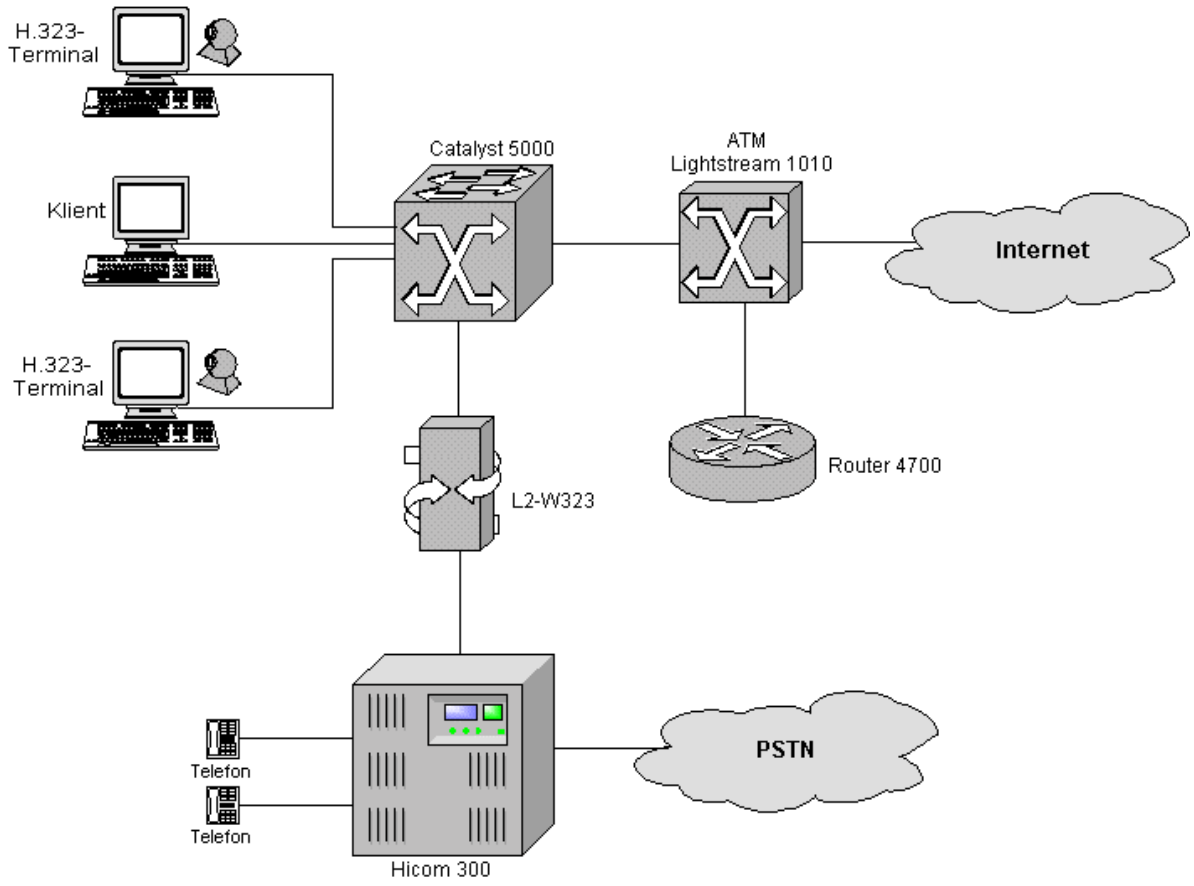

Abbildung 6.1: Netzaufbau im Labor für Kommunikationssysteme (April 2001)

Für die Testversuche mit VoIP über verschiedene Implementierungen einer Firewall ist ein kleines Netzwerk mit verschiedenen Firewallösungen im Labor konfiguriert geworden. Im Wesentlichen besteht es aus drei Rechnern, die in den nächsten Kapiteln erklärt werden.

6.2 Linux Firewall Testnetzwerk

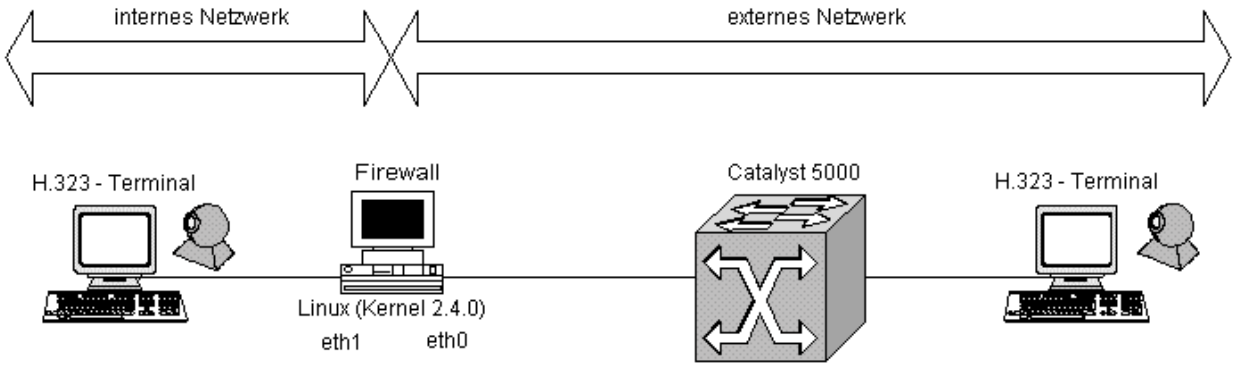

Abbildung 6.2: Linux Testnetzwerk

Der Switch Cisco Catalyst 5000 ist weiterhin wie in Abbildung 6.1 verbunden. Die Firewall ist auf einem Rechner mit x86-Plattform installiert, dessen Betriebssystem Linux SuSE 7.1 (Kernel 2.4.0) ist. Darüber hinaus sind zwei Netzwerkkarten eingebaut. Die erste ist eth0, die mit dem Switch und damit auch mit dem Internet verbunden ist. Die zweite ist eth1, die mit dem geschützten, internen Netzwerk verbunden ist. An den beiden H.323-Terminals ist Microsoft Windows 98 als Betriebsystem installiert, und als Kommuniktationssoftware zwischen den Terminals wird Microsoft Netmeeting benutzt.

Konfiguration:

Linux Firewall:

eth0:	IP:	139.30.208.12
	Netmask:	255.255.255.224
	Gateway:	139.30.208.27 (Cisco 4700 Router)
	DNS:	139.30.208.2, 139.30.8.40

eth1:	IP:	192.168.1.1
	Netmask:	255.255.255.0
	Gateway:	139.30.208.27 (Cisco 4700 Router)
	DNS:	139.30.208.2, 139.30.8.40
H.323-Terminal (intern):	IP:	192.168.1.10
	Netmask:	255.255.255.0
	Gateway:	192.168.1.1
	DNS:	139.30.208.2, 139.30.8.40
H.323-Terminal (extern):	IP:	139.30.208.9
	Netmask:	255.255.255.224
	Gateway:	139.30.208.27 (Cisco 4700 Router)
	DNS:	139.30.208.2, 139.30.8.40

Die IP-Adressen im internen Netzwerk (192.168.1.x) sind sogenannte private IP-Adressen. Sie werden häufig als nicht vermittelbare IP-Adressen benannt und für Netzwerke ohne Internetanschluß oder für mehreren Clients, die einen Internetanschluß teilen, benutzt. Insgesamt gibt es drei verschiedene IP-Adressenblöcke, die für private Netzwerke reserviert sind. Diese sind:

10.0.0.0	–	10.255.255.255	Klasse A
172.16.0.0	–	172.31.255.255	Klasse B
192.168.0.0	–	192.168.255.255	Klasse C

In diesem Testnetzwerk werden für das interne Netzwerk die privaten IP-Adressen der Klasse C benutzt. Hier wäre es möglich bis zu 253 Clients zu konfigurieren. 192.168.1.0 und 192.168.1.255 sind für das Netzwerk reservierte IP-Adressen. Die IP-Adresse 192.168.1.1 wird vom Gateway genutzt.

Für Rechner mit privaten, nicht vermittelbaren IP-Adressen, wie in diesem Testnetzwerk, gibt es trotzdem eine Möglichkeit, das Internet zu erreichen.

Unter Linux wird dieser Vorgang Masquerading bzw. SNAT. Im Testnetzwerk wird beim Linux Firewall-Rechner der Kernel so kompiliert, daß er Masquerading unterstützt.

Wenn die Datenpakete des internen Rechners die Firewall passieren, wandelt die Firewall die Header der internen Adresse um in ihre eigene öffentliche, vermittelbare IP-Adresse, welche an der eth0 konfiguriert ist. Wenn die Antwortpakete zurückkommen, übersetzt die Firewall die Zieladresse mit der Adresse des internen Rechners. Damit ist es möglich, von dem internen Netzwerk, Verbindungen mit verschiedenen Diensten des Internets herzustellen. Umgekehrt ist es aber nicht möglich, eine Verbindung aus dem Internet mit einem Rechner des privaten Netzwerkes aufzubauen. Deswegen ist es den Anwendern unmöglich, sich direkt ins Internet und ohne Adressenumsetzung zu schalten. Das bedeutet z.B., daß VoIP nur in eine Richtung funktionieren würde. Eine Lösung für Kommunikation in beide Richtungen wird in Kapitel 6.3 gegeben.

Ein Vorteil mit privaten IP-Adressen die Masquerading nutzen ist, daß die internen IP-Adressen nicht angezeigt werden, sondern nur die der externen Netzwerkkarte. Das erschwert Attacken gegen das interne Netzwerk.

6.3 Phonepatch

Es gibt eine Lösung für das oben genannte Problem der privaten IP-Adressen im internen Netzwerk, die eigentlich nicht von außen aufgerufen werden können. Diese kommt von der australischen Firma Equivalence Pty Ltd [EQU01] und wird Phonepatch genannt. Die Firma koordiniert auch die Arbeit des allgemeinen *Open H323 Project*, in welchem sie eine frei verfügbare Open Source Implementierung für das H.323-Protokoll bereitstellen.

Dieses Programm funktioniert wie eine Telefonvermittlungsstelle. Es vermittelt Anrufe zwischen dem internen und externen (Internet) Netzwerk. Bei einem Anruf, entweder vom externen oder internen H.323-Terminal, wird erst eine Verbindung mit Phonepatch hergestellt, das an der Firewall zwischen internem und externem Netzwerk installiert ist. Dann vermittelt und sendet Phonepatch die Datenpakete zurück an die beteiligten Teilnehmer.

Auf dem Firewall Rechner ist Phonepatch zusammen mit der Firewall Software installiert. Zu bemerken ist hier, daß das Phonepatch selbst keine Firewall Software enthält. Damit die H.323-Verbindungen aufgebaut werden können, müssen die ILS und HTTP-Protokolle geöffnet sein. Phonepatch ist so konfiguriert, daß es die unten aufgelisteten Portnummern benutzt:

- 389 zum ILS und zur LDAP-Unterstützung
- 1084 zur Vermittlungsstelle und Administrationsport
- 1503 für T.120 Konferenzunterstützung
- 1720 für H.225 Call Setup
- 3000 als Startportnummer, um einen freien TCP-Port zum H.245 Kontrollkanal zu finden
- 3000 als Startportnummer, um einen freien UDP-Port für die RTP/RTCP-Medienübertragung zu finden

Abbildung 6.3 zeigt, wie Phonepatch die ILS, HTTP und H.323-Protokolle behandelt. Die Linux Netfilter Firewall-Software wird hier separat im Firewall-Rechner konfiguriert.

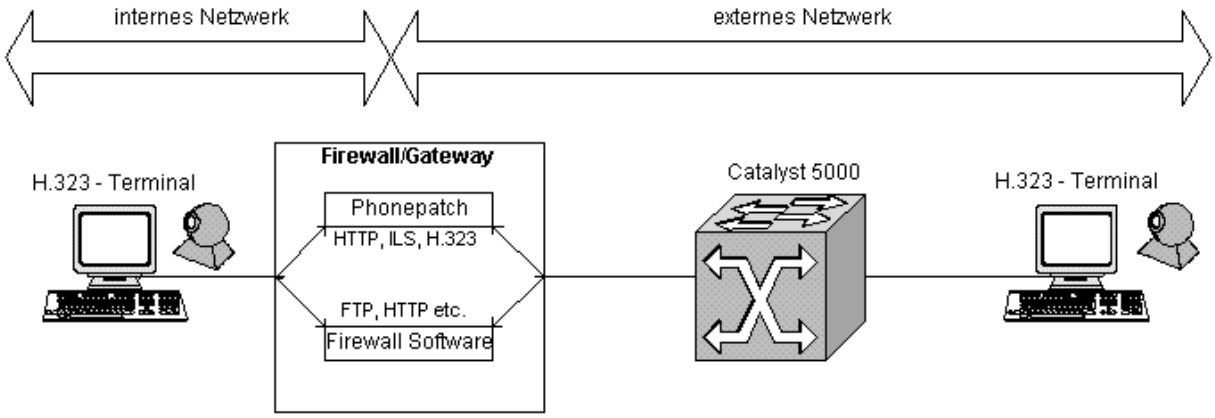

Abbildung 6.3: Phonepatch in der Firewall

Es können entweder eine IP-Adresse oder der Name eines Benutzers, der an einem ILS registriert ist, angerufen werden. Der interne H.323-Terminal schickt seine Datenpakete mit dem ILS-Protokoll durch den Phonepatch. Daher weiß er, welcher Name zu welcher IP-Adresse gehört und kann entsprechende Verbindungen herstellen.

Allerdings gibt es noch Begrenzungen in Phonepatch: wenn z.B. ein Name für einen Anruf benutzt wird, ist Microsoft Netmeeting das einzige Programm, das die Verbindung herstellen kann. Außerdem muß jeder Teilnehmer in der gleichen ILS angemeldet sein.

Ein Anruf kann vom externen zum internen Rechner oder umgekehrt durchgeführt werden.

Zum Beispiel können in diesem Testnetzwerk in den Webbrowser des externen Rechners die IP-Adresse und Portnummer der Firewall eingegeben werden. In diesem Fall lautet die IP-Firewall-Adresse: com12.comlab.uni-rostock.de:1084. Dann erscheint die Vermittlungsstelle auf der Firewall wie in Abbildung 6.4 dargestellt.

Abbildung 6.4: External Switchboard im Phonepatch

Von dort folgt entweder die IP-Adresse des gesuchten Teilnehmers im Call Destination Feld, in diesem Fall 192.168.1.10, oder man benutzt den Namen des gesuchten Anwenders, der im ILS registriert ist.

Eine Begrenzung gibt es auch, wenn das T.120-Protokoll für Chat und Whiteboard benutzt wird. Das T.120-Protokoll benutzt, wie in Tabelle 4.3, einen konstanten TCP-Port. Bisher arbeiten nur die T.120- Funktionen, wenn ein Anruf über einen ILS gemacht wird. Dies hat sich auch in der Testimplementierung gezeigt.

Die sicherste Variante wäre Phonepatch in Verbindung mit einem LDAP-Server auf dem Rechner in der DMZ zu konfigurieren und die Firewall von anderen Server-Diensten ganz zu trennen und auf einem eigenen Rechner laufen zu lassen. Auf dem LDAP-Server können die Mitarbeiter mit Namen registriert werden und sind damit gut vom Internet aus erreichbar.

6.4 Linux Netfilter iptables

Für die Unterstützung von Phonepatch ist Linux Netfilter iptables als Modul im Kernel kompiliert. Iptables arbeitet an der Firewall parallel mit dem Phonepatch, siehe Abbildung 6.3. Linux iptables muß das interne Netzwerk schützen und die H.323-Protokolle zulassen. Dafür wird das Firewallskript als bootskript kompiliert. Dieses ist im Anhang 1 aufgeführt.

Das Firewallskript erlaubt allen Anwendern im internen Netz, erst eine Verbindung zum Phonepatch und von dort nach außen, zu initiieren. Für externe Nutzer sind nur einige notwendige Portnummern geöffnet, wie z.B ssh-Server und 1084. Die letzere wird von Phonepatch benutzt um eine HTTP-Verbindung aufzusetzen. Wenn ein Anwender von außen eine Verbindung zu einer der geöffneten Portnummern herstellt, werden die weiteren benötigten Portnummern für die Verbindung automatisch geöffnet, welche im Normalfall geschlossen sind. Iptables hat den ESTABLISHED-Befehl, welcher die FLAGS in jedem Paket liest und dann für die Antwortpakete weiß, ob die Verbindung schon initiiert ist.

Eine kleine Schwachstelle ist hier in der Sicherheit, denn wenn eine ausgehende Verbindung initiiert wird, muß das Firewallskript neue, von der Firewall ausgesendete TCP-Verbindungen über Portnummer 1023 zum aufgerufenen Clients zulassen.

Dies ist für den H.225 Call Control Channel und H.245-Kontrollkanal notwendig. Das gleiche gilt für eingehende UDP-Datenpakete an der Firewallportnummer von 3000 bis 3010, sie müssen offen bleiben.

Es ist nicht die Aufgabe des Firewallskriptes die Rechner gegen alle Attacken zu schützen, sondern es soll nur zeigen, wie die H.323-Datenpakete mit Hilfe von Phonepatch durchgelassen werden, ohne daß man alle Ports öffnen muß.

6.5 Checkpoint Firewall-1 Testnetzwerk

Checkpoint Firewall-1 ist die in Unternehmen am häufigsten angewendete Firewall. Daher ist es sinnvoll, auch diese Lösung mit dem Testnetzwerk zu testen. Firewall-1 läuft mit verschiedenen Betriebsystemen. Für den professionellen Einsatz wird aber die SUN Sparc-Plattform empfohlen, die auch in diesem Fall verwendet wird. Die SUN Technologie zeichnet sich gegenüber x86-Plattformen durch besondere Laufzeitstabilität aus. Das installierte Betriebsystem ist Solaris 8, das von SUN mit optimierten Leistungs- und Netzwerkfähigkeiten mit der SUN Sparc-Plattform als Firewall entwickelt worden ist.

In der Datenspezifikation wird angegeben, daß Firewall-1 im Einsatz mit dem Solaris Betriebsystemen 2.6 und 2.7 unterstützt ist. In diesem Versuch hat es sich geziegt, daß auch Solaris 8 kompatibel ist.

Das Solaris Betriebsystem ist ein UNIX Betriebsystem, ähnlich Linux, und hat eine grafische Oberfläche, welche die Handhabung des Systems vereinfacht. Trotz dieser grafischen Oberfläche, ist es nicht so einfach und flexibel zu bedienen wie Linux.

SUN Solaris Rechner mit zwei Netzwerkkarten unterstützt das IP-Forwarding im Kernel automatisch, wenn zwei oder mehr Netzwerkkarten vorhanden sind. Der Rechner bekommt dann eine Vermittlungsfunktion zwischen den verschiedenen Netzwerken. Automatisches Masquerading oder automatische Adressenumsetzung wird vom Kernel nicht unterstützt und muß von einem anderen Programm durchgeführt werden. Frei verfügbare Programme sind selten und meistens nicht besonders kompatibel und benutzerfreundlich. Unter anderem wurde die Axent Raptor Firewall, welche im Kapitel 5.2 beschrieben ist, getestet.

Dabei funktionierte die Adressenumsetzung und damit auch die Paketvermittlung nicht. Mit der auf dem SUN Solaris Rechner installierten Firewall-1 ist die Kompatibilität und die Benutzerfreundlichkeit größer und es funktionieren Adressenumsetzung und Paketvermittlung.

Bei der Installation von Firewall-1 kann der Aufbau des Netzwerkes sehr komplex sein und an verschiedenen Rechnern durchgeführt werden.

In Kapitel 5.1 sind die Benutzeroberfläche, Management-Server und Inspect Engine beschrieben, welche separat und je nach Anforderung installiert werden können. In diesem Testfall wurde eine sogenannte standalone Installation mit allen Modulen an dem SUN Solaris Rechner gemacht. Das gesamte Testnetzwerk ist mit dem Linux Testnetzwerk vergleichbar, welches bereits in Kapitel 6.2 näher beschrieben ist. Das Testnetzwerk mit Checkpoint Firewall-1 als Firewall ist in Abbildung 6.5 dargestellt.

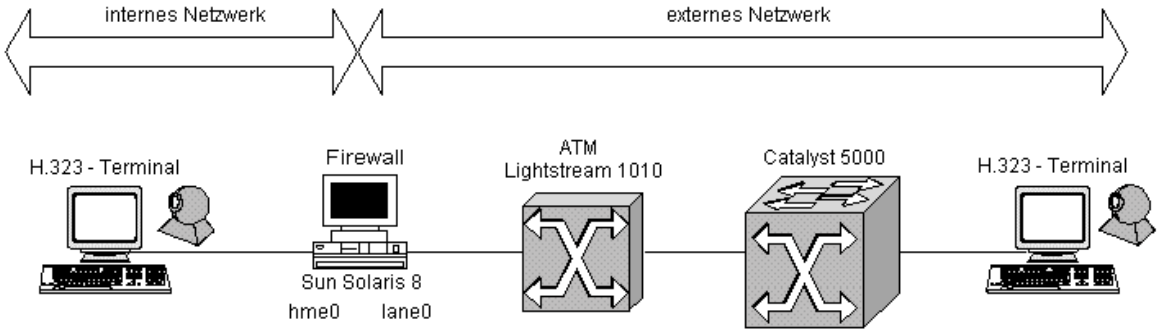

Abbildung 6.5: Testnetzwerk mit Checkpoint Firewall-1

Auf dem Firewall-Rechner ist das Betriebsystem Solaris 8 installiert, und es sind dort zwei Netzwerkkarten (Fast Ethernet und ATM) vorhanden. Die ATM-Karte ist (lane0), mit dem Lightstream 1010 ATM Switch und damit auch mit dem Internet verbunden. Das zweite interface ist hme0, welches mit dem geschützten, internen Netzwerk verbunden ist. An den beiden H.323-Terminals ist Microsoft Windows 98 als Betriebsystem installiert, und als Kommuniktationssoftware zwischen den Terminals wird Microsoft Netmeeting benutzt.

Konfiguration:

lane0:	IP:	139.30.208.1
	Netmask:	255.255.255.224
	Gateway:	139.30.208.27 (Cisco 4700 Router)
	DNS:	139.30.8.2, 139.30.8.40
Hme0:	IP:	192.168.1.1
	Netmask:	255.255.255.0
H.323-Terminal (intern):	IP:	192.168.1.10
	Netmask:	255.255.255.0
	Gateway:	192.168.1.1
	DNS:	139.30.8.2, 139.30.8.40
H.323-Terminal (extern):	IP:	139.30.208.9
	Netmask:	255.255.255.224
	Gateway:	139.30.208.27 (Cisco 4700 Router)
	DNS:	139.30.208.2, 139.30.8.40

Das H.323-Protokoll wird vollständig von der Firewall-1 unterstützt und öffnet nur die benutzten Portnummern für eine bestimmte Verbindung. Firewall-1 unterstützt NAT, die es ermöglicht, Applikationen des privaten, internen Netzes wie Netmeeting extern zu benutzen.

In der Firewallpolicy, siehe Abbildung 6.6 für ausgehende Verbindungen, müssen das Netmeeting und die T.120-Services gewählt werden. Im Netmeeting sind schon H.323- und LDAP-Services vorhanden. Um mit dem internen Rechner via NAT kommunizieren zu können, muß man entweder den Hide- oder Static-Befehl nutzen.

Hide wird für ausgehende Datenpakete dann verwendet, wenn sich mehrere Rechner im internen Netzwerk befinden, die die gleiche IP-Adresse der externen Netzwerkkarte des Gateways (hier 139.30.208.1) benutzen. Für Static NAT gilt das Verhältnis eins zu eins, da jede private IP-Adresse eine statisch zugeordnete, öffentliche IP-Adresse haben muß.

Für eingehende Anrufe durch Firewall-1 müssen die gleichen Services wie für die ausgehenden Verbindungen gewährleistet sein. Erst wird eine öffentliche IP-Adresse in der Firewall-1 von einem externen Anrufer aus dem Internet aufgerufen und dann von dort aus zur privaten IP-Adresse vermittelt. Bei eingehenden Anrufen ist nur Static NAT geeignet, um herauszufinden, an wen im internen Netzwerk der Anruf gerichtet ist.

Static NAT bringt aber auch einige Nachteile mit sich:

- Benötigt viele öffentliche IP-Adressen.
- Das interne Netz ist einfacher zu finden, um sich einen Überblick über den Aufbau zu verschaffen.
- Jeder interne Rechner ist von außen erreichbar.
- Lange Routingtabellen in Router und Firewall.

6.5.1 Konfiguration Firewall-1

Die Firewall-1 wurde, wie zuvor erläutert, mit Bedieneroberfläche, Management-Server und Firewall Modulen auf dem SUN Solaris Rechner installiert. Die Firewall-1 führt die Adressenumsetzung durch und bestimmt die Regeln für die Vermittlung der Datenpakete durch die Firewall.

Die grafische Bedieneroberfläche der Firewall-1 bietet einen übersichtlichen Firewall Policy Editor, welcher die geltenden Regeln und die Adressenumsetzungen definiert.

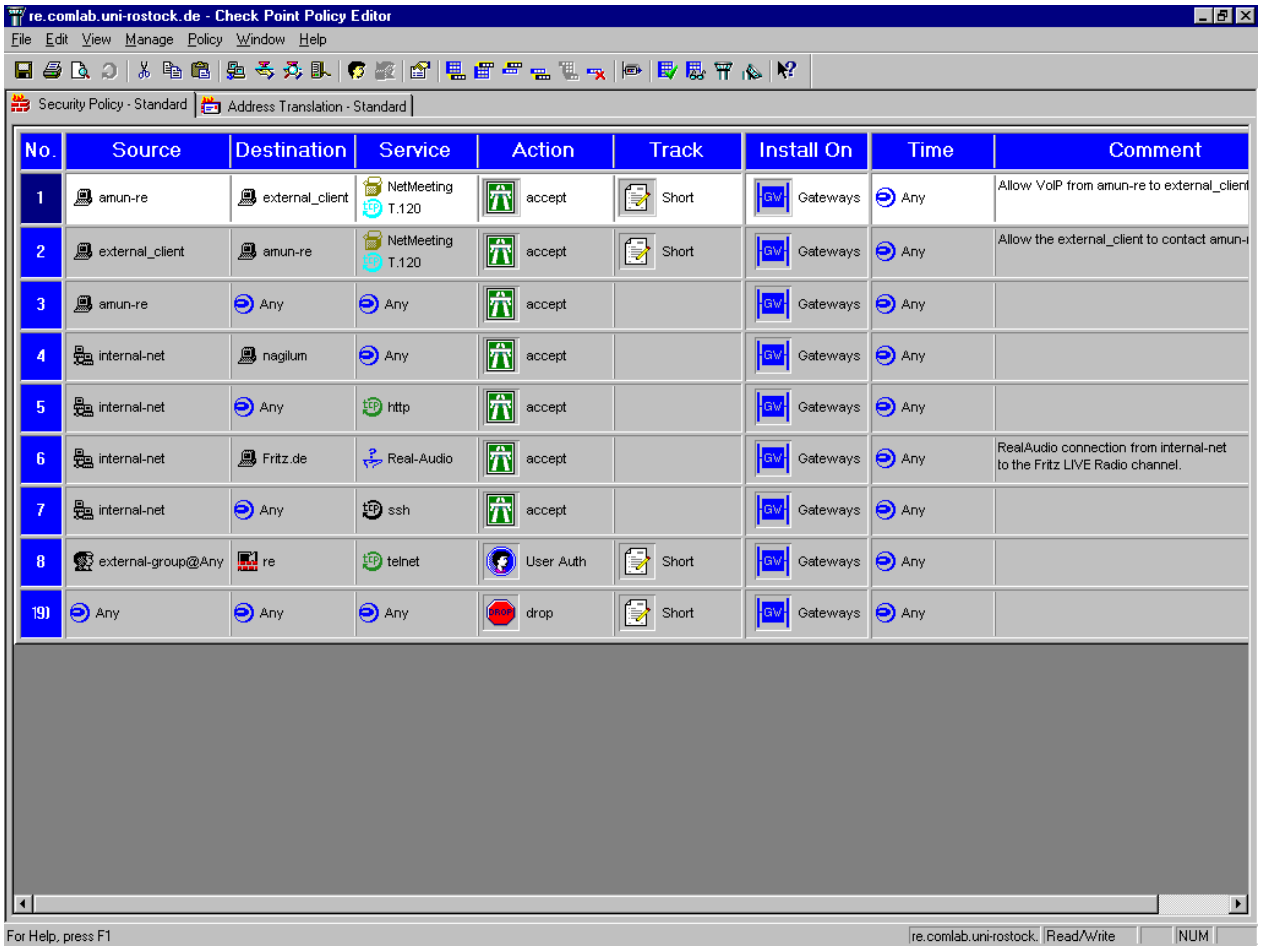

Abbildung 6.6: Checkpoint Firewall-1 Policy Editor

Die durchgehenden Datenpakete werden nach den festgelegten Firewallregeln getestet, wenn eines diesen Regeln entspricht, entscheidet die Firewall-1, was mit dem Paket zu tun ist.
Für die Definition der Regeln, müssen zunächst die Objekte bestimmt und konfiguriert werden. Die Objekte können z.B. Workstations, Networks, Domains, Users oder Groups sein. Zur Bestimmung der Regeln gibt es folgende Optionsspalten:

Optionsspalten	Eigene Daten
• Source	amun-re
• Destination	external-client
• Service	Netmeeting, T.120
• Action	Accept
• Install On	Gateways
• Time	Any

Mit dieser Lösung ist es möglich, synchrone Dienste wie das H.323-Protokoll zu nutzen, ohne daß für jede, beim Verbindungsaufbau dynamisch bestimmte Portnummer, eine Öffnung erfolgen müßte. Die Firewall-1 analysiert den Verbindungsaufbau und läßt nur die für die IP-Adresse und Portnummern spezifizierten Datenpakete durch. Wenn die Verbindung beendet wird, werden die geöffneten Ports nach einer Zeit automatisch geschlossen. Bei UDP-Datenpaketen beträgt die Standardzeit 40 Sekunden und für TCP-Datenpakete 50 Sekunden.

6.5.2 Scanning der Firewall

Durch den Einsatz von verschiedenen Netzwerkadministratoren und Hackerwerkzeugen, wurden die beiden Firewalls gegen verschiedene Angriffe vom externen Netzwerk getestet. Diese Werkzeuge sind meist als frei verfügbare Programme im Internet und relativ einfach zu beschaffen und anzuwenden.

Häufig vorkommende Programme sind hier: NMAP, SAINT, Nessus und Hping.

NMAP ist der beste auf dem Markt befindliche Portscanner. Er enthüllt offene Portnummern und die laufenden Dienste auf dem gescannten Ziel. Das Scanning kann mit verschiedenen Protokollen wie z.B. TCP, UDP und ICMP realisiert werden. Weiter werden verschiedene FLAG-Bits gesetzt und auch das Betriebsystem kann mittels NMAP enthüllt werden. Aufgrund der überbrachten Informationen hat der Angreifer nun einen klaren Überblick über sein Ziel gewonnen und kann sich danach auf weniger geschützte Stellen spezialisieren.

SAINT ist ein Netzwerkadministratorwerkzeug, welches die Schwachpunkte in einem Rechner oder Netzwerk aufzeigt. Hier wird auch ein Portscanner benutzt.

Nessus ist funktional mit SAINT zu vergleichen. Der größte Unterschied ist, daß Nessus über 650 Exploits enthält, welche nach bestimmten Anforderungen ausgewählt und einfach gegen das Angriffziel gerichtet werden. Die Exploits können z.B. DoS, backdoors, Remote file Access, und Firewall-Attacken ausführen. Als Portscanner benutzt Nessus NMAP.

Hping ist ein Netzwerkwerkzeug, ähnlich dem berühmten Pingprogramm, aber mit der Möglichkeit benutzerdefinierte TCP- UDP- und ICMP –Datenpakete zu schicken und eine Antwort zu erhalten. Mit Hping ist es weiterhin möglich, verschiedene Paketgrößen, Portnummern, Fragmentierungen, spoof-adressen, FLAG-Bits usw. selbst festzulegen.

Beispielsweise bei UDP-Verbindungen im H.323-Protokoll bleiben die gewählten Portnummern für eine gewisse Zeitspanne offen. Das macht es außenstehenden Rechnern möglich, andere Datenpakete mit gleicher Quell/Ziel-IP-Adresse und Portnummer, wie schon in der H.323-Verbindung, zu benutzen. Der Stateful Paket Filter bezieht sich in der Verbindungsliste für UDP-Datenpakete leider nicht auf die FLAG-Bits und Sequenznummer.

6.6 Schlußfolgerung

Es wurden die Firewall-Lösungen von Linux Netfilter und Firewall-1 der Firma Checkpoint, in Verbindung mit dem H.323-Protokoll, getestet und miteinander verglichen.

Vor einem möglichen Einsatz im Unternehmen, sollten folgende Fragen Beachtung finden:

- Wie wichtig ist es, das eigene Netzwerk und interne Dateien vor Außenstehenden zu schützen?
- Welche Investitionen müssen für einen solchen Schutz getätigt werden und wird dies vom Unternehmen rentabel gestaltet?

Beide Firewalls führen im Grunde auf der Netzwerk- und der Transportschicht die gleichen Kontrolldienste aus. Zusätzlich wird für Firewall-1 mit der grafischen Bedieneroberfläche das Arbeiten, bei gleichzeitiger Verringerung der Fehlerwahrscheinlichkeit, wesentlich einfacher und übersichtlicher, was als wichtig angesehen wird.

Für Linux laufen Projekte [FWB01], die versuchen, eine grafische Bedieneroberfläche für iptables zu implementieren. Wie sie funktionieren und ob sie jemals Praxisreife erreichen werden, ist schwer vorauszusagen, da die Entwicklungen von Privatpersonen vorangetrieben werden.

Eine allgemeine Schwäche des Stateful Paket Filters gegenüber dem Applikationsfilter liegt in den UDP-Verbindungen. Ist eine neue Verbindung aufgebaut, schreibt die Firewall die Adresse und Portnummer in ihre Verbindungsliste im Kernel. Wenn die Datenpakete von außen zurückkommen, werden sie von der Firewall nur mit der IP-Adresse und der Portnummer in der Verbindungsliste verglichen. Aufgrund fehlender FLAG-Bits kann keine Kontrolle erfolgen.

Die Phonepatch Vermittlungsstelle, welche zusammen mit dem Linux Netfilter eingesetzt wurde, bewirkte daß die Rechner mit privaten IP-Adressen von außen erreichbar sind. Alle dynamisch vergebenen Portnummern von 3000 bis 3010 wurden geöffnet. Über die Anzahl zur verfügung gestellten Portnummern hinaus können keine Verbindung aufgebaut werden.

Firewall-1 unterstützt das H.323-Protokoll, aber die geöffneten Verbindungen bleiben für eine gewisse Zeitspanne offen. Bei einem Angriff könnten Datenpakete mit anderem Inhalt in die Firewall gelangen und das System schädigen. Dies ist aber nur dann möglich, wenn der Inhalt nicht kontrolliert wird.

Für ein LAN mit privaten IP-Adressen, wie in diesem Beispiel, hat Firewall-1 nur eine Lösung für das H.323-Protokoll, da jede private IP-Adresse auch eine öffentliche IP-Adresse auf der externen Schnittstelle der Firewall hat. In bezug auf die Sicherheitsaspekte gibt es die in Abschnitt 6.5 dargestellten Probleme. Für ausgehende Verbindungen funktioniert Firewall-1 sehr gut.

Durch die unterschiedlichen Netzwerke und Hackerwerkzeuge konnten umfangreiche Informationen über die getesteten Firewalls gewonnen werden. Dies wurde mittels des Scanningwerkzeuges NMAP ermöglicht. Die in Kapitel 6.5.2 durchgeführten Tests, die Firewalls mit Hilfe von Exploits abstürzen lassen, führten zu keinen Abstürtzen, was positiv zu bewerten ist. Es wurden keine Einschränkungen im Datenverkehr festgestellt. Die Firewalls haben sich bewährt.

7 Ergebnisse und Ausblick

Mit der Entwicklung neuer Technologien sind Datenübertragungsgeschwindigkeit und Prozessorkapazität deutlich erhöht worden. Die Technologien eröffnen gleichzeitig die Möglichkeit, synchrone Dienste wie VoIP über Internet in Echtzeitübertragung zu nutzen. Dieses neue digitale Medium erweitert die Mittel der Kommunikation, die sich nicht länger nur auf z.B. Sprache stützen, sondern über das nun auch Video, Chat und Whiteboard möglich sind.

Das H.323-Protokoll wurde von ITU entwickelt und forciert das VoIP. Trotz anderer Vorschläge wie SIP ist das H.323-Protokoll mit seiner weiten Verbreitung die derzeit umfangreichste und am besten unterstützte Lösung für VoIP.

Das H.323-Protokoll ist eine Zusammensetzung verschiedener Unterprotokolle. Das macht es sehr komplex, bietet dafür aber auch viele Funktionen und komplexe Möglichkeiten.

Der vielleicht größte Vorteil des VoIP liegt bei den geringen Gebühren für Ferngespräche. Die geringe Implementierung des VoIP in Unternehmen hängt von unterschiedlichen Faktoren ab. Die wesentlichsten sind, daß die Technik im Vergleich zum PSTN relativ neu und vielen Menschen unbekannt ist oder es besteht nur geringer Innovationsdruck aufgrund bereits bestehender Technologien.

Heute gibt es gute Kommunikationslösungen, die aber beim Umstieg vergleichsweise hohe Investitionskosten mitsichführen. Für die vollständige Etablierung muß VoIP, im Verhältnis zu PSTN, genauso einfach benutzbar sein und mit gleichwertiger Sprachqualität dienen können.

Ein anderer Aspekt, welcher in dieser Diplomarbeit behandelt wurde, ist die Integration des VoIP in das private LAN, ohne den Sicherheitsaufwand zu vernachlässigen. Der Schwerpunkt lag auf der Firewall als letztem Gerät zwischen privatem LAN und Internet.

Was den Einsatz des H.323-Protokolls über Firewalls so schwierig gestaltet, ist, daß eine Verbindung aus verschiedenen dynamischen TCP- und UDP-Portnummern besteht, die im Verbindungsaufbau über Applikationsebenen ausgetauscht werden. Eine weitere Ursache ist, daß das Client/Server-Verhältnis im asynchronen Dienst hier nicht existiert.

Für das H.323-Protokoll sind normale statische Paketfilter, nicht geeignet. Ursache dafür ist, daß er alle Portnummern über 1023 geöffnet haben muß, um funktionieren zu können, was nicht akzeptiert werden kann.

Besser geeignet ist der weiterentwickelte Stateful Paket Filter, der ebenfalls die Möglichkeit hat, auf den FLAG-Bits basierend eine Verbindung zu stützen. Informationen über die Verbindung werden ebenfalls für eine konstante Zeit in einer Verbindungsliste gespeichert. Nachteilig ist, daß die benutzten Portnummern nicht direkt nach dem Verbindungsabbau geschlossen werden.

Die optimale Lösung für das H.323-Protokoll ist der Applikationsfilter. Ein Schwerpunkt dieser Diplomarbeit lag darin, herauszufinden, welche Applikationsfilter dieses Prokoll auf dem Markt unterstützen. Der Applikationsfilter analysiert den Dateninhalt der gewonnenen Information in der Applikationsebene und weiß somit, welche dynamischen Portnummern für die Verbindung geöffnet werden müssen. Nur bei einigen Applikationsfiltern hat sich gezeigt, daß sie das H.323-Protokoll [Tabelle 5.1] unterstützen. Die Ursache dafür liegt darin, daß das Protokoll relativ neu ist und es sich somit für den Applikationsfilter als schwierig darstellt, die wichtigen Daten in der Applikationsebene zu analysieren.

In praktischen Beispielen wurden die beiden Stateful Paket Filter "Linux Netfilter" und "Checkpoint Firewall-1" getestet. Hinter der externen Schnittstelle der Firewall zum Internet wurden private IP-Adressen benutzt und eine NAT gemacht. In den Beispielen hat die H.323-Verbindung zwischen internem und externem Rechner über die Firewalls in beide Richtungen funktioniert. Für Firewall-1 funktioniert die eingehende und ausgehende Verbindung nur, wenn jede private IP-Adresse, eine öffentliche IP-Addresse zugeteilt bekommen hat. Bei Linux Netfilter und die Phonepatch Vermittlungsstelle können eingehende und ausgehende Verbindungen realisiert werden, egal ob eine oder mehrere öffentliche IP-Adressen auf dem Firewall Gateway konfiguriert sind.

Bei einer Lösung für Firewall-1 mit static NAT, werden viele öffentliche IP-Adressen benutzt, was Nachteile mit sich bringt [Kapitel 6.5]. Die Phonepatch-Lösung für Linux Netfilter ist umständlich zu bedienen und aus diesem Grund für einen praktischen Einsatz nicht zu empfehlen. Eine Erweiterung des Phonepatches, bei der am Client die Adresse des Gateways und die Adresse bzw. der Name des Kommunikationspartners angegeben werden, wäre eine praktikable Lösung für das Problem.

Für die weitere Verbreitung von VoIP ist es sinnvoll, eine Datenbasis ähnlich dem DNS aufzubauen. Für einen Nutzer könnten so Name und das eventuell benutzte Gateway gespeichert werden, so daß die Daten für alle Nutzer erreichbar sind. Dieses sollte in nachfolgenden Arbeiten genauer untersucht werden.

Eine interessante Fortsetzung wäre auch, den gleichen Vorgang mit öffentlichen IP-Adressen zu testen, dies würde die Testversuche einfacher machen.

Desweiteren könnten Stateful Paketfilter zusammen mit einem H.323-kompatiblen Applikationsfilter getestet werden, somit würde sich die Sicherheit des internen Netzwerkes deutlich erhöhen, was für ein Unternehmen wichtig ist.

Bei einem weiteren Laborversuch könnte man eine Firewall auf einem Linux-Rechner mit Vermittlungsfunktion einrichten. Als Kommunikationsprotokoll muß nicht wie in dieser Diplomarbeit das H.323-Protokoll benutzt werden, sondern es kann ein einfacheres wie das HTTP-Protokoll gewählt werden.

Trotz der verschiedenen, getesteten Exploits, wurden die Sicherheitsregeln nicht überwunden. Es wurden lediglich Information über die Firewall-Rechner und die offenen Portnummern ermittelt. Damit besteht aber keine Garantie, daß sie vor Angriffen sicher sind. Vielmehr muß das LAN ständig verbessert werden, um dem Angreifern einen Schritt vorraus zu sein, was durch ständige Aktualisierung der Informationen im Sicherheitsbereich, durch Zulassung der nur notwendigen Serverdienste und durch permanente Softwareaktualisierung verwirklicht werden kann.

Literaturverzeichnis

[AFT01] Authenticated Firewall Traversal (aft), http://www.ietf.org/html.charters/aft-charter.html, 2001.

[CB94] Cheswick William R., Bellowin Steven M., *Firewalls and Internet Security*, Addison-Wesly, 1994.

[CBA99] Cisco Systems, *Configuring Context-Based Access Control* http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt3/sccbac.htm, 1999.

[CEN00] SIP Center, *What is SIP?*, http://www.sipcenter.com, 2000.

[CER99] CERT, http://www.cert.org/security-improvement/practices/p053.html, 1999.

[CHE00] Checkpoint, http://www.checkpoint.com/, 2000.

[DAL99] Dalgic Ismail, Fang Hanlin, *Comparison of H.323 and SIP for IP Telephony Signaling*, http://www.cs.columbia.edu/%7Ehgs/papers/others/Dalg9909 Comparison.pdf, 1999.

[DAT00] Databeam Corp., *A Primer on the H.323 Series Standard*, http://www.databeam.com, 2000.

[DEP00] Kotha Sam, Cisco Systems, *Deploying H.323 Applications in Cisco Networks*, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/mmcm/tech/h323_wp.htm, 2001.

[EQU01] Equivalence, http://www.equival.com/, 2001.

[FUH98] Fuhrberg Kai, *Internet Sicherheit,* Carl Hanser Verlag München, 1998.

[FWB01] http://www.fwbuilder.org/, 2001.

[GAU99] Network Associates, *Gauntlet Firewall for UNIX Administrators Guide*, 1999.

[H.323] ITU Telecommunication Standardization Sector Study Group 16, *Draft H.323v4*, http://www.itu.org, 2000.

[H.225] ITU Telecommunication Standardization Sector Study Group 16, *Draft H.225.0v4*, http://www.itu.org, 2000.

[HAR00] Haragutchi Ricardo, *Voice/Data Coexistence in IBM Routers*, http://www.ibm.com, 2000.

[INT00] Intel, The *Problems and Pitfalls of Getting H.323 Safely Through Firewalls*, http://www.intel.com.ve/support/videophone/trial21/h323_wpr.htm, 2000.

[ISO00] International Organisation for Standardisation, http://www.iso.ch/, 2000.

[IOS99] Cisco Systems, *Cisco IOS Firewall Overview*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt3/scfirewl.htm, 1999.

[ITU00] International Telecommunication Union, http://www.itu.org, 2000.

[MCM00] Multimedia Conference Manager, Cisco Systems, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/mucvmn/prodlit/ipmcm_ds.htm, 2000.

[NAI01] Network Associates, *Gauntlet homepage*, http://www.pgp.com/products/gauntlet/default.asp, 2001.

[NAT01] Cisco Systems, *Cisco IOS Network Address Translation (NAT)*, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.htm, 2001.

[NIC00] nic.at, *Geschichte des Intenet*, http://www.nic.at/german/geschichte.html, 2000.

[OPS01] http://www.checkpoint.com/opsec, 2001.

[PIX00] *Cisco Secure PIX Firewalls*, Cisco Systems http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/prodlit/pix_pa.htm, 2000.

[POD00] Podey Bernd, *Sicherheitskonzept der Informations- und Kommunikationsprozesse in strukturierten Unternehmen anhand des ISO/OSI Referenzmodells*, Diplomarbeit, Universität Rostock, 2000.

[RAD01] Radvision, *What is SIP?*, http://www.radvision.com, 2001.

[RFW99] Axent Technologies, *Raptor Firewall 6.0 Whitepaper*, 1999.

[ROA00] Roach Adam, *draft-roach-mmusic-sip-pstn-require-header-00.txt*, http://ietf.org, 2000.

[ROS00] IETF SIP Working Group, *SIP Traversal through Residential and Enterprise NATs and Firewalls*, http://www.ietf.org, 2000.

[SIP99] IETF Network Working Group, *SIP: Session Initiation Protocol*, http://www.ietf.org, 1999.

[SHO00] Shore Melinda, *H.323 and Firewalls: Problem Statement and Solution Framework*, draft-shore-h323-firewalls-00.txt, http://community.roxen.com/developers/idocs/drafts/draft-shore-h323-firewalls-00.html, 2000.

[SUR00] SURFnet ExpertiseCentrum, *Multimedia demonstration environment for creation and distribution of content in an open research and education environment,* http://www.sec.nl/, 2000.

[SOR00] Sorenson Vision, Inc, *Ubiquitous public Internet videoconferencing,* http://www.teleconferencemag.com, 2000.

[STR99] Strobel Stefan, *Firewalls Einführung Praxis Produkte*, dpunkt.verlag GmbH Heidelberg, 1999.

[SYM01] Symmantec, *Raptor Firewall 6.5*, http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=47&PID= 4153271, 2001.

[THE00] Therlenius Fredrik, *SIP, NAT and Firewalls*, Master's Thesis, Kungliga Tekniska Högskolan Stockholm, Schweden, 2000.

[VOI98] Ryan Jerry, *Voce over IP (VoIP)*, http://www.techguide.com, 1998.

[WAL00] Wallstabe Jörn, *Synchrone Dienste in IP-Netzen*, Diplomarbeit, Universität Rostock, 2000.

Anhang 1

```
#
#FIREWALL SCRIPT FOR AN INTERNAL NETWORK BEHIND A  FIREWALL WITH
#A CONNECTION  TO INTERNET

#!bin/sh


echo -e "\n      executing firewall script \n"


#-------------------------------------------------------------------------------------------
#
###constants
#
#ip-addresses
FIREWALL=139.30.208.12
INTERNAL_NETWORK=192.168.1.0/24
GATEWAY=192.168.1.1
NAGILUM=139.30.208.2

#portnumber
P_HIGH=1024:65535

#interfaces
EXT=eth0
INT=eth1


#-------------------------------------------------------------------------------------------
#
```

```
###values
#

echo "1" >      /proc/sys/net/ipv4/conf/all/rp_filter
echo "0" >      /proc/sys/net/ipv4/conf/all/accept_redirects
echo "0" >      /proc/sys/net/ipv4/conf/all/accept_source_route
echo "0" >      /proc/sys/net/ipv4/conf/all/bootp_relay
echo "1" >      /proc/sys/net/ipv4/conf/all/log_martians

echo "0" >      /proc/sys/net/ipv4/ip_forward
echo "1" >      /proc/sys/net/ipv4/tcp_syncookies
echo "1" >      /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo "1" >      /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
echo "5" >      /proc/sys/net/ipv4/icmp_destunreach_rate
echo "5" >      /proc/sys/net/ipv4/icmp_echoreply_rate
echo "5" >      /proc/sys/net/ipv4/icmp_paramprob_rate
echo "10" >     /proc/sys/net/ipv4/icmp_timeexceed_rate


#-----------------------------------------------------------------------------------------
#
###modprobe
#
modprobe ip_tables ipt_LOG
modprobe iptable_nat ip_conntrack

#-----------------------------------------------------------------------------------------
#
###default policy and flush
#
iptables -F
iptables -t nat -F
iptables -X
```

```
#-------------------------------------------------------------------------------------------
#
###local processes
#
iptables -A OUTPUT -o lo                                                    -j ACCEPT
iptables -A INPUT     -i lo                                                 -j ACCEPT


#-------------------------------------------------------------------------------------------
#
###Ssh to the firewall.
#
iptables -A INPUT               -p TCP -d $FIREWALL --dport ssh \
                                -m state --state NEW,ESTABLISHED,RELATED  -j ACCEPT


#-------------------------------------------------------------------------------------------
#
###DROP and LOG chain
#
iptables -N my_drop
iptables -A my_drop             -p ICMP -j LOG --log-prefix "DROP-ICMP"
iptables -A my_drop             -p UDP  -j LOG --log-prefix "DROP-UDP"
iptables -A my_drop             -p TCP  -j LOG --log-prefix "DROP-TCP"
iptables -A my_drop                                                         -j DROP


#-------------------------------------------------------------------------------------------
#
###masquerading
#
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
iptables -t nat -A POSTROUTING -o $EXT -j MASQUERADE


#-----------------------------------------------------------------------------------------
#
###Local rules for firewall.
#
#Allow to ping.
iptables -A OUTPUT          -p ICMP --icmp-type echo-request                -j ACCEPT

#Allow all traffic to nagilum-server.
iptables -A OUTPUT          -s $FIREWALL -d $NAGILUM                        -j ACCEPT

#Allow to HTTP.
iptables -A OUTPUT          -p TCP -s $FIREWALL --dport http \
                            -m state --state NEW                            -j ACCEPT

#Allow ssh to others.
iptables -A OUTPUT          -p TCP -s $FIREWALL --dport ssh \
                            -m state --state NEW                            -j ACCEPT

#Allow  for phonepatch switch.
iptables -A INPUT           -i $EXT -p TCP --sport $P_HIGH -d $FIREWALL \
                            --dport 1084 -m state --state NEW               -j ACCEPT


#
#Duplex H.323 communication over an ils-server or the Phonepatch interface
#on the Firewall.
#
#Phonepatch allows input for portnumber (H.225) 1720, (h.245) 3000:3010
#and (RTP/RTCP) 3000:3010. Phonepatch also need the rights to start outgoing
#connections between high portnumbers.
```

```
#

#Allow outgoing to the ils server.
iptables -A INPUT       -i $INT -p TCP -s $INTERNAL_NETWORK -d $GATEWAY \
                        --dport 389 -m state --state NEW                -j ACCEPT
iptables -A OUTPUT      -o $EXT -p TCP -s $FIREWALL --sport $P_HIGH \
                        -d ils.advalvas.be --dport 389 -m state --state NEW -j ACCEPT


#Allow external users to initiate an H.225 and H.245 connection
#to the phonepatch.
iptables -A INPUT       -i $EXT -p TCP -d $FIREWALL --dport 1720 \
                        -m state --state NEW                            -j ACCEPT
iptables -A INPUT       -i $EXT -p TCP -d $FIREWALL --dport 3000:3010 \
                        -m state --state NEW                            -j ACCEPT


#Allow incomming RTP/RTCP both from external and internal user.
iptables -A INPUT       -p UDP --sport $P_HIGH --dport 3000:3010 \
                        -m state --state NEW                            -j ACCEPT


#Allow H.225 and H.245 to an internal user from phonepatch.
iptables -A OUTPUT      -o $INT -p TCP  -s $GATEWAY --sport $P_HIGH \
                        -d $INTERNAL_NETWORK --dport $P_HIGH -m state \
                        --state NEW                                     -j ACCEPT


#Allow internal users to establish a new outgoing connection over phonepatch.
iptables -A INPUT       -i $INT -p TCP -s $INTERNAL_NETWORK \
                        -d $GATEWAY --dport 1720 -m state --state NEW-j ACCEPT
iptables -A INPUT       -i $INT -p TCP -s $INTERNAL_NETWORK \
                        -d $GATEWAY --dport 3000:3010 \
```

```
                        -m state --state NEW                    -j ACCEPT


#From the initiated internal connection, we also need to build a connection
#from phonepatch to the external user. This is for H.225 and H.245 necessary.
iptables -A OUTPUT      -o $EXT -p TCP -s $FIREWALL --sport $P_HIGH \
                        --dport $P_HIGH -m state --state NEW    -j ACCEPT



#-------------------------------------------------------------------------------------
#
###rules for internal LAN
#

#Allow to ping.
iptables -A FORWARD     -p ICMP --icmp-type echo-request        -j ACCEPT

#Allow all traffic to NAGILUM.
iptables -A FORWARD     -s $INTERNAL_NETWORK -d $NAGILUM        -j ACCEPT

#HTTP/HTTPS
iptables -A FORWARD     -o $EXT -p TCP -s $INTERNAL_NETWORK --sport
                        $P_HIGH --dport http -m state --state NEW   -j ACCEPT
iptables -A FORWARD     -o $EXT -p TCP -s $INTERNAL_NETWORK \
                        --sport $P_HIGH --dport https -m state --state NEW -j ACCEPT

#Allow to ssh.
iptables -A FORWARD     -o $EXT -p TCP -s $INTERNAL_NETWORK \
                        --dport ssh -m state --state NEW        -j ACCEPT

#Allow incomming to phonepatch switch.
iptables -A INPUT       -i $INT -p TCP -s $INTERNAL_NETWORK \
                        --sport $P_HIGH -d $GATEWAY --dport 1084 \
```

```
                                -m state --state NEW                    -j ACCEPT


#-----------------------------------------------------------------------------------------
#
###Responses to established connections.
#
#Outgoing packets for an established connection.
iptables -A OUTPUT      -m state --state ESTABLISHED,RELATED        -j ACCEPT
iptables -A FORWARD     -i $INT -o $EXT -m state \
                        --state ESTABLISHED,RELATED                 -j ACCEPT


#Incomming packets for an established connection.
iptables -A INPUT       -m state --state ESTABLISHED,RELATED        -j ACCEPT
iptables -A INPUT       -m state --state NEW,INVALID                -j my_drop

iptables -A FORWARD     -i $EXT -o $INT -m state \
                        --state ESTABLISHED,RELATED                 -j ACCEPT
iptables -A FORWARD     -i $EXT -o $INT -m state --state NEW,INVALID -j my_drop


#-----------------------------------------------------------------------------------------
#
###Block known exploit and scanning tools.
#
#Block xmas packets.
iptables -A INPUT       -p TCP --tcp-flags ALL ALL                  -j DROP
iptables -A FORWARD     -p TCP --tcp-flags ALL ALL                  -j DROP

iptables -A INPUT       -p TCP --tcp-flags ALL NONE                 -j DROP
iptables -A FORWARD     -p TCP --tcp-flags ALL NONE                 -j DROP

#Block syn-flood.
```

```
iptables -A INPUT       -p TCP --syn -m limit --limit 1/s       -j ACCEPT
iptables -A FORWARD     -p TCP --syn -m limit --limit 1/s       -j ACCEPT

#Furtive port scanner.
iptables -A INPUT       -p TCP --tcp-flags SYN,ACK,FIN,RST RST \
                        -m limit --limit 1/s                    -j ACCEPT
iptables -A FORWARD     -p TCP --tcp-flags SYN,ACK,FIN,RST RST \
                        -m limit --limit 1/s                    -j ACCEPT

#Ping of death.
iptables -A INPUT       -p ICMP --icmp-type echo-request \
                        -m limit --limit 1/s                    -j ACCEPT
iptables -A FORWARD     -p ICMP --icmp-type echo-request \
                        -m limit --limit 1/s                    -j ACCEPT

#Kill spoofed packets.
for f in /proc/sys/net/ipv4/conf/*/rp_filter;
do
echo 1 > $f
done

#Anything coming from internal network must have a valid address.
iptables -A FORWARD     -i $INT -s ! $INTERNAL_NETWORK          -j DROP

#Anything coming from the Internet should have real ip-addresses.
iptables -A INPUT       -i $EXT -s 192.168.0.0/16               -j DROP
iptables -A INPUT       -i $EXT -s 172.16.0.0/12                -j DROP
iptables -A INPUT       -i $EXT -s 10.0.0.0/8                   -j DROP
iptables -A FORWARD     -i $EXT -s 192.168.0.0/16               -j DROP
iptables -A FORWARD     -i $EXT -s 172.16.0.0/12                -j DROP
iptables -A FORWARD     -i $EXT -s 10.0.0.0/8                   -j DROP
```

```
#-------------------------------------------------------------------------------------------
#
###Drop and log all the other packets.
#
iptables -A INPUT                                   -j my_drop
iptables -A OUTPUT                                  -j my_drop
iptables -A FORWARD                                 -j my_drop

iptables -A INPUT                                   -j DROP
iptables -A OUTPUT                                  -j DROP
iptables -A FORWARD                                 -j DROP
```

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig angefertigt und keine weiteren als die angegebenen Quellen benutzt zu haben.

Rostock, den 30. 06. 2001

Andreas Olsson