# IP-Telefonie und Firewalls, Probleme und Lösungen

Utz Roedig[1], Ralf Ackermann[1], Ralf Steinmetz[1,2]

1
Technische Universität Darmstadt
Merckstr. 25 • 64283 Darmstadt

2
GMD IPSI
Dolivostr. 15 • 64293 Darmstadt

email: {Utz.Roedig,Ralf.Ackermann,Ralf.Steinmetz}@KOM.tu-darmstadt.de

**Kurzfassung**

Im Rahmen einer umfassenden Security-Policy stellen Firewall-Systeme eine wichtige Maßnahme zum Schutz eines privaten Netzes vor Angriffen aus dem Internet dar. Durch die Einführung neuer Applikationstypen, zu denen auch IP-Telefonie Applikationen gehören, ergeben sich neue Anforderungen denen ein Firewall-System gerecht werden muß. Diesen neuen Anforderungen werden existierende Firewall-Systeme nicht gerecht, weshalb IP-Telefonie Applikationen von Firewalls zur Zeit nicht zufriedenstellend unterstützt werden können.

In diesem Beitrag werden wir zeigen, welche speziellen Probleme sich bei der Integration von IP-Telefonie Unterstützung in eine Firewall ergeben. Dazu werden wir ausgewählte, von einer Firewall zu unterstützenden Telefonieszenarien, erläutern, sowie ausgewählte vorhandene Firewall-Lösungen und ihre existierenden Beschränkungen beschreiben. Nachdem die Probleme identifiziert und klassifiziert sind, werden wir die daraus resultierenden Anforderungen, denen eine IP-Telefonie fähige Firewall gerecht werden muß, herleiten. Abschließend werden wir eine mögliche technische Umsetzung dieser Anforderungen, sowie den entsprechenden realisierten Prototypen beschreiben.

**Schlüsselwörter** Firewall, IP-Telefonie, H.323, SIP

# 1 Einführung

Die einer Firewall zugrunde liegende Architektur sowie die internen Mechanismen und die Betriebsart einer Firewall hat sich in den letzten Jahre nicht wesentlich geändert. Durch die Einführung neuer Applikationstypen, zu denen auch IP-Telefonie Applikationen gehören, ergeben sich jedoch neue Anforderungen denen eine Firewall gerecht werden muß. Existierende Firewalls haben große Probleme diesen Anforderungen gerecht zu werden, wenn sie IP-Telefonie Anwendungen - wie es im Augenblick die Regel ist - wie “herkömmliche Applikationen” behandeln.

Zur Zeit entwickelt sich die IP-Telefonie vom experimentellen Status hin zu einer echten Alternative zur konventionellen Telefonie. Es wird allgemein angenommen, daß IP-Telefonie Applikationen ein großes wirtschaftliches Potential besitzen. Damit diese Applikationen aber im täglichen Betrieb mit dem erwarteten wirtschaftlichen Erfolg eingesetzt werden können, ist es notwendig, ein harmonisches Zusammenwirken mit vorhandenen Security-Policies sicherzustellen.

In diesem Beitrag werden wir zeigen, daß und wie sich IP-Telefonie Applikationen von herkömmlichen Applikationen in wesentlichen Aspekten unterscheiden, und daß sie daher innerhalb einer Firewall auch speziell behandelt werden müssen. Wir werden ausgewählte von einer Firewall zu unterstützenden Telefonieszenarien erläutern, die darin auftretenden Probleme identifiziert und hinsichtlich ihrer Ursachen klassifizieren. Mit Hilfe dieser Klassifizierung wird nachfolgend eine Definition einer “IP-Telefonie kompatiblen Firewall” geben. Danach werden wir ausgewählte vorhandene Firewall - Lösungen und ihre existierenden Beschränkungen beschreiben. Der Beitrag schließt mit der Vorstellung einer möglichen technischen Umsetzung einer “IP-Telefonie kompatiblen Firewall”, sowie eines entsprechenden realisierten Prototypen.

## 1.1 IP-Telefonie

IP-Telefonie Applikationen werden verwendet, um eine Audioverbindung zwischen zwei Endsystemen aufzubauen. Dabei wird, anders als bei der klassischen Telefonie, ein paketvermitteltes IP-Netzwerk als Trägermedium der Sprach- und Signalisierungsdaten verwendet. Die Applikationen können auf verschiedenen Protokollfamilien basieren, wobei im wesentlichen die H.323- [1] und die SIP-Protokollfamilie [2] eingesetzt wird. Zur Zeit kann eine unterschiedliche Verbreitung dieser Protokollfamilien beobachtet werden, wobei sich die Anteile aber zunehmend verschieben. Heute verwenden die Mehrzahl der Applikationen und IP-Telefonie Szenarien das H.323 Protokoll, weshalb dieser Beitrag hauptsächlich auf Anwendungen, die auf dieser Protokollfamilie basieren, fokussiert. Es wird aber allgemein angenommen, daß in der nahen Zukunft die SIP Protokollfamilie an Bedeutung und Verbreitung gewinnen wird [3]. Es ist sogar möglich, wenn entsprechende Gateways verwendet werden, beide Protokollfamilien in einem Szenario gemeinsam einzusetzen [4].

## 1.2 Firewalls

Innerhalb einer global vernetzten Umgebung gewinnen Sicherheitsaspekte zunehmend an Bedeutung und die Zugangskontrolle an Netzwerkgrenzen wird mittlerweile als absolut notwendige Maßnahme betrachtet. Aus diesem Grund haben die meisten Organisationen ihre einfachen, ohne spezielle Schutzmechanismen ausgestatteten Internet-Router durch Firewall-Systeme ersetzt.

Diese Firewall-Systeme bestehen in der Regel aus Paketfiltern, Proxies, “Stateful Filtern” oder aus einer Kombination dieser Komponenten. Eine Firewall analysiert den gesamten Netzwerkverkehr zwischen den an sie angeschlossenen Netzwerken und leitetet nur Datenströme in das jeweilige andere Netz weiter, für die dies innerhalb der entsprechenden Sicherheits-Policy spezifiziert wurde [5],[6]. Zusätzlich zur Analyse der Datenströme werden Firewalls teilweise auch dazu eingesetzt, die interne Netzwerkstruktur einer Organisation zu verbergen. Aus dem externen Netz (z.B. dem Internet) ist der einzig “sichtbare” und deshalb direkt angreifbare Rechner die Firewall selbst. Dieses Verbergen der internen Strukturen wird mit Hilfe eines als Network Adress Translation (NAT) [7] bezeichneten Mechanismus realisiert.

Zur Analyse der Datenströme benötigen die einzelnen Firewall-Komponenten (Paketfilter, Proxies und “Stateful Filter”) ein entsprechendes, für den zu unterstützenden Dienst optimiertes und angepaßtes Element. Dieses Element bezeichnen wir in diesem Kontext als Parser. Basierend auf der Analyse der Kommunikationsdaten entscheidet die Firewall, ob die Daten die Netzgrenze passieren dürfen oder nicht. Der Parser ist dabei kein isolierter Systembestandteil, sondern kann mit anderen Elementen innerhalb der Firewall, z.B. dem NAT Mechanismus, interagieren.

### 1.3 Referenzszenario

Abbildung 1 zeigt ein IP-Telefonie Szenario, basierend auf der H.323 Protokollfamilie.

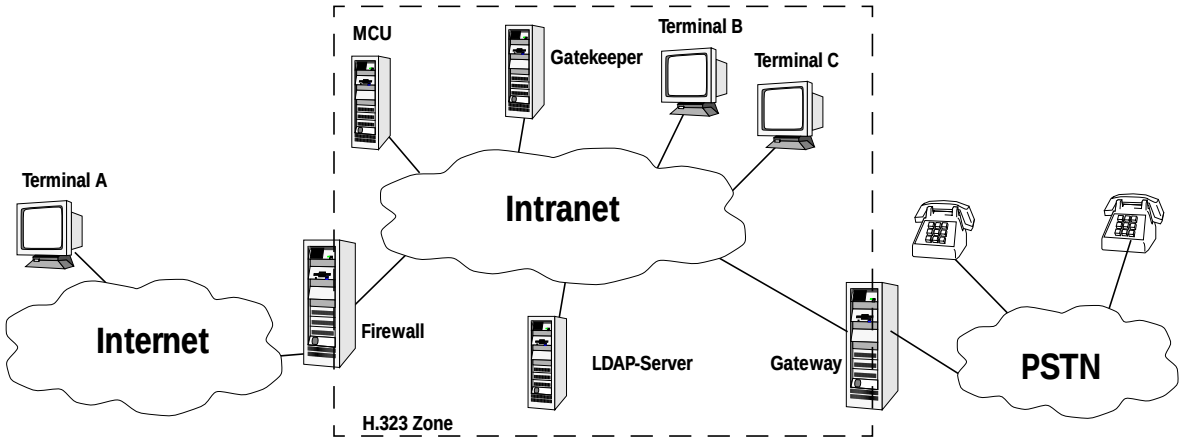


Abbildung 1: H.323 Standard-Szenario

Darin werden typische H.323 Komponenten im Zusammenwirken mit einer Firewall dargestellt. Allgemein wird angenommen, daß ein solches Szenario repräsentativ für normale Einsatzszenarien ist. Auf modifizierte individuelle Anforderungen kann es leicht angepaßt werden. Das Bild zeigt das private Netz einer Organisation, welches zum Internet hin durch eine Firewall geschützt wird. Innerhalb des Intranets existiert eine (alternativ auch mehrere) H.323-Zone, welche einen Gatekeeper und mehrere optionale Geräten wie zum Beispiel einer Multi Conference Unit (MCU), Gateways und Terminals umfaßt. Diese Abbildung werden wir innerhalb dieses Beitrages als Referenzszenario verwenden.

## 2 Klassifizierung der Probleme

Sollen IP-Telefonie Applikationen in einer Umgebung eingesetzt werden, in der auch konventionelle Firewalls verwendet werden, so führt dies zu einer Beeinträchtigung des IP-Telefonie Dienstes, und/oder zu einer Beeinträchtigung der Funktionalität der Firewall. So können sich beispielsweise folgende funktionalen Beeinträchtigungen (Probleme) ergeben:

- Es kann keine Kommunikationsverbindung über die Netzgrenze hinweg aufgebaut werden.
- Der IP-Telefonie Dienst kann nur mit Einschränkungen verwendet werden (z.B. Störungen in der Sprachqualität, keine Gatekeeper vermittelten Rufe).
- Die Firewall muß mit Einschränkungen ihrer Schutzfunktion betrieben werden. (z.B. UDP Kommunikation muß ohne Einschränkungen zugelassen werden, es kann keine NAT Funktionalität verwenden.)

Für ein fundiertes Problemverständnis und als Basis für eine Evaluierung bestehender Lösungsansätze, sowie das Entwickeln geeigneter Lösungen ist eine Strukturierung der Probleme hinsichtlich ihrer Ursachen notwendig. In den folgenden

Abschnitten werden die einzelnen Probleme, ihre Ursachen sowie deren Lösung detailliert dargestellt.

## 2.1 Probleme durch Charakteristika von Multimedia Applikationen

Bei den von uns betrachteten Applikationen handelt es sich um Multimedia-Anwendungen, die kontinuierliche und diskrete Medienströme verarbeiten [8]. Die kontinuierlichen Medienströme können zum Beispiel Audio- und/oder Videoströme enthalten, die diskreten Medienströme transportieren zusätzliche Kontroll- oder Metainformationen. Diese Multimedia-Applikationen unterscheiden sich hinsichtlich vieler Eigenschaften signifikant von “traditionellen Applikationen”:
Speziell sind

- mehrere Flows für eine logische Session,
- komplexe Protokolle und dynamisches Protokollverhalten,
- hohe Datenrate im Zusammenhang mit Dienstgüte-Anforderungen,
- die Verwendung von Multicast Mechanismen,

zu beobachtende Charakteristika. Gerade diese führen zu Problemen in Umgebungen, in denen Firewalls verwendet werden. Eine detaillierte Beschreibung dieser Probleme, sowie die sich daraus für eine Firewall ergebenden Anforderungen sind in [9] und [11] dargestellt. Diese Anforderungen können zum einen umgesetzt werden, indem die Firewall Architektur angepaßt wird [10], [11], [12], zum anderen kann aber auch die Multimedia Applikation - bzw. die von ihr verwendeten Protokolle - angepaßt werden [13], [14]. In diesem Beitrag konzentrieren und beschränken wir uns auf die IP-Telefonie spezifischen Probleme und ihre Ursachen. Die zur Lösung dieser generellen Probleme notwendigen Mechanismen sind hier nicht dargestellt.

## 2.2 Probleme durch IP-Telefonie spezifische Charakteristika

### Szenarienvielfalt und Szenarienkomplexität

Die verwendeten Kommunikationsmechanismen innerhalb des Referenzszenarios (Abbildung 1) sind von den in die Kommunikation involvierten Geräten abhängig und ändern sich je nach Anwendungsfall. Wenn nur zwei Terminals an der Kommunikation beteiligt sind (direkter Ruf zwischen Terminal A und Terminal B), wird der in Abbildung 2 dargestellte Kommunikationsablauf verwendet. Die Pfeile stellen dabei die Richtung der Kommunikationsinitiierung dar. Bei TCP bedeutet dies, daß in Richtung des Pfeiles die Verbindung aufgebaut wird. Daten aber werden in beide Richtungen gesendet. Bei UDP wird in Richtung des Pfeiles das erste Paket der “Verbindung” transportiert. Die Abfolge der Antworten in Gegenrichtung (unter Verwendung der selben Ports) ist aus Gründen der Vereinfachung nicht vollständig aufgeführt.

Für die Definition der Security-Policy innerhalb einer Firewall ist es notwendig zu wissen, wer die Verbindung initiiert, und welche Ports dabei verwendet werden. Daher ist eine solche Darstellung hier besonders geeignet.

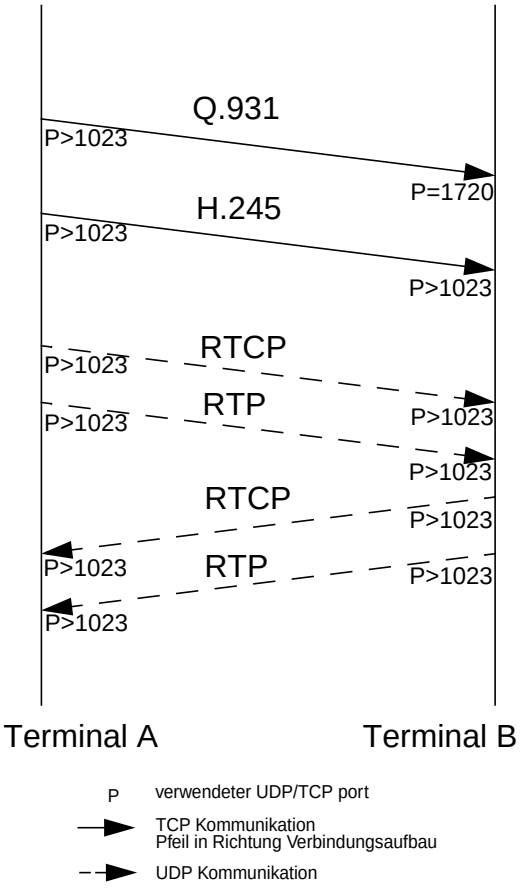


Abbildung 2: H.323 direkter Ruf

**•Q.931 *Call Signalling* (TCP):**
Eine TCP Verbindung wird zwischen Terminal A und Terminal B aufgebaut. Diese wird zum Transport der *Call Setup* Nachrichten, wie sie in H.225.0 [15] definiert sind, verwendet. Diese Nachrichten werden benötigt, um den Rufaufbau durchzuführen. Dabei werden unter anderem die Parameter (Port und IP-Adresse) für die folgende *Call Control* Verbindung an Terminal A übermittelt.

**•H.245 *Call Control* (TCP):**
Terminal A kontaktiert Terminal B via TCP unter Verwendung der zuvor übermittelten Parameter (Port und IP-Adresse). Die H.245 Verbindung wird verwendet, um *Call Control* Nachrichten (definiert in [16]) zwischen den Terminals auszutauschen. Diese Nachrichten werden unter anderem dazu verwendet, die Parameter der nachfolgenden Medienströme auszuhandeln (*OpenLogicalChannel*). Dabei werden neben den entsprechenden Medienkodierungsverfahren auch die zu verwendenden UDP-Ports vereinbart.

- **RTP/RTCP *Media* und *Mediacontrol* (UDP):**
  Zwischen den beiden Terminals werden mehrere Medienströme verwendet. Es sind mindestens 4 UDP-Ströme notwendig, um die Audiodaten zu transportieren (1 RTP- und der korrespondierende RTCP-Strom in jede Richtung). Zusätzliche Ströme werden verwendet, wenn zum Beispiel eine optionale Video-Übertragung stattfindet.

Wenn innerhalb dieses Szenarios ein Gatekeeper verwendet wird (Rufvermittlung durch den Gatekeeper), ändern sich die Kommunikationsbeziehungen und -abläufe. In dem hier dargestellten Fall (Abbildung 3) laufen die Signalisierungsnachrichten über den Gatekeeper (*gatekeeper routed call* [17]). Der H.323 Standard sieht auch eine Möglichkeit vor, den Gatekeeper ohne direkte Einbeziehung in die Signalisierung zu verwenden (*direct call model* [17]).

Der *gatekeeper routed call* wird jedoch in den meisten Gatekeeper-Szenarien verwendet, weshalb dieser Fall hier dargestellt ist.

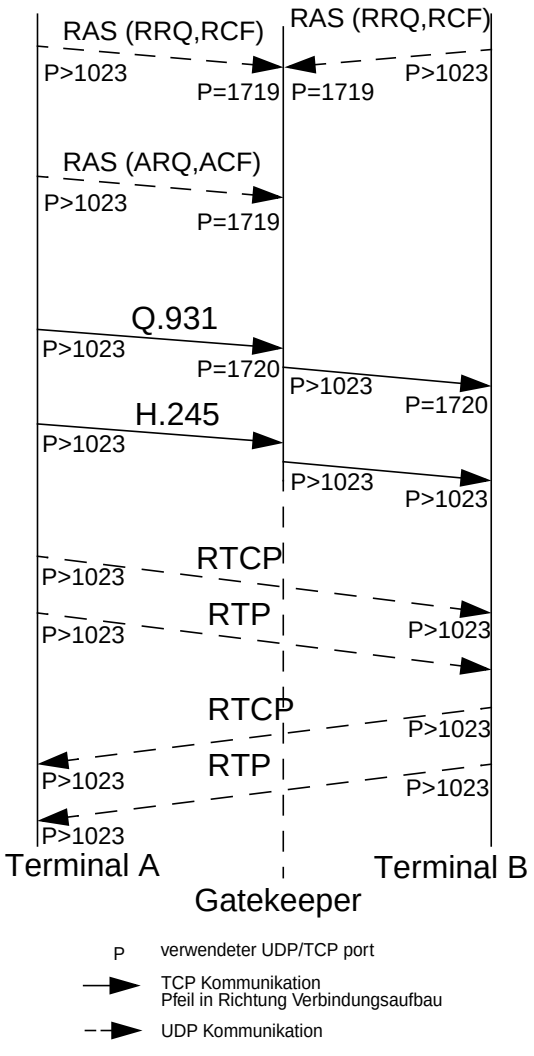


Abbildung 3: H.323 Gatekeeper vermittelter Ruf

**•RAS *Registration, Admission, Status* (UDP):**
Nach dem Start der Terminals registrieren sich diese am Gatekeeper (*registration request RRQ* und *registration confirm RCF*). Dazu wird das auf UDP basierte RAS - Protokoll [15] verwendet. Den Terminals muß dazu die Adresse des Gatekeepers bekannt sein; in diesem Beispiel nehmen wir eine statische Konfiguration an.

**•RAS *Registration, Admission, Status* (UDP):**
Bevor die Kommunikation stattfinden kann, muß das rufende Terminal beim Gatekeeper unter Verwendung des RAS-Protokolls mit einer Spezifikation der geplanten Gesprächscharakteristika eine entsprechende Erlaubnis erbitten (*admission request ARC*). Wenn die Erlaubnis erteilt wird (*admission confirm ACF*), kann der eigentliche Ruf ausgelöst werden.

**•Q.931 *Call Signalling* (TCP):**
Terminal A kontaktiert den Gatekeeper via TCP. Anhand der ersten *Call Signalling*-Nachricht kann der Gatekeeper das Zielterminal bestimmen und kontaktiert dieses. Die *Call Signalling*-Nachrichten werden in diesem Fall über den Gatekeeper “geroutet”. Die Parameter, der Port und die IP-Adresse, für die folgende *Call Control* Verbindung werden an Terminal A übermittelt.

- **H.245 *Call Control* (TCP):**
  Terminal A kontaktiert den Gatekeeper via TCP unter Verwendung der zuvor ausgehandelten Parameter. Der Gatekeeper kontaktiert seinerseits das Terminal B. Die *Call Control*-Nachrichten werden in diesem Fall ebenfalls über den Gatekeeper “geroutet”. Die *Call Control*-Nachrichten wird dazu verwendet, die Parameter der folgenden Medienströme auszuhandeln.
- **RTP/RTCP *Media* und *Mediacontrol* (UDP):**
  Wie im ersten Beispiel werden nun mehrere Medienströme zwischen beiden Terminals verwendet. Diese werden direkt gesendet, ohne den Gatekeeper in die Kommunikation einzubeziehen.

Die Kommunikationsabläufe ändern sich ebenfalls, wenn Protokollvarianten wie *H.245 Tunneling* oder *Fast Start* verwendet werden, bzw. weitere H.323 Komponenten in die Kommunikation mit einbezogen sind (z.B. MCUs).

## Herstellerspezifische Implementierungen

Nicht nur die Nutzung verschiedener H.323 Komponenten sondern auch deren unterschiedlichen herstellerspezifischen Implementierungen haben einen Einfluß auf die Kommunikationsabläufe. Unsere Experimente zeigen, daß unterschiedliche Hersteller auch unterschiedliche (und teilweise sogar nicht interoperable) Implementierungen verwenden, obwohl alle von sich behaupten, H.323-kompatibel zu sein.

Ist zum Beispiel Terminal A kein “reines” H.323- sondern ein Microsoft Netmeeting-Terminal, so werden die folgenden Erweiterungen verwendet:

- **ILS/LDAP Namens- bzw. Adreßauflösung (TCP)**
  Bevor die Kommunikation beginnt, versucht Terminal A einen Kontakt zu einem ILS-Server aufzubauen, um eine Namensauflösung durchzuführen. Auf diesem Weg können symbolische Namen in Client IP-Adressen aufgelöst werden (Telefonbuchfunktion). Nachdem das Terminal die Zieladresse bestimmt hat, startet die normale H.323 Kommunikation. Die Kommunikationsabläufe entsprechen dabei dem direkten Ruf aus dem vorhergehenden Beispiel.

Die im vorigen Abschnitt beschriebenen Beispiele zeigen, daß sich das Kommunikationsverhalten signifikant ändern kann, wenn sich das Anwendungsszenario ändert. Eine sich innerhalb der Kommunikationswege befindliche Firewall muß dieser dynamischen Vielfalt gerecht werden.

## 2.3 Probleme bei Network Address Translation (NAT)

Weitere Probleme innerhalb eines H.323 Szenarios treten dann auf, wenn eine Adreßumsetzung (NAT) von der Firewall durchgeführt wird. In diesem Fall können die internen Terminals (Terminal B und C) nicht direkt von außerhalb angerufen werden, da ihre Adressen vom Internet aus nicht “sichtbar” sind. Dabei handelt es sich um eine gewünschte Firewall-Funktionalität, da so die Strukturen und Details des internen Netzwerkes verborgen bleiben und dieses vor Angriffen besser geschützt ist. Ausgehende Rufe stellen in einer NAT-Umgebung prinzipiell kein Problem dar, da das Ziel eine gültige Adresse besitzt. Bei einem ausgehenden Ruf (z.B. Terminal B nach Terminal A) muß die Firewall die Kommunikation beobachten und alle internen (privaten) IP-Adressen auf extern gültige IP-Adressen abbilden (z.B. auf die externe Adresse der Firewall selbst).

Wenn wie in unserem Szenario (Abbildung 1) Terminal A eine Verbindung zu Terminal B aufbauen will (eingehender Ruf in den geschützten Bereich), so ist dies nicht direkt möglich. Terminal A muß sich dazu zuerst mit der Firewall verbinden und dieser mitteilen, welches das Endziel des Rufes ist. Die Firewall muß nachfolgend Terminal B kontaktieren und die Kontroll- und Audioströme zwischen beiden Terminals vermitteln. Es existieren verschiedene Methoden, um diese Aufgabe zu erfüllen. Wenn innerhalb des Szenarios kein Gatekeeper verwendet wird, kann der folgende

Kommunikationsablauf, welcher auch in [18] beschrieben ist, verwendet werden:

- Das externe Terminal muß modifiziert werden. Es ist eine Konfigurationsmöglichkeit vorzusehen, die es dem Benutzer erlaubt, die Adresse der Firewall (zusätzlich zum Rufziel) einzugeben. Die so spezifizierte Firewall kann dann den Ruf zwischen externem und internem Netz vermitteln.
- Bei einem Verbindungsaufbau kontaktiert das Terminal zunächst die Firewall. Dies kann erreicht werden, indem innerhalb der H.323 Setup Message das Feld *destcallSignalingAdress* und / oder das Feld *destinationAddress* die Adresse der Firewall beinhaltet. Der Firewall muß dann mitgeteilt werden, welches das eigentliche Zielterminal ist. Dies wiederum kann unter Verwendung des Feldes *remoteExtensionAlias* erreicht werden. Innerhalb dieses Feldes wird der Zielbenutzer eingetragen. Die Firewall muß diesen Namens-Alias in die Adresse des Zielterminals auflösen - danach kann der Ruf zwischen beiden Endsystemen vermittelt werden.

Diese Methode erlaubt es, eingehende Rufe in einer NAT-Umgebung zu unterstützen, allerdings ist eine Modifikation der Terminals nötig. Wird innerhalb des Szenarios ein Gatekeeper verwendet, so kann folgender Kommunikationsablauf [19] verwendet werden:

- Der Gatekeeper muß parallel zur Firewall installiert werden. Er muß eine IP-Adresse besitzen, die im Internet gültig ist.
- Das externe Terminal A muß so konfiguriert werden, daß es den Gatekeeper verwendet.
- Wenn Terminal A einen Ruf zu Terminal B aufbauen möchte, muß es zunächst beim Gatekeeper eine entsprechende Erlaubnis erfragen.
- Wird diese erteilt, baut Terminal A eine Signalisierungsverbindung zu dem Gatekeeper auf. Dieser vermittelt dann die Signalisierungsverbindung an das Zielterminal. Dabei werden die Q.931 und H.245-Verbindung durch den Gatekeeper “vermittelt” (Proxy-Funktionalität).
- Die Audio-Datenströme fließen nach abgeschlossener Signalisierung zwischen den Terminals. In einer NAT-Umgebung muß beim Überschreiten der Netzgrenze eine entsprechende Modifikation der IP-Ziel- und Quelladressen vorgenommen werden.
- In diesem Fall wird die NAT-Adreßmodifikation für die Kontrollkanäle durch den Gatekeeper ausgeführt. Für die Audiodatenströme muß dies die Firewall übernehmen. Dazu ist eine Synchronisation zwischen Firewall und Gatekeeper notwendig.

Die jetzt beschriebene Methode erlaubt es, eingehende Rufe in einer NAT Umgebung zu unterstützen. Im Unterschied zu der zuerst genannten Methode findet hier die Handhabung der Datenströme in der Firewall transparent für die Endbenutzer statt. Dabei ist die Verwendung eines Gatekeepers zwingend notwendig, um das interne Terminal adressieren zu können.

Wir nehmen an, daß sich in einem typischen realen Szenario beide Parteien (rufendes und gerufenes Terminal) hinter einer Firmen-Firewall befinden werden. Deshalb stellt das Problem der eingehenden Rufe ein wichtiges Problem dar. Wie gezeigt wurde, benötigen alle Lösungen für die Problematik des eingehenden Rufes eine Interaktion zwischen Firewall und den Komponenten, die eine Namensauflösung durchführen. Die Namensauflösung kann hierbei von H.323-Komponenten selbst

(z.B. dem Gatekeeper) oder anderen Diensten (z.B. DNS, LDAP, ILS,...) ausgeführt werden. Aus diesem Grund muß die Parser-Komponente innerhalb der Firewall in der Lage sein, mit diesen Komponenten zu interagieren.

## 2.4 Parser abhängige Probleme

Die Aufgabe der Klassifizierung der Datenströme wird innerhalb der Firewall von einem Parser ausgeführt. Herkömmliche Firewalls verwenden dazu einen statischen und in das System fest integrierten Protokoll-Parser. Diese Parser werden oft in einer firewallspezifischen Sprache notiert bzw. kodiert (z.B. in INSPECT für die Firewall-1 [20]). Normalerweise werden diese Beschreibungen compiliert und in die Firewall eingebunden. Parser können mit der Firewall interagieren, Datenströme zur Analyse anfordern - basierend auf den Ergebnissen der Verkehrsanalyse - oder aber die Firewall neu konfigurieren. Eine Firewall diesen Typs ist in Abbildung 4 dargestellt

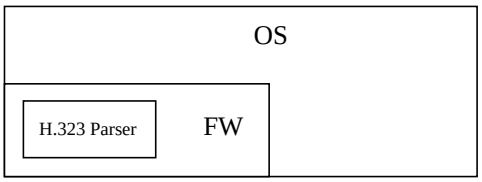

Abbildung 4: Eingebetteter H.323 Parser

Das Bild zeigt einen H.323-Parser, der direkt in die Firewall eingebettet ist, so wie dies für Protokolle anderer Art ebenfalls der Fall sein kann. Die Firewall benutzt dabei Funktionen des Betriebssystems (OS) des Firewallhosts. Die IP-Telefonie Datenströme werden der Firewall-Komponente übergeben (z.B. durch die Konfiguration der OS spezifischen Sockets / Packet Filter) und der Parser innerhalb der Firewall ist dafür verantwortlich, deren Inhalt zu analysieren. Innerhalb dieses Beitrages werden wir dieses einfache Schema verwenden, um die Unterschiede der verschiedenen Architekturen zu erläutern.

Ein solcher “integraler Ansatz” funktioniert sehr gut im Zusammenwirken mit herkömmlichen Applikationen - für IP-Telefonie Szenarien ist er jedoch nur unzureichend geeignet. Dies läßt sich anhand der zuvor beschriebenen Probleme erklären:

- **IP-Telefonie spezifische Charakteristika:**
  Für verschiedene H.323-Szenarien sind verschiedene Parser notwendig. Wenn das Szenario nur geringfügig geändert wird, kann oftmals der Parser nicht an die neuen Anforderungen angepaßt werden und ein neuer Parser ist notwendig. Statische und integrierte Parser können sich nicht diesen geänderten Anforderungen anpassen.
- **Network Address Translation:**
  Der Parser selbst kann nur mit der Firewall selbst interagieren, nicht aber mit anderen Komponenten. Wie gezeigt ist es aber notwendig, mit anderen Komponenten zu interagieren, um NAT-Umgebungen unterstützen zu können.

## 2.5 Schlußfolgerung

Wie beschrieben lassen sich die in einem IP-Telefonie Szenario, in dem Firewalls verwendet werden, auftretenden Probleme nach folgenden Ursachen gruppieren:

- Charakteristika von Multimedia-Applikationen
- IP-Telefonie spezifische Charakteristika
- Network Address Translation (NAT)
- Interner Aufbau des Parserelements

Um IP-Telefonie Szenarien sinnvoll betreiben zu können, ist für jede der beschriebenen Ursachengruppen zumindest ein Lösungsansatz notwendig. Zunächst muß eine Firewall generell in der Lage sein, den Anforderungen einer Multimedia-Applikation gerecht zu werden. Um den IP-Telefonie spezifischen Charakteristika gerecht werden zu können, muß die Firewall in der Lage sein, eine dynamische Anpassung an sich verändernde Szenarien oder spezifische Implementierungen vorzunehmen. Um NAT unterstützen zu können, muß die Firewall in der Lage sein mit anderen Komponenten zu interagieren. Um diese beiden Anforderungen zu erfüllen, muß der Standardaufbau eines Parsers angepaßt werden. Wie bereits gezeigt, kann ein herkömmlicher Parser zur Lösung dieser Probleme nicht verwendet werden. Zusammenfassend muß demnach eine “IP-Telefonie kompatible Firewall” folgende Kriterien erfüllen:

1. Umsetzung der Anforderungen von Multimedia-Applikationen
2. Dynamische Anpassbarkeit an sich verändernde Szenarien
3. Kommunikation mit anderen IP-Telephonie Komponenten
4. Verwendung einer geeigneten Parser Architektur

Im nachfolgenden Abschnitt werden wir einige für die Unterstützung von IP-Telefonie Applikationen vorgesehene Firewall-Implementierungen hinsichtlich der Erfüllung der IP-Telefonie spezifischen Kriterien (2,3 und 4) analysieren. Danach wird ein eigener Ansatz vorgestellt, der diesen Kriterien optimal gerecht wird.

# 3 Evaluation existierender Lösungsansätze

Eine “konventionelle” Firewall- bzw. Parser-Architektur, wie sie in Abbildung 4 dargestellt ist, ist nicht ausreichend, um IP-Telefonie Szenarien sinnvoll zu unterstützen. Dies wurde von verschiedenen Firewall-Herstellern erkannt und hat dazu geführt, daß verschiedene Lösungen entwickelt wurden, um diesem Problem adäquat gerecht zu werden. Das erste hier dargestellte Beispiel beschreibt die H.323 Lösung des Firewall-Marktführers (80% Marktanteil). Die beiden weiteren Beispiele stellen Lösungen dar, die explizit das IP-Telefonie Problem adressieren und für dieses Einsatzgebiet optimiert sind.

## 3.1 Firewall-1

Die Architektur des Firewall-1 [20] Produktes entspricht im wesentlichen einer Architektur, wie sie in Abbildung 4 dargestellt ist. Aus diesem Grund treten alle beschriebe-

nen Probleme in einer Firewall-1 geschützten H.323-Zone auf. Da der Parser statisch realisiert ist, wird ein dedizierter Parser für jedes zu unterstützende Szenario notwendig. Zur Zeit sind für die Firewall-1 zwei Parser verfügbar, einer für Microsoft Netmeeting und ein zweiter generischer H.323-Parser. Der Microsoft Netmeeting Parser enthält dabei den generischen H.323-Parser, kann aber zusätzlich ILS Anfragen handhaben. Beide Parser wurden untersucht und es wurden die nachfolgend vorgestellten Ergebnisse erzielt.

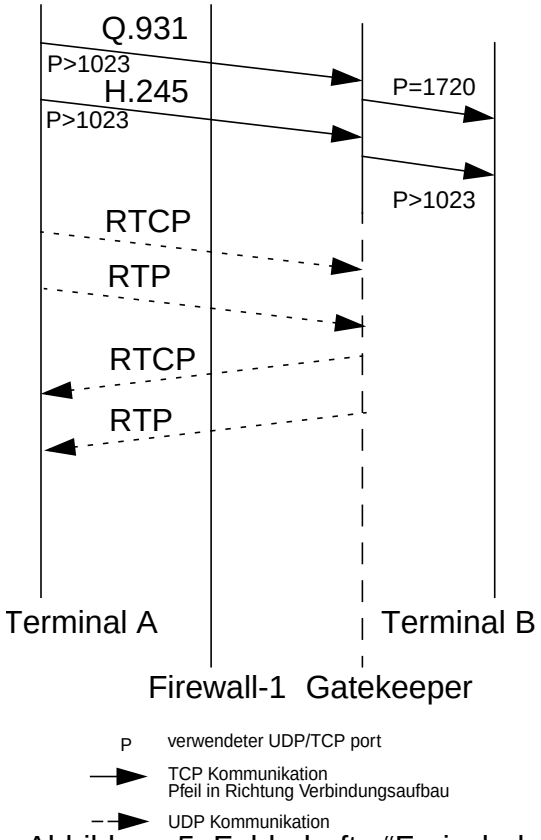


Abbildung 5: Fehlerhafte “Freischaltung” der Medienströme

•Eine direkte Verbindung zwischen zwei Netmeeting Terminals kann ohne Probleme unterstützt werden (sowohl für Netmeting v2 als auch Netmeeting v3).
•Beim Ersetzen der Terminals durch Produkte anderer Hersteller (wie z.B. mit dem Innovaphone [22] IP400 v1) traten teilweise Probleme auf. Diese waren auf den verwendeten spezifischen H.323-Protokollstack des Innovaphone zurückzuführen. Version 2 der Software des Gerätes war mit der Firewall-1 kompatibel.
•NAT-Szenarios können nur für ausgehende Rufe unterstützt werden. Eingehende Anrufe können nicht geroutet werden, da eine Kommunikation mit entsprechenden H.323- oder anderen namensauflösenden Komponenten nicht vorgesehen ist.
•Szenarios, in denen ein Gatekeeper verwendet wird, können nicht unterstützt werden. Dies liegt an der Konzeption des Parsers, der nur eine mögliche Kommunikationsform, die direkte zwischen zwei Terminals kennt.

Der Firewall-1 Parser analysiert zuerst die Q.931-Verbindung (Abbildung 5). Er erkennt die ausgehandelten Ports der folgenden H.245-Verbindung und schaltet eine Verbindung zwischen dem Gatekeeper und Terminal B mit den entsprechenden Ports frei. Danach analysiert der Parser die H.245-Verbindung und erkennt die ausgehandelten Ports der Medienströme. Danach werden die Wege für diese Medienströme freigeschaltet. Bei dieser Freischaltung wird aber angenommen, daß die Medienströme zwischen denselben Komponenten fließen, zwischen denen auch die Signalisierung stattfand. Für einen durch einen Gatekeeper vermittelten Ruf gilt dies jedoch nicht. Der Parser müßte in diesem Fall nicht nur die ausgehandelten Ports sondern auch die ausgehandelten Quell- und Zieladressen beachten.

**Fazit:**

Die im betrachteten Beispiel verwendeten Parser-Komponenten weisen eine sehr statische Struktur auf. Aus diesem Grund konnte in unseren Experimenten nur ein Basisszenario unterstützt werden, der direkte Ruf. Aufgrund der fehlenden Interaktion der Firewall mit H.323 (oder funktionsäquivalenten) Komponenten können in NAT Umgebungen keine eingehenden Rufe unterstützt werden. Die Firewall-1 ist dementsprechend nach unserer Definition (siehe 2.5) nicht “IP-Telefonie kompatibel”.

## 3.2 Cisco MCM

Der Cisco Multimedia Conference Manager (MCM) [19] stellt sowohl Proxy- als auch Gatekeeper-Funktionalität bereit. Dadurch bildet er ein System, das dazu verwendet werden kann, eine existierende Firewall um IP-Telefonie Funktionen zu erweitern. Der MCM kann auf einem CISCO System (z.B. auf einem Cisco Router der CISCO IOS unterstützt) parallel (oder hinter) einer Firewall installiert werden. Seine innere Architektur ist in der folgenden Abbildung 6 dargestellt.

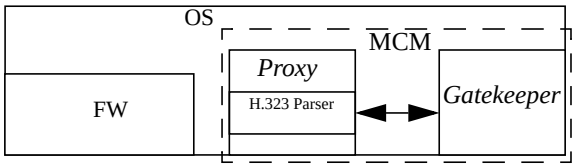


Abbildung 6: MCM Architektur

Der gesamte IP-Telefonie Verkehr wird durch den MCM bearbeitet und “umgeht” dadurch die eigentliche Firewall. Eine Interaktion zwischen der Firewall und dem MCM ist nicht vorgesehen. Wenn der MCM parallel zu einer Firewall betrieben wird, ist eine Unterstützung von NAT-Szenarien sowohl für eingehende als auch für ausgehende Rufe möglich. Dies gilt, da der MCM den Gatekeeper beinhaltet, der ein Routing der eingehenden Rufe ermöglicht. Somit ist eine Interaktion mit dem enthaltenen Proxy realisiert.

**Fazit:**

Dieser Ansatz adressiert hauptsächlich das NAT-Problem. Alle möglichen NAT-Szenarien können mit seiner Hilfe unterstützt werden. Der Proxy innerhalb des MCM ist aber ebenfalls statisch, er kann nicht an dedizierte Szenarien und Applikationen angepaßt werden. Bei Verwendung des Systems muß innerhalb der H.323-Zone stets der Gatekeeper innerhalb des MCM verwendet werden. Eine freie Wahl eines Gatekeepers (z.B. eines nicht Cisco Gatekeepers) ist nicht möglich. Diese Firewall ist ebenfalls nach unserer Definition nicht “IP-Telefonie kompatibel”, da ein Anpassen der Firewall an verschiedene Szenarien (z.B. Verwendung eines anderen Gatekeepers) nicht möglich ist.

## 3.3 PhonePatch

Das PhonePatch Produkt [21] adressiert Netmeeting-Szenarien und arbeitet wie ein Proxy mit einigen zusätzlichen (PBX ähnlichen, z.B. Callback) Funktionen. PhonePatch wird parallel zu einer existierenden Firewall verwendet und ist verantwortlich für die Behandlung des IP-Telefonie Verkehrs. Eine Interaktion zwischen Firewall und PhonePatch findet nicht statt.

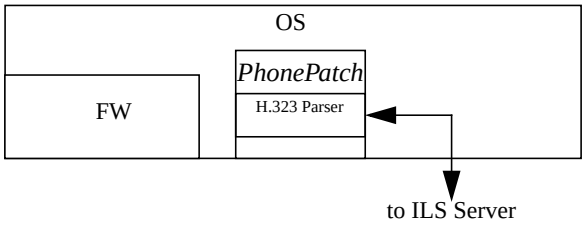


Abbildung 7: PhonePatch Architektur

Alle Internet Location Service (ILS) Anfragen werden durch das PhonePatch Programm geleitet. Dies ermöglicht es, die IP-Zieladressen, die mit Hilfe des ILS Protokolls übermittelt werden, herauszufiltern und zu verändern. Dadurch können die Rufe zum PhonePatch-Host umgeleitet werden. Wenn nachfolgend der eigentliche Verbindungsaufbau erfolgt, kann PhonePatch eine Verbindung zu dem gewünschten Ziel aufbauen. Diese Vorgehensweise zwingt Microsoft Netmeeting dazu, transparent einen Ruf über einen Proxy aufzubauen. Dies funktioniert selbst dann, wenn die Applikationskonfiguration die Verwendung eines Proxies nicht explizit unterstützt (erst Netmeeting v3 unterstützt die Konfiguration eines Proxies für ausgehende Rufe).

**Fazit:**

Dieser Ansatz adressiert NAT-Szenarien, in denen Microsoft Netmeeting mit ILS verwendet wird. Andere Protokollszenarien und generische H.323-Applikationen werden nicht adressiert und können deshalb auch nicht unterstützt werden. Auch hier ist nach unserer Definition keine “IP-Telefonie kompatible Firewall” gegeben, da ein Anpassen der Firewall an verschiedene Szenarien (z.B. andere Terminals) nicht möglich ist.

# 4 Eigener erweiterter Ansatz und prototypische Implementierung

Wie wir gezeigt haben, ist eine gewöhnliche integrierte Firewall-Struktur wie in Abbildung 4 nur bedingt hilfreich für die Behandlung von IP-Telefonie Szenarien. Aus diesem Grund haben verschiedene Hersteller andere Architekturen vorgeschlagen oder implementiert. Diese können bis jetzt stets nur einen Teil der beschriebenen Probleme lösen, ein genereller Ansatz zur Lösung der Probleme nach unserer Definition (siehe 2.5) ist aber nicht verfügbar.

Aus diesem Grund führen wir eine neue Parser-Architektur ein, die die gezeigten Probleme auf allgemeinere Art löst. Die neue Parser-Architektur ist notwendig, um die Kriterien 2, 3 und 4 unsere Definition einer “IP-Telefonie kompatiblen Firewall” zu erfüllen. Wir haben uns entschieden, den Parser außerhalb der konventionellen Firewall zu plazieren.

- Dies ermöglicht es dem Parser-Element mit anderen IP-Telefonie Komponenten zu interagieren. Die Konsequenz daraus ist, daß alle relevanten NAT Szenarien, sowohl eingehender als auch ausgehender Ruf, unterstützt werden können. Dadurch kann Punkt 3 unseres Kriterienkatalogs erfüllt werden.

- Zusätzlich kann das Parser-Element dynamisch geladen und separat von der Firewall konfiguriert werden, z.B. mit einer dafür optimierten Konfigurationssprache. Dies ermöglicht es, Punkt 2 unseres Kriterienkatalogs zu erfüllen, eine Anpassung des Systems an sich ändernde Szenarien.

Die Design-Überlegungen beeinflussen direkt unsere Architektur und Implementierung. Um den Parser aus der Firewall herauslösen zu können, ist ein Firewall-Interface notwendig. Dieses erlaubt es dem Parser wie zuvor, als er integraler Bestandteil des Systems war, mit der Firewall zu interagieren. Dabei lassen sich folgende Schnittstellen definieren:

- ***Firewall Adaption Layer*:**
  Ein Adaptions-Layer (*Firewall Adaption Layer*) wird verwendet, um die generischen Firewall-Konfigurationanweisungen des Parsers in eine für die jeweils verwendete Firewall - die dazu dieses Interface implementieren muß - verständliche Semantik umzusetzen. Die Verwendung eines solchen Layers ermöglicht es ebenfalls, den Parser zusammen mit verschiedenen Firewall-Typen zu verwenden. Dazu ist nur dieser Adaptions-Layer anzupassen. Ein Beispiel für ein generisches Kommando, das über diese Schnittstelle übergeben werden kann, ist die Anweisung zum Öffnen und Schließen von Ports für bestimmte Verbindungen.

- ***Data Adaption Layer*:**
  Zur Übergabe der Daten an den Parser verwenden wir einen sogenannten *Data Adaption Layer*, der für die Übergabe der zu untersuchenden Daten an den Parser verantwortlich ist. Durch die Verwendung eines solchen Layers ist es möglich, die Quelle (und Senke) der zu untersuchenden Daten gezielt zu beeinflussen. Zum Beispiel können die Daten von der Firewall oder direkt von dem darunter liegenden Betriebssystem bereitgestellt werden.

Um die Kommunikation mit anderen IP-Telefonie Komponenten zu ermöglichen wurde zusätzlich eine weitere Schnittstelle implementiert:

- ***IP-Telephony Adaption Layer*:**
  Wir verwenden einen Adaptions-Layer, um mit externen IP-Telefonie Komponenten zu kommunizieren. Der Parser kann generische Anfragen an diese Schnittstelle stellen und der *IP-Telephony Adaption Layer* kann diese in die entsprechende Protokoll Sprache umsetzen und an die gewünschte Komponente weiterleiten. Dies erlaubt es zum Beispiel, eine Parser-Anfrage der Art “Bestimme die Zieladresse für Teilnehmer A” in eine spezifischen DNS, LDAP oder Gatekeeper Anfrage umzusetzen.

Das aus diesen Überlegungen resultierende System ist in Abbildung 8 schematisch-dargestellt.

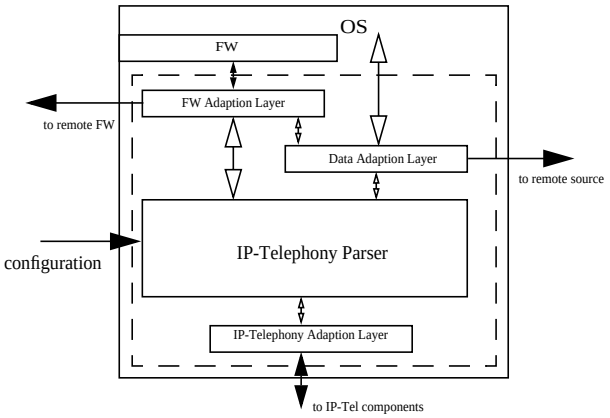


Abbildung 8: Alternative Architektur

Aus diesem Architekturdesign ergeben sich weitere Vorteile. Nicht nur der H.323 Parser kann an bestimmte H.323-Szenarien angepaßt werden, sondern dieser kann zusätzlich ebenfalls an Szenarien angepaßt werden, die auf einem anderen IP-Telefonie Signalisierungsprotokoll aufbauen. Die Unterstützung für SIP- oder heterogene Szenarien kann auf diese Weise durch eine einfache gezielte Modifikation des IP-Telefonie-Parsers realisiert werden. Wie unsere prototypische Implementierung zeigt, kann der Parser verschiedene Firewalls unterstützen. Der Parser muß nicht neu implementiert werden, wenn er für eine andere Firewall portiert wird.

Der nach dem hier beschriebenen Design implementierte Prototyp ist nach unserer Definition “IP-Telefonie kompatibel”, da er alle entsprechenden Kriterien erfüllt.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag haben wir gezeigt, warum die Verwendung von Firewalls innerhalb von IP-Telefonie Szenarien zu Problemen führt. Basierend auf den beschriebenen Problemen wurde eine Definition einer “IP-Telefonie kompatiblen Firewall” gegeben. Es wurden verschiedene Firewall Produkte analysiert und deren Limitierungen dargestellt. Um IP-Telefonie uneingeschränkt in Firewall-Umgebungen einsetzen zu können, schlagen wir eine neue Architektur vor, die den nötigen Anforderungen entspricht. Die prototypische Implementierung dieser Architektur wird von uns zur Zeit evaluiert.

## 6 Literatur

[1] ITU: ITU-T Recommendation H.323, Packet-Based Multimedia Communication Systems. 1998.

[2] Handley, M., Schulzrinne, H., Schooler, E., Rosenberg, J.: RFC 2543, SIP: Session Initiation Protocol. March 1999.

[3] Douskalis, B.: IP Telephony - The Integration of Robust VoIP Services. Prentice Hall, 2000.

[4] Agrawal, H., Roy, R.,Palawat, V., Johnston, A., Agboh, C.,Wang, D., Singh, K., Schulzrinne, H.: SIP-H.323 Interworking Requirements. Internet Engineering Task Force, Jul. 2000.

[5] Chapman, D. B.: Building Internet Firewalls. O’Reilly, Cambridge, 1995.

[6] Cheswick, W. R., Bellovin S. M.: Firewalls and Internet Security. Addison Wesley, 1994.

[7] Egevang, K., Francis, P.: RFC 1631, The IP Network Address Translator. May 1994.

[8] Steinmetz, R.: Multimedia-Technologie. Springer, 1999.

[9] Finlayson, R.: Internet Draft draft-ietf-mboned-mcast-firewall-02.txt, IP Multicast and Firewalls. 1998.

[10] Knobbe, R.; Purtell, A.; Schwab, S.: Advanced security proxies: an architecture and implementation for high-performance network firewalls. In DARPA Information Survivability Conference and Exposition, 1999.

[11] Roedig, U., Ackermann, R., Rensing, C., Steinmetz, R.: A Distributed Firewall for Multimedia Applications. In Proceedings of the Workshop "Sicherheit in Mediendaten", September 2000.

[12] Ellermann, U., Benecke, C.: Parallele Firewalls - skalierbare Lösungen für Hochgeschwindigkeitsnetze. DFN-CERT Workshop Sicherheit in vernetzten Systemen, Hamburg, 1998.

[13] Kuthan, J., Rosenberg, J.: Internet Draft draft-kuthan-fcp-01.txt, Firewall Control Protocol Framework and Requirements. June 2000.

[14] Mercer, S., Moilitor, A., Hurry, M., Ngo, T.: Internet Draft draft-rfced-inf-mercer-00.txt, H.323 Firewall Control Interface (HFCI). June 1999.

[15] ITU: ITU-T Recommendation H.225.0, Call signaling protocols and media stream packetization for packet-based multimedia communications systems. 1998.

[16] ITU: ITU-T Recommendation H.245, Control protocol for multimedia communication. 1998

[17] Hersent, O., Gurle, D., Petit, J.: IP Telephony. Addison Wesley, 2000

[18] Intel:http://support.intel.com/support/videophone/trial21/H323_WPR.HTM.

[19] Cisco: MCM, http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113na/1137na/mcm_cfg.htm.

[20] Goncalves, M., Brown, S.: Checkpoint Firewall 1 Administration Guide. McGraw-Hill, 1999.

[21] PhonePatch: http://www.phonepatch.com.

[22] IInnovaPhone: http://www.innovaphone.com.