H.323 und Firewalls

Probleme und Lösungen

Utz Roedig, Manuel Görtz, Ralf Steinmetz

KIMK GmbH
Philipp-März-Straße 15
64342 Seeheim-Jugenheim

Utz.Roedig@KOM.tu-darmstadt.de
Manuel.Goertz@KIMK.de
Ralf.Steinmetz@KIMK.de

Inhaltsverzeichnis

1. Einleitung

1.1 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 beispielsweise auch Multimedia-Applikationen gehören, ergeben sich neue Anforderungen, die ein Firewall-System erfüllen muß. Diesen neuen Anforderungen werden existierende Firewall-Systeme nicht gerecht, weshalb Multimedia-Applikationen von Firewalls zur Zeit nicht zufriedenstellend unterstützt werden können.
In dieser Studie wird gezeigt, welche speziellen Probleme sich bei der Integration einer Unterstützung von Multimedia-Applikationen in eine Firewall ergeben. Des weiteren wird der Grund für die auftretenden Probleme erläutert. Abschließend wird diskutiert, wie diese Probleme gelöst werden können.
Bei der Beschreibung der Probleme sowie der entsprechenden Lösungen wird das im DFN verwendete Videokonferenzsystem betrachtet. Dieses Videokonferenzsystem basiert auf dem H.323-Protokoll, weshalb in dieser Studie im wesentlichen H.323-Applikationen als spezielle Vertreter von Multimedia-Applikationen betrachtet werden.

1.2 Inhalt

Die vorliegende Studie ist folgendermaßen aufgebaut:

Kapitel 2. beschreibt die zum Verständnis dieser Studie notwendigen Grundlagen der Themengebiete Sicherheit, Firewalls, Multimedia-Applikationen sowie H.323-Applikationen.

Kapitel 3. beschreibt das in dieser Studie verwendete H.323 Referenzszenario.

Kapitel 4. beschreibt, welchen Bedrohungen ein H.323-Szenario ausgesetzt ist. Es werden Beispiele von Angriffen auf einzelne H.323-Komponenten vorgestellt. Es wird gezeigt, welche Angriffe auf H.323-Systeme mit Hilfe einer Firewall verhindert werden können.

Kapitel 5. beschreibt die grundlegenden Probleme, die bei der Verwendung von Multimedia Applikationen im Firewall-Umfeld auftreten.

Kapitel 6. beschreibt die Probleme, die speziell bei der Verwendung des H.323-Protokolls im Zusammenhang mit Firewalls entstehen.

Kapitel 7. beschreibt generell, wie die zuvor in Kapitel 6 beschriebenen Probleme gelöst werden können.

Kapitel 8 . beschreibt spezifische Lösungen für den Einsatz von H.323-Applikationen in einem H.323-Umfeld.

Kapitel 9. beschreibt das im DFN eingesetzte H.323-Szenario. Es wird gezeigt, wie die in Kapitel 7 und 8 aufgezeigten Lösungen im DFN-Umfeld umgesetzt werden können.

Kapitel 10. faßt die in dieser Studie gesammelten Erkenntnisse zusammen und gibt Empfehlungen für den Einsatz von H.323-Applikationen im DFN-Umfeld im Zusammenhang mit Firewalls.

2. Grundlagen

In diesem Kapitel werden die Grundlagen, welche zum Verständnis der Studie notwendig sind, erläutert. Es werden die notwendigen Teilbereiche der Themengebiete *Sicherheit*, *Firewalls* und *Multimedia-Kommunikation* erläutert. Für die wesentlichen Begriffe und Konzepte werden jeweils Definitionen gegeben, insbesondere wenn aus der Literatur keine oder nicht eindeutige Definitionen bekannt sind. Eine umfangreiche Darstellung der einzelnen Themengebiete kann in diesem Dokument nicht erfolgen, weshalb an den entsprechenden Stellen auf weiterführende Literatur verwiesen wird.

2.1 IT-Sicherheit

An ein IT-System werden die unterschiedlichsten Anforderungen gestellt. An erster Stelle stehen dabei zunächst Anforderungen hinsichtlich Funktionsumfang, Funktionalität und Wirtschaftlichkeit. Die Tatsache, daß ein IT-System ebenfalls auch Anforderungen hinsichtlich der Sicherheit gerecht werden muß, wird allerdings aus verschiedenen Gründen nicht in genügendem Maß berücksichtigt. Ein wesentlicher Grund dafür mag sein, daß Sicherheit eine nicht-funktionale Eigenschaft ist. Ein Benutzer, dessen System korrekt funktioniert, kann eigentlich nicht sagen, ob es nun auch sicher ist oder nicht[1]. Dennoch ist für den letztendlichen Erfolg eines IT-Systems auch dessen Sicherheit von Bedeutung, da dies Einfluß auf die primären Anforderungen wie z.B. Funktionalität oder Wirtschaftlichkeit hat.
Insbesondere die zweitrangige Betrachtung der Sicherheit führt dazu, daß IT-Systeme erst nachträglich in ein Sicherheitskonzept integriert werden, bzw. ein Sicherheitskonzept für sie erstellt wird. Dies führt in der Regel dazu, daß Sicherheitsanforderungen nicht mehr in dem Maße umgesetzt werden können, wie dies bei einer Beachtung dieser Anforderungen schon bei der Konzeption des IT-Systems möglich gewesen wäre. Diese Problematik kann durch ein Beispiel belegt werden. Das für die IP-Telefonie verwendete H.323-Protokoll [1], - und damit auch die darauf aufbauenden IP-Telefoniesysteme - wurde zunächst hinsichtlich der Anforderung Funktionalität entwickelt. Sicherheitsanforderungen spielten bei der Entwicklung der ersten Version des H.323-Protokolls (fast) keine Rolle. Dadurch ergeben sich zwei Probleme:

- Die nachträgliche Integration von Sicherheitsanforderungen innerhalb der H.323-IT-Systeme ist nur unter Berücksichtigung der schon zuvor festgelegten Gegebenheiten möglich. Beispielsweise muß eine Authentifizierung von Gesprächsteilnehmern in die gegebene H.323-Signalisierung integriert werden, obwohl diese zu diesem Zweck nicht optimal ausgelegt ist. Daß dies ein Problem darstellt zeigt sich mitunter an den zahlreichen Standardisierungsbemühungen auf diesem Gebiet [5] [6] [7].

- Die nachträgliche Anpassung der H.323-IT-Systeme an gegebene Sicherheitsanforderungen ist nicht möglich, ohne die gegebenen Sicherheitsanforderungen zu verändern.

1. Von Edsger W. Dijkstra stammt die Aussage, daß das Testen immer nur das Vorhandensein von Fehlern zeigen kann, aber niemals die Fehlerfreiheit (=Sicherheit).

Beispielsweise müssen die Sicherheitsanforderungen (Sicherheits-Level) dahingehend eingeschränkt werden, daß Firewalls mit H.323-IT-Systemen zurecht kommen.

Aber selbst wenn man annimmt, daß Sicherheitsanforderungen im ersten Schritt zusammen mit anderen Anforderungen bei der Konzeption eines IT-Systems beachtet werden[1], ergibt sich eine Wechselwirkung zwischen den verschiedenen Anforderungen. Einzig der mögliche Suchraum für eine optimale Gesamtlösung vergrößert sich.
Diese Studie beschäftigt sich mit der Kombination von IT-Systemanforderungen mit Sicherheitsanforderungen. Dabei beschränkt sich die Studie auf die Betrachtung der Anforderungen von Multimedia-Applikationen sowie der Sicherheitsanforderungen die in Firewalls umgesetzt werden.
Die für diese Studie relevante Sicht auf das Themengebiet Sicherheit ist ausführlicher in [8] und [9] beschrieben. Weiteres zu diesem Themengebiet findet sich in [10], [11] und anderen.

2.1.1 Begriffe aus der IT-Sicherheit

Im Folgenden werden die aus dem Bereich IT-Sicherheit verwendeten Begriffe, die innerhalb dieser Studie verwendet werden definiert. Eine Beschreibung dieser Begriffe ist notwendig, da in der Literatur unterschiedliche Definitionen existieren.

Sicherheit. Der Begriff Sicherheit wird in den unterschiedlichsten Bereichen mehr oder weniger überlegt verwendet. Wesentlich für eine Begriffsbildung ist, daß IT-Sicherheit nur auf den ersten Blick ein rein technisches Problem ist. IT Sicherheit wird von Menschen für Menschen umgesetzt, dementsprechend muß den sich daraus ergebenden Aspekten ebenfalls Rechnung getragen werden [12]. Der technische - und für die vorliegende Studie wesentliche - Aspekt der Sicherheit kann folgendermaßen beschrieben werden: *Sicherheit von IT-Systemen ist der Schutz von Verfügbarkeit, Vertraulichkeit und Integrität.*

Schutzziel. Entsprechend der obigen Definition sind die Schutzziele in IT-Systemen: *Verfügbarkeit, Vertraulichkeit und Integrität*.

Bedrohung. In Anlehnung an [13] kann der Begriff Bedrohung folgendermaßen beschrieben werden: *Eine Bedrohung tritt dann auf, wenn es entsprechende Umstände, Möglichkeiten, Aktionen oder Ereignisse gibt, die die Sicherheit gefährden.*

Angriff. Eine mögliche Bedrohung stellen Angriffe auf IT-Systeme dar, d.h. ein Angreifer versucht, die Sicherheit eines IT-Systems zu unterwandern und zu seinem Vorteil auszunutzen. Eine spezielle Form des Angriffs ist ein sog. Denial of Service (DoS) Angriff. Ziel eines DoS Angriffs ist es, ein IT-System (zeitweise) unbenutzbar zu machen.

Verletzbarkeiten. Damit ein Angriff überhaupt stattfinden kann, muß ein IT-System Sicherheitslücken aufweisen, die ausgenutzt werden können.

1. Was in der Regel nicht der Realität entspricht, und deshalb eine Betrachtungsweise mit geringem praktischen Wert darstellt.

Maßnahmen. Es gibt eine Reihe alternativer Strategien, um Maßnahmen zum Schutz vor Bedrohungen zu erreichen. Es ergänzen sich dabei proaktive und reaktive Maßnahmen. Primäres Ziel ist es, die Auswirkungen von Verletzbarkeiten zu eliminieren oder aber den Schaden zu begrenzen. Proaktive Maßnahmen richten sich gegen Bedrohungen, die vorhergesehen wurden und gegen die bereits im Vorfeld Vorkehrungen getroffen werden können. Reaktive Maßnahmen zielen darauf ab, einen Angriff als solchen überhaupt zu erkennen. Nur wenn ein Angriff erkannt worden ist, kann korrigierend eingegriffen werden, d.h. es werden entsprechende Schritte als Reaktion auf den Angriff eingeleitet.

2.2 Firewalls

Eine Firewall ist eine von vielen möglichen Maßnahmen zum Schutz vor speziellen Bedrohungen. In erster Linie handelt es sich bei einer Firewall um eine proaktive Maßnahme[1] zum Schutz vor Bedrohungen. Eine Firewall ermöglicht es, an einem Übergangspunkt zwischen zwei, meist unter getrennter administrativer Verwaltung stehender Netzwerke, Überprüfungen und/oder Modifikation der über die Netzgrenze fließenden Daten anhand der gegebenen Anforderungen (Security Policy) vorzunehmen. Dementsprechend kann eine Firewall nur als Maßnahme gegen Bedrohungen eingesetzt werden, die an dem entsprechenden Netzübergang, an dem die Firewall eingesetzt wird, neutralisiert werden kann. Eine Firewall kann also nicht die alleinige Maßnahme zur Beseitigung aller Bedrohungen darstellen und muß durch andere Maßnahmen ergänzt werden.
Eine Firewall implementiert bestimmte Funktionen, um bestimmte Bedrohungen zu neutralisieren. In der Literatur (z.B. [14], [15], [16]) herrscht dabei Uneinigkeit, welche Aufgaben bzw. Funktionen genau von einer Firewall übernommen werden bzw. dieser zugerechnet werden müssen. Im wesentlichen können alle Maßnahmen die der Vermeidung von Bedrohungen an einer Netzgrenze dienen, einer Firewall zugesprochen werden. Innerhalb der Studie werden folgende Kernfunktionen als Definition für eine Firewall verwendet:

- **Authentifizierung:**
  Die Firewall muß in der Lage sein, die Identitäten von Sender und Empfänger der über die Firewall laufenden Daten festzustellen. Diese Prüfung kann auf verschiedene Arten erfolgen, die sich hinsichtlich ihrer Parameter - z.B. technische Umsetzung der Prüfung, Güte, Granularität usw. - unterscheiden. Beispielsweise kann als Identität ein Benutzer, ein bestimmter Prozeß auf einem Rechner oder ein Rechner verwendet werden.

- **Analyse:**
  Die Firewall muß in der Lage sein, die über sie fließenden Daten zu analysieren, um zu entscheiden, ob durch sie eine Bedrohung entstehen kann oder nicht. Beispielsweise ist eine Analyse der Semantik der Header einzelner Protokolle, die Analyse der durchlaufenden Zustände einer Protokollzustandsmaschine, die zwischen zwei Kommunika-

1. Eine Firewall ist durch die Integration bestimmter Intrusion Detection Mechanismen auch in begrenztem Umfang eine Reaktive Maßnahme.

tionspartnern abgewickelt wird, oder die Analyse der Zusammenhänge verschiedener parallel aktiver Kommunikationsverbindungen denkbar.

- **Filterung und Modifikation:**
  Die Firewall muß in der Lage sein, zu entscheiden ob die Daten weitergeleitet werden oder nicht (Filterung). Als Entscheidungsgrundlage dienen die durch *Analyse* und *Authentifizierung* gewonnenen Informationen. Zusätzlich muß die Firewall entscheiden, ob Daten vor einer Weiterleitung modifiziert werden müssen, um Bedrohungen zu entfernen.

- **Verbergen:**
  Eine Firewall muß in der Lage sein, die auf der einen Seite der Netzgrenze liegenden Strukturen vor der anderen Seite zu verbergen. Dies entzieht einem Angreifer die Möglichkeit, sich Informationen über Ziele zu verschaffen. Beispielsweise die heute in fast jeder kommerziellen Firewall verwendete Network Address Translation (NAT) stellt die Umsetzung dieser Aufgabe dar.

Diese Kernfunktionen einer Firewall stellen ebenfalls die wesentlichen Bereiche dar, in denen es zu Problemen bei einem Datenaustausch von Multimedia-Applikationen über eine Firewall kommt (siehe Kapitel 5. auf Seite 22). Aus diesem Grund eignet sich die gegebene Definition für die vorliegende Studie.
Die von einer Firewall zu implementierenden Funktionen müssen technisch in einem Firewall-System - beschrieben durch die Firewall-Architektur und bestehend aus den einzelnen Firewall-Komponenten sowie der Festlegung ihrer Interaktionsprinzipien - realisiert werden. Bei der Festlegung der grundlegenden Funktionen einer Firewall findet sich in der Literatur noch eine relativ hohe Übereinstimmung. Hingegen ist die technische Umsetzung der Funktionen - und damit die Definition der einzelnen Firewall-Komponenten und insbesondere deren Interaktion - nicht allgemeingültig festgelegt. Demnach ergeben sich für die Optimierung einer Firewall für einen bestimmten Zweck Variablen hinsichtlich der Implementierung, nicht aber hinsichtlich ihrer Funktion[1]. Prinzipiell ist jede beliebige Implementierung, die die nötigen Funktionen bereitstellt, ausreichend. Dennoch gibt es einige bewährte Konzepte, die zur Implementierung verwendet können, welche im folgenden beschrieben sind.

1. Es sei denn man ändert die Sicherheitsanforderungen, was zu einer Änderung der benötigten Funktionen innerhalb einer Firewall führt.

2.2.1 Firewall Komponenten

Zur Umsetzung der zuvor beschriebenen Firewall-Funktionen werden in der Regel die hier beschriebenen Standard-Firewall-Komponenten verwendet.

Paketfilter. Ein Paketfilter besteht aus einer Netzwerkkomponente - z.B. einer Bridge oder einem Router - die zum Verbinden zweier Netzwerke verwendet wird und die durch einen Filter erweitert wird. Der Filter prüft anhand zuvor festgelegter Kriterien (den Filterregeln), ob die von der Netzwerkkomponente weiterzuleitenden Datenpakete eine Bedrohung darstellen.

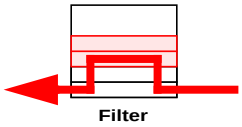

Abbildung 1: Paketfilter

Stellt ein Datenpaket eine Bedrohung dar, wird es verworfen und nicht weitergeleitet. Die an bestimmten Positionen enthaltenen Werte innerhalb der Datenpakete werden dabei mit den zuvor festgelegten Filterregeln verglichen, um zu entscheiden, ob eine Bedrohung vorliegt oder nicht. Im wesentlichen werden die in den Header-Feldern der Vermittlungs- und Transportschicht enthaltenen Werte innerhalb der Datenpakete geprüft. Da jede Prüfung eines Paketes ohne Berücksichtigung der bereits analysierten Pakete durchgeführt wird, können die Daten der Applikationsebene nicht geprüft werden. Eine Analyse der Daten der Applikationsebene würde es erfordern, Applikationsdaten sowie Zustände der jeweiligen Applikationsprotokollautomaten vorzuhalten. Für solche Analysen müssen Stateful Filter oder Proxies verwendet werden. IP-Paketfilter untersuchen beispielsweise die Pakete nach Typ des Pakets (z.B. TCP oder UDP), nach IP-Adressen des Senders und Empfängers, sowie nach den Port-Nummern des Senders und Empfängers. Pakete, die dann beispielsweise an einen bestimmten Empfänger gerichtet sind, können dann als Bedrohung erkannt (wenn dies in den Filterregeln so festgelegt wurde) und die Weiterleitung dieser Pakete an den Empfänger verhindert werden. Eine Firewall, die nur durch die Verwendung eines Paketfilters realisiert wird, kann demnach folgende Funktionen umsetzen:

- **Authentifizierung:** Der Paketfilter kann die Identität der Kommunikationsteilnehmer anhand der verwendeten Port-Nummern und der IP-Adressen bestimmen[1].
- **Analyse:** Die Header-Felder der Vermittlungs- und Transportschicht können überprüft werden.
- **Filterung und Modifikation:** Die Firewall entscheidet ob die Daten weitergeleitet werden oder nicht (Filterung). Eine Modifikation der Daten ist vor dem Weiterleiten möglich.
- **Verbergen:** Ein Verbergen der internen Netzstrukturen ist mit einem Filter allein nicht möglich.

Stateful Filter. Bei einem Stateful Filter handelt es sich um einen einfachen Paketfilter (siehe oben), der zusätzlich in der Lage ist, bei der Prüfung der Pakete auch auf Informationen zurückzugreifen, die durch die Prüfung von zuvor verarbeiteten Paketen gewonnen wurden.

1. Die Ports/IP-Adressen können dabei aber auch gefälscht worden sein.

Ein Stateful Filter für das IP-Protokoll ist beispielsweise in der Lage zu prüfen, ob ein TCP-Paket zu einer bereits ordnungsgemäß geöffneten TCP-Verbindung gehört oder nicht.

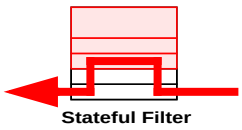

Abbildung 2: Stateful Filter

Der Stateful Filter "merkt" sich, daß zwischen dem Sender und Empfänger bereits TCP-Pakete für den TCP-Verbindungsaufbau in der richtigen Reihenfolge ausgetauscht wurden. Wird dann zu einem späteren Zeitpunkt ein TCP-Paket durch den Stateful Filter geprüft, kann dieser feststellen, ob das Paket den festgelegten Filterregeln entspricht (wie bei einem normalen IP-Paketfilter) und zusätzlich im Kontext der aktuellen TCP-Verbindungen Sinn macht. Ein Stateful Filter kann auch in der Lage sein, einzelne Protokolle der Applikationsschicht zu "verstehen" und dadurch auch Informationen zur Prüfung der Daten heranziehen, die nur auf dieser Schicht verfügbar sind. Beispielsweise kann ein Stateful Filter, der die Applikationssemantik des FTP-Protokolls "versteht", die auf dem FTP Kontrollkanal ausgehandelten IP-Adressen und Portnummern für den FTP-Datenkanal ermitteln. Diese Informationen stehen dann für die Prüfung der folgenden TCP-Pakete zur Verfügung; TCP-Pakete können als Pakete einer FTP-Datenverbindung identifiziert werden. Eine Firewall, die nur durch die Verwendung eines Stateful Filter realisiert wird, kann demnach folgende Funktionen umsetzen:

- **Authentifizierung:** Der Stateful Filter kann die Identität der Kommunikationsteilnehmer anhand der verwendeten Portnummern und der IP-Adressen bestimmen. Zusätzlich können benutzerbezogene Informationen, die innerhalb der Applikationsschicht eventuell vorhanden sind, verwendet werden.

- **Analyse:** Die Header-Felder der Vermittlungs-, Transport- und Applikationsschicht können überprüft werden. Zusätzlich kann die Verbindungssemantik überprüft werden.

- **Filterung und Modifikation:** Die Firewall entscheidet, ob die Daten weitergeleitet werden oder nicht (Filterung). Eine Modifikation der Daten ist vor dem Weiterleiten möglich.

- **Verbergen:** Ein Verbergen der internen Netzstrukturen ist mit einem Stateful Filter allein nicht möglich.

Proxy. Ein Proxy wird in den Kommunikationspfad zwischen zwei Kommunikationspartnern eingebracht, indem er für die jeweiligen Kommunikationsteilnehmer den Kommunikationsendpunkt widerspiegelt.

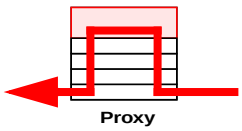

Abbildung 3: Proxy

Dadurch wird die Verbindung zwischen zwei Kommunikationspartnern durch den Proxy aufgetrennt. Alle Datenpakete, die an den Kommunikationsteilnehmer gesendet werden, werden durch den Proxy entgegengenommen und auf Applikationsebene verarbeitet. Dazu benötigt der Proxy "Wissen" über das der Kommunikation zugrunde liegende Applikationsprotokoll. Bei der Verarbeitung wird geprüft, ob sich durch die Daten Bedrohungen ergeben können. Dies geschieht

anhand zuvor festgelegter Kriterien. Als Entscheidungsgrundlage, ob die Daten an den Empfänger weitergeleitet werden oder nicht, können nur die aus der Applikationsschicht zu gewinnenden Informationen verwendet werden. Informationen, die in den Daten der unteren Schichten enthalten sind, stehen einem Proxy nicht zur Verfügung. Eine Firewall, die nur durch die Verwendung eines Proxies realisiert wird, kann demnach folgende Funktionen umsetzen:

- **Authentifizierung:** Der Proxy kann die Identität der Kommunikationsteilnehmer anhand der benutzerbezogenen Informationen, die innerhalb der Applikationsschicht eventuell vorhanden sind, prüfen.

- **Analyse:** Die Header/Daten der Applikationsschicht können überprüft werden. Zusätzlich kann die Applikationssemantik überprüft werden.

- **Filterung und Modifikation:** Die Firewall entscheidet, ob die Daten weitergeleitet werden oder nicht (Filterung). Eine Modifikation der Daten ist vor dem Weiterleiten möglich.

- **Verbergen:** Da die Kommunikationsteilnehmer durch den Proxy auf allen Schichten vollständig voneinander getrennt sind, können durch einen Proxy die Netzstrukturen verborgen werden.

Network Address Translation (NAT). Eine NAT-Komponente ermöglicht es, in einem Subnetz IP-Adressen zu verwenden, die im Internet keine Gültigkeit besitzen. Schickt ein Sender innerhalb dieses Subnetzes ein IP-Paket an einen Empfänger im Internet, so wird die an der Grenze des Subnetzes verwendete NAT-Komponente die im Internet ungültige Adresse des Absenders durch eine gültige ersetzen. Die NAT-Komponente muß sich diese Adreßumsetzung merken, um die eintreffenden Antwortpakete an den korrekten Empfänger durch eine weitere Umschreibung der IP-Adresse weiterleiten zu können. Neben dem Effekt, daß auf diese Weise ein gesamtes Subnetz nur eine gültige IP-Adresse benötigt, um mit Rechnern im Internet zu kommunizieren, beinhaltet dieser Mechanismus auch eine Schutzfunktion. Rechner innerhalb des privaten Subnetzes können aus dem Internet nicht adressiert werden, da sie keine gültige IP-Adresse besitzen. Eine Kommunikation mit internen Geräten ist erst möglich, nachdem eine Verbindung von innen initiiert wurde, und dieses Mapping in der Adreßumsetzungstabelle der NAT-Komponente eingetragen ist. Dadurch ermöglicht ein solches Gerät das Verbergen der internen Strukturen eines Netzwerks.

2.2.2 Firewall Systeme

Um alle für eine Firewall notwendigen Funktionen zu implementieren, werden in der Regel verschiedene der zuvor beschriebenen Firewall-Komponenten - die meist nur einen Teil der Funktionen bereitstellen - zu einem Firewall System zusammengefaßt.

Hybridsystem. Eine gängige Kombination besteht darin, einen Proxy mit einem Stateful Filter und einer NAT-Komponente zu verbinden. Der Proxy (für verschiedene Applikationstypen) befindet sich dabei auf demselben Gerät, auf dem sich auch der Stateful Filter und die NAT-Komponente befinden. Die Komponenten sind dabei in der Lage, über eine geeignete Schnittstelle miteinander zu interagieren. Dadurch werden die durch die einzelnen Komponenten bereitgestellten Funktionen von dem entstehenden Hybridsystem erbracht. Dadurch, daß eine Interaktionsmöglichkeit zwischen Proxy und Stateful Filter besteht, kann der Stateful Filter sich darauf beschränken, die Analyse der Daten auf Vermittlungs- und Transportschicht zu übernehmen. Die Analyse der Applikationsdaten wird durch den Proxy übernommen. Es ist in einer weiteren Variation ebenfalls möglich, auf den Proxy zu verzichten, indem seine Funktionen durch den Stateful Filter vollständig erbracht werden. Es besteht demnach die Wahl den Stateful Filter entsprechend komplex zu gestalten und dafür auf den Proxy zu verzichten oder aber den Stateful Filter einfach zu gestalten und einen Proxy bereitzustellen. Heutige auf dem Markt erhältliche Firewall-Systeme entsprechen von ihrem inneren Aufbau diesen Hybridsystemen. Beispiele dafür sind Checkpoints Firewall-1 [17]oder Ciscos PIX [18]. Hybridsysteme werden in der Praxis häufig als einziges Element verwendet, um eine Firewall aufzubauen.

Verteilte Systeme. Oft werden die zuvor beschriebenen Komponenten so an einer Netzgrenze verwendet, daß die einzelnen Komponenten von den Datenströmen nacheinander durchlaufen werden.

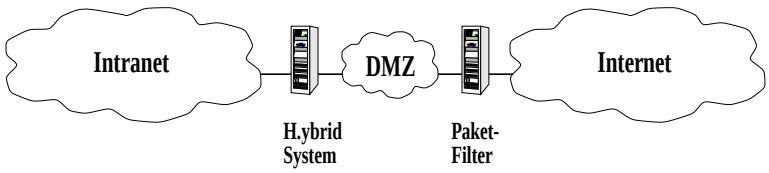

Abbildung 4: Demilitarisierte Zone (DMZ)

Zwischen den einzelnen verwendeten Komponenten besteht dabei aber meist keine Möglichkeit der Interaktion. Durch die Hintereinanderschaltung wird zum einen erreicht, daß alle für eine Firewall nötigen Funktionen erbracht werden. Zum anderen aber kann durch die Hintereinanderschaltung verschiedener Komponenten erreicht werden, daß ein möglicher Fehler in einer Komponente nicht zu einem Verlust der Schutzfunktion der gesamten Firewall führt. Eine häufig verwendete Hintereinanderschaltung von Komponenten ist in Abbildung 4 dargestellt. Der Paketfilter ist dem Hybridsystem vorgeschaltet. Sollte der Paketfilter fehlerhaft sein, so muß von einem Angreifer immer noch das Hybridsystem überwunden werden. Weitere Beschreibungen solcher verteilter Firewall-Systeme finden sich in [14], [15], [16].

Parallele Systeme. Die herkömmlichen Firewall-Architekturen bestehen zumeist aus jeweils einer der oben beschriebenen Firewall-Komponenten. Jedoch in Hochgeschwindigkeitsnetzen mit

Datenraten von bis zu 1Gbps reichen die von diesen Systemen bereitgestellten Übertragungsraten nicht mehr aus.

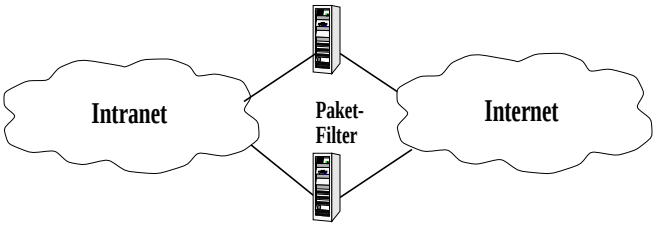

Abbildung 5: Parallele Paketfilter

Eine Möglichkeit, den hohen Datenraten gerecht zu werden, ist die parallele Verwendung mehrerer gleicher Firewall-Komponenten. Einfache Paketfilter mit gleicher Konfiguration können wie in [19] beschrieben parallel eingesetzt werden, um die mögliche Datenrate zu erhöhen. Mehrere Stateful Filter sowie Proxies können so betrieben werden, daß eine Lastverteilung zwischen ihnen stattfindet [19]. Diese Techniken können nicht mehr ohne weiteres verwendet werden, wenn sich Konfigurationen dynamisch ändern, und/oder aber nicht sichergestellt werden kann, daß alle Flows einer Session über dieselben Komponenten laufen.

2.3 Multimedia-Applikationen

Um erläutern zu können, wie der Begriff “Multimedia-Applikation” bzw. “Multimedia-Protokoll” im Kontext dieser Studie verstanden wird, müssen zuerst die Begriffe *Flow* und *Session* erläutert werden. Beide Begriffe können dazu verwendet werden den von einer Multimedia-Applikation verarbeiteten Datenstrom auf verschiedenen Granularitätsstufen zu beschreiben.

Flow. Ein Flow ist ein einzelner Datenstrom, der durch ein Tupel (Quelladresse, Quellport, Zieladresse, Zielport und Protokollnummer) beschrieben ist. In vielen Dokumenten wird der Begriff Kanal (Channel) verwendet, um einen einzelnen Datenstrom zu beschreiben. In diesem Dokument werden wir diese Begriffe synonym verwenden.

Session. Eine Session wird von einer Multimedia-Applikation dazu verwendet, eine multimediale Kommunikationsinstanz zu verwalten. Eine Session beinhaltet mehrere Flows, welche die innerhalb der Kommunikationsinstanz anfallenden Daten transportieren.

Multimedia-Protokoll. Ein Multimedia-Protokoll definiert unter anderem die Struktur einer Session. Das Multimedia-Protokoll definiert, wie viele Flows für eine Session verwendet werden, wie die einzelnen Flows initiiert und verwendet werden, sowie welche Inhalte und in welcher Form diese von den Flows transportiert werden. Ein Multimedia-Protokoll definiert mindestens zwei Flows pro Session. Ein Flow wird für den Transport von Signalisierungsdaten verwendet, der andere zum Transport eines kontinuierlichen Mediums.

Multimedia-Applikation. Eine Multimedia-Applikation wird verwendet um über ein Netzwerk mindestens ein kontinuierliches Medium zu transportieren und um dieses dann zu verarbeiten. Der Transport des Mediums erfolgt mit Hilfe der innnerhalb des entsprechenden Multimedia-Protokolls festgelegten Session. Zusätzlich kann eine Multimedia-Applikation diskrete Medien trans-

portieren und darstellen. Außerdem kann die Multimedia-Applikation zur Erbringung ihrer Funktionen zusätzliche Protokolle verwenden.

Kommunikationsverhalten. Die meisten Multimedia-Applikationen kommunizieren auf die im folgenden beschriebene Art und Weise. Der Client verbindet sich mit dem Server über eine TCP-Verbindung, den sogenannten Kontrollkanal.

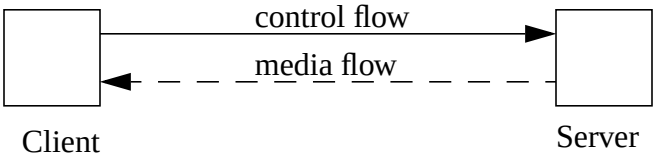

Abbildung 6: Multimedia-Applikation

Nachdem der Kontrollkanal aufgebaut ist, öffnet die Multimedia-Applikation, wie in Abbildung 6 gezeigt, einen oder mehrere weitere Kanäle, über die Audio und/oder Video Daten übertragen werden. Die Portnummern, die dabei für den Medienkanal verwendet werden, werden dabei in der Regel dynamisch über den Kontrollkanal ausgehandelt.

2.4 H.323-Applikationen

IP-Telefonieapplikationen 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 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].

2.4.1 H.323 Grundlagen

Im Jahr 1996 wurde von der ITU-T der Standard H.323 mit dem Ziel verabschiedet, multimediale Kommunikation über lokale Netzwerke, die keine Dienstgüte gewährleisten, zu ermöglichen. Die Kommunikationsdienste beinhalten Übertragungen von Audio in Echtzeit, Video (optional) und Daten (optional) für Punkt-zu-Punkt und Multipoint-Konferenzen. Der H.323-Standard wird laufend weiter entwickelt; die einzelnen Entwicklungsstufen äußern sich in den verschiedenen Versionen des Standards. Aktuell ist die Version 4 des H.323-Standards.
Der H.323-Standard beschreibt alle Komponenten eines H.323-Systems und die Kommunikation dieser Komponenten untereinander durch die Verwendung von Kontrollnachrichten und Prozeduren. Da zu diesem Zweck auch andere Protokolle wie RTP, weitere ITU-Protokolle wie H.225.0,

H.245 und Kodierungsprotokolle einbezogen sind, wird H.323 auch als Protokollfamilie bezeichnet. Außerhalb des Rahmens von H.323 sind Definitionen von Netzwerkschnittstellen, physischen Netzwerken oder verwendeten Netzwerkprotokollen. Außerhalb des Rahmens befinden sich dementsprechend auch die Interaktion mit Firewalls bzw. Firewall-Systemen.

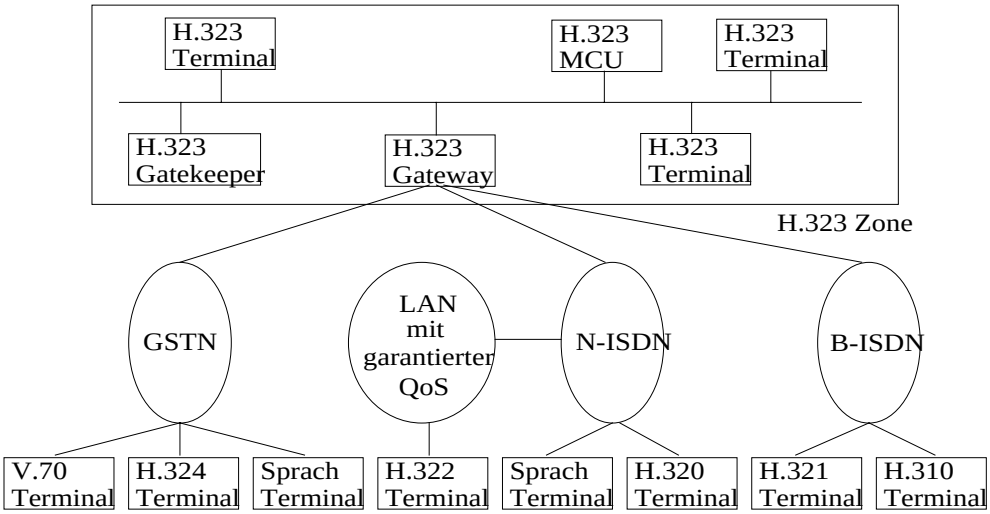

Abbildung 7: H.323-Architektur

In Abbildung 7 ist die Gesamtarchitektur dargestellt. Der H.323-Standard beschreibt die Komponenten und Mechanismen der H.323-Zone. Diese werden im folgenden soweit beschrieben, wie es für das Verständnis der Firewall-/H.323-Problematik notwendig ist. Details der H.323-Protokollfamilie, sowie der darin verwendeten Standards, sind in den jeweils angegebenen Referenzen zu finden.

RAS Kontrolle - H.225.0. Die RAS Kontrolle erfolgt mit H.225.0-Nachrichten für die Registrierung und Zulassung von Teilnehmern, die Veränderung der Bandbreite, den Status und die Auflösung von Verbindungen zwischen Endpunkten und Gatekeepern. Der RAS-Kanal ist unabhängig vom H.225.0-Gesprächssignalisierungskanal und dem H.245-Kontrollkanal und kommt ausschließlich in Netzwerken zum Einsatz, in denen sich ein Gatekeeper befindet. In einem Netzwerk mit Gatekeeper wird der RAS-Kanal zwischen dem Gatekeeper und den zugehörigen Endpunkten vor der Öffnung weiterer Kanäle zwischen den H.323-Endpunkten eingerichtet und bleibt solange geöffnet, bis ein Endpunkt seine Registrierung beim Gatekeeper auflöst. Der Transport von RAS-Nachrichten erfolgt über UDP.

Verbindungskontrolle H.225.0/Q.931. Der Standard H.225.0 [21] spezifiziert den Verbindungsaufbau zwischen zwei H.323-Endpunkten durch Kontrollpakete des Protokolls Q.931. Nach der Einrichtung einer TCP-Verbindung zwischen den Kommunikationspartnern werden über den H.225.0-Gesprächssignalisierungskanal die für den Gesprächsaufbau nötigen Q.931-Nachrichten ausgetauscht. Der H.225.0-Gesprächssignalisierungskanal ist unabhängig vom RAS-Kanal und H.245-Kontrollkanal und wird vor der Einrichtung eines H.245-Kontrollkanals oder anderer logischer Kanäle geöffnet. In Systemen ohne Gatekeeper wird der Gesprächssignalisierungskanal

unmittelbar zwischen den Endpunkten eingerichtet, in Systemen mit Gatekeepern hingegen erfolgt die Einrichtung zwischen den Endpunkten und den zugehörigen Gatekeepern.

Kommunikationskontrolle H.245. Über den H.245-Kontrollkanal werden Ende-Zu-Ende Kontrollnachrichten ausgetauscht, um die Arbeitsweise von H.323-Einheiten zu steuern [22]. Dazu gehören der Austausch der technischen Fähigkeiten (z.B. die möglichen Codecs), das Öffnen und Schließen der logischer Kanäle (z.B. der Audio Kanäle), Anfragen nach Moduspräferenzen, Flußkontrollnachrichten und weitere Befehle oder Verbindungsmerkmale. Der Austausch von H.245-Nachrichten findet entweder zwischen zwei Endpunkten, einem Endpunkt und einem Multipoint Controlle (MC) oder aber einem Endpunkt und einem Gatekeeper statt. Jeder Endpunkt richtet nur einen H.245- Kontrollkanal für eine Verbindung ein. Ein Terminal, eine MCU, ein Gateway oder ein Gatekeeper können viele Verbindungen und daher auch viele H.245-Kontrollkanäle unterstützen. Ein H.245-Kontrollkanal ist von der Verbindungseinrichtung an bis zum Verbindungsende geöffnet.

2.4.2 Medien

RTP/RTCP. Das Realtime Transport Protocol (RTP) [23] hat die Aufgabe, Datenströme in Echtzeit (z. B. Audio oder Video) zu transportieren. In erster Linie wurde RTP für Multicast von Daten entwickelt. Eine Verwendung für Unicast ist jedoch auch möglich. RTP verwendet das User Datagram Protocol (UDP) für den verbindungslosen und unzuverlässigen Transport von Daten über paketbasierte Netzwerke. Zusätzliche Kontrollinformationen während des Datenaustauschs liefert das Realtime Transport Control Protocol (RTCP) [23].
Damit multimediale Datenströme bei den Empfängern in Echtzeit verarbeitet werden können, stellt RTP verschiedene Mechanismen zur Verfügung: Zeitmarkierung, Sequenznumerierung, Typ-Identifikation der Nutzlast, Sender-Identifikation.
Während einer RTP-Sitzung liefert RTCP-Informationen über die Teilnehmer und die Qualität der ankommenden Daten. Zu diesem Zweck unterscheidet RTCP fünf unterschiedliche Paket-Typen, die Kontrollinformationen übertragen: Receiver Report, Sender Report, Source Description Items, BYE, Application Specific Functions.

Audio Codec. Alle H.323-Terminals müssen über mindestens einen Audio-Codec verfügen. Mit dem Audio- Codec werden Audiosignale eines Mikrofons kodiert und umgekehrt eingehende kodierte Audiosignale für die Ausgabe am Lautsprecher dekodiert. Für die Kodierung und Dekodierung gibt es verschiedene ITU-Standards. Gemäß H.323 müssen Terminals den Standard G.711 unterstützen; optional können auch die Codecs die Standards G.722, G.728, G.729, MPEG 1 Audio und G.723.1 implementiert werden. Der verwendete Audio-Algorithmus, sowie die notwendigen Parameter, werden während des Austauschs der Fähigkeiten unter H.245 ermittelt.

Video Codec. Die Implementierung eines Video Codec in einem H.323-Terminal ist optional. Falls jedoch H.323-Terminals Videokommunikation unterstützen, sollte die Kodierung und Dekodierung von Video gemäß H.261 CIF implementiert sein. Optional können zusätzlich andere Modi von H.261 oder von H.263 integriert sein. Andere Video-Codecs und Bildformate können

über den H.245-Kontrollkanal ebenso wie mehrere Videokanäle zum gleichzeitigen Senden oder Empfangen vereinbart werden. Die Video-Bitrate, das Bildformat sowie Algorithmusoptionen werden während des Austauschs der technischen Fähigkeiten unter H.245 festgelegt.

T.120. Der Standard T.120 ist die Basis für die Austauschbarkeit von Daten zwischen einem H.323-Terminal und anderen H.323, H.324, H.320 oder H.310-Terminals. Es können ein oder mehrere Datenkanäle für den Austausch von Daten optional geöffnet werden. Entsprechend der jeweils verwendeten Datenanwendung kann der Datenkanal uni- oder bidirektional sein. Die Öffnung eines Datenkanals erfolgt durch das Senden einer *OpenLogicalChannel* Nachricht über den H.245-Kontrollkanal, in welcher weitere Parameter zur Spezifikation des Datenkanals mitgeteilt werden. T.120 kann die H.225.0-Schicht zum Senden und Verpacken von Datenpaketen nutzen oder mit Hilfe eigener Mechanismen Daten direkt ins Netzwerk versenden.

2.4.3 Komponenten

Terminal. Ein Terminal ist der Endpunkt einer Kommunikationsverbindung. Es muß mindestens über ein Audio Codec, eine System-Kontrolleinheit, eine H.225.0-Schicht und eine Netzwerkschnittstelle verfügen. Video Codec und Datenanwendungen können optional hinzugefügt werden.

Gatekeeper. Ein Gatekeeper ermöglicht zusätzliche Dienste für die Verbindungskontrolle zur Unterstützung von H.323-Endpunkten. Die Dienste sind über die RAS-Funktionen in H.225.0 definiert und lassen sich wie folgt einteilen: Adreßübersetzung, Zugangskontrolle, Bandbreitenkontrolle. Ferner kann ein Gatekeeper folgende Dienste optional realisieren: Verbindungskontrolle, Gesprächsautorisierung, Bandbreitenmanagement.

Gateway. Ein H.323-Gateway ist ein Endpunkt im Netzwerk, der Zwei-Wege-Kommunikation zwischen H.323-Terminals in paketbasierten Netzen und anderen ITU-Terminals in leitungsvermittelten Netzen oder zu einem anderen H.323-Gateway ermöglicht. Zu den anderen ITU-Terminals gehören z.B. H.320 (ISDN).

Multipoint Control Unit. Eine Multipoint Control Unit (MCU) bietet Unterstützung für Multipoint-Konferenzen. Sie sollte aus einem MC und entweder keinem oder mehreren Multipoint Processors (MPs) bestehen. Die klassischen Bestandteile einer MCU in einer zentralen Multipoint-Konferenz sind ein MC und jeweils ein MP für Audio, Video und Daten. In einer dezentralen Multipoint Konferenz besteht eine typische MCU aus einem MC und einem Daten MP gemäß des T.120-Standards. Die Audio- und Video-Verarbeitung erfolgt dann dezentral. Mit den entsprechenden Prozeduren können H.323-Endpunkte Verbindungen zur MCU aufbauen.
Ein Multipoint-Controller (MC) bietet Kontrollfunktionen zur Unterstützung von Konferenzen zwischen drei oder mehr Endpunkten. Er führt den Austausch der technischen Möglichkeiten mit jedem Endpunkt durch und sendet jedem Endpunkt die möglichen Übertragungsmodi innerhalb einer Konferenz. Während einer Konferenz kann der MC die Übertragungsmodi ändern, falls Ter-

minals mit bestimmten Fähigkeiten hinzukommen oder sich abmelden. Auf diese Weise bestimmt der MC den Übertragungsmodus für eine Konferenz.
Ein Multipoint-Prozessor (MP) ist Teil einer MCU und empfängt Audio-, Video- und Datenströme von Endpunkten und sendet diese nach erfolgter Verarbeitung zu den Endpunkten zurück. Zur Unterstützung von Terminals, die an einer Konferenz mit unterschiedlich ausgewählten Konferenzmodi teilnehmen, kann ein MP verschiedene Algorithmen und Formatkonvertierungen anbieten.

3. H.323-Referenzszenario

In Abbildung 8 ist ein IP-Telefonieszenario, basierend auf der H.323-Protokollfamilie dargestellt. Darin werden typische H.323-Komponenten im Zusammenwirken mit einer Firewall dargestellt.

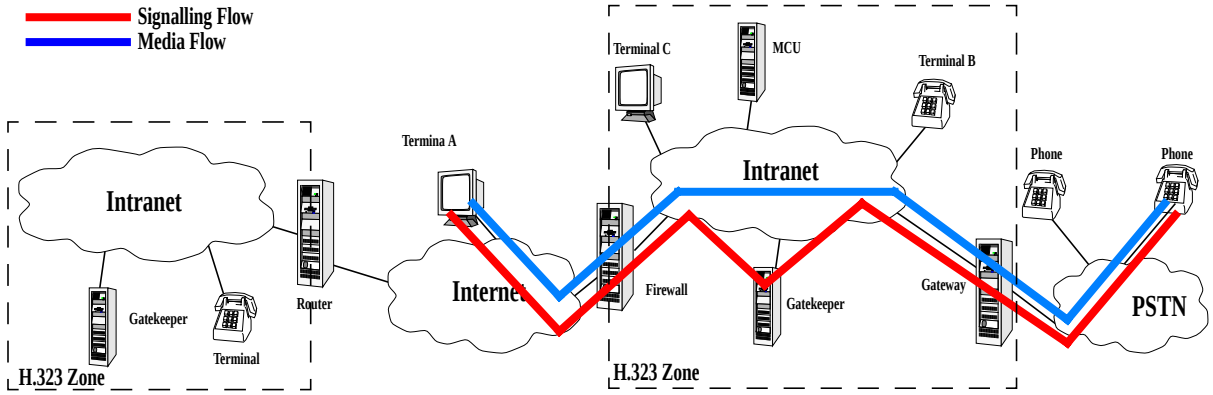

Abbildung 8: H.323-Szenario

Normalerweise umfaßt ein solches IP-Telefonie Szenario - unabhängig vom verwendeten Protokoll (H.323, SIP oder sonstige) - eine Signalisierungs- und eine Medientransportebene sowie verschiedene Telefoniekomponenten. Die Signalisierungsebene wird verwendet, um die notwendigen Signalisierungsinformationen zwischen den Komponenten zu transportieren. Nachdem ein Ruf aufgebaut wurde, wird die Medientransportebene dazu benutzt, um die Sprachdaten zwischen den Komponenten, z.B. Terminals oder Gateways, zu befördern. Oft erfolgt die Konfiguration der Komponenten über Fernzugriff, so das auch Verwaltungsdaten transportiert werden müssen. Dies könnte als zusätzliche dritte Ebene betrachtet werden, oft wird aber diese Funktion als Ergänzung zur Signalebene angesehen.

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 Geräte, wie zum Beispiel einer Multipoint Control Unit (MCU), Gateways und Terminals umfaßt. Diese Abbildung werden wir innerhalb dieser Studie als Referenzszenario verwenden.

4. Angriffe auf H.323-Systeme

Für die Nutzung eines H.323-Dienstes in einem über einen Experimentalbetrieb herausgehenden Grad ist die Sicherheit entsprechender Lösungen eine wichtige und zu hinterfragende Anforderung. Da mittlerweile von einer Reihe von Herstellern entsprechende Lösungen verfügbar werden, wurden entsprechende Überlegungen zur formalen Klassifizierung von Angriffspunkten und Verletzlichkeiten sowie praktische Experimente zu deren Ausnutzung ausgeführt und sind hier nachfolgend beschrieben. Eine ausführliche Beschreibung dieser Sachverhalte ist [20] gegeben.

4.1 Generelle Angriffsmöglichkeiten auf H.323-Systeme

Für eine Sicherheitsbewertung von H.323-Szenarien muß eine Reihe von korrespondierenden Fakten berücksichtigt werden. Zunächst sind alle verwendeten Ebenen abhängig von derselben Infrastruktur - dem IP-Netzwerk. Gefährdungen auf Signalisierungsebene, Medienebene und Verwaltungsebene können daher auf unzuverlässigen Netzwerkteilen, Komponenten oder Betreibern beruhen. Letztlich wird die Infrastruktur auch von anderen Diensten genutzt, und es kann darauf geschlossen werden, daß nicht nur Telefonie bezogene Sicherheitsprobleme auftreten können. Generell können die möglichen Gefährdungen und die davon abgeleiteten Angriffspunkte folgendermaßen klassifiziert werden:

- **Signalisierungsebene, Medientransportebene, Verwaltungsebene:** In erster Linie können die H.323-Systeme selbst auf den verschiedenen Ebenen angegriffen werden. Signalisierungsebene, Medientransportebene und Verwaltungsebene können dabei Ziel eines Angriffs sein.

- **Umgebung:** Die Umgebung, die für einen spezifischen H.323-Dienst verwendet wird, kann Ziel eines Angriffs sein. Dazu gehören beispielsweise Betriebssysteme, welche die H.323-Komponenten tragen. Diese Betriebssysteme selbst sind natürlich ebenfalls angreifbar.

- **Sicherheitssysteme:** Vorhandene Sicherheitssysteme, die nicht unmittelbar dem H.323-System zugerechnet werden, können angegriffen werden. So könnte z.B. eine von einem H.323-Gespräch durchlaufene Firewall geschwächt werden, weil sie H.323-Kommunikation unterstützt und zeitweise gewisse Kommunikationswege öffnet, die dann nicht für die normale erwünschte Konversationen, sondern für Angriffe verwendet werden können.

- **Menschen:** Da die meisten Systeme wie auch ein H.323-System von Menschen verwendet wird, ist es möglich an diesem Punkt anzugreifen.

Der nächste Abschnitt gibt eine Auswahl aus den untersuchten Verletzlichkeitsbeispielen wieder, die entsprechend der Klassifizierung eingeordnet werden können. Bei den gegebenen Beispielen handelt es sich um Angriffe, die sich in die erste Kategorie (Angriffe auf Signalisierungsebene, Medientransportebene, Verwaltungsebene) einordnen lassen. Angriffe, die sich in die restlichen

Kategorien einordnen lassen sind nicht H.323 spezifisch - solche Angriffsmöglichkeiten treten bei allen vernetzten Systemen auf - und werden deshalb in dieser Studie nicht betrachtet.

4.2 Exemplarische Angriffe auf H.323-Systeme

Im folgenden werden einige exemplarische Angriffe auf verschiedene Elemente eines H.323-Szenarios (siehe Abbildung 8) dargestellt. Die dargestellten Angriffe beziehen sich auf die für die Tests verwendeten herstellerspezifischen Komponenten. Es hat sich aber gezeigt, das eine Vielzahl der beschriebenen Angriffe oft auf die verschiedenen Implementierungen der Hersteller anwendbar sind.

DoS-Angriff auf ein Terminal über die H.323-Signalisierung. Die von uns ausgewerteten Endsysteme waren nicht in der Lage, einem Angriff, bei dem unerwartete oder inkorrekte H.323-Signalisierungs-PDUs versandt wurden, zu widerstehen. Dies hatte zur Folge, daß ein System entweder zeitweise nicht verfügbar war, oder daß das System sogar ganz ausfiel, weil das Gerät blokkierte, abstürzte oder neu startete.

DoS-Angriff auf ein Terminal über die administrative Schnittstelle. Das untersuchte IP-Telefon benutzt einen integrierten WWW Server, mit dem das Gerät verwaltet und einige seiner Einstellungen abgefragt werden können. Dieser WWW-Server und seine Implementierungsmängel (obwohl nicht grundlegend verantwortlich für die IP-Telefoniefunktionen des Gerätes) machen es für böswillige Angriffe anfällig. Wenn an den integrierten WWW-Server eine ausreichend lange URL versandt wird, kann das Gerät (abhängig von der Länge des URL-Strings) entweder außer Betrieb gesetzt werden oder es wird neu gestartet.

Übernahme eines Terminals über die administrative Schnittstelle. Die Schnittstelle für die Fernverwaltung über HTTP, die das IP-Telefon verwendet, ist für Angriffe anfällig. Das Administrator-Paßwort wird im Klartext versandt, wodurch dieses abhörbar ist. Darüber hinaus kann das Administrator-Paßwort auch anhand einer Reihe von automatisierten Brute-Force Versuchen angegriffen werden. Eine Begrenzung der Rateversuche erfolgt nicht, wegen seiner begrenzten Länge und eingeschränktem Alphabet - da das Paßwort normalerweise über die Telefontastatur eingegeben werden muß - ist dieses auch leicht zu erraten. Damit erhält der Angreifer wiederum vollen Zugriff auf alle Konfigurationsmöglichkeiten des Geräts.

Angriff auf die Vertraulichkeit der H.323 Medienströme. H.323-Anwendungen benutzen mit dem UDP-Protokoll versandte RTP-Pakete, um Audio-Datenströme zu befördern. Ein Angreifer, der Zugang zu dem verwendeten Kommunikationsweg hat, muß die Datenströme, die die Audioverbindung(en) repräsentieren, zunächst identifizieren. Danach ist ein Abhören/Aufzeichnen von Gesprächen ohne Probleme möglich.

DoS-Angriffe auf Gatekeeper über die H.323-Signalisierung. Wir konnten alle betrachteten Gatekeeper daran hindern, ihre normalen Aufgaben auszuführen, indem wir ihnen eine große Anzahl von entweder regulären (zyklischen Endgeräteanmeldungen bzw. -abmeldungen) oder irregulären H.323-PDUs zusandten. Dadurch steht der H.323-Dienst entweder nur für eine

gewisse Zeit oder überhaupt nicht mehr zur Verfügung (falls der Gatekeeper zum Absturz gezwungen wurde).

Angriff auf die Gatekeeper Anmeldung über die H.323-Signalisierung. Während der Anmeldung an einem Gatekeeper registriert ein H.323-Terminal seine IP-Adresse, seine E.164 Nummer und eine beliebige Anzahl zusätzlicher symbolischer Namen (sogenannte Aliase). Eine Abmeldung eines angemeldeten Terminals ist durch Fälschen einer RAS-Meldung von beliebiger anderer Stelle im Netzwerk möglich. Dadurch kann das Terminal keine Verbindungen mehr entgegen nehmen oder selbst absetzen. Des weiteren kann der Angreifer sich nun unter der E.164 Nummer des angegriffenen Terminals registrieren und unter dieser Identität Gespräche führen und empfangen.

4.3 Mögliche Maßnahmen

Die zu ergreifenden Maßnahmen beziehen sich auf die verschiedensten Punkte innerhalb des H.323-Szenarios. Im folgenden wird betrachtet, welche der oben beschriebenen Angriffe sich durch die Maßnahme “Firewall” am Übergang zwischen internem und externem Netz neutralisieren lassen. Prinzipiell kann eine Firewall nur dann von Nutzen sein, wenn die betrachtete Kommunikation über sie abgewickelt wird. Angriffe, bei denen beispielsweise nur interne Komponenten beteiligt sind, können durch die Maßnahme “Firewall” nicht unterbunden werden.

DoS Angriff auf ein Terminal über die H.323-Signalisierung. Es ist möglich, alle H.323-Signalisierungs-PDUs auf ihre Korrektheit zu prüfen, bevor diese von der Firewall weitergeleitet werden. Zusätzlich kann geprüft werden, ob die gesendeten PDUs in den Kontext der aktuell stattfindenden Kommunikation passen, um unerwartete PDU-Abfolgen zu verhindern. Dieses Prüfen der Signalisierungssemantik erfordert es, daß die Firewall über eine absolut korrekte und vollständige H.323-Implementierung verfügt. Dies ist prinzipiell möglich, aber nicht einfach umzusetzen. Dadurch lassen sich Angriffe dieser Art unterbinden.

DoS Angriff auf ein Terminal über die Administrative Schnittstelle, Übernahme eines Terminals über die Administrative Schnittstelle. Eine Firewall ist in der Lage bestimmte Kommunikationswege zu blockieren. Da auf die Administrative Schnittstelle eines Terminals in der Regel nicht von außerhalb zugegriffen werden muß, können alle Zugriffe auf diese Schnittstelle von außerhalb präventiv unterbunden werden. Dadurch lassen sich Angriffe auf die administrativen Schnittstellen unterbinden.

Angriff auf die Vertraulichkeit der H.323-Medienströme. Diese Art des Angriffes läßt sich durch eine Firewall nicht verhindern. Als Maßnahme kann hier die Verschlüsselung der Medienströme eingesetzt werden.

DoS Angriffe auf Gatekeeper über die H.323-Signalisierung, Angriff auf die Gatekeeper Anmeldung über die H.323-Signalisierung. Sofern es sich um Angriffe über irreguläre H.323-PDUs handelt, können diese wie schon in “DoS Angriff auf ein Terminal über die H.323-Signalisierung” beschrieben durch eine Firewall verhindert werden. Werden reguläre H.323-PDUs ver-

wendet, müssen entsprechende Maßnahmen innerhalb des Gatekeepers verwendet werden. Teilweise können solche Angriffe auch über die Integration von Intrusion Detection System-Komponenten (IDS) in die Firewall erkannt und verhindert werden [27].

5. Multimedia-Applikationen und Firewalls

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

Verhinderung der Nutzung des Multimedia-Dienstes. Es kann keine Kommunikationsverbindung über die Firewall hinweg aufgebaut werden. In diesem Fall kann der Multimedia-Dienst nicht genutzt werden, sowie sich die Kommunikationsteilnehmer auf verschiedenen Seiten der Firewall befinden. Diese funktionale Beeinträchtigung kann beispielsweise eintreten, wenn die Firewall nicht in der Lage ist alle zu einer Session gehörenden Flows zu identifizieren und bestimmte Flows nicht passieren läßt.

Beeinträchtigung des Multimedia-Dienstes. Der Multimedia-Dienst kann nur mit Einschränkungen verwendet werden. Bestimmte funktionale Eigenschaften des Multimedia-Dienstes stehen zwar zur Verfügung, nicht aber in dem Maße, wie dies benötigt wird. Beispielsweise wird ein Audio-Flow von einem Sender zum Empfänger zwar transportiert, die Firewall verzögert aber die Datenpakete in solch einem Maß, daß die Audiodaten nicht mehr sinnvoll wiedergegeben werden können.

Beeinträchtigung der Schutzfunktion der Firewall. Die Firewall muß mit Einschränkungen ihrer Schutzfunktion betrieben werden. Der Multimedia-Dienst kann dann ohne die zuvor beschriebenen Einschränkungen verwendet werden, aber es muß auf einen Teil der Schutzfunktionen, die die Firewall bereitstellt, verzichtet werden. Ist die Firewall beispielsweise nicht in der Lage, alle zu einer Session gehörenden Flows zu identifizieren, kann die Firewall statisch so eingestellt werden, daß sie alle potentiell auftretenden Flows passieren läßt. Dies ermöglicht es einem Angreifer diese dauerhaft geöffneten Kommunikationswege durch die Firewall für einen Angriff zu 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 und ihre Ursachen detailliert dargestellt.

5.1 Ursachen der Probleme

Die zuvor beschriebenen Probleme ergeben sich im wesentlichen durch zwei grundlegende Tatsachen:

1. Heute verwendete Firewalls, sowie die ihnen zugrundeliegenden Techniken wurden entwickelt um mit herkömmlichen Applikationen zusammenzuarbeiten. Multimedia-Applikationen standen beim Entwurf dieser Systeme, bzw. der zugrunde liegenden Techniken nicht im Vordergrund. Dementsprechend sind diese nicht für den Betrieb im Zusammenhang mit Multimedia-Applikationen ausgelegt oder optimiert.
2. Multimedia-Applikationen, bzw. die zugrunde liegenden Protokolle werden in der Regel entworfen, ohne zu berücksichtigen, daß diese in einem Umfeld betrieben werden müssen, in dem Firewalls verwendet werden. Dementsprechend ist es nicht verwunderlich, daß bei einem Betrieb der Multimedia-Applikationen in einer Umgebung, in welcher Firewalls verwendet werden, Probleme zu erwarten sind.

Diese beiden Tatsachen lassen demnach zwei Optimierungsmöglichkeiten zu, welche genutzt werden können, um Multimedia-Applikationen und Firewalls für einen gemeinsamen Betrieb optimal auszulegen.

1. Firewalls müssen so ausgelegt werden, daß sie mit den Eigenschaften einer Multimedia-Applikation zurecht kommen. Es existieren bewährte Firewall-Techniken, um klassische Applikationen sinnvoll zu handhaben. Entsprechend müssen Methoden gefunden werden, um mit den Charakteristika von Multimedia-Applikationen zurechtzukommen, die diese von klassischen Applikationen unterscheiden. Bei der Auslegung der Firewall ist zu beachten, daß die in Kapitel 2.2 auf Seite 5 beschriebenen Firewall-Funktionen erhalten bleiben. Die Firewall Architektur kann unter Beibehaltung der Firewall Funktionen verändert werden, damit diese optimal mit den Charakteristika von Multimedia-Applikationen zurecht kommt.
2. Die Multimedia-Applikationen sowie die zugrunde liegenden Multimedia-Protokolle müssen dahingehend verändert werden, daß diese mit Firewalls zurecht kommen. Diese Forderung kann als schwierig erfüllbar angesehen werden, da es in der Praxis kaum möglich ist bestehende Applikationen oder Protokollstandards zu verändern.

Beide Optimierungsmöglichkeiten können nicht getrennt und/oder als Ersatz für die jeweils andere Methode gesehen werden. Für eine optimale Gesamtlösung müssen beide Optimierungsmöglichkeiten genutzt werden. Bei der Optimierung muß aber immer die Randbedingung *Sicherheit* beachtet werden.

5.2 Charakteristika von Multimedia-Applikationen

Multimedia-Applikationen unterscheiden sich hinsichtlich vieler Eigenschaften signifikant von “traditionellen Applikationen”. Die im folgenden aufgeführten Charakteristika, welche im Zusammenhang mit Firewalls wesentlich für die bestehenden Probleme verantwortlich sind, können in drei Gruppen unterteilt werden. Die innerhalb der Gruppen zusammengefaßten Charakteristika besitzen eine starke Abhängigkeit dahingehend, daß für sie gemeinsam Lösungen gefunden werden müssen.

Protokoll. Die von Multimedia-Applikationen verwendeten Multimedia-Protokolle besitzen in der Regel eine deutlich höhere Komplexität als “traditionelle Applikationen”. Dies ist im wesentlichen dadurch begründet, das für die Behandlung der verschiedenen Flows einer Session zum einen verschiedene Teilprotokolle verwendet werden und zum anderen die Verwaltung der verschiedenen Flows eine sehr komplexe Protokollmaschine benötigt.
Des weiteren besteht eine Multimedia-Session aus mehreren einzelnen Flows, die für verschiedene Aufgaben verwendet werden. Dies unterscheidet eine Multimedia-Session von einer Session einer “traditionellen Applikationen”, die in der Regel nur einen Flow pro Session verwendet.
Außerdem besitzt ein Multimedia-Protokoll ein sehr dynamisches Verhalten. Viele der verwendeten Flows benutzen dynamisch zwischen Sender und Empfänger ausgehandelte Ports. Auch die verwendete Bandbreite einzelner Flows kann sich während der Dauer einer Session dynamisch ändern. Ebenso kann die Anzahl der verwendeten Flows während der Dauer einer Session geändert werden.
Zusammenfassend können folgende Charakteristika der Protokolle, die zu Problemen in Firewalls führen, genannt werden:

- **Komplexität der Protokolle**
- **Mehrere Flows in einer Session**
- **Dynamisches Verhalten**

Applikation. Multimedia-Applikationen verwenden in der Regel eine im Vergleich zu traditionellen Applikationen komplexe Infrastruktur. Das Design bzw. der Umfang der für die Applikation verwendeten Infrastruktur hat dabei starken Einfluß auf das Routing der Signalisierungs- bzw. Medien-Flows. Eine Firewall, die einen Teil der Infrastruktur darstellt, muß um ihre Aufgaben erfüllen zu können zum einen in dieses Routing entsprechend einbezogen werden. Zum anderen muß die Firewall mit den verschiedenen Infrastrukturszenarien zurecht kommen.
Verschiedene Multimedia-Applikationen verwenden die Multimedia-Protokolle meist auf verschiedene Arten. Dies liegt daran, daß die sehr umfangreichen Standards, die die Multimedia-Protokolle beschreiben an manchen Stellen Interpretationsmöglichkeiten zulassen. Dadurch entstehen Multimedia-Applikationen, die dasselbe Multimedia-Protokoll verwenden, aber ein sehr unterschiedliches Kommunikationsverhalten aufweisen.
Außerdem können Multimedia-Applikationen zusätzliche Protokolle verwenden um bestimmte Aufgaben zu erfüllen, die nicht durch das verwendete Multimedia-Protokoll abgedeckt sind. Beispielsweise können bestimmte Informationen, die zur Initiierung einer Multimedia-Session nötig sind, in einer Datenbank abgelegt werden. Die Abfrage der Datenbank geschieht über ein Protokoll, das nichts mit dem Multimedia-Protokoll zu tun hat. Dennoch gehört die Abfrage der Datenbank zu der Multimedia-Session.
Zusammenfassend können folgende Charakteristika der Applikationen die zu Problemen in Firewalls führen genannt werden:

- **Signalisierungs- und Medien-Routing**

- **Szenarienvielfalt und Szenarienkomplexität**
- **Interpretationsmöglichkeiten der Protokollstandards**
- **Verwendung zusätzlicher Protokolle**

Performance-Anforderungen. Multimedia-Applikationen verwenden Flows, die eine sehr hohe Bandbreite - teilweise über einen sehr langen Zeitraum - benötigen. Insbesondere Videoübertragungen benötigen eine hohe Bandbreite.
Zusätzlich unterliegt die Übertragung der kontinuierlichen Medien bestimmten Dienstgüteanforderungen. Diese Dienstgüteanforderungen müssen eingehalten werden, damit die transportierten Inhalte verwendet werden können.
Zusammenfassend können folgende Charakteristika der Performance Anforderungen die zu Problemen in Firewalls führen genannt werden:

- **hohe Datenrate**
- **Dienstgüteanforderungen**

Diese Charakteristika, müssen bei der Optimierung einer Firewall für die Unterstützung von Multimedia-Protokollen berücksichtigt werden.

5.3 Anforderungen an eine Multimedia-Firewall

Um Multimedia-Applikationen in Firewall Umgebungen sinnvoll betreiben zu können, ist es notwendig, die zuvor beschriebenen Charakteristika zu beachten. Aus diesen Charakteristika können dann folgende generelle Design-Pattern abgeleitet werden:

- Aufgrund der hohen Komplexität der Multimedia-Protokolle ist es nicht sinnvoll einen integrierten Firewall Ansatz zu verwenden [28]. Eine monolithische Architektur verhindert es, durch Schichtung bzw. Verteilung die Komplexität in einen durchschaubaren und beherrschbaren Rahmen zu bringen. Ein Hybridsystem ist aus diesem Grund nicht geeignet.

- Eine Firewall muß in der Lage sein, Zustände über die Session-Zugehörigkeit der verwendeten Flows zu halten. Ein Paketfilter beispielsweise ist aus diesem Grund ungeeignet.

- Statische Elemente innerhalb einer Firewall können nicht verwendet werden, da sie nicht den dynamischen Charakteristika der Multimedia-Protokolle entsprechen. Beispielsweise ist der klassische zustandslose Paketfilter aus diesem Grund nicht geeignet als Firewall in einer Umgebung mit Multimedia-Applikationen eingesetzt zu werden.

- Damit eine Firewall in die Infrastruktur einbezogen werden kann, muß diese in der Lage sein die jeweiligen Routing Mechanismen der entsprechenden Multimedia-Applikation zu unterstützen. Da diese Mechanismen von den Applikationstypen und der verwendeten Infrastruktur abhängen, ist innerhalb der Firewall ein flexibles und anpaßbares Routing nötig. Diese Betrachtung führt letztlich wieder zu der Aussage,

einen nicht integrierten Firewall Ansatz zu verwenden, der es ermöglicht die applikationsspezifischen Elemente flexibel und anpaßbar zu gestalten.

- Durch die verschiedenen Interpretationsmöglichkeiten der Protokolle durch die Applikationen ist es nötig eine Firewall flexibel und anpaßbar zu gestalten. Dies spricht wiederum gegen einen integrierten Ansatz.

- Durch die mögliche Verwendung zusätzlicher Protokolle ist es nötig, daß die Firewall Zustände über die Zugehörigkeit verschiedener Flows zu einer Applikation halten kann. Ein Paketfilter beispielsweise ist aus diesem Grund ungeeignet.

- Medienströme sowie Signalisierungsströme können hinsichtlich der notwendigen Sicherheitsüberprüfungen unterschiedlich durch eine Firewall behandelt werden. Dies ermöglicht es, die Medien- und Signalisierungsströme auch auf verschiedenen Wegen durch eine Firewall zu leiten. Es ist dann möglich, diese verschiedenen Pfade für unterschiedliche Ziele - z.B. Performance oder Sicherheit - zu optimieren. Ein Hybridsystem oder eine verteilte Firewall sind aus diesem Grund sinnvoll, da hier eine unterschiedliche Wegewahl für die verschiedenen Ströme möglich ist.

Eine Firewall, die sich an diese Pattern hält, kann mit Multimedia-Applikationen prinzipiell umgehen. Viele der heute eingesetzten Firewalls entsprechen dem Typ “Hybridsystem”. Dieser Typ Firewall ist prinzipiell nicht geeignet Multimedia-Applikationen zu unterstützen, wodurch sich die bei zur Zeit eingesetzten Firewalls auftretenden Probleme erklären lassen.

6. H.323-Applikationen und Firewalls

Wie in Kapitel 5.2 auf Seite 23 beschrieben, sind bestimmte Charakteristika von Multimedia-Applikationen dafür verantwortlich, daß deren Verwendung problematisch in einer Umgebung, in welcher auch Firewalls eingesetzt werden, ist. Diese Charakteristika bzw. die daraus resultierenden Probleme sind auch in H.323-Szenarien zu finden. Im Folgenden werden diese Probleme genauer erläutert.

Um nachfolgend die durch die einzelnen Charakteristika hervorgerufenen Probleme verständlich erläutern zu können, ist es nötig, die H.323-Kommunikation genauer zu beschreiben. Dies kann nicht in vollem Umfang innerhalb dieser Studie geschehen (Siehe [1] für eine genaue Beschreibung aller möglichen Fälle). Wir beschränken uns auf die im Folgenden gegebene vereinfachte Darstellung, die zur Erläuterung der folgenden Abschnitte dient. Wenn nur zwei Terminals an der Kommunikation beteiligt sind (direkter Ruf zwischen Terminal A und Terminal B), wird der in Abbildung 9 dargestellte Kommunikationsablauf verwendet.

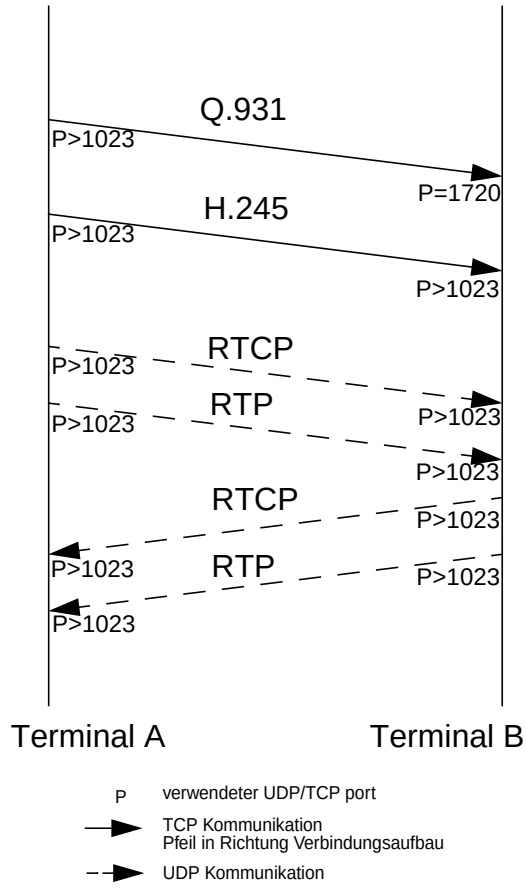

Abbildung 9: H.323 direkter Ruf

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 derselben 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.

•Q.931 Call Signalling (TCP):

Eine TCP Verbindung wird zwischen Terminal A und Terminal B aufgebaut. Diese wird zum Transport der *Call Setu*p Nachrichten, wie sie in H.225.0 [21] 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 [22] Verbindung wird verwendet, um *Call Control* Nachrichten 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 IP-Adressen und 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 Videoübertragung stattfindet.

6.1 Protokollcharakteristika

6.1.1 Komplexität der Protokolle

Wie in Kapitel 2.4 beschrieben, stellt das H.323-Protokoll eine Protokollfamilie dar. Innerhalb des H.323-Standards sind für einzelne Teilbereiche weitere Protokolle definiert bzw. angegeben (z.B. H.225, H.245, RTP/RTCP, usw.). Dementsprechend verwendet eine H.323-Komponente einen Teil - oder sogar alle - der innerhalb des H.323-Standards beschriebenen Protokolle um ihre vorgesehene Aufgabe innerhalb eines H.323-Szenarios zu erfüllen. Daraus ergibt sich, daß eine H.323-Komponente zur Erfüllung ihrer Aufgabe verschiedene Protokollautomaten besitzt. Die einzelnen, innerhalb einer Komponente verwendeten Protokollautomaten sind zusätzlich untereinander verbunden. Die Zustände der einzelnen Protokollautomaten hängen jeweils vom Zustand der anderen Protokollautomaten ab. Beispielsweise ist der Zustand des H.245-Protokollautomaten vom Zustand des RTP/RTCP-Protokollautomaten abhängig und umgekehrt. Eine Firewall, die innerhalb eines H.323-Szenarios eingesetzt wird, muß in der Lage sein, solche komplexen und mehrdimensionalen Zustandsautomaten zu implementieren und zu verwalten. Dies ist beispielsweise notwendig, um alle zu einer Session gehörenden Flows zu identifizieren. Die hohe Komplexität des H.323-Protokolls macht innerhalb der Firewall ein ebenso komplexes Element zur Verarbeitung des H.323-Protokolls notwendig.
Bestehende Firewalls sind in der Regel nicht dafür ausgelegt solch komplexe Zustandsautomaten zu implementieren. Es kann beispielsweise vorkommen, das die notwendigen Zustandsautomaten nicht mehr in ein bestehendes System integriert werden können, da die verfügbaren Ressourcen (Speicherplatz, Prozessorleistung,...) für die Abbildung dieser komplexen Zustandsautomaten nicht ausreichen.
Ein weiteres Problem stellt die Verwendung einer ASN.1 Kodierung für die Nachrichten verschiedener Teilprotokolle (H.225, H.245) dar. Durch die Verwendung von ASN.1 werden die zu übermittelnden Informationen sehr dicht in die Nachrichten gepackt, wodurch die vorhandenen Übertragungskapazitäten sehr effizient genutzt werden können. Allerdings müssen die Nachrichten kodiert und dekodiert werden. Dazu muß eine H.323-Element innerhalb der Firewall in der Lage sein, die ASN.1 Kodierung/Dekodierung vornehmen zu können. Die dazu notwendigen Funktionen und Mechanismen erhöhen zusätzlich die Komplexität des Elements zur H.323-Verarbeitung innerhalb der Firewall.

6.1.2 Mehrere Flows in einer Session

Ein wesentliches Merkmal von Multimedia-Protokollen ist die Verwendung mehrerer Flows, die zusammen eine logische Session bilden. Dieses Merkmal ist innerhalb von H.323 besonders stark ausgeprägt. Im Unterschied zu anderen Multimedia-Protokollen (siehe Kapitel 2.3) besteht eine H.323-Session aus mehreren Signalisierungs- und Medien-Flows. Für eine Firewall stellt sich nun das Problem, die verschiedenen Flows, welche zusammen eine Session formen, identifizieren zu können. Diese Identifizierung ist aus zwei Gründen notwendig. Erstens muß eine Firewall in der Lage sein zu identifizieren, welche Kommunikationspartner innerhalb einer Session Daten austauschen. Diese Information stellt die Grundlage der sicherheitsrelevanten Entscheidung (z.B. ob die entsprechenden Daten weitergeleitet werden) innerhalb einer Firewall dar. Zweitens muß die Firewall in der Lage sein, den (zeitlichen) Zusammenhang der auftretenden Flows innerhalb einer Session zu prüfen, um zu verhindern daß ein Flow von einem Angreifer in einem ungültigen Kontext verwendet wird.
Die Erkennung dieser Zusammenhänge stellt an eine Firewall besondere Anforderungen. Es ist erforderlich, daß die Firewall fähig ist die Teile des zwischen den Kommunikationspartnern ablaufenden H.323-Protokolls zu interpretieren, welche Aufschluß über die aktuell verwendeten Flows einer Session geben. Dies betrifft innerhalb des H.323-Protokolls das H.225/Q.931 Protokoll und das H.245 Protokoll. Mit Hilfe des Q.931-Protokolls werden Informationen zwischen den Kommunikationspartnern ausgetauscht, die dazu verwendet werden den H.245-Signalisierungs-Flow zu etablieren. Dies geschieht über die Q.931 Nachricht *Setup*. Es besteht die Möglichkeit, auf den zweiten Signalisierungs Flow zu verzichten und die H.245-Nachrichten über den gleichen Flow zu transportieren, über den auch die Q.931 Nachrichten ausgetauscht werden (bezeichnet als *H.245Tunneling*). Damit eine Firewall diesen Unterschied zwischen diesen beiden Operationsmodi erkennen kann muß innerhalb der Q.931-Nachrichten das Feld *H.245Tunneling* überprüft werden. Innerhalb der dann folgenden H.245-Nachrichten werden zwischen den Kommunikationspartnern die notwendigen Flows für die Medienströme vereinbart. Je nach Beschaffenheit der Kommunikationspartner kann es sich um Video, Audio und Daten flows handeln. Die entsprechenden Flows werden durch den Austausch von *OpenLogicalChannel*, *OpenLogicalChannelAck* bzw. *OpenLogicalChannelReject* H.245-Nachrichten festgelegt. Während einer H.323-Session können verschiedene Flows neu zu einer Session hinzukommen, bzw. bestehende abgebaut werden. Dies geschieht durch denn Austausch von entsprechenden *OpenLogicalChannel, OpenLogicalChannelAck* bzw. *OpenLogicalChannelReject* H.245 Nachrichten. Bei der Beendigung der H.323-Session werden zuerst die Medien-Flows geschlossen. Dies geschieht durch den Austausch von *CloseLogicalChannel* H.245 Nachrichten. Die jeweiligen Nachrichten werden durch entsprechende *CloseLogicalChannelAck* bestätigt. Danach wird der H.245-Flow, falls nicht *H.245Tunneling* verwendet wird - geschlossen. Als Letztes wird dann der Q.931-Flow terminiert, dazu wird die *ReleaseComplete* Q.931-Nachricht ausgetauscht und anschließend der Kanal geschlossen. Es ist ebenfalls möglich, daß Medien-Flows schon innerhalb des initialen Austausches von Q.931-Nachrichten vereinbart werden (bezeichnet als *Faststart*). Dieser Operationsmodi muß, wenn verwendet, von einer Firewall ebenfalls berücksichtigt werden.

Die genaue Abfolge der einzelnen Nachrichten, sowie die möglichen Varianten (*H245Tunneling*, *Faststart*), die für einen H.323-Session-Aufbau verwendet werden, sind genau in [1] dargestellt. Eine H.323-Firewall muß, um die verschiedenen Flows einer Session zuordnen zu können, mindestens diese Nachrichten interpretieren können. Darüber hinaus muß die Firewall sich entsprechende Zustände über den Kontext der Nachrichten behalten. Dies bedeutet letztendlich, daß die Firewall für das Q.931- sowie H.245-Protokoll einen Protokollautomaten implementieren muß, der zumindest auf den oben beschriebenen Nachrichten basieren muß. Eine H.323-Firewall benötigt dementsprechend zumindest einen reduzierten H.323-Protokoll-Stack. Tabelle 1 faßt zusammen, welche Nachrichten im wesentlichen zur Identifizierung der einzelnen Flows von einer Firewall interpretiert werden müssen.

Tabelle 1: H.323-Nachrichten zur Beschreibung der verwendeten Flows

Protokoll	Nachrichten	Beschreibung
Q.931	*Setup*	Vereinbarung des Call Control Flows (H.245)
Q.931	*ReleaseComplete*	Aufheben des Call Signalling Flows, sowie der davon abhängigen Flows Call Control, Media und Media Control (Q.931, H.245, RTP/RTCP))
Q.931	*(H.245Tunneling)* [a]	Tunneln der Call Control-Nachrichten über den Call Signalling Flow.
Q.931	*(Faststart)*[b]	Vereinbarung der Media Control Flows innerhalb von Call Signalling-Nachrichten
H.245	*OpenLogicalChannel*, *OpenLogicalChannelAck*, *OpenLogicalChannelReject*	Vereinbarung der Media und Media Control Flows (RTP/RTCP)
H.245	*CloseLogicalChannel*, *CloseLogicalChannelAck*	Aufheben der Media und Media Control Flows (RTP/RTCP)

a. Diese Option wird in verschiedenen Nachrichten übermittelt
b. Diese Parameter werden innerhalb verschiedener Q.931 Nachrichten übermittelt

Es existieren weitere Nachrichten die ebenfalls zur Steuerung der einzelnen Flows verwendet werden können (z.B. *requestChannelClose*). Alle Nachrichten und Operationsmodi sind genau in [1] beschrieben.
Die hier beschriebenen Nachrichten enthalten neben der Information, daß ein bestimmter Flow verwendet werden soll, ebenfalls Informationen über die Charakteristika der entsprechenden Flows. Diese dynamisch festgelegten Charakteristika werden im nächsten Abschnitt erläutert.

6.1.3 Dynamisches Verhalten

Multimedia-Protokolle besitzen viele dynamische Anteile; dies gilt insbesondere für das H.323-Protokoll. Wie bereits erwähnt werden die Charakteristika der verwendeten Flows größtenteils

dynamisch festgelegt. Die dynamischen Festlegungen schließen das verwendete Protokoll, die verwendeten Quell- und Ziel-Ports sowie die Quell- und Zieladressen ein. Darüber hinaus wird für die Medienströme dynamisch festgelegt, welche Bandbreite diese verwenden. Weitere Dynamik entsteht durch die Tatsache, daß die Anzahl der Flows sich während der Dauer einer Session ändern kann. Die wesentlichen Nachrichten zur Übermittlung der dynamischen Parameter sind ebenfalls in Tabelle 1 zusammengefaßt.
Diese Dynamik erschwert es der Firewall zusätzlich, die Zuordnung der Flows zu ihren entsprechenden Sessions vorzunehmen.

6.2 Applikationscharakteristika

6.2.1 Szenarienvielfalt und Szenarienkomplexität

Die verwendeten Kommunikationsmechanismen innerhalb des Referenzszenarios (Abbildung 8) 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 9 dargestellte Kommunikationsablauf verwendet.
Wenn innerhalb dieses Szenarios ein Gatekeeper verwendet wird (Rufvermittlung durch den Gatekeeper), ändern sich die Kommunikationsbeziehungen und Kommunikationsabläufe. In dem hier dargestellten Fall (Abbildung 10) laufen die Signalisierungsnachrichten über den Gatekeeper (*gatekeeper routed call* [24]).

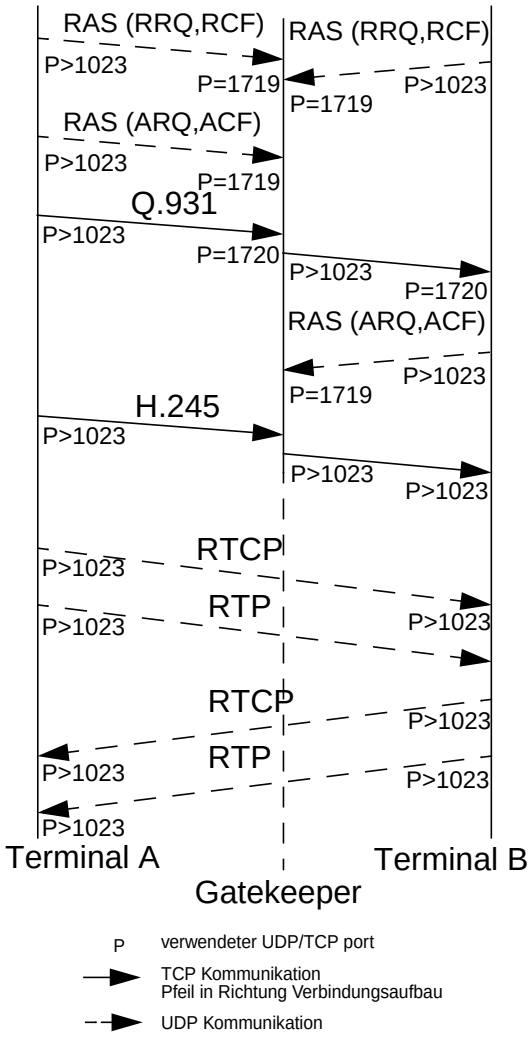

Abbildung 10: H.323 Gatekeeper vermittelter Ruf

Der H.323-Standard sieht auch eine Möglichkeit vor, den Gatekeeper ohne direkte Einbeziehung in die Signalisierung zu verwenden (*direct call model* [24]). Der *gatekeeper routed call* wird jedoch in den meisten Gatekeeper-Szenarien verwendet.

•RAS Registration, Admission, Status (UDP):
Nach dem Start der Terminals registrieren sich diese am Gatekeeper (*RegistrationRequest RRQ* und *RegistrationConfirm RCF*). Dazu wird das auf UDP basierte RAS - Protokoll [21] 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 (*AdmissionRequest ARC*). Wenn die Erlaubnis erteilt wird (*AdmissionConfirm 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 kon-

taktiert dieses. Das Zielterminal führt daraufhin ebenfalls eine Erlaubnisanfrage durch, ob es den Ruf annehmen darf. 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 (innerhalb der Call Signalling-Nachricht *Setup*).

- **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 werden dazu verwendet, die Parameter der folgenden Medienströme auszuhandeln.

- **RTP/RTCP Media und Media Control (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.245Tunneling* oder *Faststart* verwendet werden, bzw. weitere H.323-Komponenten in die Kommunikation mit einbezogen sind. Es zeigt sich, 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.

6.2.2 Call Routing

Damit eine Firewall ihre Aufgaben wahrnehmen kann, muß sie (teilweise) in die ablaufende Kommunikation involviert sein. Wie bereits in den vorhergehenden Abschnitten gezeigt, muß die Firewall zumindest in die Signalisierung der H.323-Kommunikation eingreifen können. Es ergibt sich dabei die Fragestellung, wie eine Firewall in den Signalisierungsweg eingebracht werden kann. Diese Frage ist nicht einfach zu beantworten. Zum einen ist das Einbringen einer Firewall in den Signalisierungsweg einer H.323-Kommunikation innerhalb des H.323-Standards nicht vorgesehen. Zum anderen ist - wie im Folgenden dargestellt - es nicht möglich eine Methode anzugeben, die alle möglichen Szenarien abdecken kann. Es ergibt sich das Problem, daß das Call Routing derart beeinflußt werden muß, daß die Firewall in den Signalisierungsweg eingebunden wird. Es sind die nachfolgend beschriebenen Methoden denkbar, eine Firewall in den Kommunikationsweg einzubringen. Die hier aufgezählten Methoden sind nicht vollständig, umfassen aber die gängigsten zur Zeit angedachten Variationen.

Explizit - Modifikation von H.323-Komponenten. Die *Explizite* Variante zeichnet sich dadurch aus, daß eine oder mehrere H.323-Komponenten (beispielsweise ein Terminal) so verändert werden, daß die Firewall in den Kommunikationsweg eingebracht wird. Dies geschieht dann nicht transparent, sondern explizit. Ein solches Verfahren ist oft auch nicht transparent für den Benutzer, der das H.323-System verwendet.

Wenn beispielsweise ein Terminal A eine Verbindung zu einem Terminal B (direkter Ruf, ohne Verwendung eines Gatekeepers, siehe Abbildung 11) über eine dazwischenliegende Firewall aufbauen will, so ist dies nicht direkt möglich. Terminal A muß sich dazu zuerst mit der Firewall verbinden und dieser explizit mitteilen, welches das Endziel des Rufes ist.

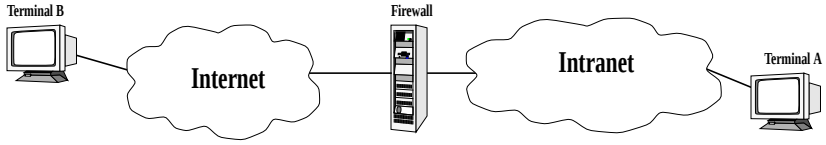

Abbildung 11: Explizite Signalisierung (I)

Die Firewall muß nachfolgend Terminal B kontaktieren und die Kontroll- und Audio-Flows zwischen beiden Terminals vermitteln. Es ergibt sich der folgende Kommunikationsablauf (siehe auch [25]):

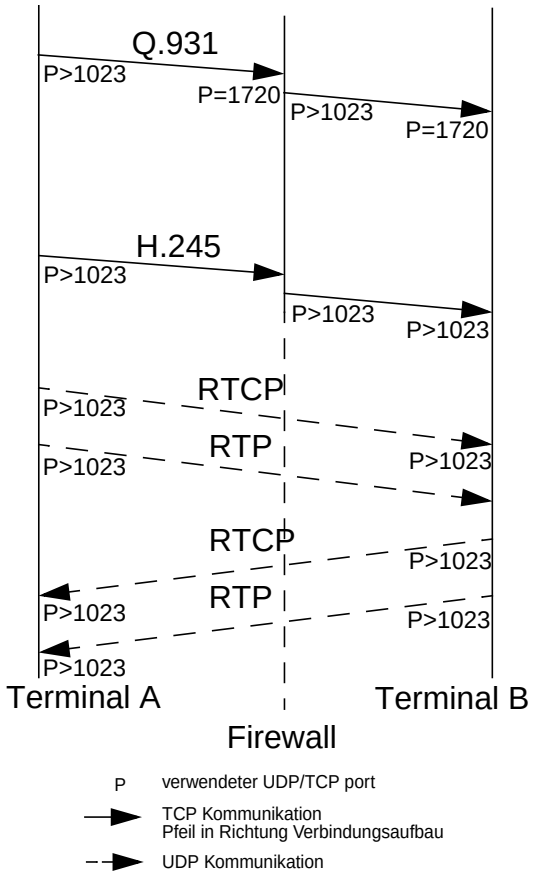

Abbildung 12: Explizite Signalisierung (II)

•Voraussetzung:
Das externe Terminal A 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.

•Q.931 Call Signalling (TCP):
Bei einem Verbindungsaufbau kontaktiert das Terminal A zunächst die Firewall. In diesem Fall wird die Q.931-Verbindung von Terminal A zur Firewall aufgebaut. Innerhalb der Q.931-Verbindung wird dann als erstes eine Q.931-*Setup* Nachricht von Terminal A an die Firewall übermittelt. Innerhalb dieser Nachricht ist angegeben, welches die eigentliche Zieladresse des Rufes ist. Die Zieladresse kann dabei in verschiedenen Formaten angegeben sein, beispielsweise als E.164-Nummer, als IP-Adresse, als H323-ID oder in einer anderen Form. Die Firewall muß nun in der Lage sein, diese Zieladresse in eine IP-Adresse umzusetzen. Danach kann die Firewall das Terminal B kontaktieren und die jeweils auftretenden Q.931-Nachrichten zum jeweiligen Terminal weiterleiten.

- **H.245 Call Control (TCP), RTP/RTCP Media und Mediacontrol (UDP):**
  Analog zu den in Kapitel 6.2.1 beschriebenen Mechanismen kann die Firewall sich

nun in die gesamte Kommunikation einschalten, indem die übermittelten Flow Spezifikationen in den Signalisierungsnachrichten modifiziert werden.

Diese Methode ist nicht transparent, das Terminal muß verändert werden. Rufe die über die Firewall vermittelt werden, werden explizit anders behandelt als direkte Rufe zwischen zwei Terminals.
Ein Beispiel, welches diese Form des Routing über eine Firewall unterstützt, ist die Microsoft Netmeeting Terminalsoftware. Diese Terminalsoftware besitzt die Möglichkeit eine Firewall (dort als Gateway bezeichnet) für die ausgehenden Rufe anzugeben.

Transparent - Firewall Redirect . Bei dieser Variante wird ebenfalls das in Abbildung 11 dargestellte Szenario zugrunde gelegt. Allerdings wird in diesem Fall keine H.323-Komponente verändert. Lediglich die Firewall besitzt eine zusätzliche Funktion, die es ihr ermöglicht bestimmte Flows umzuleiten. Es ergibt sich der folgende Kommunikationsablauf:

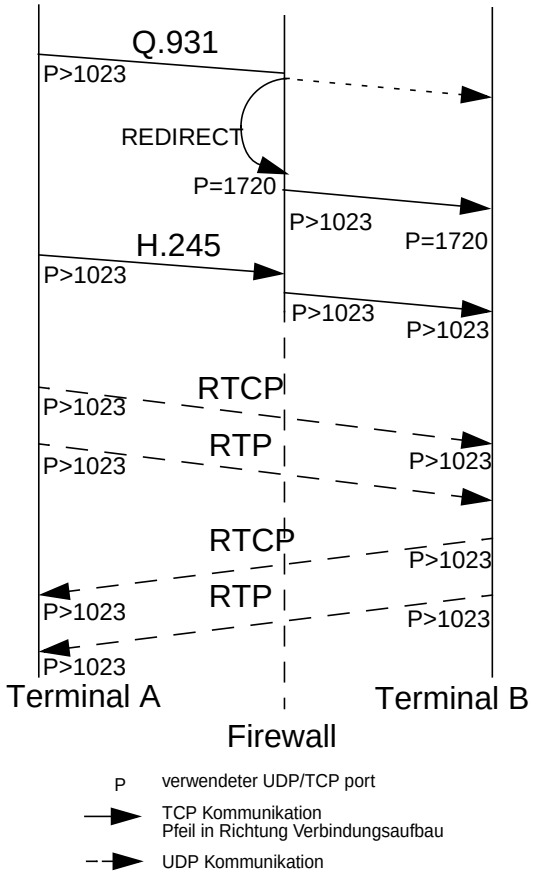

Abbildung 13: Firewall Redirect

•Voraussetzung:
Die Firewall ist in der Lage, bestimmte Flows - identifizierbar an Protokoll, Ziel-Port, Ziel-IP-Adresse, Quell-Port und Quell IP-Adresse - umzuleiten. Dies geschieht indem die Firewall die IP-Adressen und Port-Nummern der einzelnen Pakete eines Flows modifiziert und sich den Zustand der umgeleiteten Flows in einer Tabelle (genannt *Redirect-Table*) merkt.

•Q.931 Call Signalling (TCP):
Bei einem Verbindungsaufbau kontaktiert das Terminal A direkt Terminal B. Die Firewall verwendet ihre Redirect-Fähigkeit, um den Q.931-Flow transparent umzuleiten. Dabei wird der Q.931-Flow an die Firewall selbst umgelenkt, wodurch diese in die Signalisierung eingebunden wird. Um nun das eigentliche Ziel (Terminal B) zu bestimmen und dann zu kontaktieren steht der Firewall zwei Möglichkeiten zur Verfügung. Erstens kann, wie im vorhergehend beschriebenen Verfahren ebenfalls durchgeführt, die Q.931-*Setup* Nachricht analysiert werden, die Informationen über das Ziel enthält. Zweitens kann die *Redirect-Table* der Firewall verwendet werden um das Ziel zu bestimmen, da dort die Zuordnung zwischen Quelle (Terminal A) und eigentlichem Ziel (Terminal B) festgehalten ist. Nach Bestimmung des Ziels kann die Firewall das Terminal B kontaktieren und die jeweils auftretenden Q.931-Nachrichten zum jeweiligen Terminal weiterleiten.

- **H.245 Call Control (TCP), RTP/RTCP Media und Mediacontrol (UDP):**
  Analog zu den in Kapitel 6.2.1 beschriebenen Mechanismen kann die Firewall sich nun in die gesamte Kommunikation einschalten, indem die übermittelten Flow Spezifikationen in den Signalisierungsnachrichten modifiziert werden.

Diese Methode ist transparent, das Terminal A adressiert das Terminal B in gewohnter Weise. Die Umlenkung des initialen Q.931-TCP-Flows ist für das Terminal A nicht erkennbar und damit transparent. Allerdings kann diese Methode nur dann verwendet werden, wenn Terminal B durch Terminal A direkt adressiert werden kann. Aus diesem Grund kann dieses Verfahren nicht in NAT Umgebungen für eingehende Rufe verwendet werden.
Ein Beispiel, welches diese Form des Routing über eine Firewall unterstützt, ist die Firewall Software IP-Filter [29]. Diese Firewall Software besitzt die Möglichkeit spezifische Flows über die Angabe der Flow Spezifizierungen umzuleiten.

Transparent - Nutzung von Infrastrukturkomponenten. Um die Firewall in die Kommunikation einzubinden, können auch H.323-Infrastruktur Komponenten in die Firewall integriert werden. Es ist beispielsweise möglich, einen Gatekeeper in die Firewall zu integrieren. Wird dann *ein Gatekeeper Routed Call Model* verwendet (siehe Kapitel 6.2.1), so ist die Firewall transparent in die Kommunikation mit eingebunden. Ein solches Szenario ist in Abbildung 14 gezeigt.

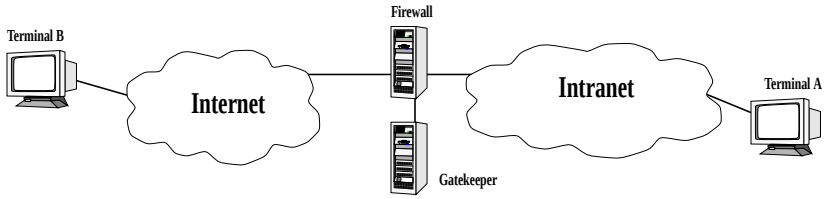

Abbildung 14: Nutzung von Infrastrukturkomponenten

Es ergibt sich ein Kommunikationsablauf (siehe auch [26]) der exakt dem in Kapitel 6.2.1 beschriebenen Ablauf gleicht.

- **Voraussetzung:**
  Der Gatekeeper muß eine Interaktionsmöglichkeit mit der Firewall besitzen, um dieser die zur Erfüllung ihrer Aufgaben notwendigen Informationen zukommen lassen zu können.

Dieses Verfahren ermöglicht es die Firewall auf eine Art in die H.323-Kommunikation einzubringen, die sich sehr gut mit den H.323-Konzepten vereinbaren läßt. Innerhalb des H.323-Standards ist ein Gatekeeper als zentrales Element vorgesehen, eine Firewall aber nicht. Der Nachteil dieses Verfahrens besteht darin, daß ein vollständiger Gatekeeper in eine Firewall zu integrieren ist. Dies ist oft bei bestehenden Firewallsystemen nicht möglich.
Ein Beispiel, welches diese Form des Routing über eine Firewall unterstützt, ist der Cisco MCM. Dieses Firewall System besteht aus einer Verbindung eines Gatekeepers mit einer Firewall.

Transparent - Nutzung der Gatekeeper-Gatekeeper Kommunikation . Betrachtet man ein Szenario, indem eine Kommunikation über verschiedene H.323-Zonen läuft, ergeben sich spezi-

elle Möglichkeiten eine Firewall in den Kommunikationsweg einzubringen. In Abbildung 15 ist ein solches Szenario dargestellt.

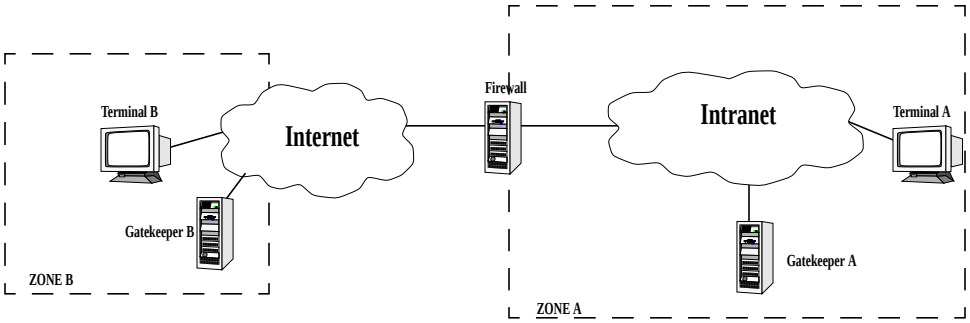

Abbildung 15: Gatekeeper-Gatekeeper Kommunikation (I)

Terminal A ist dabei an Gatekeeper A angemeldet, Terminal B an Gatekeeper B. In dem dargestellten Beispiel wird ein Ruf, der von Terminal A nach Terminal B durchgeführt wird, durch beide dargestellten H.323-Zonen laufen. Die jeweiligen Zonengrenzen entsprechen den Bereichen, die durch die jeweilige Firewall - wenn verwendet - geschützt werden. Dies entspricht in vielen Fällen der Realität, da die H.323-Zone sowie der durch die Firewall zu schützende Bereich von derselben administrativen Einheit betrieben wird. Es ergibt sich der folgende Kommunikationsablauf:

- **Voraussetzung:**
  Rufendes und gerufenes Terminal verwenden jeweils einen eigenen Gatekeeper. Die verwendeten Gatekeeper sind in der Lage über sog. RAS *LocationRequests* (LRQ) mit anderen Gatekeepern zu kommunizieren. Entsprechend benötigen die Gatekeeper eine Konfigurationsmöglichkeit, die ihnen angibt, welche *LRQ* an welchen externen Gatekeeper weitergeleitet werden müssen. Dies wird innerhalb der Gatekeeper in sog. *Neighbour Tables* festgelegt.

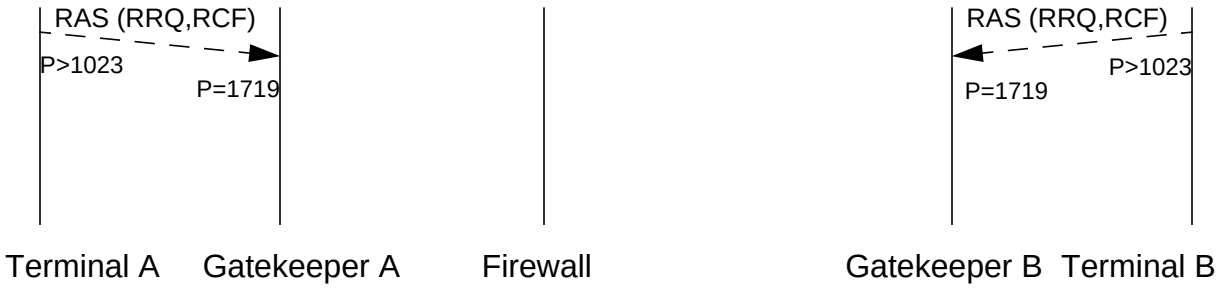

Abbildung 16: Gatekeeper-Gatekeeper Kommunikation (II.a)

- **RAS Registration, Admission, Status (UDP):**
  Nach dem Start der Terminals registrieren sich diese jeweils an dem ihnen zugewiesenen Gatekeeper (*RegistrationRequest RRQ* und *RegistrationConfirm RCF*).

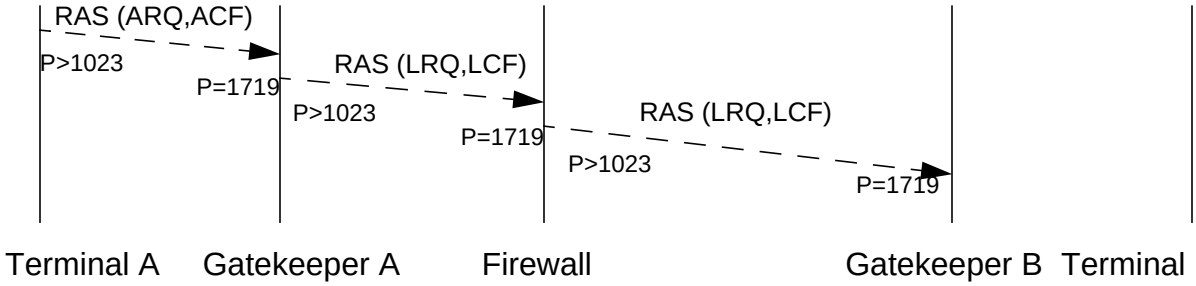

Abbildung 17: Gatekeeper-Gatekeeper Kommunikation (II.b)

- **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 (*AdmissionRequest ARQ*).

- **RAS Registration, Admission, Status (UDP):**
  Innerhalb der *ARQ*-Nachricht ist die Zieladresse von Terminal B enthalten. Die Zieladresse kann dabei in verschiedenen Formaten angegeben sein, beispielsweise als E.164-Nummer, als IP-Adresse, als H323-ID oder in einer anderen Form. Wir nehmen in diesem Beispiel an, das die Zieladresse von Terminal B als E.164-Nummer angegeben ist. Da kein Terminal unter dieser Nummer bei Gatekeeper A registriert ist, verwendet Gatekeeper A nun seine *Neighbour Table* um herauszufinden, ob ein erreichbarer Gatekeeper nach einem registrierten Terminal mit dieser Nummer gefragt werden kann. Diese Anfrage wird innerhalb einer *LRQ*-Nachricht an den entfernten Gatekeeper übermittelt. In diesem Beispiel existiert in der *Neighbour Table* ein einziger Eintrag: alle durch den Gatekeeper generierten *LRQ*-Anfragen werden an die Firewall weitergeleitet.

- **RAS Registration, Admission, Status (UDP):**
  Die Firewall erhält die *LRQ*-Nachricht. In der Nachricht sind die folgenden Werte enthalten: Eine Sequenznummer der Nachricht; die E.164-Nummer, zu der das entsprechende Terminal gesucht wird; Die IP-Adresse, an welche die Antwort auf die *LRQ*-Nachricht gesendet werden soll. Die Firewall selbst benötigt nun ebenfalls eine *Neighbour Table* um zu bestimmen, wohin die Anfrage weitergeleitet werden soll. Wir nehmen an, das die Firewall anhand des Nummern Prefixes der übermittelten E.164-Nummer Gatekeeper B als Ziel der *LRQ*-Nachricht ermittelt. Bevor die *LRQ*-Nachricht von der Firewall an Gatekeeper B weitergeleitet wird, merkt sich die Firewall die original *LRQ*-Nachricht und ersetzt die enthaltene IP-Adresse, an die die Antwort gesendet werden soll, durch die IP-Adresse der Firewall. Dadurch wird Gatekeeper B veranlaßt, die Antwort an die Firewall und nicht direkt an Gatekeeper A zu senden. Danach wird die *LRQ*-Nachricht an den Gatekeeper B weitergeleitet.

- **RAS Registration, Admission, Status (UDP):**
  Gatekeeper B erhält nun die *LRQ*-Nachricht. Da Terminal B unter der übermittelten

E.164-Nummer angemeldet und damit erreichbar ist, sendet Gatekeeper B an die Firewall eine *LocationConfirm- (LCF)* Nachricht an die Firewall zurück. Diese Nachricht enthält folgende Informationen: Die Sequenznummer des *LRQ*, auf welche sich das *LCF* bezieht; die Signalisierungs Zieladresse (IP Adresse und Port für den folgenden Call Signalling Flow). Wir nehmen in diesem Beispiel an, daß Gatekeeper B einen nicht Gatekeeper vermittelten Ruf verwenden möchte, und deshalb als Call Signalling Adresse direkt die IP-Adresse des Terminals B übermittelt. Die Firewall benutzt nun die in der *LCF* und der zuvor gespeicherten *LRQ*-Nachricht enthaltenen Informationen um eine Call-Routing Tabelle aufzubauen. Die Tabelle enthält nun für die E.164-Nummer die Ziel IP-Adresse für das Call Signalling, in diesem Fall die IP-Adresse von Terminal B. Bevor die *LCF* an den Gatekeeper A weitergeleitet wird, ersetzt die Firewall die Call Signalling Adresse durch die eigene IP-Adresse.

- **RAS Registration, Admission, Status (UDP):**
  Gatekeeper A sendet nun auf die initiale *ARQ*-Anfrage des Terminals A eine *ACF* Nachricht, welche dem Terminal bestätigt, daß das gewünschte Ziel Terminal erreichbar ist. Zusätzlich enthält die *ACF*-Nachricht die Call Signalling Adresse, welche für den Ruf verwendet werden soll. In diesem Beispiel nehmen wir an, daß Gatekeeper A einen Gatekeeper vermittelten Ruf verlangt. Als Call Signaling Adresse wird deshalb die IP Adresse von Gatekeeper A in der *LCF*-Nachricht angegeben.

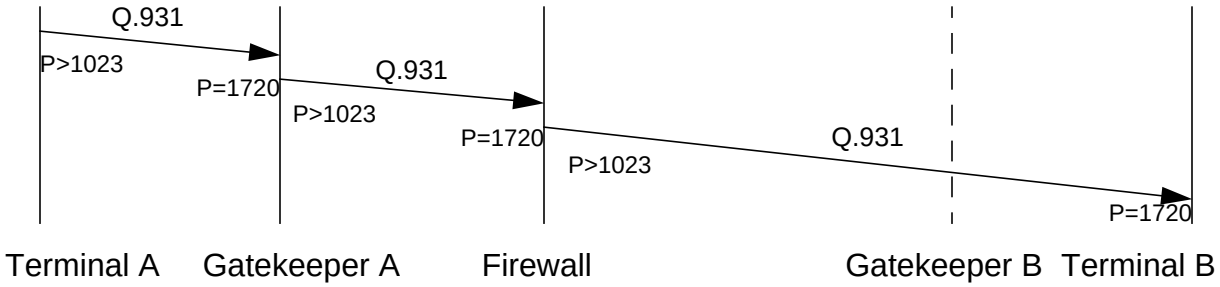

Abbildung 18: Gatekeeper-Gatekeeper Kommunikation (II.c)

- **Q.931 Call Signalling (TCP):**
  Bei einem Verbindungsaufbau kontaktiert das Terminal A nun entsprechend der in der *ACF*-Nachricht übermittelten Parameter Gatekeeper A. Danach wird die initiale Q.931-*Setup*-Nachricht übermittelt, welche die E.164-Adresse des gewünschten Ziels enthält.

- **Q.931 Call Signalling (TCP):**
  Gatekeeper A verwendet die übermittelte E.164-Adresse um zu erkennen, daß das zu kontaktierende Ziel die Firewall ist. Diese Information wurde zuvor über die *LCF*-Nachricht dem Gatekeeper A mitgeteilt. Die *Setup*-Nachricht wird nun an die Firewall weitergeleitet.

- **Q.931 Call Signalling (TCP):**
  Die Firewall verwendet wiederum die E.164-Adresse um das Ziel des Rufes zu verwenden. Wie zuvor beschrieben, wurden die *LRQ*- und *LCF*-Nachrichten dazu ver-

wendet, eine Call-Routing Tabelle in der Firewall aufzubauen. Diese wird nun verwendet um das Ziel zu bestimmen. Die Firewall kontaktiert daraufhin Terminal B und leitet die CallSetup Nachricht an das Terminal weiter.
Das Call Signalling ist nun vollständig aufgebaut, und die folgenden Q.931-Nachrichten können zwischen den Terminals über die jeweilig involvierten Infrastrukturkomponenten ausgetauscht werden.

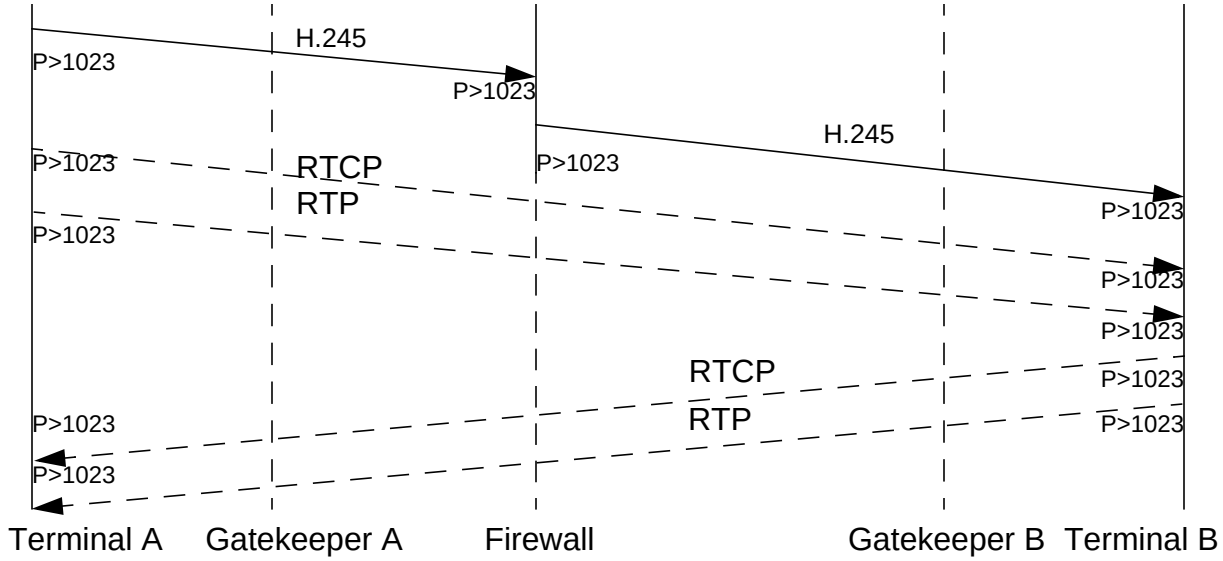

Abbildung 19: Gatekeeper-Gatekeeper Kommunikation (II.d)

- **H.245 Call Control (TCP), RTP/RTCP Media und Mediacontrol (UDP):**
  Analog zu den in Kapitel 6.2.1 beschriebenen Mechanismen kann die Firewall sich nun in die gesamte Kommunikation einschalten, indem die übermittelten Flow Spezifikationen in den Signalisierungsnachrichten modifiziert werden.
  In dem hier gegebenen Beispiel wird der H.245 Flow über die Firewall geleitet, nicht aber über die Gatekeeper. Die Medienströme werden direkt zwischen den Terminals verwendet.

Dieser Mechanismus ist ebenfalls verwendbar, wenn Gatekeeper A und die Firewall integriert werden (ähnlich dem zuvor beschriebenen Verfahren). Der Vorteil dieses Verfahrens besteht darin, daß die Firewall in die H.323-Szenarien transparent eingebunden werden kann. Ein weiterer Vorteil liegt darin, daß die Firewall im Gegensatz zu dem davor beschriebenen Verfahren keine vollständige Gatekeeper Implementierung aufnehmen muß. Dadurch kann die Komplexität einer solchen Firewall reduziert werden, was die Integration in bestehende Firewall Systeme vereinfacht.
Ein Beispiel, welches diese Form des Routing über eine Firewall unterstützt, ist die Firewall Software KOMproxyd [30].

6.2.3 Herstellerspezifische Implementierungen

Nicht nur die Nutzung verschiedener H.323-Komponenten sondern auch deren unterschiedliche 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.
Wie bereits beschrieben, werden die verwendeten Medienströme über entsprechende H.245-Nachrichten zwischen den Kommunikationspartnern ausgehandelt. Ein Terminal, welches beispielsweise einen Video-Flow an das andere Terminal übermitteln möchte, sendet dazu eine entsprechende *OpenLogicalChannel* Nachricht an das andere Terminal. Dieses bestätigt die *OpenLogicalChannel* Nachricht mit einer entsprechenden *OpenLogicalChannelAck* Nachricht. Danach sind die Flow Parameter festgelegt und der Video Flow kann zwischen den Terminals ausgetauscht werden. Werden mehrere Flows verwendet, wiederholt sich diese Prozedur. Je nach Art der verwendeten Terminals können diese Nachrichten aber auch in anderer Reihenfolge ausgetauscht werden. Beispielsweise können die Terminals zuerst alle nötigen *OpenLogicalChannel* Nachrichten austauschen und danach alle *OpenLogicalChannelAck* Nachrichten. In anderen Fällen wird nach jeder *OpenLogicalChannel* Nachricht zunächst ein *OpenLogicalChannelAck* gesendet. Für eine dazwischenliegende Firewall ergibt sich nun das Problem, daß sie mit jeder Variation umgehen können muß.
Ein weiteres Beispiel ist in der Q.931-*Setup*-Nachricht zu finden. Wie bereits erwähnt enthält die *Setup*-Nachricht die Adresse des gerufenen Terminals. Der H.323-Standard läßt mehrere Möglichkeiten zu diese Zieladresse innerhalb der Nachricht unterzubringen (im Feld *CalledPartyNumber* oder innerhalb der *UserUserIE* im Feld *DestinationAddress*). Manche Terminals verwenden das *CalledPartyNumber*-Feld, andere das *DestinationAddress*-Feld und wieder andere Terminals füllen die Information in beide Felder. Eine dazwischenliegende Firewall muß in der Lage sein alle Varianten zu unterstützen um ein flexibles Call Routing zu ermöglichen.

6.2.4 Verwendung zusätzlicher Protokolle

Ist zum Beispiel Terminal A kein “reines” H.323-Terminal sondern ein Microsoft Netmeeting-Terminal, so kann eine ILS/LDAP-Erweiterung verwendet werden. Bevor die Kommunikation beginnt, versucht das rufende Terminal 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. In diesem Fall werden teilweise die Funktionen des Gatekeepers durch den ILS-Server erbracht. Die dabei verwendeten Kommunikationsmechanismen sind außerhalb des H.323-Standards festgelegt, sind aber logisch der H.323-Kommunikation zuzurechnen.

Bei der Verwendung dieser speziellen H.323-Applikation muß eine im Kommunikationsweg befindliche Firewall in der Lage sein, die für die Kommunikation mit dem ILS Server verwendeten Flows und die H.323 spezifischen Flows einer logischen Session zuzuordnen.

6.3 Performance-Anforderungen

Um eine H.323-Applikation sinnvoll zu betreiben, sind verschiedene Dienstgüteanforderungen zu erfüllen. Hinsichtlich der Unterstützung der H.323-Applikationen durch Firewalls ist die durch die Firewall zusätzlich eingebrachte Verzögerung der Signalisierungs- und Mediendaten sowie der erreichbare Datendurchsatz wesentlich. Diese Aspekte werden nachfolgend behandelt.

6.3.1 Signalisierung

Die Signalisierung wird dazu verwendet, die Medienströme auszuhandeln und zu initiieren. An die Geschwindigkeit, mit der diese Aushandlung geschieht sind gewisse Mindestanforderungen zu stellen. Folgendes Beispiel soll dies verdeutlichen: Ein Terminal A ruft (direkter Ruf) ein Terminal B an. Die Q.931 Call Signalling-Verbindung wird zwischen beiden Terminals aufgebaut und die zur Initiierung notwendigen Q.931-Nachrichten werden ausgetauscht. Zu diesem Zeitpunkt beginnt das Terminal B zu klingeln, um anzuzeigen das ein Verbindungswunsch besteht. Nimmt nun eine Person an Terminal B den Ruf entgegen, z.B. durch Abheben des Hörers, so wird die H.245 Call Control Verbindung aufgebaut. Über die H.245 Call Control Verbindung werden nun die Medienströme ausgehandelt und initiiert. Zur Aushandlung und Initiierung der Medienströme bleibt in diesem Beispiel gerade soviel Zeit, wie die Person an Zeit benötigt um den abgehobenen Telefonhörer zum Ohr zu befördern. Ist der Hörer am Ohr angekommen, wird die Person sich in der Regel als erstes melden, z.B. mit dem eigenen Namen. Sind zu diesem Zeitpunkt die Medienströme noch nicht geschaltet, werden diese ersten Worte verlorengehen, was zu erheblichen Irritationen der Gesprächsteilnehmer führen kann. Um diesem Effekt vorzubeugen, wurde innerhalb von H.323 die *Faststart* Variante festgelegt. Diese ermöglicht eine Festlegung der Medienströme schon innerhalb des Q.931-*Setup*. Allerdings wird diese Variante zur Zeit nicht von allen Terminaltypen unterstützt. Eine Firewall sollte dementsprechend eine möglichst geringe zusätzliche Verzögerung der Signalisierungsnachrichten verursachen um den Effekt nicht zu verstärken.

6.3.2 Medienströme

Für die über die Firewall transportierten Mediendaten müssen Grenzwerte für die Verzögerung und den Jitter eingehalten werden. Damit Audio- und Videodaten von einem Empfänger verarbeitet werden können, muß hinsichtlich der Ende-Zu-Ende Verzögerung der transportierten Daten ein Grenzwert eingehalten werden. Die Ende-Zu-Ende Verzögerung der Daten wird ohne die Verwendung einer Firewall im wesentlichen durch das dazwischenliegende Transportnetz verursacht. Eine in den Kommunikationspfad eingebrachte Firewall sollte die durch das Transportnetz verur-

sachte Verzögerung möglichst nicht erhöhen. Da sich die Ende-Zu-Ende Verzögerung der Daten aber zwangsläufig durch die Firewall erhöht,

- sollte dies vorhersagbar geschehen.
  Der Betrag um den die Ende-zu-Ende Verzögerung der einzelnen Pakete erhöht wird, sollte für jedes Paket möglichst gleich sein. Dadurch wird ein planbares bzw. voraussagbares Verhalten der Firewall gewährleistet. Außerdem wird dadurch verhindert, daß der Jitter durch die Firewall zusätzlich erhöht wird.

- sollte die Erhöhung möglichst gering ausfallen.

Zusätzlich muß eine Firewall den gewünschten Datendurchsatz ermöglichen, damit auch Medienströme mit einer hohen Bandbreite, bzw. mehrere gleichzeitige Sessions von einer Firewall gehandhabt werden können.

6.3.3 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. Die zuvor beschriebenen H.323-Charakteristika bzw. die dadurch entstehenden Probleme verstärken sich dadurch nochmals. Diese zusätzlichen Schwierigkeiten werden im folgenden genauer dargestellt.

Protokollcharakteristika - Komplexität der Protokolle. Innerhalb der Applikationsschicht (Schicht 5) der H.323-Protokolle werden IP-Adressen verwendet. Damit NAT für das H.323-Protokoll verwendet werden kann, muß nicht nur die Adressierung der Flows auf Schicht 3 (IP-Adressen) und Schicht 4 (Port-Nummern) verändert werden, sondern es müssen auch alle innerhalb der Schicht 5 übermittelten IP-Adressen sowie Port-Nummern verändert werden. Durch diese zusätzliche Aufgabe erhöht sich die Komplexität der Firewall nochmals.

Protokollcharakteristika - Mehrere Flows in einer Session. Bei der Ermittlung, welche Flows einer Session zugeordnet sind, muß die Firewall zusätzlich die aktuell gültige Adreßumsetzung beachten.

Protokollcharakteristika - Dynamisches Verhalten. Die verwendeten Adressen der meisten Flows werden dynamisch ausgehandelt. In dieser Aushandlung muß nun die Adreßumsetzung beachtet werden, alle ausgehandelten IP-Adressen und Port-Nummern müssen durch die Firewall modifiziert werden, um die Adreßumsetzung entsprechend abbilden zu können.

Applikationscharakteristika - Call Routing. Die internen Terminals können nicht direkt von außerhalb angerufen werden, da ihre Adressen vom Internet aus nicht “sichtbar” sind. Aus diesem Grund können in einer NAT Umgebung nur Call Routing Mechanismen verwendet werden, bei welchen die initiale Call Signalling Adresse (Q.931) eine öffentlich gültige IP-Adresse darstellt. Die folgenden, zuvor beschriebenen Mechanismen können verwendet werden:

- **Explizit - Modifikation von H.323-Komponenten.**
  Das rufende, externe Terminal adressiert direkt die Firewall.

- **Transparent - Nutzung von Infrastrukturkomponenten.**
  Der in der Firewall angeordnete Gatekeeper muß eine öffentlich gültige IP-Adresse verwenden.

- **Transparent - Nutzung der Gatekeeper-Gatekeeper Kommunikation.**
  Die Firewall verwendet eine gültige IP-Adresse, welche in der *Neighbour Table* der externen Gatekeeper eingetragen ist.

Ausgehende Rufe stellen in einer NAT-Umgebung prinzipiell kein Problem dar, da das Ziel eine gültige Adresse besitzt. Es ist anzunehmen, daß sich in einem typischen realen Szenario beide Parteien (rufendes und gerufenes Terminal) hinter einer Firewall befinden werden. Deshalb stellt das Problem der eingehenden Rufe ein wichtiges Problem dar.

Applikationscharakteristika - Verwendung zusätzlicher Protokolle. Für die zusätzlich verwendeten Protokolle muß ebenfalls eine Adreßumsetzung von der Firewall durchgeführt werden.

Performance-Anforderungen - Medien. In einer Umgebung, in der keine Adreßumsetzung nötig ist, kann die H.323-Firewall sich darauf beschränken für die über die Signalisierung vereinbarten Medienströme die entsprechenden Kommunikationswege zu öffnen. Nachdem die entsprechenden Kommunikationswege freigeschaltet sind, kann die Firewall sich darauf beschränken die Medien-Pakete möglichst schnell weiterzuleiten. Wird eine Adreßumsetzung notwendig, so muß die Firewall in jedem weiterzuleitenden Paket die Adresse umschreiben. Je nach verwendeter Technik kann dies zu einem erheblichen Mehraufwand bei der Weiterleitung dieser Pakete führen. Dadurch erhöht sich der Delay sowie der Jitter der entsprechenden Pakete.

6.4 Anforderungen an eine H.323-Firewall

Es bestätigen sich die bereits in Kapitel 5.3 dargestellten allgemeinen Anforderungen sowie die daraus folgenden Design-Pattern. Es wurde gezeigt, daß alle allgemein in Kapitel 5.2 aufgestellten und für Firewalls problematischen Charakteristika von Multimedia-Applikationen auch - bzw. in den meisten Fällen in verstärktem Maße - innerhalb von H.323-Applikationen auftreten. Verwendet man die in Kapitel 5.3 gegebenen allgemeinen Design-Pattern, die in diesem Kapitel vorgestellten H.323 spezifischen Erkenntnisse sowie die in Kapitel 4.3 dargestellten Security-Vorgaben so erhält man folgende Design-Pattern, die von einer H.323-Firewall erfüllt werden müssen:

- **Architektur:** Verteilte Firewall mit Interaktion der einzelnen Firewall-Komponenten.
  Aufgrund der hohen Komplexität des H.323-Protokolls ist es sinnvoll eine verteilte, aus mehreren Komponenten bestehende Firewall zu verwenden. Dies ermöglicht es, die H.323 spezifischen Firewall-Aufgaben in eine (bzw. mehrere) dafür optimierte Firewall-Komponenten zu verlegen. Die Verteilung ermöglicht es zusätzlich, für die Bearbeitung der verschiedenen Flows einer H.323-Session (Signalisierungs- und Medien-Flows) ebenfalls jeweils optimierte Firewall-Komponenten zu verwenden. Dies erlaubt eine Optimierung der Firewall hinsichtlich der Performance, erfordert aber einen Informationsaustausch zwischen den einzelnen Firewall-Komponenten. Als Firewall Architektur bietet sich demnach eine verteilte Firewall mit Interaktionsmög-

lichkeit der einzelnen Komponenten an, wobei die einzelnen Komponenten in ihrer Gesamtheit in der Lage sind, der Dynamik einer H.323-Kommunikation zu folgen. Architekturen dieser Art sind beispielsweise in [28], [31], [32], [33], [34] beschrieben.

- **Call Routing:** Integration verschiedener Call Routing-Methoden.
  Die Firewall muß in die H.323-Kommunikation mit einbezogen werden, damit sie ihre Aufgaben erfüllen kann. Dazu muß die Firewall in das Call Routing integriert werden. Da die Call Routing-Möglichkeiten von dem verwendeten H.323-Szenario abhängen, muß eine H.323-Firewall verschiedene Mechanismen unterstützen, um an das jeweilig verwendete Szenario angepaßt werden zu können.

- **Konfiguration:** Anpaßbarkeit der Firewall an unterschiedliche Szenarien.
  Eine H.323-Firewall muß, um an verschiedene Szenarien angepaßt werden zu können, verschiedene H.323 spezifische Konfigurationsmöglichkeiten bieten. Zum einen muß es möglich sein, das Call Routing-Verhalten beeinflussen zu können. Zum anderen sind Konfigurationsmöglichkeiten nötig, die es ermöglichen die Firewall an die gegebenen Applikationscharakteristika (herstellerspezifische Implementierungen, Verwendung zusätzlicher Protokolle) anzupassen.

- **Sicherheit:** Korrektheit der Signalisierung.
  Eine H.323-Firewall muß die Korrektheit der Signalisierung überprüfen. Dabei ist die Überprüfung auf Korrektheit der einzelnen Nachrichten notwendig, sowie die Korrektheit der zeitlichen Zusammenhänge einzelner Nachrichten. Dies betrifft RAS-, Q.931- und H.245-Nachrichten.
  Desweitern muß die Firewall in der Lage sein, die Weiterleitung bestimmter Nachrichten explizit zu erlauben, bzw. zu verbieten. Es sollte beispielsweise möglich sein, die Weiterleitung bestimmter RAS-Nachrichten an einen internen Gatekeeper zu erlauben (z.B. *LRQ, LCF, LRJ*), aber die Weiterleitung anderer RAS-Nachrichten zu verbieten (z.B. *RRQ*).
  Außerdem sollte die Firewall es ermöglichen bestimmte Dienste und Funktionen, die ein H.323-Szenario bietet, an der Firewall einzuschränken. Es sollte beispielsweise möglich sein, den Aufbau eines T.120 Flows über die Firewall zwischen einem internen und einem externen Terminal zu verhindern.
  Schließlich sollte die Firewall in der Lage sein, mit in H.323 integrierten Sicherheitsfunktionen zurecht zu kommen (z.B. H.235).

Es existieren verschiedene Möglichkeiten, diese Pattern technisch umzusetzen. Einige ausgewählte Realisierungen werden in den folgenden Kapiteln vorgestellt und diskutiert. Die meisten Realisierungen bieten nicht die Möglichkeit, alle Pattern in gleichem Maße umzusetzen bzw. zu berücksichtigen, so daß eine Optimierung hinsichtlich verschiedener Kriterien vorgenommen werden muß. Es ist ebenfalls zu beachten, daß zusätzlich zu den hier aufgeführten Design-Pattern die grundlegenden Design-Pattern einer Firewall (z.B. [14],[15]) zu erfüllen sind. Teilweise widersprechen sich die verschiedenen zu erfüllenden Pattern (z.B. Pattern1: Die Firewall muß Zustände über die Session-Zugehörigkeit der verwendeten Flows halten; Pattern2: Eine Firewall

sollte möglichst einfach aufgebaut sein) was wiederum zu verschiedensten Optimierungsproblemen führt.

7. Firewall-Architekturen

Wie in Kapitel 6.4 dargestellt, ist es sinnvoll für die Unterstützung einer Multimedia-Applikation eine verteilte Firewall mit Interaktion der einzelnen Firewall-Komponenten zu verwenden. Es ist dabei möglich, verschiedene Architekturen zu verwenden, die die in Kapitel 2.2 beschriebenen Firewall-Funktionen erbringen. Im folgenden wird ein Klassifizierungsschema vorgestellt, das es ermöglicht verschiedene verteilte Firewall Architekturen einzuordnen und zu bewerten. Am Ende dieses Kapitels werden die verschiedenen prinzipiell möglichen Architekturen diskutiert. Die gewählte Architektur beeinflußt - neben anderen Parametern - wesentlich die Performance der Firewall, weshalb es im Hinblick auf die zu unterstützenden Multimedia-Applikationen wichtig ist, diesen Aspekt genauer zu betrachten.

7.1 Klassifizierungsschema für Firewall-Architekturen

Damit die einzelnen Komponenten einer verteilten Firewall für Multimedia-Applikationen ihre Funktionen erfüllen können, ist ein Informationsaustausch zwischen den einzelnen Komponenten notwendig. Im wesentlichen sind dabei die Informationen über die verwendeten Flows einer Multimedia-Session von Bedeutung.
Das folgende Beispiel soll dies verdeutlichen. Angenommen, die Firewall besteht aus einem Paketfilter (durchlaufen von den Signalisierungs- und Medien-Flows), sowie einem Proxy (durchlaufen von den Signalisierungs-Flows) der Multimedia-Applikation. In diesem Fall benötigt der Paketfilter Informationen über den aktuellen Zustand der Kommunikations-Flows, um seine Konfiguration an den aktuellen Zustand anpassen zu können. Diese Informationen können innerhalb der Firewall nur durch den Proxy bereitgestellt werden, da nur diese Komponente die nötigen Informationen besitzt. Es ist also eine Übergabe der aktuellen Flow-Spezifikationen von dem verwendeten Proxy an den Paketfilter notwendig. Entsprechende Schnittstellen zwischen Proxy und Paketfilter müssen dazu vorhanden sein.
Die einzelnen logischen Bausteine, die an diesem Informationsaustausch beteiligt sind, sind in Abbildung 20 dargestellt. Die Aufgabe der einzelnen Bausteine wird nachfolgend erläutert.

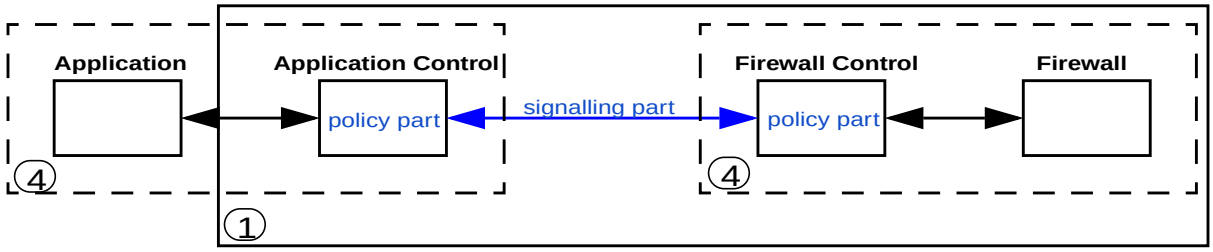

Abbildung 20: Logische Bausteine einer Firewall

Wie bereits erwähnt, ist es notwendig, die Flow-Spezifikationen der einzelnen Flows von der Applikation zur Firewall (bzw. auf die entsprechenden in der Firewall verwendeten Komponenten

zu transportieren[1]. Die Flow-Spezifikation kann in ihrer einfachsten Form durch das in Abbildung 21 dargestellte Tupel beschrieben werden.

F={PROTOCOL, SRC-IP,SRC-PORT,DST-IP,DST-PORT}

Abbildung 21: Tupel zur Flow-Beschreibung

Falls es zur Umsetzung der für die Firewall gültigen Security Policy notwendig ist, muß das angegebene Tupel entsprechend erweitert werden (z.B. um eine Angabe, die den Benutzer des entsprechenden Flows ausweist).

Application. Dieses Element beschreibt die Applikationsseite. Es ist damit nicht unbedingt die Applikation selbst gemeint, sondern vielmehr ein Element innerhalb des Applikationsszenarios, welches für die Kommunikation verwendet wird.

Application Control. Das als “Application Control” bezeichnete Element beinhaltet die Funktionen die nötig sind, um die Flow-Spezifikationen aus der “Applikationswelt” zu extrahieren. Dieses Element stellt damit die Schnittstelle zwischen Multimedia-Applikation und Firewall dar. Die technische Realisierung kann auf verschiedene Arten umgesetzt werden. Es ist beispielsweise möglich die Application Control innerhalb eines Endgerätes (z.B. eines H.323-Terminals) unterzubringen. Die Flow-Spezifikationen werden in diesem Fall von einem Endgerät an die Firewall übergeben. Eine andere Methode ist die Realisierung der Application Control durch einen im Kommunikationsweg liegenden Proxy. In diesem Fall werden die Flow-Spezifikationen aus der ablaufenden Kommunikation extrahiert und an die Firewall weitergegeben. Es sind weitere technische Realisierungen dieser Einheit möglich, die aber dem gleichen Zweck dienen.

Application Control - Policy Part. Der Policy Part innerhalb der Application Control legt fest, nach welchen Regeln die Flow-Spezifikationen übermittelt werden. Zusätzlich werden die Flow-Spezifikationen in ein für die Weitergabe geeignetes Format gebracht.

Signalling Part. Der Signalling Part wird dazu verwendet die von der Application Control an die Firewall Control zu übergebenden Flow-Spezifikationen zu transportieren. Befinden sich die technischen Umsetzungen von Firewall Control und Application Control innerhalb der gleichen Firewall-Komponente, so besteht der Signalling Part aus einer einfachen API. Sind die durch den Signalling Part zu verbindenden logischen Einheiten auf unterschiedlichen Firewall-Komponenten untergebracht, so beinhaltet der Signalling Part ein Netzwerkprotokoll, um die Informationen über das zwischen den Komponenten liegende Netzwerk auszutauschen.

Firewall Control. Dieses Element ist dafür verantwortlich die - entsprechend der übermittelten Flow-Spezifikationen - notwendigen Konfigurationsanpassungen innerhalb der Firewall-Komponenten vorzunehmen.

1. Es werden auch Nachrichten in Gegenrichtung ausgetauscht. Z.B. wird ein Flow durch die Firewall terminiert, so muß dies in Gegenrichtung in die Applikationswelt kommuniziert werden.

Firewall Control - Policy Part. Der Policy Part innerhalb der Firewall Control legt fest, nach welchen Regeln die übermittelten Flow-Spezifikationen an die Firewall-Komponenten übermittelt werden. Beispielsweise kann hier festgelegt werden, keine Flow-Spezifikationen an die Komponenten weiterzuleiten, die eine Port-Nummer kleiner 1024 für den SRC-Port oder DST-Port angeben. Außerdem wird die Umwandlung der übermittelten generischen Flow-Spezifikationen in die jeweiligen, für Firewall-Komponenten spezifischen Formate, durchgeführt.

Firewall. Dieses Element beschreibt die Firewall-Seite. Die Firewall-Seite kann wiederum selbst aus mehreren Firewall-Komponenten bestehen.

Bei einer Firewall, welche aus mehreren Komponenten besteht, kann es nötig sein, die Flow-Spezifikationen auf mehrere Firewall-Komponenten zu verteilen. Dies kann auf verschiedene Arten geschehen. Die von der Application Control extrahierten Flow-Spezifikationen können an mehrere Firewall-Control Instanzen übergeben werden, die jeweils die zugeordnete Firewall-Komponente ansteuern. Eine andere Möglichkeit besteht darin, daß die Flow-Spezifikation an nur eine Firewall-Control Instanz übergeben wird, welche in der Lage ist eine Unterverteilung der Information auf die verschiedenen Firewall-Komponenten durchzuführen. Bei beiden Varianten wird nur eine Instanz der Application Control verwendet; dies ist sinnvoll, da so die Flow-Spezifikationen nur ein einziges Mal aus der Applikationswelt gewonnen werden müssen. Eine weitere Möglichkeit, die Flow-Spezifikationen auf die einzelnen Firewall-Komponenten zu verteilen besteht darin, in jeder Firewall-Komponente eine Application Control und eine Firewall Control-Instanz zu verwenden. Diese Variation wird in der Praxis oft verwendet, aus verschiedenen Gründen (Performance, Komplexität, Anpaßbarkeit des Gesamtsystems [31]) ist diese Variante aber nicht sinnvoll.

7.2 Mögliche Firewall-Architekturen

Im Folgenden werden verschiedene Firewall-Architekturen dargestellt, sowie ihre Vor- und Nachteile besprochen. Die hier gezeigten Architekturen ergeben sich durch die entsprechende örtliche Anordnung in Firewall-Komponenten der zuvor beschriebenen logischen Bausteine des Firewall-Klassifizierungsschemas. Die hier gezeigten Architekturen entsprechen den zur Zeit in der Literatur diskutierten Lösungsansätzen. Zur Darstellung werden die folgenden Symbole verwendet:

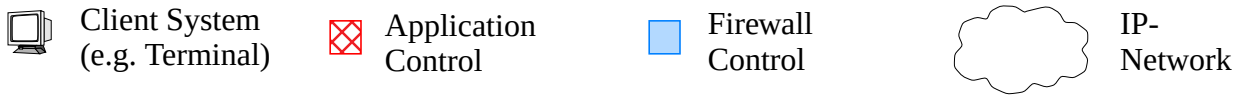

Abbildung 22: Verwendete Symbole zur Beschreibung der Firewall-Architekturen

I. Hybridsystem. Bei einem Hybridsystem werden die verschiedenen Firewall-Komponenten innerhalb einer Physikalischen Komponente realisiert.

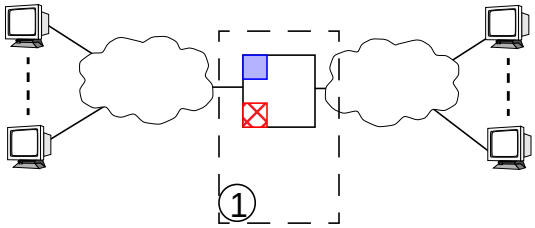

Abbildung 23: Hybridsystem

Die Application Control ist in diesem Fall als Protokoll-Parser ausgeführt, der die Kontroll-Flows des Multimedia-Protokolls verarbeiten und analysieren kann. Für jedes durch die Firewall zu unterstützende Multimedia-Protokoll ist ein spezialisierter Protokoll-Parser notwendig. Die an die Firewall Control übermittelten Flow-Spezifikationen werden verwendet um die im Hybrid-System verwendeten Firewall-Komponenten (z.B. Stateful Filter, NAT-Komponente) an die Kommunikation anzupassen.

Zusammenfassung:

- Realisierungen:
  Klassische, heute verwendete kommerzielle Firewall-Systeme entsprechen dieser Architektur. Bekannte Vertreter dieser Variante sind: Checkpoint´s Firewall-1 [39], Cisco PIX [38].

- Anforderungen:
  Aus den in Kapitel 5.3 bzw. in Kapitel 6.4 beschriebenen Anforderungen geht hervor, daß ein Hybridsystem keine geeignete Firewall Architektur darstellt, um Multimedia-Applikationen bzw. H.323 Applikationen zu unterstützen. Dennoch werden diese Systeme zur Zeit - mit geringem Erfolg - für diesen Zweck eingesetzt.

- Performance:
  Alle Flows (Signalisierungs- und Medien-Flows) müssen durch dasselbe System verarbeitet werden. Die Verwendung unterschiedlicher und optimierter Komponenten für Signalisierungs- bzw. Medienverarbeitung ist nicht möglich. Es ist nur möglich die verschiedenen Flows auf jeweils unterschiedlichen und optimierten Wegen innerhalb des Hybridsystems weiterzuleiten.
  Parallele Sessions müssen von der gleichen Komponente hinsichtlich Signalisierungs- und Medien-Flows behandelt werden. Insbesondere durch die hohe Ressourcennut-

zung der Application Control (die linear mit der Anzahl der Sessions steigt) werden Grenzen bei der parallelen Verarbeitung vieler Sessions erreicht. Diese Grenzen können nach oben verschoben werden (durch die Verwendung entsprechend leistungsfähigerer Systeme), dies ist aber nicht beliebig möglich und verursacht zunehmende Kosten.
Ein Vorteil dieses Systems liegt darin, daß bei diesem System die Flow-Spezifikationen nicht über ein zwischenliegendes Netz transportiert werden müssen. Der Signalling Part kann hier durch relativ simple Funktionsaufrufe umgesetzt werden.

II. Verteiltes System (I). Das in Abbildung 24 dargestellte verteilte System besteht aus mehreren Proxies sowie einem Stateful Filter (inklusive NAT-Komponente). Die Proxies sind mit dem Filter über ein spezielles Netzwerk verbunden. Dieser Netzwerkabschnitt[1] wird nur innerhalb der Firewall verwendet.

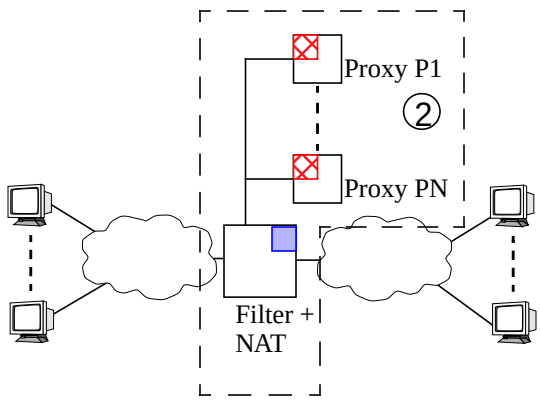

Abbildung 24: Verteiltes System (I)

Die Application Control ist als Protokoll-Parser innerhalb der verwendeten Proxies ausgeführt, der die Kontroll-Flows des Multimedia-Protokolls verarbeiten und analysieren kann. Die Firewall Control ist in dieser Architektur durch zwei Instanzen vertreten. Die erste Instanz befindet sich innerhalb des Proxies. Der Proxy, welcher die Application Control beinhaltet, benötigt selbst die Flow-Spezifikationen, um sich an die Kommunikationsbedingungen anpassen zu können. Die Flow-Spezifikationen werden in diesem Fall durch einen Funktionsaufruf übergeben. Die zweite Instanz der Firewall Control befindet sich innerhalb des Filters. Die Flow-Spezifikationen müssen in diesem Fall über das Firewall interne Kommunikationsnetzwerk von dem verwendeten Proxy zum Filter transportiert werden. Die Flow-Spezifikationen werden vom Filter (sowie der NAT-Komponente) verwendet, um sich an die aktuelle Kommunikation anzupassen.
Eine weitere Möglichkeit innerhalb dieser Architektur ist dadurch gekennzeichnet, daß die Proxies durch Infrastrukturelemente des zu unterstützenden Multimedia-Szenarios ersetzt werden können. In einem H.323-Szenario ist es beispielsweise möglich, den Proxy durch einen H.323-Gatekeeper zu ersetzen. Der H.323-Proxy enthält dann das Application Control-Element, sowie die Möglichkeit mit dem Filter zu interagieren.

- Realisierungen:
  Spezielle Proxy-Systeme, die durch einen Paketfilter gesichert sind, sind zum Teil nach dem hier dargestellten Schema aufgebaut. Allerdings fehlt bei diesen Systemen die Interaktion zwischen Proxy und Filter. Der Filter muß in solchen Fällen statisch konfiguriert werden.
  Das System KOMproxyd kann so eingesetzt werden, daß eine Interaktion nach dem

1. Dieses Netzwerk wird in der Literatur häufig als Demilitarisierte Zone (DMZ) bezeichnet.

oben dargestellten Muster möglich ist [30].
Weitere Systeme dieser Art sind in [33] und [34] beschrieben.

- Anforderungen:
  Die hier beschriebene Architektur erfüllt die in Kapitel 5.3 bzw. in Kapitel 6.4 beschriebenen Anforderungen.

- Performance:
  Verglichen mit einem Hybridsystem ist es möglich, mehrere Proxies zu verwenden, um bei parallel zu verarbeitenden Sessions die Last zu verteilen. Es ist dabei aber zu beachten, daß die Signalisierung zwischen Application Control und Firewall Control über ein Netzwerk erfolgen muß. Der Austausch der Flow-Spezifikationen ist verglichen mit dem Hybridsystem aufwendiger.
  Im Vergleich zu einem Hybridsystem ist es möglich, Signalisierungs-Flow und Medien-Flow durch spezialisierte Komponenten zu bearbeiten. Der verwendete Filter kann beispielsweise für die Weiterleitung der Medien-Flows optimiert werden (siehe [31]).

III. Verteiltes System (II). Das hier beschriebene System entspricht in großen Teilen dem zuvor beschriebenen “Verteilten System I”.

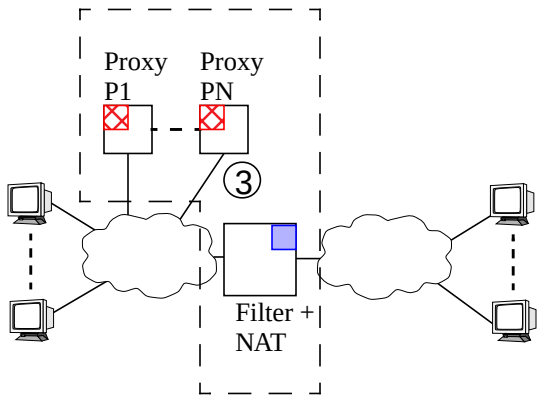

Abbildung 25: Verteiltes System (II)

Der wesentliche Unterschied besteht darin, daß die zwischen Application Control und der innerhalb des Filters liegenden Firewall Control ausgetauschten Flow-Spezifikationen über das gleiche Netz transportiert werden müssen, die auch zum Transport der Medien-Flows verwendet werden. In diesem Fall tritt eine Beeinflussung (z.B. durch die verfügbare maximale Bandbreite) der Medien-Flows durch die zur Firewall-Steuerung (und umgekehrt) verwendeten Flows auf.

- Realisierungen:
  Die innerhalb der IETF vorgeschlagenen Mechanismen [33] zur Unterstützung von IP-Telephony Szenarien basieren auf dieser Architektur.
  Das System KOMproxyd kann so eingesetzt werden, daß eine Interaktion nach dem oben dargestellten Muster möglich ist [30].
  Weitere Systeme dieser Art sind in [33] und [34] beschrieben.

- Anforderungen:
  Die hier beschriebene Architektur erfüllt die in Kapitel 5.3 bzw. in Kapitel 6.4 beschriebenen Anforderungen.

- Performance:
  Es gelten ebenfalls die zuvor beschriebenen (unter “Verteilten System I”) Eigenschaf-

ten. Es ist aber zu beachten, daß eine Beeinflussung der Medien-Flows durch die Firewall-Steuerung auftritt.

IV. Verteiltes System (III). In diesem Fall ist die Application Control innerhalb der Endsysteme angesiedelt. Abgesehen von diesem Detail ist die Architektur aber mit der Architektur “Verteiltes System II” identisch.

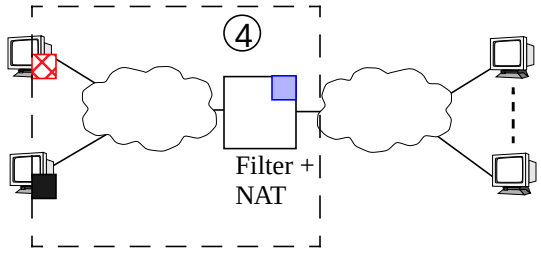

Abbildung 26: Verteiltes System (III)

Die Application Control lässt sich innerhalb der Endsysteme relativ einfach realisieren, da dort die notwendigen Flow-Spezifikationen einfach zu extrahieren sind. Verglichen mit dem zuvor beschriebenen Hybridsystem müssen die Flow-Spezifikation nicht aus der ablaufenden Kommunikation gewonnen werden, sondern können direkt über einen Funktionsaufruf von der Applikation an die Application Control übergeben werden. Allerdings benötigt jedes Endsystem die entsprechende Modifikation, um mit der Firewall Control interagieren zu können.

- Realisierungen:
  Da es relativ schwierig ist, eine bestehende Infrastruktur zu ändern (Eingriffe in jedem Endgerät sind notwendig), sind wenige Arbeiten, die diese Architektur verwenden, durchgeführt worden. In [40] ist eine Methode beschrieben, die es ermöglicht eine eventuell vorhandene RSVP-Architektur zur Realisierung der hier beschriebenen Firewall-Architektur zu verwenden. Die RSVP-API innerhalb des Endgerätes wird als Application Control und die RSVP-Nachrichten werden als Signalling Part verwendet. Dadurch kann der Aufwand zur Realisierung einer solchen Firewall-Architektur reduziert werden, falls eine RSVP-Architektur bereits existiert.

- Anforderungen:
  Die hier beschriebene Architektur erfüllt die in Kapitel 5.3 bzw. in Kapitel 6.4 beschriebenen Anforderungen.

- Performance:
  Es gelten ebenfalls die zuvor beschriebenen (unter “Verteilten System II”) Eigenschaften. Es ist aber zu beachten, daß eine Beeinflussung der Medien-Flows durch die Firewall-Steuerung auftritt.
  Zusätzlich muß jeweils nur eine Application Control pro Endsystem instanziiert werden. Dadurch ist der Aufwand geringer, als in den zuvor beschriebenen Architekturen. Dort werden mehrere Sessions von einem Proxy-System parallel verarbeitet.

7.3 Vergleich der verschiedenen Firewall-Architekturen

Die heute verwendeten und diskutierten Architekturen lassen sich jeweils im wesentlichen auf eine der vier zuvor beschriebenen Architekturen abbilden. Ein grundsätzlicher Vergleich dieser vier generellen Architekturen hinsichtlich ihrer Möglichkeiten ist notwendig, denn dies erlaubt es,

die Möglichkeiten einer bestimmter Firewall-Implementierung abzuschätzen, sobald diese anhand des Schemas klassifiziert ist.

Realisierbarkeit. Architektur I ist in der Praxis am einfachsten zu realisieren. Dies liegt zum einen daran, daß alle Firewall-Komponenten in einem Gerät realisiert sind. Dadurch ist keine Interaktion verschiedener (möglicherweise von verschiedenen Herstellern) Geräte notwendig. Zum anderen entspricht diese Architektur vorhandenen Firewall-Systemen, wodurch auf entsprechend viel bestehendes Wissen/Implementierungen zurückgegriffen werden kann.
Architektur II und Architektur III bieten den wesentlichen Vorteil, daß spezialisierte Komponenten für die Realisierung der Application Control verwendet werden können. Beispielsweise kann eine bereits vorhandene Infrastrukturkomponente (z.B. ein Gatekeeper) für die Realisierung der Application Control verwendet werden. Allerdings erfordert diese Architektur eine Interaktion verschiedener Komponenten. Soll eine Infrastrukturkomponente mit einem Filter interagieren, scheitert die Umsetzung oft an der Tatsache, daß die Komponenten (z.B. Gatekeeper und Filter) von verschiedenen Herstellern angeboten werden.
Architektur IV ist in der Praxis schwer umzusetzen, da alle Endgeräte an die Firewall angepaßt werden müssen. Da dies einen sehr hohen Aufwand darstellt, ist nicht damit zu rechnen, daß diese Architektur Verwendung findet. Da aber nicht nur für die Realisierung einer verteilten Firewall die Signalisierung der Flow-Spezifikationen notwendig ist (z.B. QoS), könnte diese Architektur interessant werden, wenn mehrere notwendige Signalisierungen in einem integrierten Ansatz zusammengefaßt werden.

Umsetzung der Anforderungen. Der wesentliche Nachteil der Architektur I besteht darin, daß die Anforderungen, die durch die Charakteristika von Multimedia-Applikationen entstehen, schlecht umgesetzt werden können. Mit Architektur II, III und IV können die gegebenen Anforderungen umgesetzt werden.

Performance. Es ist zu erkennen, daß der Aufwand der Bearbeitung der Signalisierung pro Recheneinheit durch entsprechenden Transportaufwand über das Netzwerk ersetzt werden kann.
Die verschiedenen Architekturen haben demnach eine möglicherweise erhebliche Auswirkung auf die Performance.

Sicherheit. Komplexe Systeme sind in der Regel unsicherer als einfach aufgebaute Systeme. Aus diesem Grund kann man prinzipiell annehmen, daß Architektur I die sicherste Variante darstellt. Bei Architektur II besteht eine klare Trennung zwischen Firewall und internem Netzwerk. Hier stellt die Aufteilung der Firewall auf verschiedene Komponenten eine zusätzliche Komplexität dar, die möglicherweise zu zusätzlichen Sicherheitsproblemen führt. Bei Architektur III und IV sind Teile (Application Control) innerhalb des internen Netzwerkes angeordnet. Der Bereich der Firewall ist bei diesen Architekturen nicht hart auf die Netzgrenze beschränkt. Dies könnte zu zusätzlichen Sicherheitsproblemen führen. Die Komplexität der Architektur III und IV ist ebenfalls höher als die der Architektur I.

Betrieb. Für den Betrieb ist es von entscheidender Bedeutung, in welchem Zuständigkeitsbereich sich die einzelnen Firewall Komponenten befinden. Bei Architektur I werden alle Konfigurationen in einer zentralen Komponente durchgeführt. Dies gestaltet die Abstimmung der Konfiguration der Application Control mit der Konfiguration der Firewall Control relativ einfach. Bei den Architekturen II und III gestaltet sich dies insbesondere dann schwierig, wenn die Application Control innerhalb einer Infrastrukturkomponente untergebracht ist. In einem solchen Fall wird die Infrastrukturkomponente, welche einem bestimmten Multimedia Szenario zugeordnet ist, in der Regel von einer anderen Gruppe verwaltet als die Filter/NAT Komponente. Gleiches gilt für die Architektur IV.

8. H.323 spezifische Architekturen

Im folgenden werden einige bestehende Firewall-Systeme, die das H.323- Protokoll unterstützen, dargestellt und diskutiert. Die verschiedenen Firewall-Systeme werden in das zuvor beschriebene Klassifizierungsschema eingeordnet, was eine Bewertung bzw. einen Vergleich der verschiedenen Systeme vereinfacht. In dieser Studie werden die Firewall-Systeme beschrieben, welche innerhalb des DFN-Videokonferenzszenarios als mögliche Lösung angesehen wurden bzw. angesehen werden. Eine vollständige Übersicht über alle zur Zeit verfügbaren Produkte bzw. Prototypen kann im Rahmen dieser Studie nicht gegeben werden.
Es werden nachfolgend verschiedene Vertreter aus der Gruppe der “Proxy-Systeme” (Architektur II und III) sowie einige Lösungen aus dem Bereich “Hybridsysteme” (Architektur I) dargestellt. Zur Zeit verwendete bzw. verfügbare Systeme sind Vertreter dieser Klassen, andere Systeme sind aber theoretisch möglich.

8.1 Proxy

Die zur Zeit verfügbaren Proxy-Lösungen entsprechen den zuvor beschriebene Architekturen II und III. Die einzelnen Implementierungen unterscheiden sich dabei in vielen wesentlichen Punkten. Nachfolgend ist die Funktionsweise bzw. Architektur von mehreren zur Zeit verfügbaren Systemen beschrieben.

8.1.1 OpenH323Proxy

Die in diesem Abschnitt beschriebene Softwarelösung basiert auf dem OpenH323-Protokollstack und ist in [35] beschrieben bzw. erhältlich.

Architektur. Der OpenH323Proxy stellt ein System dar, welches auf Architektur II beruht. Die Architektur, dargestellt in Abbildung 27, besitzt folgende spezielle Merkmale:

- Die Application Control ist in diesem System innerhalb einer H.323-Infrastrukturkomponente implementiert; der OpenH323Proxy ist eine Firewall-Komponente, welche im wesentlichen aus einem H.323-Gatekeeper besteht. Das System wurde im Rahmen dieser Studie in einem Laborversuch getestet.

- Zwischen dem OpenH323Proxy sowie der “Standard Firewall” (welche beispielsweise aus einem Stateful Filter sowie einer NAT-Komponente bestehen kann)

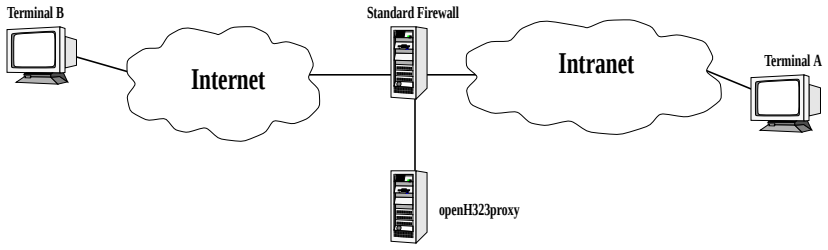

Abbildung 27: OpenH323Proxy Szenario

besteht keine Möglichkeit der Interaktion. Flow-Spezifikationen können vom OpenH323Proxy nicht an die Standard Firewall weitergegeben werden.
Dadurch ist es der Standard Firewall nicht möglich, seine Konfiguration an die Medien-Flows, die zwischen Terminal A und Terminal B fließen, dynamisch anzupassen. Dies betrifft die Konfiguration des Filters sowie der NAT-Komponente. Um diesem Problem zu begegnen, ist der OpenH323Proxy in der Lage, zusätzlich zu den Signalisierungs-Flows auch die Medien-Flows zu verarbeiten. Die Medien-Flows, die zwischen Terminal A und Terminal B ausgetauscht werden, werden über den OpenH323Proxy geleitet. Die Standard Firewall muß nun statisch so konfiguriert werden, daß die Pakete aller möglicherweise auftretenden Flows aus internem und externem Netz von und zu dem OpenH323Proxy weitergeleitet werden.

- Verwendet der OpenH323Proxy eine im Internet gültige IP-Adresse, wird durch die Umsetzung der Medien-Flows innerhalb des OpenH323Proxies auch die NAT-Funktion bereitgestellt.

- Hinsichtlich der Performance ist diese Architektur von Nachteil, da der OpenH323Proxy die Medienströme verarbeiten muß.

Call Routing. Das Firewall-System kann in die Kommunikation eingebunden werden, indem der OpenH323Proxy entsprechend seiner H.323-Gatekeeperfunktionalität in die Kommunikation eingebunden wird. Dadurch kann das Call Routing Modell “Transparent - Nutzung von Infrastrukturkomponenten”, welches in Kapitel 6.2.2 beschrieben ist, verwendet werden.

- **Transparent - Nutzung von Infrastrukturkomponenten (I):**
  Terminal A und Terminal B sind jeweils an dem OpenH323Proxy angemeldet. Der Ruf wird dann vollständig über diesen Gatekeeper abgewickelt.

- **Transparent - Nutzung von Infrastrukturkomponenten (II):**
  Terminal A ist am OpenH323Proxy-Gatekeeper angemeldet; Terminal B ist an einem eigenen Gatekeeper angemeldet. In diesem Fall wird bei einem Rufaufbau von Terminal A zu Terminal B ein *LRQ* von dem OpenH323Proxy an den Gatekeeper B geschickt. Der Ruf wird dann vollständig (Signalisierung und Medien) durch den OpenH323Proxy abgewickelt, der Gatekeeper B ist entsprechend seiner Policy eventuell ebenfalls in die Kommunikation eingebunden.

Konfiguration. Eine Anpassung des Firewall-Systems an verschiedene mögliche Szenarien ist nur in sehr beschränktem Umfang möglich.

- **Call Routing:**
  Das Call Routing kann über die Angabe einer *Neighbours* Sektion (innerhalb des Openh323proxy) festgelegt werden. Damit wird angegeben, welche benachbarten Gatekeeper für *LRQ*-Anfragen verwendet werden sollen. Dabei ist allerdings die Berücksichtigung eines Prefixes nicht möglich, *LRQ*-Anfragen werden sequentiell an die Gatekeeper in dieser Tabelle gesendet bis eine positive Antwort empfangen wird oder die Tabelle abgearbeitet ist.

- **Szenarien:**
  Eine Anpassung an verschiedene Szenarien ist nicht möglich. Die Anpassung erfolgt im wesentlichen über die Positionierung des Systems im verwendeten Szenario und dem damit beeinflußbaren Call Routing.

- **Sicherheit:**
  Es ist nicht möglich, hinsichtlich der Sicherheit die H.323-Protokollbearbeitung (siehe Kapitel 6.4) zu beeinflussen. Einzig die im folgenden erläuterten Einstellungen können vorgenommen werden.

Sicherheit. Folgende sicherheitsrelevanten Einstellungen können für den OpenH323Proxy verwendet werden:

- **Check IP:**
  Durch diese Option kann festgelegt werden, ob geprüft werden soll, daß die Endpunkte für Signalling- und Medien-Flows die gleichen IP-Adressen verwenden. Damit kann sichergestellt werden, daß Signalisierungs- und Medien-Flows an den gleichen Kommunikationsendpunkten terminieren (im obigen Beispiel bei Terminal A bzw. B). Hier können Probleme auftreten, falls beispielsweise im internen Netz weitere Gatekeeper verwendet werden, da dann eine Terminierung von Signalisierungs- und Medienströmen am gleichen Endpunkt eventuell gewünscht ist, aber verhindert wird.

- **Check Port Number:**
  Diese Option ermöglicht es eine Überprüfung der verwendeten Portnummern für die Medienströme durchzuführen. Es wird geprüft ob Portnummern größer 1023 für die Medien verwendet werden.

Kommunikationsbeispiel. Im folgenden Kommunikationsbeispiel wird das in Abbildung 27 dargestellte Szenario zugrunde gelegt. Zusätzlich wird angenommen, daß Terminal B seinerseits einen eigenen Gatekeeper verwendet (Gatekeeper B) um sich zu registrieren. Im Folgenden ist der Rufaufbau zwischen Terminal A und Terminal B dargestellt (vergl. Kapitel 6.2.2). Innerhalb des Beispieles kann angenommen werden, daß das interne Netzwerk einen privaten Adressraum verwendet.

- **Voraussetzung:**
  Der verwendete Gatekeeper B ist in der Lage über sog. RAS *LocationRequests* (LRQ) mit anderen Gatekeepern (z.B. dem OpenH323Proxy) zu kommunizieren. .

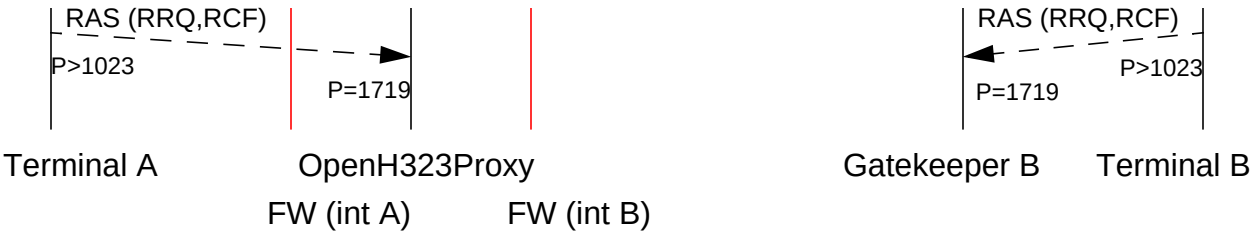

Abbildung 28: OpenH323Proxy Kommunikation (a)

- **RAS Registration, Admission, Status (UDP):**
  Nach dem Start der Terminals registrieren sich diese jeweils an dem ihnen zugewiesenen Gatekeeper (*RegistrationRequest RRQ* und *RegistrationConfirm RCF*)

- **RAS Registration, Admission, Status (UDP):**
  Terminal A meldet den Ruf bei seinem Gatekeeper (OpenH323Proxy) an (*AdmissionRequest ARQ*).

- **RAS Registration, Admission, Status (UDP):**
  Innerhalb der *ARQ*-Nachricht ist die Zieladresse von Terminal B enthalten. Da Terminal B nicht am OpenH323Proxy angemeldet ist wird entsprechend der *Neighbour Table* eine *LRQ*-Nachricht an Gatekeeper B übermittelt.

- **RAS Registration, Admission, Status (UDP):**
  Gatekeeper B antwortet mit der entsprechenden *LCF*-Nachricht.

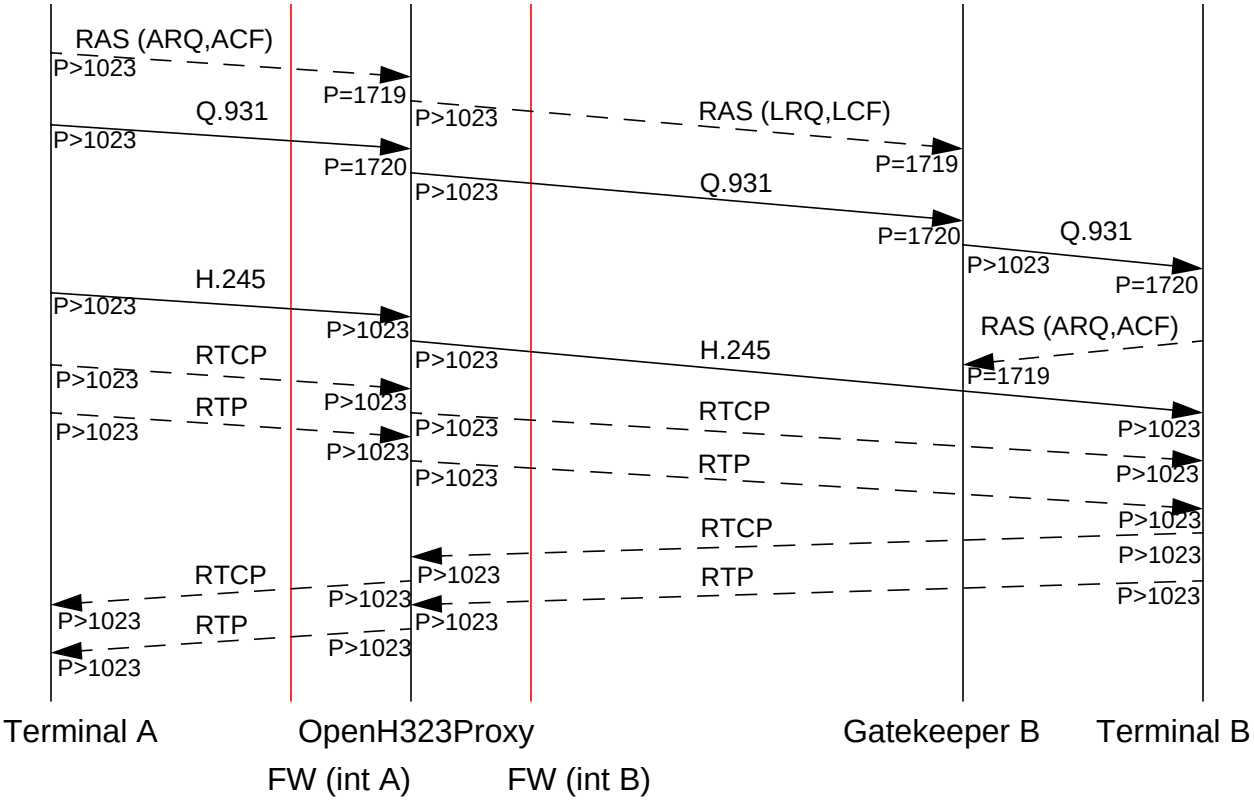

Abbildung 29: OpenH323Proxy Kommunikation (b)

- **Q.931 Call Signalling (TCP):**
  Bei einem Verbindungsaufbau kontaktiert das Terminal A nun entsprechend der in der *ACF*-Nachricht übermittelten Parameter den OpenH323Proxy.

- **Q.931 Call Signalling (TCP):**
  In dem hier angenommenen Beispiel wird die Q.931-Verbindung über beide Gate-

keeper geleitet. Dementsprechend kontaktiert der OpenH323Proxy den Gatekeeper B um die Call Signalling Verbindung aufzubauen.

- **Q.931 Call Signalling (TCP):**
  Gatekeeper B kontaktiert schließlich das Rufziel. Die Call Signalling Verbindung ist damit zwischen beiden Terminals vollständig aufgebaut.

- **RAS Registration, Admission, Status (UDP):**
  Terminal B meldet den Ruf bei seinem Gatekeeper an (*AdmissionRequest ARQ*).

- **H.245 Call Control (TCP):**
  Zwischen beiden Terminals werden nun die Parameter (z.B. Port-Nummern und IP-Adressen) für die folgende H.245 Verbindung ausgehandelt (innerhalb der Q.931 Verbindung).
  Terminal A kontaktiert den OpenH323Proxy unter Verwendung der zuvor ausgehandelten Parameter.

- **H.245 Call Control (TCP):**
  In dem hier dargestellten Beispiel wird davon ausgegangen, das Gatekeeper B nur die Q.931-Verbindung kontrolliert, nicht aber in die H.245-Verbindung mit einbezogen wird. Dementsprechend kontaktiert der OpenH323Proxy direkt Terminal B um die Call Control Verbindung vollständig aufzubauen.
  Die Call Control Nachrichten werden dazu verwendet, die Parameter der folgenden Medienströme auszuhandeln.
  Die Nachrichten zur Aushandlung der Medienströme werden durch den OpenH323Proxy so manipuliert (Port-Nummern und IP-Adressen), daß dieser ebenfalls die Kontrolle über die Medienströme erlangen kann.

- **RTP/RTCP Media und Mediacontrol (UDP):**
  Zwischen den beiden Terminals (vermittelt durch den OpenH323Proxy) werden mehrere Medienströme verwendet. Es sind mindestens 4 UDP-Ströme notwendig, um die Audiodaten zu transportieren (RTP- und der korrespondierende RTCP-Strom in jede Richtung). Zusätzliche Ströme werden verwendet, wenn zum Beispiel eine optionale Videoübertragung stattfindet.

Der Rufabbau erfolgt dann entsprechend den in Kapitel 2.4 beschriebenen Mechanismen. Für die verwendete Firewall ergibt sich folgender statischer Regelsatz[1]:

Table 2: Filtereinstellungen für OpenH323Proxy

Interface	From IP	From Port	To IP	To Port	Proto
A	INTERN	>1023	OpenH323Proxy	>1023	UDP
A	OpenH323Proxy	>1023	INTERN	>1023	UDP
A	INTERN	>1023	OpenH323Proxy	>1023	TCP

Table 2: Filtereinstellungen für OpenH323Proxy

Interface	From IP	From Port	To IP	To Port	Proto
A	OpenH323Proxy	>1023	INTERN	>1023	TCP
B	OpenH323Proxy	>1023	EXTERN	>1023	UDP
B	EXTERN	>1023	OpenH323Proxy	>1023	UDP
B	OpenH323Proxy	>1023	EXTERN	>1023	TCP
B	EXTERN	>1023	OpenH323Proxy	>1023	TCP

Bewertung . Generell ist es möglich, die beschriebene OpenH323Proxy-Komponente im Zusammenhang mit einer Standard Firewall zu verwenden. Da keine Interaktion zwischen OpenH323Proxy und der Standard Firewall vorgesehen ist, kann der OpenH323Proxy ohne größere Probleme in bestehende Firewall-Systeme integriert werden. Der Vorteil der einfachen Integration in bestehende Strukturen wird durch Nachteile hinsichtlich der Performance erkauft. Alle Medien-Flows müssen durch den OpenH323Proxy geleitet werden, was die Anzahl der parallel über die Firewall geführten Rufe beschränkt.
Die folgenden Beschränkungen existieren innerhalb der beschriebenen Lösung:

- **Gatekeeper innerhalb der Firewall:**
  Innerhalb des Firewall Systems wird eine vollständige Gatekeeper Implementierung verwendet. Eine klare Trennung zwischen Firewall Funktionalität und Gatekeeeper Funktionalität ist nicht vorhanden. Dadurch ist es beispielsweise nicht möglich den Gatekeeper durch eine andere Gatekeeper-Lösung zu ersetzen, ohne die gesamte Firewall zu verändern. Ein weiteres Problem stellen rein intern geführte Rufe dar. Bei allen intern geführten Rufen ist die Firewall beteiligt, da der Gatekeeper innerhalb der Firewall verwendet wird. Dies führt zu einer unnötigen Belastung der Firewall.

- **Performance:**
  Alle Medien-Flows werden durch den OpenH323Proxy geleitet. Dies führt zu erheblichen Performance Problemen bei der Bearbeitung paralleler Rufe durch den OpenH323Proxy. Die Anzahl der möglichen parallelen Gespräche ist begrenzt.

- **Sicherheit:**
  Möglichkeiten zur Einschränkung der verwendeten Ports sind nicht vorhanden. Dies führt, wie oben beschrieben, zu einer Filter-Tabelle innerhalb der Standard Firewall, die eine Vielzahl von Kommunikationsmöglichkeiten offen hält. Eine Einschränkung der weiterzuleitenden H.323-Nachrichten (insbesondere der RAS-Nachnachrichten) kann nicht durchgeführt werden. Es ist nicht möglich das Weiterleiten eventuell unerwünschter PDUs zu verhindern.

1. Unter der Annahme, das die Firewall nur aus einem einfachen Paketfilter besteht; sowie der Annahme, daß verschiedene Ziele im externen Netz angerufen werden können (andere Zonen).

- **Call Routing:**
  Der OpenH323Proxy kann nicht in hierarchisch organisierten Zonen eingesetzt werden. Wenn das System eine *LRQ*-Anfrage von einem benachbarten Gatekeeper bekommt, wird diese Anfrage direkt positiv oder negativ beantwortet. Das Weiterreichen der *LRQ*-Anfrage an einen dritten Gatekeeper ist nicht möglich. Dadurch läßt sich dieses System zur Zeit nicht in größere hierarchische Zonenstrukturen integrieren. Ein weiteres Problem stellt die Art der *Neighbour*-Tabellen dar. Eine Berücksichtigung eines Prefixes ist nicht möglich, *LRQ*-Anfragen werden sequentiell an die Gatekeeper in dieser Tabelle gesendet bis eine positive Antwort empfangen wird oder die Tabelle abgearbeitet ist.

- **Sonstiges:**
  Das System kann nicht an herstellerspezifische Implementierungen angepaßt werden (siehe Kapitel 6.2.3). Verwendet beispielsweise ein Terminal das Feld *CalledPartyNumber* innerhalb der Q.931-Setup-Nachricht anstelle des Feldes *DestinationAddress* innerhalb des *UserUserIE*-Feld kann ein Ruf nicht über den OpenH323Proxy aufgebaut werden.

8.1.2 KOMproxyd

Die Software KOMproxyd basiert ebenfalls auf dem OpenH323-Protokollstack und ist in [30] beschrieben, bzw. erhältlich.

Architektur. KOMproxyd stellt ein “Baukastensystem” bereit mit dessen Hilfe verschiedene (z.B die in Kapitel 7.2 dargestellten) verteilten Firewall-Architekturen erstellt werden können. Im folgenden wird der Fall betrachtet, in welchem KOMproxyd verwendet wird um ein System entsprechend Architektur II zu erstellen.

- Das KOMproxyd unterstützt verschiedene Multimedia-Applikationen, unter anderem auch H.323 basierende Multimedia-Applikationen. Im folgenden wird nur auf H.323-Applikationen eingegangen. Das System wurde im Rahmen dieser Studie in einem Laborversuch getestet.

- KOMproxyd kann in verschiedenen Operationsmodi betrieben werden. Ein Operationsmodus ermöglicht eine Interaktion zwischen dem KOMproxyd und der verwendeten Firewall..

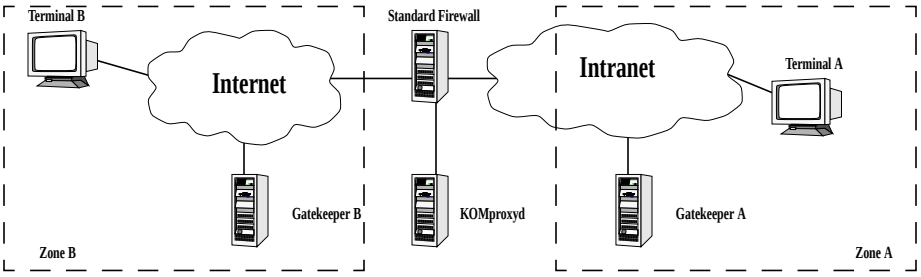

Abbildung 30: KOMproxyd Szenario

In diesem Fall werden die Spezifikationen der verwendeten Flows von dem Proxy an

die Firewall übermittelt. Eine entsprechend angepaßte Schnittstelle innerhalb der Firewall ist dazu notwendig. Ein anderer Operationsmodus ermöglicht es auf den Austausch von Flow-Spezifikationen zwischen Proxy und Firewall zu verzichten. In diesem Fall muß die Filtertabelle der Firewall entsprechend statisch konfiguriert werdenZusätzlich zu den oben beschriebenen Entscheidungsmöglichkeiten kann jeweils festgelegt werden, welche Flows durch den Proxy bearbeitet werden sollen. Es kann festgelegt werden, ob zusätzlich zu den Signalisierungs-Flows auch die Medien-Flows durch das Proxy-Element geleitet werden sollen. Nachfolgend wird folgender Operationsmodus betrachtet: Keine Interaktion zwischen Proxy und Standard-Firewall; Alle Medien-Flows werden durch das Proxy-Element geleitet. Diese Betrachtung ermöglicht es, das System KOMproxyd mit dem zuvor beschriebenen System OpenH323Proxy zu vergleichen.

- Verwendet der Proxy eine im Internet gültige IP-Adresse, wird durch die Umsetzung der Medien-Flows innerhalb des Proxies auch die NAT-Funktion bereitgestellt.

- Hinsichtlich der Performance ist der beschriebene Operationsmodus von Nachteil, da der Proxy die Medienströme verarbeiten muß. Dieser Nachteil kann ausgeglichen werden, indem das System so betrieben wird das die Medien-Flows den Proxy nicht durchlaufen und eine Kommunikation zwischen der Standard Firewall und dem Proxy stattfindet.

- Innerhalb des Proxies werden zwei verschiedene Module für die Bearbeitung der H.323-Kommunikation verwendet. Das sogenannte RAS-Modul wird verwendet um die RAS-Nachrichten, die über das Firewall-System versendet werden, zu bearbeiten. Das sogenannte H.323-Modul ist dafür verantwortlich die restliche H.323-Kommunikation zu bearbeiten. Innerhalb des KOMproxyd Systems wird eine Instanz des RAS-Moduls verwendet. Für jedes Gespräch, das über die Firewall geführt wird, wird eine Instanz des H.323-Moduls verwendet.

Call Routing. Die Firewall kann in die H.323-Kommunikation eingebunden werden, indem die verschiedenen Call Routing Modelle verwendet werden, die das KOMproxyd System unterstützt. Es können die folgenden Modelle (beschrieben in Kapitel 6.2.2) verwendet werden:

- **Explizit - Modifikation von H.323-Komponenten**
  Terminal A ist nicht an einem Gatekeeper angemeldet. Terminal A besitzt die Möglichkeit, ein Gateway für ausgehende Rufe zu spezifizieren (z.B. Terminal A ist ein Microsoft Netmeeting Terminal). Wird als Gateway der Proxy spezifiziert, so wird der H.323 Proxy in die Kommunikation zwischen Terminal A und Terminal B einbezogen. Der Proxy kann nun so konfiguriert werden, daß er anhand der übermittelten Zielrufnummer die Ziel-IP-Adresse der Kommunikation ermitteln kann (siehe Kapitel 6.2.2).

- **Transparent - Firewall Redirect**
  Besitzt die Standard-Firewall die Möglichkeit, eine TCP-Verbindung “Umzulenken”, so ist es möglich alle Q.931-Verbindungen, die über die Firewall aufgebaut werden, an den Proxy umzuleiten. Auf diese Weise kann der Proxy in die Kommunikation einbezogen werden. Der Proxy kann nun so konfiguriert werden, das er anhand der übermit-

telten Zielrufnummer die Ziel-IP-Adresse der Kommunikation ermitteln kann (siehe Kapitel 6.2.2).

- **Transparent - Nutzung der Gatekeeper-Gatekeeper Kommunikation**
  Terminal A ist an Gatekeeper A angemeldet; Terminal B ist an Gatekeeper B angemeldet. Gatekeeper A und B können so konfiguriert werden, das bei einem Ruf in die jeweils andere Zone der Proxy innerhalb der Firewall in die RAS Kommunikation einbezogen wird (durch entsprechende Anpassung der Neighbour Tabellen der Gatekeeper). Der Proxy kann sich durch Veränderung der RAS Meldungen in die H.323 Kommunikation einschalten (siehe Kapitel 6.2.2).

Konfiguration. Eine Konfiguration des Systems ist in folgenden Punkten möglich:

- **Call Routing:**
  Wie beschrieben können verschiedene Call Routing Modelle verwendet werden. Dazu müssen in der Konfiguration des Proxies das H.323- sowie das RAS-Modul konfiguriert werden. Unabhängig von dem verwendeten Call Routing Modell muß bei einem am Proxy eingehenden Ruf das Ziel des Rufes bestimmt werden. Es kann in der Konfiguration festgelegt werden, welche Mechanismen (und in welcher Reihenfolge) verwendet werden sollen um das Ziel zu bestimmen.
  Mögliche Verfahren sind:
  FW: Die Firewall wird nach der Ziel-IP-Adresse für die Quell-IP-Adresse der eingehenden Q.931-Verbindung gefragt. Dieses Verfahren wird verwendet wenn “Firewall Redirect” als Call Routing verwendet wird.
  PDU: Die Ziel-IP-Adresse wird anhand der innerhalb der *Q.931-Setup* Nachricht enthaltenen E.164 Nummer bestimmt. Dazu muß innerhalb des Proxies in der Konfiguration festgelegt sein, welche E.164-Nummer welcher IP-Adresse zugeordnet wird. Diese Konfiguration innerhalb des Proxies wird entweder statisch bei Systemstart festgelegt, oder aber sie wird dynamisch aus der Analyse der *LRQ/LCF* Nachrichten erstellt. Dieses Verfahren wird verwendet wenn “Firewall Redirect”, “Modifikation von H.323-Komponenten” oder “Nutzung der Gatekeeper-Gatekeeper Kommunikation” als Call Routing verwendet wird.
  CONF: Eine bei Systemstart in der Konfiguration festgelegte IP-Zieladresse wird für die Quell-IP-Adresse der eingehenden Q.931-Verbindung verwendet. Dieses Verfahren wird verwendet wenn “Firewall Redirect”, “Modifikation von H.323-Komponenten” oder “Nutzung der Gatekeeper-Gatekeeper Kommunikation” als Call Routing verwendet wird.

- **Szenarien:**
  Eine Anpassung an verschiedene Szenarien ist über die Auswahl eines geeigneten Call Routing Modells möglich.

- **Sicherheit:**
  Es ist möglich, hinsichtlich der Sicherheit die H.323 Protokollbearbeitung (siehe Kapitel 6.4) zu beeinflussen. Die im folgenden erläuterten Einstellungen können vorgenommen werden.

Sicherheit. Folgende sicherheitsrelevanten Einstellungen können für den OpenH323Proxy verwendet werden:

- **Port Einschränkung:**
  Es kann festgelegt werden, welche Portnummern innerhalb des Systems verwendet werden (MIN und MAX). Als Standard-Einstellung werden durch den Proxy nur Portnummern zwischen 10000 und 20000 verwendet.

- **RAS-Modul:**
  Das RAS-Modul kann so konfiguriert werden, daß nur bestimmte RAS-Meldungen das Proxy-System passieren können (Option *ras_0_ALLOW_XXX*). Es kann dadurch beispielsweise verhindert werden, daß *RRQ*-Nachrichten von außerhalb an den internen Gatekeeper gelangen. Es ist möglich die RAS-Kommunikation zwischen internem und externem Netzwerk auf den Austausch von *LRQ/LCF/LRJ*-Nachrichten zu beschränken.

- **H.323-Modul:**
  Das H.323-Modul kann so beeinflußt werden, daß bestimmte Nachrichten verworfen werden (z.B. Q.931-Information Nachrichten über die Option *h323_0_ALLOW_Q931_information*). Zusätzlich kann angegeben werden, welche Medienströme über die Firewall ermöglicht werden sollen (audio/video/t.120 *h323_0_ALLOW_CHANNEL_XXX*). Es ist möglich die Verwendung bestimmter Medienströme zu unterbinden.

Kommunikationsbeispiel. Im folgenden Kommunikationsbeispiel wird das in Abbildung 30 dargestellte Szenario zugrunde gelegt. Im Folgenden ist der Rufaufbau zwischen Terminal A und Terminal B dargestellt (vergl. Kapitel 6.2.2). Innerhalb des Beispieles kann angenommen werden, daß das interne Netzwerk einen privaten Adressraum verwendet.

- **Voraussetzung:**
  Die verwendeten Gatekeeper A und B sind in der Lage über sog. RAS *LocationRequests* (*LRQ*) mit anderen Gatekeepern zu kommunizieren.

- **RAS Registration, Admission, Status (UDP):**
  Nach dem Start der Terminals registrieren sich diese jeweils an dem ihnen zugewiesenen Gatekeeper (*RegistrationRequest RRQ* und *RegistrationConfirm RCF*).

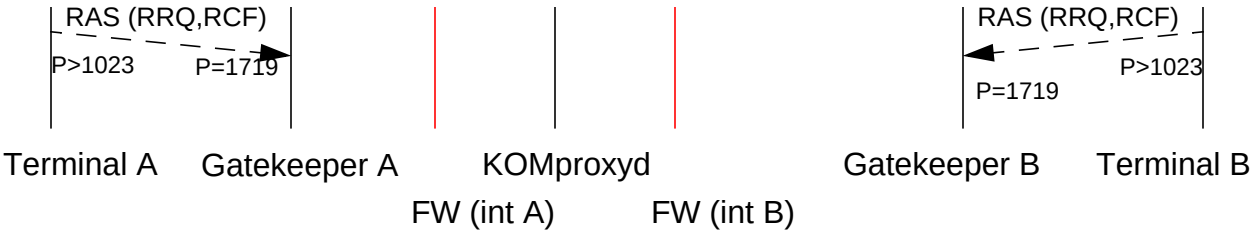

Abbildung 31: KOMproxyd Kommunikation (a)

- **RAS Registration, Admission, Status (UDP):**
  Terminal A meldet den Ruf bei seinem Gatekeeper an (*AdmissionRequest ARQ*).

- **RAS Registration, Admission, Status (UDP):**
  Innerhalb der *ARQ*-Nachricht ist die Zieladresse von Terminal B enthalten. Da Terminal B nicht am Gatekeeper A angemeldet ist wird entsprechend der *Neighbour Table* eine *LRQ*-Nachricht an den Proxy übermittelt.

- **RAS Registration, Admission, Status (UDP):**
  Der Proxy Modifiziert die empfangene *LRQ*-Nachricht und schickt sie entsprechend seiner *Neighbour Table* an Gatekeeper B weiter.

- **RAS Registration, Admission, Status (UDP):**
  Gatekeeper B antwortet mit der entsprechenden *LCF*-Nachricht

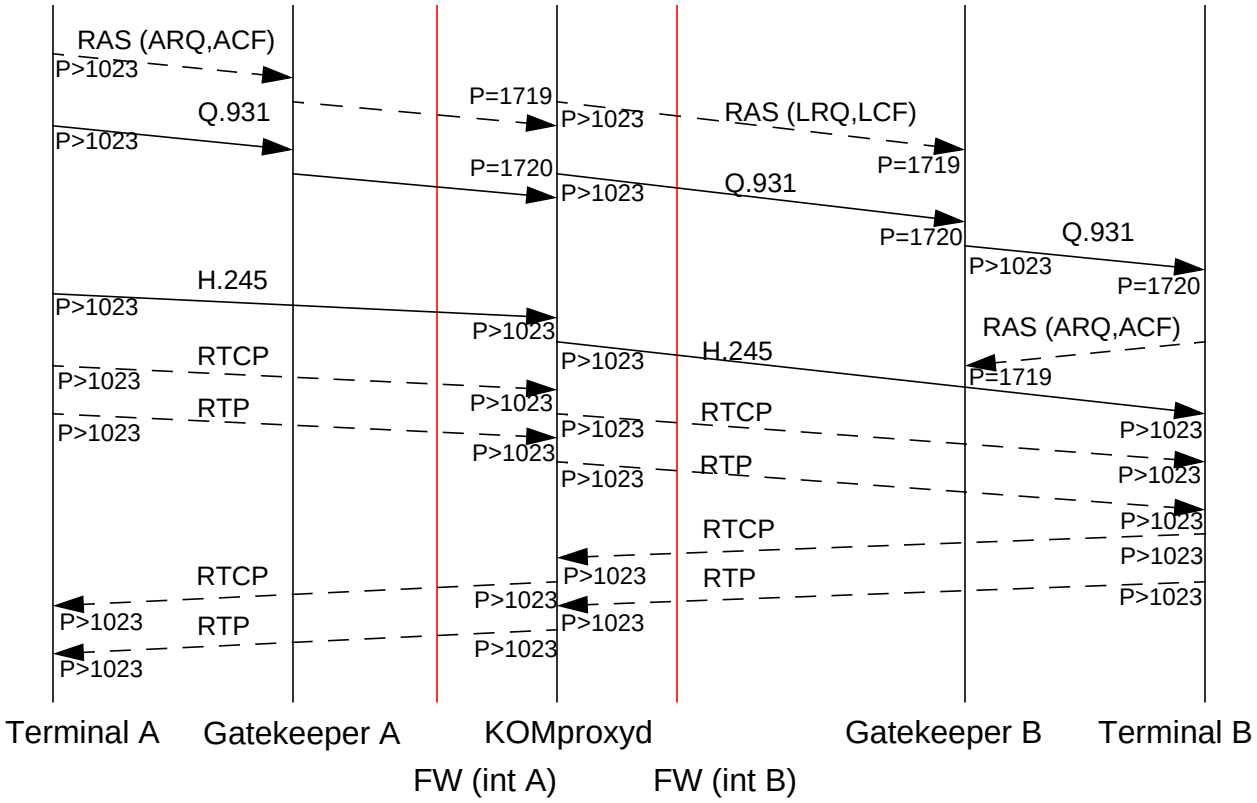

Abbildung 32: KOMproxyd Kommunikation (b)

- **RAS Registration, Admission, Status (UDP):**
  Die in der *LCF*-Nachricht enthaltenen Informationen werden innerhalb des Proxies (RAS-Modul) dazu verwendet, die enthaltene E.164-Nummer sowie die zugehörige IP-Adresse des Terminals in der Konfiguration des Proxies abzulegen. Danach wird die *LCF*-Nachricht modifiziert und an Gatekeeper A weitergeleitet.

- **Q.931 Call Signalling (TCP):**
  Bei einem Verbindungsaufbau kontaktiert das Terminal A nun entsprechend der in der *ACF*-Nachricht übermittelten Parameter den Gatekeeper A.

- **Q.931 Call Signalling (TCP):**
  Gatekeeper A kontaktiert daraufhin den Proxy.

- **Q.931 Call Signalling (TCP):**
  In dem hier angenommenen Beispiel wird die Q.931-Verbindung über beide Gatekeeper geleitet. Dementsprechend kontaktiert der Proxy den Gatekeeper B um die Call Signalling Verbindung aufzubauen (Diese Zieladresse wurde zuvor innerhalb der *LCF*-Nachricht übermittelt).

- **Q.931 Call Signalling (TCP):**
  Gatekeeper B kontaktiert schließlich das Rufziel. Die Call Signalling Verbindung ist damit zwischen beiden Terminals vollständig aufgebaut.

- **RAS Registration, Admission, Status (UDP):**
  Terminal B meldet den Ruf bei seinem Gatekeeper an (*AdmissionRequest ARQ*).

- **H.245 Call Control (TCP):**
  Zwischen beiden Terminals werden nun die Parameter (z.B. Port-Nummern und IP-Adressen) für die folgende H.245 Verbindung ausgehandelt (Innerhalb der Q.931 Verbindung).
  Terminal A kontaktiert den Proxy unter Verwendung der zuvor ausgehandelten Parameter.

- **H.245 Call Control (TCP):**
  In dem hier dargestellten Beispiel wird davon ausgegangen, das Gatekeeper A und B nur die Q.931-Verbindung kontrollieren, nicht aber in die H.245-Verbindung mit einbezogen werden. Dementsprechend kontaktiert der Proxy direkt Terminal B um die Call Control Verbindung vollständig aufzubauen.
  Die Call Control Nachrichten werden dazu verwendet, die Parameter der folgenden Medienströme auszuhandeln.
  Die Nachrichten zur Aushandlung der Medienströme werden von dem Proxy dementsprechend manipuliert (Port-Nummern und IP-Adressen), daß dieser ebenfalls die Kontrolle über die Medienströme erlangen kann.

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

Der Rufabbau erfolgt dann entsprechend den in Kapitel 2.4 beschriebenen Mechanismen. Für die verwendete Firewall ergibt sich folgender statischer Regelsatz[1]:

Table 3: Filtereinstellungen für KOMproxyd

Interface	From IP	From Port	To IP	To Port	Proto
A	INTERN	>1023	KOMproxyd	MIN<>MAX	UDP
A	KOMproxyd	MIN<>MAX	INTERN	>1023	UDP
A	INTERN	>1023	KOMproxyd	MIN<>MAX	TCP
A	KOMproxyd	MIN<>MAX	INTERN	>1023	TCP
B	KOMproxyd	MIN<>MAX	EXTERN	>1023	UDP
B	EXTERN	>1023	KOMproxyd	MIN<>MAX	UDP
B	KOMproxyd	MIN<>MAX	EXTERN	>1023	TCP
B	EXTERN	>1023	KOMproxyd	MIN<>MAX	TCP

Bewertung. Es ist möglich, das beschriebene KOMproxyd-System im Zusammenhang mit einer Standard Firewall zu verwenden. In dem hier beschriebenen Operationsmodus ist keine Interaktion zwischen Firewall und Proxy vorgesehen; es ist deshalb ein Vergleich zwischen KOMproxyd und OpenH323Proxy möglich. Das KOMproxyd System unterscheidet sich im wesentlichen von dem zuvor beschriebenen OpenH323Proxy darin, das die aufgezeigten Beschränkungen dieses Systems hier aufgehoben sind:

- **Gatekeeper innerhalb der Firewall:**
  Innerhalb des Firewall Systems wird keine vollständige Gatekeeper Implementierung verwendet.

- **Performance:**
  Alle Medien-Flows können bei verwendeter Interaktion von Firewall und Proxy direkt zwischen den Endpunkten ausgetauscht werden

- **Sicherheit:**
  Möglichkeiten zur Einschränkung der verwendeten Ports sind vorhanden. Eine Einschränkung der weiterzuleitenden H.323-Nachrichten kann vorgenommen werden.

- **Call Routing:**
  Das System kann in hierarchisch organisierten Zonen eingesetzt werden. Eine Berücksichtigung eines Prefixes in den Neighbour Tabellen ist möglich.

1. Unter der Annahme, das die Firewall nur aus einem einfachen Paketfilter besteht; sowie der Annahme, daß verschiedene Ziele im externen Netz angerufen werden können (Andere Zonen).

- **Sonstiges:**
  Das System beachtet automatisch herstellerspezifische Implementierungen (siehe Kapitel 6.2.3).

8.1.3 Weitere Proxy-Systeme

PhonePatch. Das PhonePatch Produkt [36] adressiert Netmeeting-Szenarien und entspricht der Firewall-Architektur II. PhonePatch wird parallel zu einer existierenden Firewall verwendet und ist verantwortlich für die Behandlung des H.323 und ILS-Verkehrs. Eine Interaktion zwischen Firewall und Phonepatch findet nicht statt.
Alle Internet Location Service (ILS) Anfragen werden durch das PhonePatch Programm geleitet. Dies ermöglicht es, die IP-Zieladressen herauszufiltern und zu verändern, die mit Hilfe des ILS-Protokolls übermittelt werden. Dadurch können die Rufe zum PhonePatch-Host umgeleitet werden (Call Routing). 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 (Call Routing: Explizit - Modifikation von H.323-Komponenten).
Dieser Ansatz adressiert 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.

Cisco Multimedia Conference Manager. Der Cisco Multimedia Conference Manager (MCM) [37] 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 zu (oder hinter) einer Firewall installiert werden.
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.
Der Proxy innerhalb des MCM ist 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.

8.2 Hybridsysteme

Kommerzielle Firewall-Systeme sind zur Zeit in den meisten Fällen als Hybridsysteme ausgeführt, die der Architektur I entsprechen. Wie bereits in den vorhergegangenen Kapiteln dargelegt, eignet sich eine solche Architektur nicht optimal um Multimedia-Applikationen zu unterstützen. Dies wird im Folgenden bei der Betrachtung einzelner ausgewählter Produkte die zur Zeit am Markt erhältlich sind ebenfalls deutlich.

8.2.1 Cisco PIX

Die in diesem Abschnitt beschriebene Firewall Lösung der Firma Cisco ist genauer in [38] beschrieben.

Architektur. Bei der von Cisco angebotenen Firewall Lösung “PIX” handelt es sich um eine spezialisierte Hardware mit entsprechender Software. Im Gegensatz zu vielen anderen Firewall Produkten basiert die Firewall damit nicht auf Standard Rechner Hardware und dem dazugehörigen Betriebssystem. Die Cisco PIX integriert einen Stateful Filter, eine NAT-Komponente sowie Proxies für verschiedene Protokolle. Die Cisco PIX unterstützt einen Mechanismus Namens “Cut-Through-Proxy”. Der Verbindungsaufbau wird auf der Höhe der Applikationsschicht abgewikkelt, danach wird die Verbindung durch den Stateful Filter überwacht. Dieses System entspricht damit der in Kapitel 7.2 beschriebenen Architektur I.

- Die Cisco PIX unterstützt verschiedene Applikationen und Protokolle, darunter auch das H.323-Protokoll. Im folgenden werden nur die H.323 relevanten Aspekte dieser Firewall betrachtet. Ein wesentliches Problem stellt die Dokumentation der Firewall dar, da dort sehr wenig über die genaue Funktionsweise der H.323 spezifischen Elemente zu finden ist. Das System wurde aus diesem Grund in einem Laborversuch getestet.

- Damit das H.323-Protokoll durch die PIX-Firewall unterstützt wird muß der entsprechende applikationsspezifische Proxy innerhalb der PIX aktiviert werden.

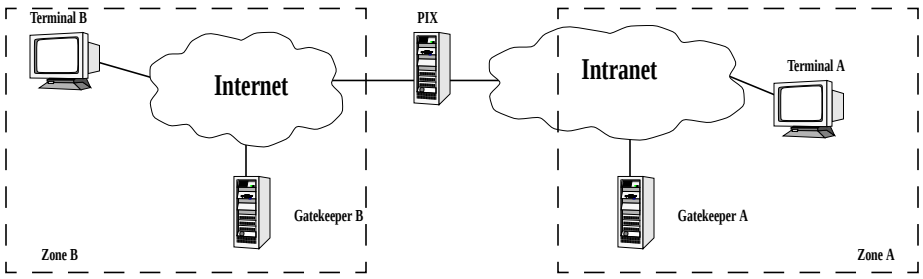

Abbildung 33: PIX Szenario

Dies geschieht durch das Hinzufügen eines sogenannten “H.323-Fixup”.

Call Routing. Die Firewall kann dann in die H.323-Kommunikation eingebunden werden, indem das Call Routing Modell “Firewall Redirect” (beschrieben in Kapitel 6.2.2) verwendet wird:

- **Transparent - Firewall Redirect**
  Alle H.323 spezifischen Flows, die über die PIX laufen werden “Umgelenkt”, so daß sie das H.323-Fixup (den H.323 spezifischen Proxy innerhalb der PIX) durchlaufen. Auf diese Weise kann die PIX in die Kommunikation einbezogen werden. Es ist zu beachten, daß das rufende Terminal in diesem Fall die IP-Adresse des Zieles verwenden muß. Eine Zuordnung von E.164 Nummern zu IP-Adressen findet innerhalb der Firewall nicht statt. Daraus folgt, daß in NAT-Umgebungen keine Rufe abgesetzt werden können.

Konfiguration.

- **Call Routing:**
  Wie beschrieben kann das Call Routing Modell “Firewall Redirect” verwendet werden. Durch das Kommando “fixup protocol h323 1720” wird das “Umlenken” der initialen Q.931-Verbindungen an den internen H.323-Proxy veranlaßt. Dadurch wird ebenfalls veranlaßt die RAS-Meldungen an den H.323-Proxy umzulenken.

- **Szenarien:**
  Eine Anpassung der Firewall an verschiedene Szenarien ist nicht möglich.

- **Sicherheit:**
  Hinsichtlich spezifischer Sicherheitseinstellungen für das H.323-Protokoll stehen keine Konfigurationsmöglichkeiten zur Verfügung.

Sicherheit. Es stehen die Sicherheitsmechanismen zur Verfügung, die allen durch die Firewall unterstützen Protokollen zur Verfügung stehen. Der H.323-Fixup kann in seinem Verhalten aber nicht beeinflußt werden. Dadurch ist es nur möglich, entweder die H.323-Kommunikation über die Firewall zu ermöglichen oder zu unterbinden. Das Unterdrücken bestimmter H.323 spezifischer Merkmale ist nicht möglich.

Kommunikationsbeispiel . Im folgenden Kommunikationsbeispiel wird das in Abbildung 33 dargestellte Szenario zugrunde gelegt. Im Folgenden ist der Rufaufbau zwischen Terminal A und Terminal B dargestellt (vergl. Kapitel 6.2.2).

- **Voraussetzung:**
  Die verwendeten Gatekeeper A und B sind in der Lage über sog. RAS *LocationRequests* (*LRQ*) mit anderen Gatekeepern zu kommunizieren.

Abbildung 34: PIX Kommunikation (a)

- **RAS Registration, Admission, Status (UDP):**
  Nach dem Start der Terminals registrieren sich diese sich jeweils an dem ihnen zugewiesenen Gatekeeper (*RegistrationRequest RRQ* und *RegistrationConfirm RCF*)

- **RAS Registration, Admission, Status (UDP):**
  Terminal A meldet den Ruf bei seinem Gatekeeper an (*AdmissionRequest ARQ*).

- **RAS Registration, Admission, Status (UDP):**
  Innerhalb der *ARQ*-Nachricht ist die Zieladresse von Terminal B enthalten. Da Terminal B nicht am Gatekeeper A angemeldet ist wird entsprechend der *Neighbour Table* eine *LRQ*-Nachricht an Gatekeeper B übermittelt.

- **RAS Registration, Admission, Status (UDP):**
  Gatekeeper B antwortet mit der entsprechenden *LCF*-Nachricht

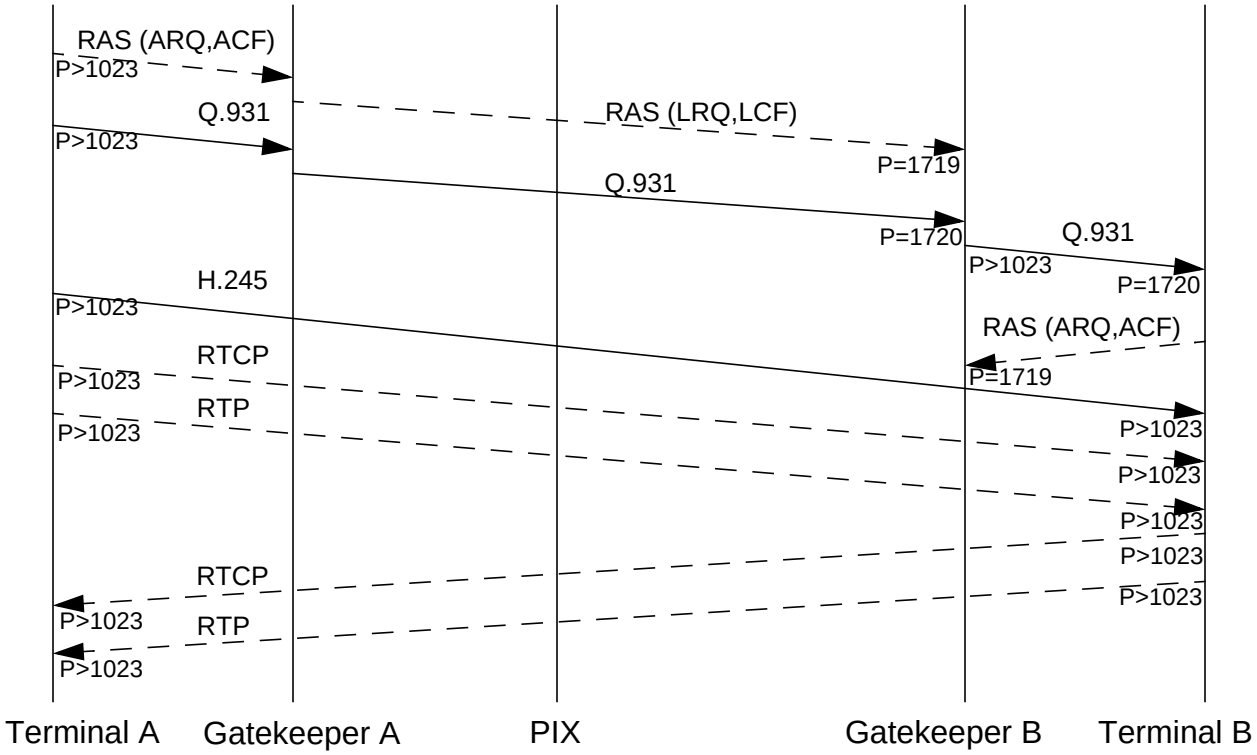

Abbildung 35: PIX Kommunikation (b)

- **Q.931 Call Signalling (TCP):**
  Bei einem Verbindungsaufbau kontaktiert das Terminal A nun entsprechend der in der *ACF*-Nachricht übermittelten Parameter den Gatekeeper A.

- **Q.931 Call Signalling (TCP):**
  Gatekeeper A kontaktiert daraufhin den Gatekeeper B.

- **Q.931 Call Signalling (TCP):**
  Gatekeeper B kontaktiert schließlich das Rufziel. Die Call Signalling Verbindung ist damit zwischen beiden Terminals vollständig aufgebaut.

- **RAS Registration, Admission, Status (UDP):**
  Terminal B meldet den Ruf bei seinem Gatekeeper an (*AdmissionRequest ARQ*).

- **H.245 Call Control (TCP):**
  Zwischen beiden Terminals werden nun die Parameter (z.B. Port-Nummern und IP-Adressen) für die folgende H.245 Verbindung ausgehandelt (Innerhalb der Q.931 Verbindung). Terminal A kontaktiert Terminal B unter Verwendung der zuvor ausgehandelten Parameter. Die Nachrichten zur Aushandlung der Medienströme werden von der PIX analysiert um die folgenden Medienströme freizuschalten

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

Der Rufabbau erfolgt dann entsprechend den in Kapitel Kapitel 2.4 beschriebenen Mechanismen.

Bewertung. Das PIX-System kann zur Zeit nur in einem Umfeld ohne NAT eingesetzt werden. Da das Call Routing nicht H.323 spezifisch ist, ist es schwer möglich dieses System sinnvoll in einem hierarchischen H.323-Umfeld einzusetzen. Es ergeben sich die folgenden Beschränkungen:

- **Performance:**
  Alle Medien-Flows werden durch die PIX geleitet. Dies führt zu Performance Problemen bei der Bearbeitung paralleler Rufe durch das System. Die Anzahl der möglichen parallelen Gespräche ist begrenzt.

- **Sicherheit:**
  Eine Einschränkung der weiterzuleitenden H.323-Nachrichten (insbesondere der RAS-Nachnachrichten) kann nicht durchgeführt werden. Es ist nicht möglich, das Weiterleiten eventuell unerwünschter PDUs zu verhindern.

- **Call Routing:**
  Das verwendete Call Routing ist nicht H.323 spezifisch. Dadurch kann das System nicht in hierarchisch organisierten Zonen eingesetzt werden. Diese Tatsache verhindert den Einsatz des Systems in NAT-Umgebungen.

- **Sonstiges:**
  Wenn der Fixup für H.323 aktiviert ist, sind RAS Verbindungen über UDP-Port 1719 auch ohne Freigabe über eine Access Control Liste möglich. Dieses Verhalten ist bei Cisco als Fehler dokumentiert und bekannt.

8.2.2 Weitere Hybridsysteme

Checkpoint Firewall-1. Die Architektur des Firewall-1 [39] Produktes entspricht im wesentlichen der Architektur I wie sie in Kapitel 7.2 dargestellt ist. Die Firewall unterstützt verschiedene Protokolle, darunter auch das H.323-Protokoll. Für jedes zu unterstützende Protokoll ist ein entsprechender Parser innerhalb der Firewall zu aktivieren. Als Call Routing Modell wird wie auch bei der Cisco PIX das Modell “Firewall Redirect” verwendet.
Der H.323 spezifische Parser wurde untersucht, es ergaben sich folgende Ergebnisse:

- 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 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 kein H.323 spezifisches Call Routing verwendet wird.

- 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 H.323 Parser analysiert zuerst die Q.931-Verbindung. 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 den selben 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 Ziel-Adressen beachten.

Die im betrachteten Beispiel verwendeten Parser-Komponenten weisen eine sehr statische Struktur auf. Es konnte in unseren Experimenten nur ein Basisszenario unterstützt werden, der direkte Ruf. Aus diesem Grund ist es nicht möglich dieses System in einem hierarchischen H.323-Umfeld einzusetzen.

9. Mögliche Lösungen im DFN-Umfeld

Im folgenden Abschnitt wird das im DFN verwendete H.323-Szenario, soweit für das Verständnis dieser Studie nötig, beschrieben. Danach werden verschiedene Firewall-Lösungen diskutiert, die innerhalb des DFN-Szenarios verwendet werden können. Der letzte Abschnitt dieses Kapitels beschreibt die Verwendung des KOMproxyd Systems innerhalb des DFN-Szenario. Dieses System wurde im DFN-H.323-Szenario getestet und stellt eine verwendbare Firewall Lösung dar.

9.1 Einsatzszenario

Wie bereits in den vorhergehenden Kapiteln dargelegt, hängt die Auswahl einer geeigneten H.323-Firewall-Lösung stark von dem vorgegebenen H.323-Einsatzszenario ab. Aus diesem Grund ist es nötig zuerst das H.323-Szenario innerhalb des DFN - dies umfaßt die verwendete H.323-Architektur sowie bereits existierende Firewall Systeme - zu beschreiben, um danach die dafür geeignete Firewall Lösung festlegen zu können.

H.323-Struktur innerhalb des DFN. Ziel des DFN ist es, einen Videokonferenzdienst basierend auf dem H.323-Protokoll anzubieten. Dazu werden vom DFN die notwendigen MCU-Einheiten (siehe Kapitel 2.4), sowie eine für das Call Routing notwendige Gatekeeper-Struktur bereitgestellt. Die bereitgestellte Gatekeeper-Struktur ist in Abbildung 36 dargestellt.

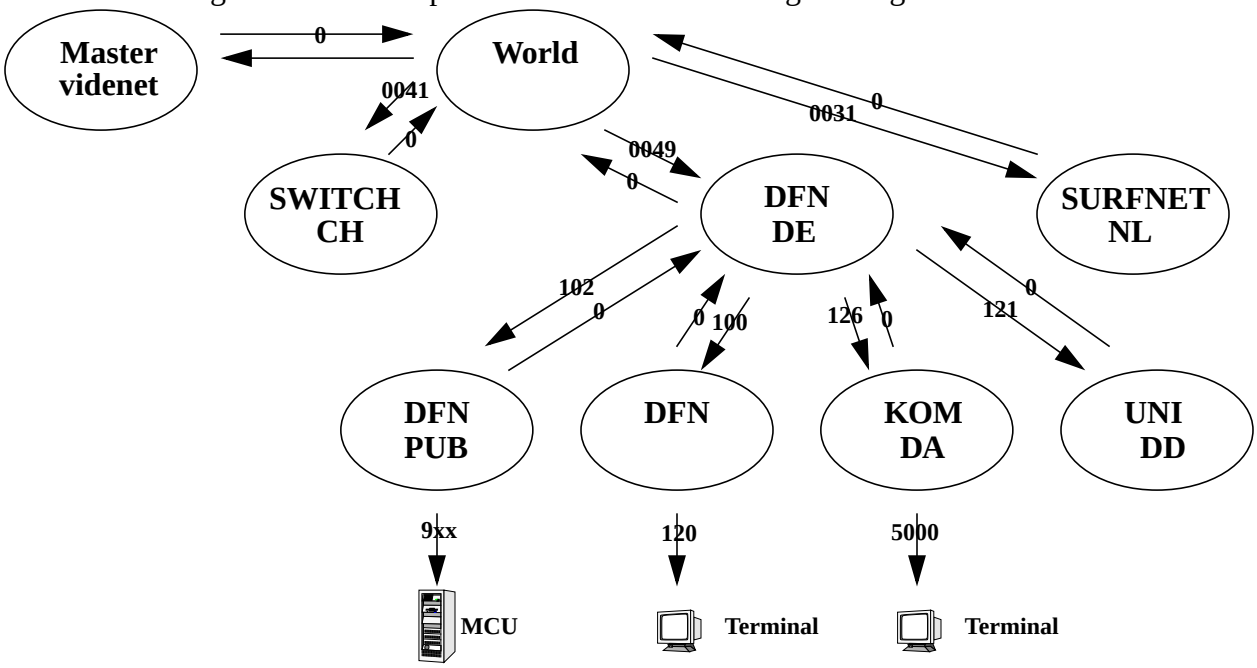

Abbildung 36: Internationale Gatekeeper Struktur

Der DFN stellt einen Gatekeeper bereit (DFN DE), der zwischen den einzelnen H.323-Zonen innerhalb Deutschlands (z.B. KOM DA, UNI DD,...) für das Call Routing verwendet werden kann, und eine internationale Anbindung bereitstellt. Zusätzlich wird vom DFN ein Nummernplan bereitgestellt, der es ermöglicht neue Zonen in das Gesamtsystem zu integrieren. Der DFN betreibt für den operator-unterstützten Dienst eine eigene H.323-Zone (DFN), sowie eine öffentli-

che Zone (DFN PUB), in welcher die für den Videokonferenzdienst notwendigen MCUs angeordnet sind. Das folgende Beispiel stellt dar, wie ein Ruf zwischen zwei Zonen innerhalb Deutschlands in diesem System vermittelt wird:

- In diesem Beispiel wird angenommen, daß das Terminal (Terminal A) mit der Nummer 5000 innerhalb der Zone KOM-DA (das Institut KOM an der Technischen Universität Darmstadt) das Terminal (Terminal B) mit der Rufnummer 120 in der Zone DFN (DFN Geschäftsstelle in Berlin) anruft.

- Das rufende Terminal A ist an dem Gatekeeper der Zone KOM-DA angemeldet, Terminal B ist an dem Gatekeeper der Zone DFN angemeldet.

- Um Terminal B zu kontaktieren, wählt Terminal A die Rufnummer 0100120.

- Der Gatekeeper der Zone KOM-DA erkennt, daß das Ziel-Terminal mit der Rufnummer 0100120 nicht bei ihm selbst registriert ist. Anhand der Neighbour-Tabelle des Gatekeepers muß nun entschieden werden, wie das Ziel gefunden wird. Der Gatekeeper besitzt einen Neighbour-Eintrag, der für den Prefix 0 auf den DFN-DE Gatekeeper verweist. Der Gatekeeper der Zone KOM-DA sendet nun einen entsprechenden *Location Request (LRQ)* für die Rufnummer 100120[1] an den DFN-DE Gatekeeper.

- Der DFN-DE Gatekeeper, welcher Neighbour-Einträge für alle innerhalb Deutschlands verwendeten H.323-Zonen besitzt, leitet die *LRQ*-Anfrage entsprechend seines Neighbour-Eintrags für die Zone mit dem Prefix 100 an den Gatekeeper der Zone DFN weiter.

- Der Gatekeeper der Zone DFN bestätigt nun die Anfrage mit einem *Location Confirm (LCF)*, da ein Terminal mit der Rufnummer 120 (bzw. 100120) bei ihm registriert ist. Die *LCF*-Nachricht wird über den DFN-DE Gatekeeper an den KOM-DA Gatekeeper weitergeleitet. Die *LCF*-Nachricht enthält die *Q.931*-Adresse von Terminal B. Je nach Gatekeeper Policy (z.b. gatekeeper routed call) wird diese Adresse von den Gatekeepern vor dem Weiterleiten eventuell durch die eigene ersetzt.

- Schließlich kann die H.323 Verbindung zwischen den Terminals aufgebaut werden.

Soll eine MCU verwendet werden, um beispielsweise eine Konferenz mit mehreren Teilnehmern durchzuführen, verbinden sich alle Konferenzteilnehmer mit der MCU. Zuvor muß über geeignete Mechanismen gesichert werden, daß die Ressourcen innerhalb der MCU zum gewünschten Zeitpunkt für die Dauer der Konferenz bereitgestellt werden können. Die einzelnen Konferenzteilnehmer bauen zwischen ihrem Terminal und der MCU jeweils eine Standard-H.323-Verbindung auf, die MCU übernimmt das Mischen der Audio und Video Daten. Die einzelnen zur MCU aufgebauten Verbindungen der Konferenzteilnehmer unterscheiden sich dabei nicht von einer Verbindung zu einem normalen Terminal. Eine in dieser Architektur verwendete Firewall, welche

1. Gatekeeper können den Prefix vor Weiterleitung der Anfrage von der Rufnummer entfernen. Dies hängt von der Konfiguration des Gatekeepers sowie der Gatekeeper-Implementierung ab.

die Kommunikation zwischen zwei Terminals unterstützt, unterstützt damit auch automatisch die Kommunikation mit einer MCU. Eine Firewall, die innerhalb des DFN-H.323-Szenarios eingesetzt werden soll, muß sich in die in Abbildung 36 beschriebenen Architektur einfügen.

Verwendete Firewall Systeme innerhalb des DFN. Die innerhalb des DFN verwendeten Firewall Architekturen bestehen meistens aus einem Paketfilter oder Stateful Filter. Der Filter ist dabei meistens innerhalb eines Routers (basierend auf einem Standard PC oder dedizierter Hardware) realisiert. NAT-Komponenten finden oft keine Anwendung, da viele Forschungseinrichtungen über genügend öffentlich gültige IP-Adressen verfügen. Kommerzielle Firewall-Systeme, wie beispielsweise eine PIX oder eine Firewall-1, kommen meist nicht zum Einsatz, da diese im Vergleich zu einem einfachen Paketfilter erheblich teurer sind. Kommerzielle Systeme sind innerhalb des DFN deshalb nur bei größeren Forschungseinrichtungen vorhanden. Folgende Voraussetzungen sind innerhalb des DFN anzutreffen:

- **Paketfilter:**
  Ein Standard PC und Betriebssystem mit Paketfilter Software (z.B. FreeBSD mit IP-Filter Software)
  Ein Standard Router mit integrierter Paketfilter Software (z.B. ein Cisco Router)

- **Kommerzielle Firewall-Systeme:**
  Es werden die gängigen Firewalls verwendet (z.B. PIX, Firewall-1). In diesen Fällen wird in der Regel auch NAT eingesetzt.

Eine H.323-Firewall-Lösung für das DFN sollte berücksichtigen, daß viele der oben beschriebenen Firewall Systeme bereits verwendet werden. Eine H.323-Firewall-Lösung sollte wenn möglich sich in diese schon bestehenden Systeme integrieren lassen.

9.2 Lösungsmöglichkeiten

Eine H.323-Firewall-Lösung innerhalb des DFN muß sich in die im vorigen Abschnitt beschriebene H.323-Struktur einfügen. Darüber hinaus sollte eine H.323-Firewall-Lösung sich in bereits bestehenden Firewall-Systeme integrieren lassen, damit bereits getätigte Investitionen geschützt werden können. Eine Lösung sollte zusätzlich kostengünstig sein, da dies im universitären Umfeld eine erhebliche Rolle spielt.

Integration von Firewalls in die H.323-Struktur des DFN. Um festlegen zu können, wie in die in Abbildung 36 dargestellte H.323-Struktur eine Firewall einfügt werden kann, müssen zuerst einige Randbedingungen betrachtet werden.

- Eine H.323-Zone entspricht in der Regel dem durch eine Firewall geschützen Bereich. Eine Institution wird eine eigene H.323-Zone innerhalb des eigenen Netzes verwenden, um dort den verwendeten Nummernraum bzw. die möglichen H.323-Dienste eigenständig verwalten und konfigurieren zu können[1]. Dieselbe Institution wird auch

1. Diese Annahme gilt für den Bereich des DFN (z.B. universitäres Umfeld), nicht aber zwingend in anderen Bereichen.

eine Firewall verwenden um das interne Netz gegen Angriffe von außerhalb zu schützen.

- Eine H.323-Zone wird normalerweise nicht von derselben Person/Personengruppe verwaltet wie die entsprechende Firewall. Beide Arbeitsfelder - Telefonie und Sicherheit - liegen nicht so dicht beieinander, daß diese automatisch von der selben Person verwaltet werden müssen. Um den Betrieb einer Firewall in einer H.323-Zone zu erleichtern, sollte die Firewall möglichst wenig H.323-Funktionalität enthalten.

Aus den oben genannten Randbedingungen ergibt sich, daß eine H.323-Firewall innerhalb des

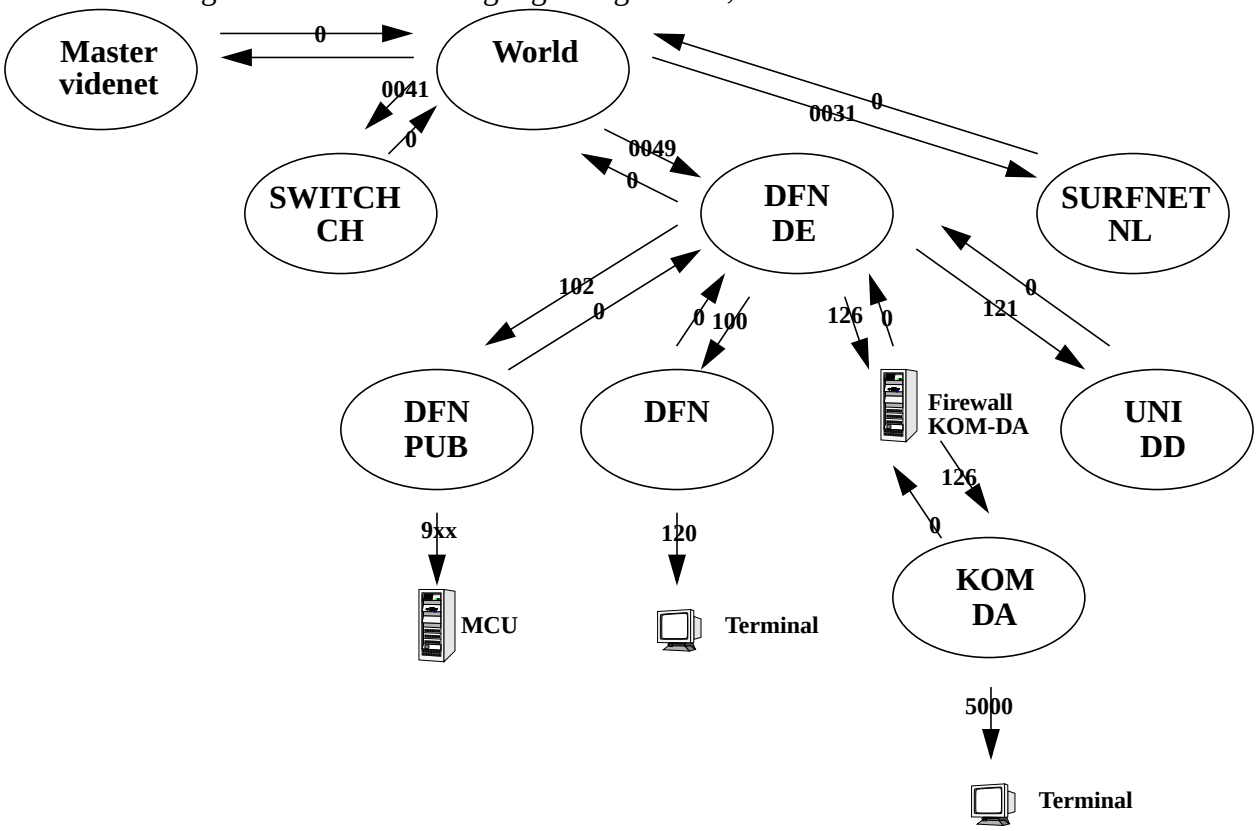

Abbildung 37: Internationale Gatekeeper Struktur mit Firewall

DFN-Szenarios an den Übergängen zwischen verschiedenen H.323-Zonen angeordnet werden sollte. Dabei sollte eine Firewall möglichst nur die für die Sicherheit relevanten H.323 Funktionen implementieren, nicht aber darüber hinaus reichende H.323-Funktionen.
Abbildung 37 zeigt die DFN-H.323-Struktur mit einer Firewall, die das interne Netzwerk (und damit auch die H.323 Zone KOM-DA) des Institutes KOM in Darmstadt absichert. Die so integrierte Firewall muß in der Lage sein, das Call Routing Modell “Transparent - Nutzung der Gatekeeper-Gatekeeper Kommunikation” (vgl. Kapitel 6.2.2) zu unterstützen. Es ergibt sich nun folgende veränderte Ruf Vermittlung:

- In diesem Beispiel wird wiederum angenommen, daß das Terminal (Terminal A) mit der Nummer 5000 innerhalb der Zone KOM-DA das Terminal (Terminal B) mit der Rufnummer 120 in der Zone DFN anruft.

- Das rufende Terminal A ist an dem Gatekeeper der Zone KOM-DA angemeldet, Terminal B ist an dem Gatekeeper der Zone DFN angemeldet.

- Um Terminal B zu kontaktieren, wählt Terminal A die Rufnummer 0100120.

- Der Gatekeeper der Zone KOM-DA erkennt, das das Ziel-Terminal mit der Rufnummer 0100120 nicht bei ihm selbst registriert ist. Der Gatekeeper besitzt einen Neighbour-Eintrag, der für den Prefix 0 auf die Firewall der Zone KOM-DA verweist. Der Gatekeeper der Zone KOM-DA sendet nun einen entsprechenden *Location Request (LRQ)* für die Rufnummer 0100120 an die KOM-DA Firewall.

- Die Firewall erkennt, das das Ziel-Terminal mit der Rufnummer 0100120 über den Gatekeeper DFN-DE zu erreichen ist (Neighbour-Tabelle). Die Firewall der Zone KOM-DA sendet nun einen entsprechenden *Location Request (LRQ)* für die Rufnummer 100120 an den DFN-DE Gatekeeper.

- Der DFN-DE Gatekeeper, welcher Neighbour-Einträge für alle innerhalb Deutschlands verwendeten H.323-Zonen besitzt, leitet die *LRQ*-Anfrage entsprechend seines Neighbour-Eintrags für die Zone mit dem Prefix 100 an den Gatekeeper der Zone DFN weiter.

- Der Gatekeeper der Zone DFN bestätigt nun die Anfrage mit einem *Location Confirm (LCF)*, da ein Terminal mit der Rufnummer 120 (bzw. 100120) bei ihm registriert ist. Die *LCF*-Nachricht wird über den DFN-DE Gatekeeper und über die KOM-DA Firewall an den KOM-DA Gatekeeper weitergeleitet. Die *LCF*-Nachricht enthält die *Q.931*-Adresse von Terminal B. Je nach Gatekeeper Policy (z.b. gatekeeper routed call) wird diese Adresse von den Gatekeepern und der Firewall vor dem Weiterleiten eventuell durch die eigene ersetzt.

- Schließlich kann die H.323 Verbindung zwischen den Terminals über die Firewall aufgebaut werden.

Eine Firewall, die das hier beschriebene Verhalten aufweist, kann innerhalb des DFN-H.323-Szenarios verwendet werden.
Von den in Kapitel 8. untersuchten Firewall-Systemen wird dieses Kriterium von den Lösungen KOMproxyd, OpenH323Proxy und Ciscos MCM erfüllt. Alle anderen beschriebenen Lösungen können nicht auf die oben beschriebene Weise in das DFN-Szenario integriert werden.

Erweiterung von bestehenden Firewall Systemen um H.323-Komponenten. Prinzipiell ist es schwierig, kommerzielle zur Zeit auf dem Markt befindliche Systeme (z.B. PIX, Firewall-1) als H.323-Firewall zu verwenden. Zum einen müssen in einem solchen Fall die bestehenden Firewall Systeme ausgetauscht werden, zum anderen bieten diese Systeme zur Zeit noch nicht die notwen-

dige H.323-Funktionalität (vgl. Kapitel 8.2). Darüber hinaus stellt die einem Hybridsystem zugrunde liegende Architektur keine für H.323 geeignete Architektur dar (vgl. Kapitel 7.3).
Eine verteilte Firewall entsprechend den in Kapitel 7.2 dargestellten Architekturen II, III und IV läßt sich ebenfalls nur dann realisieren, wenn die bestehenden Firewall-Komponenten modifiziert oder ersetzt werden. Um dies zu vermeiden, kann auf die Interaktion zwischen den verschiedenen Firewall-Komponenten verzichtet werden. Daraus ergibt sich dann eine Firewall Architektur, welche nicht hinsichtlich aller Kriterien (z.B. Performance) optimiert ist, aber auf bestehende Firewall Komponenten aufbauen kann.
Die aus dieser Betrachtung resultierenden Systeme entsprechen den in Kapitel 8.1 beschriebenen Proxy-Systemen KOMproxyd, OpenH323Proxy, PhonePatch und Ciscos MCM. Solche Proxy-Systeme eignen sich für den Einsatz innerhalb des DFN-Szenarios.

9.3 Zusammenfassung

Aus den durch das vorhandene DFN-Umfeld vorgegebenen Randbedingungen sowie den durchgeführten Untersuchungen zur Zeit verfügbarer Lösungen, ergeben sich folgende praktisch umsetzbare Lösungen:

- **KOMproxyd:**
  Das unter [30] verfügbare System kann innerhalb des DFN-Szenarios verwendet werden. Dieses Firewall-System wurde innerhalb des DFN-Szenarios erfolgreich getestet. Es ist aber zu beachten, das diese frei verfügbare Software nicht mit einem Produkt kommerzieller Firewall Anbieter - hinsichtlich Support, Funktionsumfang,... - verglichen werden kann.

- **Cisco Multimedia Conference Manager:**
  Dieses Produkt ähnelt von seiner Funktionsweise sehr stark der KOMproxyd Lösung. Innerhalb der Studie konnte dieses Produkt aber nicht praktisch innerhalb des DFN-Szenarios getestet werden. Dieses Produkt kann als alternative Möglichkeit zu KOMproxyd gesehen werden und besitzt nicht dessen Beschränkungen - z.B. Support - ist aber natürlich mit sehr viel höherem finanziellen Aufwand verbunden.

- **Andere Lösungen:**
  Die hier vorgeschlagenen Lösungen wurden innerhalb dieser Studie betrachtet. Nicht alle zur Zeit verfügbaren Produkte oder Prototypen konnten untersucht werden. Es ist daher nicht auszuschließen, das weitere Lösungen existieren. Diese Lösungen werden aber den allgemeingültigen Anforderungen, die innerhalb dieser Studie beschrieben sind, genügen müssen.

10. Zusammenfassung

Innerhalb der vorliegenden Studie wurden die Probleme analysiert und dokumentiert, welche auftreten, wenn Firewalls in einem H.323-Szenario verwendet werden. Es wurde aufgezeigt, wie diese Probleme theoretisch beseitigt werden können. Darüber hinaus wurden einige zur Zeit verfügbare Firewall-Produkte getestet, die für eine praktische Lösung der Probleme in Frage kommen. Es wurde ebenfalls beschrieben, wie die in Frage kommenden Produkte innerhalb des DFN-Szenarios integriert und eingesetzt werden können.

10.1 Empfehlungen für den DFN-Videokonferenzeinsatz

Wie in Kapitel 9. beschrieben, kann innerhalb des DFN-Videokonferenzdienstes die KOMproxyd-Firewall-Komponente verwendet werden. Dieses System wurde innerhalb des DFN-Videokonferenzdienstes erfolgreich getestet. Es stellt eine kostengünstige Möglichkeit dar, bestehende Firewall-Systeme um die Fähigkeit, den DFN-Videokonferenzdienst zu unterstützen, zu erweitern (siehe Kapitel 8. und Kapitel 9.).

10.2 Ausblick

Einige der untersuchten Firewall-Lösungen (z.B. Ciscos Multimedia Conference Manager) konnten im Rahmen dieser Studie nicht in praktischen Versuchen getestet werden. Es wäre wünschenswert, die aus der Dokumentation und aus Erfahrungsberichten dieser Produkte gewonnenen Erkenntnisse praktisch zu überprüfen. Bei vielen Firewall-Produkten hat sich gezeigt, daß eine erhebliche Differenz zwischen Produktbeschreibung und tatsächlichem Verhalten - bezüglich der Unterstützung des H.323 Protokolls - besteht.
Manche aufgetretenen Fragen konnten innerhalb der vorliegenden Studie nicht vollständig beantwortet werden. Als Beispiel sei hier die Frage der Performance (z.B. Call Setup, Media Delay,...) von verschiedenen Firewalls angeführt. Um Antworten auf diese verbleibenden Fragen zu finden, sind weitere Untersuchungen notwendig.
Die meisten der innerhalb der Studie untersuchten H.323-spezifischen Probleme im Zusammenhang mit Firewalls treten auch bei der Verwendung anderer Multimedia-Applikationen (z.B. Video Streaming Applikationen) auf. Weitere Arbeiten sind nötig, um allgemeingültige Firewall Lösungen zu entwickeln, welche in der Lage sind die Klasse der Multimedia-Applikationen sinnvoll zu unterstützen.

11. Referenzen

[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] Draft Recommendation H.235 v2 Security and Encryption for H Series (H.323 and other H.245 based) multimedia terminals, 2000.

[6] M. Euchner. Draft H.323 Annex J: Security for H.323 Annex F, 1999.

[7] H.323 Annex J: Security for H.323 Annex F, 2000.

[8] M. Schumacher and U. Roedig. Security Engineering with Patterns. 8th Conference on Pattern Languages of Programs (PLoP 2001), Monticello, Illinois, USA, September 2001.

[9] Schumacher, M.; Roedig, U.; Moschgath, M.-L.: Hacker Contest · Sicherheitsprobleme, Lösungen, Beispiele. ISBN: 3-540-41164-X, 2000.

[10] Stephan Fischer, Christoph Rensing, and Utz Roedig. Open Internet Security - Von den Grundlagen zu den Anwendungen (in German). Springer Verlag, Heidelberg, Germany, January 2000. ISBN 3-540-66814-4.

[11] B. Schneier, Secrets & Lies: Digital Security in a networked World, John Wiley & Sons, 2000.

[12] D. Doerner, Die Logik des Mißlingens - Strategisches Denken in komplexen Situationen, Rowohlt Verlag, 1989.

[13] R. Shirey, RFC 2828, Internet Security Glossary, 2000.

[14] Chapman, D. B.: Building Internet Firewalls. O'Reilly, Cambridge, 1995.

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

[16] Norbert Pohlmann.: Firewall Systeme. MITP-Verlag, Bonn, 2000. ISBN 3-8266-4075-6.

[17] http://www.checkpoint.com/.

[18] http://www.cisco.com/.

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

[20] Ackermann, M. Schumacher, U. Roedig, and R. Steinmetz.Vulnerabilities and Security Limitations of current IP Telephony Systems. In Proceedings of the Conference on Communications and Multimedia Security (CMS), Germany, May 2001.

[21] Recommendation H.225.0 - Call Signalling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems, Feb. 1998.

[22] Recommendation H.245 - Control Protocol for Multimedia Communication, Sep. 1998.

[23] H. Schulzrinne. RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control, Jan. 1996.

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

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

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

[27] U. Roedig, R. Ackermann, M. Tresse, L. Wolf, and R. Steinmetz. Verbesserte Systemsicherheit durch Kombination von IDS und Firewall. In Systemsicherheit, pages 117-128, March 2000. ISBN 3-528-05745-9.

[28] U. Roedig, R. Ackermann, and R. Steinmetz. Evaluating and Improving Firewalls for IP-Telephony Environments. In Proceedings of the 1st IP-Telephony Workshop (IPTel2000), number ISSN 1435-2702, pages 161-166. GMD-Forschungszentrum Informationstechnik GmbH, April 2000.

[29] D. Reed. IP-Filter - TCP/IP Packet Filtering Package. http://coombs.anu.edu.au/~avalon/ip-filter.htm, 2001.

[30] U. Roedig. KOMproxyd. http://www.kom.e-technik.tu-darmstadt.de/KOMproxyd, 2001.

[31] U. Roedig, R. Ackermann, C. Rensing, and R. Steinmetz. A Distributed Firewall for Multimedia Applications. In Proceedings of the Workshop “Sicherheit in Mediendaten”, September 2000.

[32] A.Molitor. Firewall Control for IP Telephony. ARAVOX Technologies. http://www.aravox.com.

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

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

[35] OpenGatekeeper H.323 Proxy: http://openh323proxy.sourceforge.net.

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

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

[38] Cisco: http://www.cisco.com/warp/public/cc/pd/fw/sqfw500.

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

[40] Utz Roedig, Manuel Görtz, Martin Karsten, and Ralf Steinmetz. RSVP as Firewall Signalling Protocol. In Proceedings of the 6th IEEE Symposium on Computers and Communications, Hammamet, Tunisia, pages 57-62. IEEE, July 2001.