Architekturen verteilter Firewalls für IP-Telefonie-Plattformen

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Sebastian Kiesel

geb. in Hechingen

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Mitberichter:	Prof. Dr.-Ing. Thomas Magedanz, TU Berlin
Tag der Einreichung:	11. Dezember 2007
Tag der mündlichen Prüfung:	15. September 2008

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2008

Abstract

The history of the Internet has been an impressive success story. The underlying IP protocol suite originally has been designed for non-realtime computer communications. However, practical experience has shown that these protocols are also suitable for the realtime transmission of multimedia sessions in many scenarios. Today, "Voice over IP" (VoIP) is widely used to replace conventional private branch exchanges or for making cheap long-distance calls on the Internet. However, this "open" federation of interconnected networks operated by various parties has also many problems, especially in the area of network security. Therefore, security is a key issue when discussing plans for a future-generation all-IP network capable of offering all types of services, including those currently provided by the conventional PSTN/ISDN networks.

Access control on network traffic traversing boundaries between domains with different security properties and requirements plays an important role when securing networks. In IP networks this concept is implemented in so-called firewalls. In the context of IP-based multimedia applications the tasks of a firewall can be classified into two main functions: access control on the signaling protocols and access control on the actual media flows. For the coordination between the two modules implementing these functions, a signaling protocol is needed. Two basic architectural approaches exist for performing this task, the path-decoupled and the path-coupled firewall signaling.

Working groups of the Internet Engineering Task Force are currently working on the specifications of two specific signaling architectures and corresponding protocols, each following one of these approaches, respectively (MIDCOM and NSIS). Other standardization organizations work on similar solutions. In scientific literature, this classification is known, too. However, contributions usually itemize only some rather obvious advantages and disadvantages of the architectures. The aim of this thesis is to perform a comprehensive analysis and comparison of both approaches. In addition to the protection goals, functional aspects and operational requirements shall be considered. Furthermore, the impact of firewall control on quality of service and scalability of the network infrastructure is studied.

Chapter 2 gives an overview on IP-telephony, the Session Initiation Protocol (SIP), network security, and firewalls. In chapter 3, security threats for IP-telephony are investigated. It is shown that the "usual" approach for securing Internet applications – establishing cryptographic channels on top of an insecure physical infrastructure – cannot defend against all types of attacks. This applies in particular to so-called denial-of-service (DoS) attacks, which aim at disrupting services by flooding parts of the infrastructure with large amounts of messages, thus overwhelming devices and deliberately causing congestion. Firewalls can contribute to the mitigation of

these problems. As an alternative to Internet-telephony, IP-telephony can also be performed in IP-telephony-platforms. This term is used to describe networks, which – though using the IP protocol – are completely separated from the public Internet, possibly interconnected by gateways, only. Establishment of these networks may be for economic reasons, or for increasing security or quality of service. Several consortia of equipment manufacturers and network operators consider this approach; the 3GPP IMS and ETSI TISPAN architectures are taken as an example and their main features are described.

Chapter 4 presents a classification of firewall architectures suitable for handling multimedia applications with separate protocols for signaling and media transport. If the corresponding signaling function and media function of the firewall are placed in different network elements, a control protocol is needed in between. The two basic architectural approaches and the corresponding IETF protocols are introduced. When using path-decoupled signaling, the signaling functions, which are often placed in the middle of the network topology, send control commands to the media function at the edge. These commands, which are sent to a specific media function, add or remove policy rules, which allow a specific media stream to traverse this media function. In contrast, when using path-coupled signaling, signaling messages are sent along the future media path, in order to announce the new flow and establish the corresponding policy rules in any firewall on the path. These messages are forwarded using the same routing tables which are used for the media streams, too.

Chapter 5 describes design and implementation of a prototype for the SIMCO protocol. A testbed demonstrates the feasibility of integrating SIP-based IP telephony and path-decoupled firewall control. However, it is shown that the presence of SIMCO/MIDCOM is not as transparent to SIP as suggested by the MIDCOM architecture specification. The most important issue is how to ensure that an error condition in the context of one of these protocols causes a reasonable reaction by the other protocol, respectively. It is investigated and shown how optional SIP extensions, which have been designed for other purposes, can be used for handling these error conditions. In addition, performance measurement results for the Linux based prototype implementation are presented.

In IP-telephony-platforms, which are supposed to replace the legacy PSTN/ISDN networks, there are a lot of signaling tasks to be performed. In addition to SIP-based signaling for call control, there is a need for authentication, authorization, routing, call forwarding, number portability, capturing call detail records, dynamic firewall control, etc. Therefore, establishment of a new multimedia session involves several signaling protocols. Delays caused by them contribute to the post-selection delay or to the answer signal delay, which are unpleasant to the subscribers. In order to minimize these delays, both local processing and transport of signaling messages have to be optimized. Therefore, chapter 6 gives an overview on several transport layer protocols and configuration options for signaling transport. At the example of SIMCO it is investigated whether they require modifications at the signaling protocol.

One degree of freedom when choosing and configuring the transport layer protocol is whether none, partial, or complete protection against message reordering is required, as message sequence protection mechanisms may cause so-called head-of-line blocking. This delaying effect occurs if packets get lost in the network and have to be retransmitted. Subsequent packets have to be buffered at the receiver until the retransmission has finished, in order to protect the message sequence. This impacts in particular signaling associations with high message rates

and completely ordered transport. Even moderate packet loss may affect many sessions. The Stream Control Transmission Protocol (SCTP) allows for a partially ordered transport, which is beneficial for SIMCO. The necessary modifications to SIMCO, which usually uses TCP-based transport, are investigated and specified; a "SIMCO over SCTP" prototype implementation demonstrates the feasibility of this proposal.

Delays caused by head-of-line blocking are a well-known effect, e. g., in the context of satellite communications. However, existing models do not consider the specific mechanisms of IP-based transport layer protocols. In the course of this work an analytical model for head-of-line blocking has been developed, which covers the specific mechanisms of TCP and SCTP, such as packet loss detection by means of a combination of timers, positive and negative acknowledgments, as well as an arbitrary number of SCTP streams. Results from measurements using a WAN emulator, a load generator, and the "SIMCO over SCTP" prototype match very well the values predicted by the model and show a significantly reduced transaction response time, compared to TCP-based transport. As only few aspects of the study are specific to SIMCO, results can be adapted easily to other signaling protocols, such as SIP or NSIS.

Based on these results, a comparison between the path-decoupled and the path-coupled approach for firewall control is performed in chapter 7, for a scenario with interconnected IP-telephony-platforms under the control of different operators. It is assumed that there is a SIP-based control plane, which authorizes new multimedia sessions and which knows the parameters of the corresponding media streams, which have to be allowed by the media functions in the firewalls.

Concerning functional aspects, a major drawback of the path-decoupled firewall signaling approach is that it requires entities of the control plane to know the network topology and the state of the routing protocol. It is required in order to add policy rules to those media functions, which actually are on the path of a given media flow. In "carrier grade" networks operated by professional companies one has to assume that there are several interconnection points between any two domains. Media flows, which have to cross a border, will be assigned dynamically to one of these interconnections, for increasing resilience in case of link failures, for load balancing and for reducing latencies. There exist some approaches for dynamically analyzing the state of the routing protocols. However, these are complex and introduce new interfaces and dependencies. Another approach is to perform an "off-line" analysis, e. g. by simulating the behavior of the routing protocol for any possible combination of link failures. This would yield a list of all firewalls a given flow possibly could traverse. If this list is not too long, the corresponding policy rule could simply be added to all of them. As this can be done in parallel it would not cause any significant additional delay. If the list of possible interconnection points is very long, this approach is inefficient and not suitable. However, the question may arise whether such a rather volatile border is a reasonable border between security domains.

In contrast, path-coupled signaling does not require topology knowledge at a central place. However, this approach has also several drawbacks. Unlike the path-decoupled signaling it is not an intra-domain solution as a matter of principle. Instead, signaling messages have to be exchanged with untrusted entities in adjacent domains. This requires some coordination between network operators and may open security holes, which could be exploited, e. g. for denial-of-service attacks or for compromising the firewall. If path-coupled signaling is used in

an end-to-end configuration, protocol entities are needed in the multimedia endpoints, possibly restricting subscribers when choosing their equipment.

The establishment of a new multimedia session requires an interaction between the signaling protocols used for call control and firewall control. Depending on the solution used for firewall control adding policy rules may or may not be performed in parallel. Considering a scenario with several transit domains between the subscribers, signaling procedures are analyzed and the impact of different firewall control architectures on post-selection delay and answer signal delay is quantified. It can be shown that, compared to the unprotected reference scenario, the path-coupled approach causes more additional delay than the path-decoupled method – independent of the specific numbers assumed for message processing delays, packet loss probabilities, and transport latencies in the network.

Some people in favor of the path-coupled firewall signaling consider the need for topology awareness being *the* KO-criterion against path-decoupled signaling. However, when considering all criteria which were investigated in this thesis, one can conclude that this true and important fact is achieved at the cost of several other drawbacks. Therefore the complexity and the dynamics of the network topology probably will become the *determining* issue when choosing the firewall signaling architecture. If they are high, e. g. in "open," Internet-like scenarios with dynamic routing, network elements under the administrative control of the subscribers (e. g. "DSL routers"), and cascaded network address and port translation in the access network, yielding topology information may be cumbersome or impossible. Path-coupled signaling clearly is advantageous in this case. However, IP-telephony-platforms are networks which – despite using the TCP/IP protocol suite – are much more centralized and static than the Internet, for example by "route pinning" or by forcing all sessions to be signaled to the control plane. In these network domains, which are under the full administrative control of one operator, gathering topology information may be much easier. In this case, an efficient deployment of path-decoupled signaling is feasible. As this solution does not require direct interfaces to untrusted entities in other domains it fits better to the design principles of these "closed" IP-telephony-platforms.

Kurzfassung

Die Geschichte des Internet ist bis zum heutigen Tag eine beispiellose Erfolgsgeschichte. Die praktische Erfahrung hat gezeigt, dass die zugrundeliegende, ursprünglich für die nicht-echtzeitkritische Rechnerkommunikation entwickelte IP-Protokollfamilie in vielen realistischen Szenarien auch für die Übertragung echtzeitkritischer Multimedia-Ströme geeignet ist. Bereits heute ist „Voice over IP“ (VoIP) als Ersatz für private Nebenstellenanlagen oder für preisgünstige Ferngespräche über das Internet weit verbreitet. Dennoch existiert in diesem recht offenen Verbund von Netzen verschiedener Betreiber eine Reihe von Problemen, insbesondere im Bereich der Netzsicherheit. Dementsprechend sind Maßnahmen zur Erhöhung der Sicherheit ein zentrales Thema bei dem Vorhaben, in Zukunft sämtliche Telekommunikationsdienste – inklusive der bisher im ISDN erbrachten Telefoniedienste – über IP-basierte Netze abzuwickeln.

Eine wichtige Rolle bei der Absicherung von Kommunikationsnetzen gegen Angriffe spielen Zugriffskontrollen auf Netzverkehr, der Grenzen zwischen Bereichen mit unterschiedlichen Sicherheitsniveaus und -anforderungen überquert. In IP-Netzen werden solche Kontrollen mit Hilfe von Firewalls implementiert. Bei der Verwendung von Multimedia-Anwendungen wie z. B. IP-Telefonie ergeben sich für die Firewall zwei Hauptaufgaben: Zugriffskontrollen auf die Signalisierung und solche auf die Medienströme. Zwischen den beiden Modulen, die die jeweiligen Funktionen übernehmen, wird ein Signalisierprotokoll zur Koordination benötigt; hierfür existieren zwei verschiedene Grund-Architekturen, die Pfad-entkoppelte und die Pfad-gekoppelte Firewall-Signalisierung.

In Arbeitsgruppen der Internet Engineering Task Force werden derzeit zwei konkrete Signalisierarchitekturen und -protokolle spezifiziert (MIDCOM bzw. NSIS), die jeweils einem dieser Grundprinzipien folgen; andere Standardisierungsorganisationen arbeiten an ähnlichen Konzepten. Auch in der wissenschaftlichen Literatur ist diese Klassifikation bekannt, allerdings werden i. d. R. für jeden der beiden Vorschläge nur einige recht offensichtliche Vor- und Nachteile benannt. Ziel dieser Arbeit ist daher ein umfassenderer Vergleich der beiden Architekturansätze. Neben den Schutzzielen sollen hierbei auch funktionelle Aspekte und betriebliche Anforderungen berücksichtigt werden; ferner soll untersucht werden, wie sich die Firewall-Steuerung auf die von den Teilnehmern empfundene Dienstgüte und auf die Leistungsfähigkeit der Netzinfrastruktur auswirkt.

Nach einer Übersicht über die Grundlagen von Voice-over-IP (VoIP) – insbesondere dem Signalisierprotokoll SIP – und der Netzsicherheit in Kapitel 2 werden in Kapitel 3 Bedrohungsszenarien für die IP-Telefonie untersucht. Es wird aufgezeigt, dass die „übliche“ Vorgehensweise zur Erhöhung der Sicherheit von Internet-Anwendungen – das Schaffen kryptographisch separier-

ter Kanäle oberhalb einer als nicht sicher angenommenen Infrastruktur physikalischer Kanäle – nicht alle denkbaren Angriffe abwehren kann. Hierzu zählen insbesondere so genannte Denial-of-Service-Attacken, die Teile der Infrastruktur mit Nachrichtenfluten überlasten und so die Verfügbarkeit von Diensten einschränken können. Firewalls an Domänengrenzen können zur Eingrenzung solcher Probleme beitragen. Alternativ zur Internet-Telefonie kann IP-Telefonie auch in IP-Telefonie-Plattformen durchgeführt werden. Dabei handelt es sich um Netze, die zwar das IP-Protokoll verwenden, jedoch vom öffentlichen Internet vollständig getrennt bzw. nur über Gateways erreichbar sind. Die Errichtung solcher getrennter Netze kann wirtschaftliche Gründe haben oder zur Erhöhung der Sicherheit oder Dienstgüte erfolgen. Solche Ansätze werden von verschiedenen Hersteller- und Betreiberkonsortien unter verschiedenen Namen verfolgt; stellvertretend werden die Grundzüge der 3GPP IMS- und ETSI TISPAN-Architekturen aufgezeigt.

In Kapitel 4 werden Firewall-Architekturen klassifiziert, die die Verwendung von Multimedia-Anwendungen mit getrennten Protokollen für Signalisierung und Medientransport erlauben. Werden die entsprechenden Signalisier- und Medienkomponenten der Firewall in getrennten Netzelementen platziert, wird ein Steuerprotokoll benötigt. Die beiden Grundverfahren sowie die entsprechenden IETF-Protokolle werden vorgestellt. Bei der Pfad-entkoppelten Signalisierung senden die – in der Netztopologie oft zentral platzierten – Signalisierkomponenten Steuerkommandos an die dezentral installierten Medienkomponenten, um das Eintragen von Regeln für die Medienströme zu veranlassen. Dabei werden die Medienkomponenten explizit adressiert. Bei der Pfad-gekoppelten Signalisierung werden hingegen Signalisiernachrichten entlang des zukünftigen Medienpfades gesendet, die einen Medienstrom ankündigen und auf dem Pfad vorhandene Firewalls entsprechend konfigurieren. Die Weiterleitung dieser Nachrichten erfolgt auf Basis der selben Routing-Tabellen, die auch für die Weiterleitung der Medienströme verwendet werden.

Kapitel 5 beschreibt den Entwurf und die prototypische Implementierung von Protokollinstanzen des SIMCO-Protokolls. Der Aufbau einer Testumgebung demonstriert die Machbarkeit einer Integration von SIP-basierter IP-Telefonie und Pfad-entkoppelter Firewall-Steuerung. Dabei stellt sich jedoch heraus, dass das Vorhandensein von MIDCOM bzw. SIMCO für SIP nicht so transparent ist, wie in der MIDCOM-Architekturspezifikation suggeriert wird. Dies betrifft insbesondere die Frage, wie evtl. auftretende Fehler im Bereich eines der beiden Signalisierprotokolle zu einer sinnvollen Reaktion des jeweils anderen Protokolls führen können. Es wird untersucht und aufgezeigt, wie optionale SIP-Erweiterungen, die teilweise für andere Zwecke spezifiziert wurden, zur Behandlung dieser Fehlerfälle eingesetzt werden können. Ergänzend werden einige Messergebnisse zur Leistungsfähigkeit der Paketfilterung mit dem Betriebssystem Linux präsentiert.

In IP-Telefonie-Plattformen, welche die konventionellen ISDN-Netze ersetzen können, fallen neben dem Medientransport und der SIP-Signalisierung zur Verbindungssteuerung noch diverse andere Signalisieraufgaben an, z. B. zur Authentisierung und Autorisierung von Teilnehmern, zur Wegesuche, Rufweiterleitung und Rufnummernportierung, zur Erfassung von Kommunikationsdatensätzen für die Entgelterfassung, sowie zur dynamischen Steuerung von Firewalls, etc. Der Aufbau einer neuen Multimedia-Sitzung erfordert somit ein Zusammenspiel mehrerer Signalisierprotokolle. Verzögerungen, die durch diese Protokolle verursacht werden, tragen zum Ruf- bzw. Meldeverzug bei, die von den Teilnehmern als störend empfunden werden. Um diese

Beiträge zu verringern, muss neben der Verarbeitung der Signalisiernachrichten in den Netzknoten auch ihr Transport über das IP-Netz optimiert werden. In Kapitel 6 werden daher zunächst verschiedene Transportschichtprotokolle und Konfigurationsmöglichkeiten für den Transport von Signalisiernachrichten über IP vorgestellt. Am Beispiel von SIMCO wird untersucht, welche Anpassungen an einem Signalisierprotokoll dafür ggf. notwendig sind.

Ein Freiheitsgrad bei der Auswahl und ggf. Parametrisierung des Transportschichtprotokolls ist, ob keine, teilweise oder vollständige Reihenfolgesicherung für die Signalisiernachrichten benötigt wird, da diese so genanntes Head-Of-Line Blocking verursachen kann. Dieser verzögernde Effekt tritt auf, wenn infolge eines Paketverlustes Nachrichten erneut übertragen werden und darauffolgende Nachrichten zur Reihenfolgesicherung empfängerseitig gepuffert werden müssen. Insbesondere bei Signalisier-Assoziationen mit hohen Nachrichtenraten und vollständiger Reihenfolgesicherung können selbst bei nur sehr sporadisch auftretenden Paketverlusten viele Sitzungen in Mitleidenschaft gezogen werden. Das Stream Control Transmission Protocol (SCTP) erlaubt einen Transport mit teilweiser Reihenfolgesicherung, der für SIMCO besonders gut geeignet ist. Die notwendigen Anpassungen an SIMCO, welches normalerweise über TCP transportiert wird, werden untersucht und spezifiziert; eine prototypische „SIMCO over SCTP“-Implementierung demonstriert die Realisierbarkeit des Vorschlags.

Verzögerungen durch Mechanismen zur Reihenfolgesicherung sind ein allgemein bekannter Effekt, z. B. im Umfeld der Satelliten-Kommunikation; die vorhandenen Modelle bilden jedoch die speziellen Mechanismen der IP-basierten Transportschichtprotokolle nicht ab. Im Zuge dieser Arbeit wurde ein analytisches Modell für Head-Of-Line Blocking entwickelt, welches die spezifischen Mechanismen von TCP und SCTP – z. B. Erkennung von Paketverlusten durch eine Kombination von Zeitüberwachung, positiven und negativen Quittierungen – sowie eine beliebige Anzahl von SCTP Streams abdeckt. Mit Hilfe eines WAN-Emulators und eines Lastgenerators durchgeführte Messungen am „SIMCO over SCTP“-Prototypen zeigen eine gute Übereinstimmung mit den vom Modell vorhergesagten Werten, sowie eine gegenüber TCP-basiertem Transport deutlich reduzierte mittlere Transaktions-Antwortzeit. Da nur wenige Aspekte der Untersuchung spezifisch für SIMCO sind, können die Resultate auch auf andere Signalisierprotokolle, z. B. SIP oder NSIS, übertragen werden.

Basierend auf diesen Untersuchungen einzelner Mechanismen werden in Kapitel 7 die beiden grundsätzlichen Signalisierverfahren zur Steuerung der Medienkomponenten verteilter Firewalls, die Pfad-entkoppelte und die Pfad-gekoppelte Signalisierung, miteinander verglichen. Dabei wird von einem Szenario ausgegangen, in dem die IP-Telefonie-Plattformen mehrerer Betreiber zusammengeschaltet wurden. Dies bedeutet unter Anderem, dass davon ausgegangen wird, dass eine SIP-basierte Control Plane vorhanden ist. Deren SIP-Server autorisieren neue Multimedia-Sitzungen und kennen die dazugehörigen Medienströme, die von den Medienkomponenten der Firewalls erlaubt werden müssen.

Im Bereich der funktionalen Aspekte ist ein wesentlicher Nachteil der Pfad-entkoppelten Signalisierung, dass den zentralen Instanzen der Control Plane Informationen über Netztopologie und Verkehrslenkung bekannt sein müssen. Diese werden benötigt, um Policy Rules zum Erlauben eines Medienstroms in jene Medienkomponenten eintragen zu können, die tatsächlich auf seinem Pfad durch das Netz liegen. Bei kommerziell betriebenen, großen Netzen ist davon auszugehen, dass zwischen zwei Domänen mehrere Netzübergänge vorhanden sind, auf die der grenzüberschreitende Verkehr dynamisch verteilt wird, zur Erhöhung der Verfügbarkeit bei

Komponentenausfällen und ggf. zur Lastverteilung und zur Reduzierung von Latenzen. Zwar existieren Verfahren, die den Zustand des dynamischen Routingprotokolls nachzuvollziehen versuchen, diese sind jedoch aufwändig und führen zu weiteren Abhängigkeiten und Protokollschnittstellen. Alternativ kann versucht werden, durch eine „off-line"-Analyse eine Liste aller Medienkomponenten zu bestimmen, durch die ein bestimmter Medienstrom möglicherweise fließen könnte, z. B. indem das Verhalten des Routingprotokolls für beliebige Kombinationen von Komponentenausfällen simuliert wird. Ist die so gewonnene Liste nicht allzu lang, kann eine entsprechende Regel einfach in alle diese Medienkomponenten eingetragen werden; da dies zeitgleich geschehen kann, entsteht so keine wesentliche zusätzliche Verzögerung des Verbindungsaufbaus (s. u.). Ist die Liste potenzieller Netzübergänge hingegen sehr lang, erscheint dieser Ansatz ungeeignet; allerdings stellt sich dann u. U. auch die Frage, ob eine derart unscharfe Grenze eine sinnvolle Grenze zwischen Sicherheitsdomänen ist.

Bei der Pfad-gekoppelten Signalisierung wird hingegen kein zentrales Topologie-Wissen benötigt. Dieser gewichtige Vorteil wird jedoch dadurch erkauft, dass es sich – anders als bei der Pfad-entkoppelten Signalisierung – nicht um eine rein domäneninterne Lösung handelt, sondern Signalisiernachrichten mit Protokollinstanzen in nicht vertrauenswürdigen Nachbardomänen ausgetauscht werden müssen. Neben dem dadurch notwendigen Abstimmungsbedarf zwischen den Netzbetreibern kann dies u. U. Einfallstore für Angriffe öffnen, z. B. für Denial-of-Service-Attacken oder Angriffe, die Implementierungsfehler zur Kompromittierung der Protokollinstanzen nutzen. Falls Pfad-gekoppelte Signalisierung Ende-zu-Ende zwischen den Endgeräten der Teilnehmer eingesetzt werden soll, müssen dort entsprechende Protokollinstanzen vorhanden sein, was die Teilnehmer in der Wahl ihrer Endgeräte einschränken kann.

Der Aufbau einer Multimedia-Sitzung erfordert ein Zusammenspiel der Protokolle zur Verbindungs- und Firewall-Signalisierung. Je nach verwendetem Signalisierverfahren kann das Eintragen von Policy Rules in verschiedene Medienkomponenten teilweise nebenläufig erfolgen. Für ein Szenario mit mehreren Transitdomänen zwischen den Teilnehmern werden die Signalisiervorgänge analysiert und der Einfluss auf Ruf- und Meldeverzug quantifiziert. Es kann gezeigt werden, dass diese für die Teilnehmer unangenehmen Verzögerungen beim Einsatz Pfad-gekoppelter Signalisierung gegenüber dem ungeschützten Vergleichsszenario stärker anwachsen als bei Pfad-entkoppelter Signalisierung, unabhängig davon, welche konkreten Zahlenwerte für die Verzögerungen und Paketverlustwahrscheinlichkeiten auf den Pfaden sowie für die Bearbeitung der Signalisiernachrichten angenommen werden.

Die Notwendigkeit, Topologie-Wissen zentral vorhalten zu müssen, wird von Befürwortern der Pfad-gekoppelten Firewall-Signalisierung häufig als *das* KO-Kriterium gegen die Pfad-entkoppelte Signalisierung angeführt. Bei einer gemeinsamen Betrachtung aller in dieser Arbeit untersuchten Aspekte zeigt sich, dass dieses Argument durchaus richtig und gewichtig ist; jedoch wird der entsprechende Vorteil der Pfad-gekoppelten Signalisierung mit einer ganzen Reihe von Nachteilen erkauft. Somit dürfte die Komplexität und die Dynamik der Netztopologie tatsächlich das *entscheidende* Argument bei der Auswahl des Signalisierverfahrens werden. Ist diese hoch, z. B. in „offenen", Internet-ähnlichen Szenarien mit dynamischem Routing und Netzelementen unter der administrativen Kontrolle der Teilnehmer sowie kaskadierter Network Address and Port Translation (NAPT) im Zugangsnetz, ist das Sammeln von Topologie-Informationen und somit der Einsatz von Pfad-entkoppelter Signalisierung unverhältnismäßig aufwändig oder unmöglich; die Pfad-gekoppelte Signalisierung ist hier im Vorteil. IP-Telefonie-Plattformen

sind hingegen Netze, in denen zwar die Protokolle der TCP/IP-Protokollfamilie zum Einsatz kommen, die von ihrem Netzdesign jedoch zentralisierter und statischer als das Internet sind, z. B. indem erzwungen wird, dass alle Multimedia-Sitzungen über die zentralen SIP-Server der Control Plane signalisiert werden müssen, oder indem Zwangspunkte im Zugangsnetz geschaffen werden. In solchen Netzen, die unter der vollen administrativen Kontrolle eines Betreibers stehen, können Topologie-Informationen viel einfacher beschafft werden. Somit ist ein Einsatz der Pfad-entkoppelten Firewall-Signalisierung möglich, die als domäneninterne Lösung keine direkten Schnittstellen zu nicht vertrauenswürdigen Instanzen benötigt und so besser mit den Designprinzipien der „geschlossenen“ IP-Telefonie-Plattformen vereinbar ist.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen und Symbole

Abkürzungen

3GPP	Third Generation Partnership Project	11, 52
AAA	Authentication, Authorization and Accounting	144
ABC	Appropriate Byte Counting	123
ACL	Access Control List	28
ADSL	Asymmetric Digital Subscriber Line	11
AOR	Address of Record	18
API	Application Programming Interface	86
ARD	Asynchronous Policy Rule Deletion (MIDCOM/SIMCO-Nachricht)	71
ARE	Asynchronous Policy Rule Event (MIDCOM/SIMCO-Nachricht)	71
ARP	Address Resolution Protocol	40, 43
ARPA	Advanced Research Projects Agency	1
AS	Application Server	53
AST	Asynchronous Session Termination (MIDCOM/SIMCO-Nachricht)	71
B2BUA	SIP Back-to-Back User Agent	13, 61, 84, 89, 94
BGF	Border Gateway Function	54
BGP	Border Gateway Protocol	130
BSI	Bundesamt für Sicherheit in der Informationstechnik	43
C-BGF	Core BGF	54
C-Mode	Connection Mode	75
CCDF	Complementary Cumulative Distribution Function	122

CDR	Call Detail Records	139
COPS	Common Open Policy Service Protocol	78
CPU	Central Processing Unit	28, 99
CSCF	Call/Session Control Function, auch: Call State Control Function	53
D-Mode	Datagram Mode	75
DCCP	Datagram Congestion Control Protocol	112
DHT	Distributed Hash Table	25
DMZ	Demilitarized Zone	38
DNS	Domain Name System	17, 40, 43
DoS	Denial-of-Service (Attack)	27, 107, 136
DR	Data Receiver (NSIS)	74
DS	Data Sender (NSIS)	74
DTMF	Dual Tone Multi Frequency	8
ENUM	tElephone NUmber Mapping, auch: Electronic NUmber Mapping	19
ETSI	European Telecommunications Standards Institute	11, 52
FCP	Firewall Control Protocol	62
FRTX	Fast Retransmit	116
GIMPS	General Internet Messaging Protocol for Signaling	75
GIST	General Internet Signaling Transport	75
GL	Group List (MIDCOM-Nachricht)	71
GLC	Group Lifetime Change (MIDCOM-Nachricht)	71
GOS	Grade Of Service	146
GPL	GNU Public License	83
GPRS	General Packet Radio Service	52
GS	Group Status (MIDCOM-Nachricht)	71
GSM	Global System for Mobile Communications	11, 53
GSMA	GSM Association	56
HLR	Home Location Register	53

HSS	Home Subscriber Server	53
HTML	Hypertext Markup Language	1
HTTP	Hypertext Transfer Protocol	10, 35
I-BGF	Interconnection BGF	54
I-CSCF	Interrogating-CSCF	53
IANA	Internet Assigned Numbers Authority	41
IAT	Interarrival Time	90, 116
IAX2	Inter-Asterisk eXchange (Version 2) Protocol	24
IBCF	Interconnection Border Control Function	54
ICMP	Internet Control Message Protocol	38
IDS	Intrusion Detection System	73
IETF	Internet Engineering Task Force	10, 49
IMS	IP Multimedia Subsystem	52
IP	Internet Protocol	5
IPX	IP eXchange	56
ISDN	Integrated Services Digital Network	22
ISP	Internet Service Provider	20
ISUP	ISDN User Part	22, 50
LAN	Local Area Network	23
LF	Location Function	56
MEGACO	Media Gateway Control Protocol	24
MF	Media Function	56
MGC	Media Gateway Controller	22, 24
MGW	Media Gateway (auch: MG)	23, 24
MIB	Management Information Base	66
mmusic	Multiparty Multimedia Session Control	10
MRFC	Media Resource Function Controller	53
MRFP	Media Resource Function Processor	53

MUCS	Multimedia Conferencing System	10
NAPT	Network Address and Port Translation	31, 76, 134
NAS	Network Attachment Subsystem	54
NAT	Network Address Translation	31, 66
NF	NSIS Forwarder	75
NGN	Next Generation Network	11, 51
NI	NSIS Initiator	74
NIDS	Network Intrusion Detection System	66, 73
NIST	National Institute of Standards and Technology	43
NNI	Network-to-Network Interface	50, 55
NR	NSIS Responder	75
NSIS	Next Steps In Signaling	74
NSLP	NSIS Signaling Layer Protocols	74
NTLP	NSIS Transport Layer Protocol	74
OAM	Operation, Administration, and Maintenance	8
OSPF	Open Shortest Path First (Protocol)	43, 130
OVSt	Ortsvermittlungsstelle	50
P-CSCF	Proxy-CSCF	53
P2PSIP	Peer-to-Peer-SIP	25
PCM	Puls-Code-Modulation	7
PDR	Policy Disable Rule (SIMCO-Nachricht)	71
PEA	Policy Enable Rule After Reservation (SIMCO-Nachricht)	71
PER	Policy Enable Rule (MIDCOM/SIMCO-Nachricht)	71
PF	Policy Function	56
PID	Policy Rule Identifier	85, 114
PKI	Public Key Infrastructure	30, 45
PLC	Policy Rule Lifetime Change (SIMCO-Nachricht)	71
PRL	Policy Rule List (MIDCOM/SIMCO-Nachricht)	71

PRR	Policy Reserve Rule (MIDCOM/SIMCO-Nachricht)	71
PRS	Policy Rule Status (MIDCOM/SIMCO-Nachricht)	71
PSTN	Public Switched Telephone Network	49
QoS	Quality of Service	64
RACS	Resource and Admission Control Subsystem	54
RLC	Policy Rule Lifetime Change (MIDCOM-Nachricht)	71
RPC	Remote Procedure Call	40
RSVP	Resource ReServation Protocol	63
RTCP	RTP Control Protocol	12
RTO	Retransmission Timeout	116
RTP	Real-time Transport Protocol	12
RTT	Round-Trip Time	115
S-CSCF	Serving-CSCF	53
S/MIME	Secure/Multipurpose Internet Mail Extensions	17
SA	Session Authentication (SIMCO-Nachricht)	71
SACK	Selective Acknowledgment	116, 121
SAP	Service Access Point	89, 107
SBC	Session Border Controller	61, 63, 133
SCCP	Skinny Client Control Protocol	25
SCIP	Simple Conference Invitation Protocol	10
SCTP	Stream Control Transmission Protocol	9, 107
SDL	Specification and Description Language	81, 86
SDP	Session Description Protocol	12
SE	Session Establishment (MIDCOM/SIMCO-Nachricht)	71
SER	SIP Express Router	62
SF	Signaling Function	56
SGW	Signaling Gateway	22
SIGTRAN	Signaling Transport	22, 107

SIMCO	Simple Middlebox Configuration Protocol	72, 83
SIP	Session Initiation Protocol	10
SIPPING	Session Initiation Proposal Investigation	11
SLF	Subscription Locator Function	53
SMS	Short Message Service	9
SMTP	Simple Mail Transfer Protocol	10
SNMP	Simple Network Management Protocol	66
SPEERMINT	Session PEERing for Multimedia INTerconnect	56
SPIT	Spam over IP Telephony	44
SRTP	Secure Real-time Transport Protocol	17
SS7	Signaling System No. 7	49, 107
SSw	Softswitch	68
ST	Session Termination (MIDCOM/SIMCO-Nachricht)	71
SUNRPC	Sun Remote Procedure Call	40
TCP	Transmission Control Protocol	9, 107
TDM	Time Division Multiplex	23, 49
THIG	Topology Hiding Internetwork Gateway	54
TID	Transaction Identifier	68, 114
TISPAN	Telecoms and Internet converged Services and Protocols for Advanced Networks	11, 52
Tln	Teilnehmer	11
TLS	Transport Layer Security	17, 47
TLV	Type-Length-Value	72, 76, 89
TSN	Transmission Sequence Number	117
TSVWG	IETF Transport Area Working Group	107
UA	SIP User Agent	13
UA	User Adaptation Layer	107
UAC	User Agent Client	13
UAS	User Agent Server	13

UCE	Unsolicited Commercial Email	44
UDP	User Datagram Protocol	9, 106
ULP	Upper Layer Protocol	108
UNI	User-to-Network Interface	50, 55
UPnP	Universal Plug and Play	62
URI	Uniform Resource Identifier	17
URL	Uniform Resource Locator	35
VLAN	Virtual LAN	44
VPN	Virtual Private Network	30
VPN GW	Virtual Private Network Gateway	30
WAN	Wide Area Network	93
WEP	Wired Equivalent Privacy	47
Wkt	Wahrscheinlichkeit	xxviii
ZGS Nr. 7	Zeichengabesystem Nr. 7	8

Symbole

A	Meldeverzug (engl. *Answer Signal Delay*)	146
D	Zeitdauer bis zur Erkennung eines Paketverlustes	117
d	Zwischenankunftszeit der Signalisier-Transaktionen	116
d_{C}	Zwischenankunftszeit der Verbindungsanforderungen	91
h	Mittlere Verbindungsdauer	91, 100
L	Zwischenankunftszeit von Transaktionen zur Zustands-Auffrischung	91
M	Optimale Anzahl von SCTP Streams	119
m	Anzahl gleichzeitiger Multimedia-Sitzungen	91, 100
N	Anzahl der Streams (pro Richtung) in einer SCTP-Assoziation	117
N_{C}	Anzahl paralleler Netzübergänge zwischen zwei Sicherheits-Domänen	144
N_{D}	Anzahl von Sicherheits-Domänen zwischen zwei Teilnehmern	144
N_{H}	Anzahl kryptographischer Hash-Operationen	147
N_{M}	Anzahl zusätzlicher Signalisiernachrichten zur Firewall-Steuerung	147
N_{U}	Gesamtzahl der Policy Rules pro Multimedia-Sitzung in allen Paketfiltern	147
P	Rufverzug (engl. *Post-selection Delay*)	146
p_{F}	Paketverlustwahrscheinlichkeit erlaubter Pakete im Paketfilter	99
p_{L}	Paketverlustwahrscheinlichkeit auf Ende-zu-Ende-Pfad	115
p_{S}	Wkt. einer erfolgreichen Übertragung von Datensegment und Quittierung	115
Q	Anzahl der zur Reihenfolgesicherung zu puffernden Datenblöcke	119
R	Antwortzeit einer Transaktion (z. B. SIMCO)	70, 116, 146
RTO	Zeitdauer vor Übertragungswiederholung (engl. *Retransmission Timeout*)	118
RTT	Umlaufzeit von IP-Paketen im Netz (engl. *Round-Trip Time*)	115
TO	Zeitüberwachung (engl. *Timeout*)	120
U	Anzahl der Policy Rules im Paketfilter	91, 100
u	Anzahl der Policy Rules pro Multimedia-Sitzung pro Paketfilter	91, 100
V	Dauer des unidirektionalen Transports einer Signalisiernachricht	149

W	Verzögerung durch das Transportschichtprotokoll	116
Δ	Unidirektionale Ende-zu-Ende-Verzögerung auf der IP-Schicht	115
δ	Verarbeitungsdauer einer Transaktion im Server	70, 116, 148
δ_{F}	Mittlere Verzögerung eines erlaubten Pakets im Paketfilter	99
δ_{H}	Bearbeitungsdauer zum Erstellen bzw. Prüfen eines Hash-Wertes	148
δ_{S}	Bearbeitungsdauer einer SIP-Nachricht im B2BUA	148
δ_{SI}	Bearbeitungsdauer einer SIP *INVITE*-Nachricht im B2BUA	148
δ_u	Dauer zum Hinzufügen einer Policy Rule in den Paketfilter	99
λ_{C}	Rate der Verbindungsanforderungen	91, 100
λ_{F}	Rate der Regelvergleiche im Paketfilter	101
λ_{i}	Rate der Pakete, die an den Paketfilter gesendet werden	101
λ_{M}	Mittlere Paketrate eines Medienstroms (unidirektional)	101
λ_{T}	Rate der Signalisier-Transaktionen	116
λ_{u}	Rate neuer Policy Rules	91
λ_{uc}	Rate der Anforderungen zum Hinzufügen oder Löschen einer Policy Rule	100
λ_{X}	Mittlere Paketrate des abzuwehrenden Angriffsverkehrs	101

1 Einleitung

1.1 Umfeld und Motivation

Spätestens mit der Einführung der digitalen Signalübertragung in Kommunikationsnetzen kam auch der Wunsch auf, verschiedene Kommunikationsdienste auf Basis einer gemeinsamen Netzinfrastruktur anzubieten und nicht, wie in der Anfangszeit der Telekommunikation, für jeden Übertragungsdienst (z. B. Telegraphie, Telefonie, Rundfunk- und Fernsehübertragung, Videokonferenzen) eigene Netze zu schaffen. Zumindest international betrachtet blieb die Nutzung des auf kanalorientierter Vermittlung basierenden Integrated Services Digital Network (ISDN) und seines Nachfolgers Breitband-ISDN jedoch recht überschaubar; ihre Verbreitung wurde von der des Internets bei weitem übertroffen.

Basierend auf dem Prinzip der Paketvermittlung entstand dieses ab den späten 1960er-Jahren zunächst unter dem Namen Arpanet als Verbund von Datennetzen an wissenschaftlichen und militärischen Einrichtungen. Bei den angeschlossenen Endsystemen handelte es sich um Großrechner; an eine wirklich interaktive Nutzung oder gar Echtzeit- und Multimedia-Dienste war mit der damaligen Rechnertechnik nicht zu denken. Weitere Entwurfsprinzipien neben der Paketvermittlung waren von Anfang an das Streben, möglichst viele Funktionen in den Endsystemen und nicht – wie z. B. beim klassischen Telefonnetz – in zentralen Knoten „im Netz“ zu erbringen (das so genannte Ende-zu-Ende-Prinzip). Dementsprechend wurde auch auf eine Unterscheidung zwischen Teilnehmer- und Zwischenamts-Signalisierung verzichtet, überhaupt wird in Netzen, die auf dem Internet Protocol (IP) basieren, vergleichsweise wenig signalisiert. Damals wie heute erfolgt im Internet auch keine aufwändige Verwaltung und Reservierung von Ressourcen; stattdessen kommen verteilte Algorithmen zur Schätzung von Pfad-Kapazitäten und zur Überlastabwehr zum Einsatz. In den frühen 1980er Jahren wurden mit IPv4, TCP, DNS die Basisprotokolle des Internets in ihren bis heute verwendeten Versionen eingeführt. Die Verbreitung preiswerter Personal Computer, die Spezifikation von HTTP und HTML als Basistechnologien des „World Wide Web“, sowie die Freigabe für die kommerzielle Nutzung durch die US-Regierung führten ab Anfang der 1990er-Jahre zu einem explosionsartigen Wachstum. Die Kommerzialisierung ging einher mit einem massiven Ausbau der Übertragungskapazitäten des Kernnetzes – längst wird die Kapazität der konventionellen Telefonnetze um Größenordnungen übertroffen und aus dem zeitweise spöttisch als „World Wide Wait“ titulierten Netz ist eine leistungsfähige Infrastruktur geworden – das Internet ist aus unserer heutigen Welt praktisch nicht mehr wegzudenken.

Aufgrund dieser Leistungssteigerung der Infrastruktur und der angeschlossenen Endgeräte ist mittlerweile auch die Übertragung echtzeitkritischer Multimedia-Datenströme über IP üblich; in

vielen Szenarien haben sich adaptive Kodierungsverfahren sowie ggf. die Ergänzung des „best effort“-Transports um eine einfache, statische Priorisierung bestimmter IP-Pakete als vollkommen ausreichend erwiesen, um qualitativ hochwertige Sprach- und Videoübertragungen zu ermöglichen. Bereits heute ist „Voice over IP“ (VoIP) für preisgünstige Ferngespräche über das Internet oder als Ersatz für private Nebenstellenanlagen weit verbreitet. Dabei kommen teilweise proprietäre Protokolle zum Einsatz, in jüngerer Vergangenheit aber vermehrt das standardisierte Session Initiation Protocol (SIP). Noch handelt es sich bei diesen Lösungen überwiegend um „VoIP-Inseln“, die jeweils von einem einzigen Anbieter betrieben werden; domänenübergreifende Verbindungen werden häufig noch mit Umweg über die etablierten Telefonnetze realisiert, selbst wenn beide Teilnehmer VoIP nutzen.

Aufgrund dieser neuen Möglichkeiten und der in die Jahre gekommenen Infrastruktur der klassischen PSTN/ISDN-Telefonnetze wird erneut über eine Konvergenz der Netze nachgedacht; diesmal mit dem Ziel, alle Dienste oberhalb eines IP-basierten Transports abzuwickeln. Ein zentraler Problembereich bei der Umsetzung dieses Vorhabens ist die Netzsicherheit und eng damit verbunden die Frage, wie Bereiche, die unter verschiedener administrativer Kontrolle stehen, sicher zusammengeschaltet werden können. Erfahrungen mit anderen Kommunikationsdiensten im Internet (z. B. Electronic Mail) haben gezeigt, dass unter anderem die „offenen“ Strukturen des Internets sowie die komplexen und damit u. U. leichter verwundbaren Protokollinstanzen in den Endsystemen zu Sicherheitsproblemen geführt haben, die im ISDN zumindest in dieser Schärfe nicht aufgetreten sind, z. B. die Kompromittierung von Endsystemen oder die milliardenfache Verbreitung unerwünschter Werbebotschaften. Viele dieser Probleme können durch kryptographischen Schutz der Nachrichten gelöst werden. Andere Bedrohungsszenarien, z. B. Angriffe gegen die Verfügbarkeit der Infrastruktur durch Senden von Paketfluten, können aufgrund der Strukturen des Internets kaum abgewehrt werden.

Ein Ansatz zur Lösung dieser Probleme ist die Errichtung von Netzen, die zwar das IP-Protokoll und insbesondere SIP verwenden, jedoch vom Internet vollständig getrennt sind und – verglichen mit dem Internet – viel stärker in Domänen mit ggf. unterschiedlichen Sicherheitsrichtlinien eingeteilt sind. Die Durchsetzung dieser Richtlinien erfolgt vor allem an den Bereichsgrenzen durch Zugriffskontrollen mit Hilfe von Firewalls bzw. Gateways. Bei der ITU-T werden Überlegungen zu solchen Netzarchitekturen unter dem Stichwort *Next Generation Network* (NGN) zusammengefasst, es gibt aber auch ähnliche Ansätze bei anderen Hersteller- oder Betreiber-Konsortien, so dass der Sammelbegriff „IP-Telefonie-Plattform“ verwendet werden soll.

Die Aufgaben der Firewalls an den Domänengrenzen lassen sich grob in zwei Bereiche gliedern: Zugriffskontrollen auf die Signalisierung und solche auf die Medienströme. Nur wenn der Aufbau einer Multimedia-Sitzung ordnungsgemäß signalisiert wurde, zu den Sicherheitsrichtlinien konform ist und alle Betroffenen zugestimmt haben, soll auch das Fließen der entsprechenden Medienströme erlaubt sein. Die signalisierungs- bzw. medienbezogenen Funktionen einer Firewall können in einem gemeinsamen oder in getrennten Netzelementen platziert werden. Bei getrennter Platzierung wird für die Koordination zwischen beiden Funktionen ein Steuerprotokoll benötigt. Hierfür existieren verschiedene Grund-Architekturen, die sich u. a. darin unterscheiden, von welcher Instanz die Signalisierung ausgeht und wie die entsprechenden Nachrichten durch das Netz geleitet werden. Diese wurden in verschiedenen Protokollen standardisiert. Ziel dieser Arbeit ist eine Bewertung dieser Ansätze, sowohl bezüglich funktionaler und sicherheitsbezogener Aspekte, als auch bezüglich eventueller Auswirkungen auf die Dienstgüte.

1.2 Gliederung der Arbeit

Ziel dieser Arbeit ist die Bewertung verschiedener Architekturen zur Steuerung von Netzelementen, die durch Zugriffskontrollen an Bereichsgrenzen die Sicherheit IP-basierter Telefonie erhöhen sollen. In Kapitel 2 werden daher die Grundkonzepte der SIP-basierten IP-Telefonie, sowie der Netzsicherheit und Firewalls zunächst unabhängig voneinander dargestellt, bevor in Kapitel 3 gezeigt wird, wie Firewalls die Sicherheit der IP-Telefonie erhöhen können, aber auch welche Probleme dabei auftreten können. Ferner wird aufgezeigt, dass auf dem Internet Protocol basierende Telefonie nicht notwenigerweise im Internet stattfinden muss; am Beispiel der 3GPP IMS-Architektur werden Ansätze für separate Infrastrukturen illustriert.

In Kapitel 4 werden verschiedene Architekturvarianten für die Zugriffskontrolle auf Signalisierung und Medienströme klassifiziert. Es werden die beiden grundsätzlichen Signalisierverfahren zur Steuerung der Medienkomponenten verteilter Firewalls, die Pfad-entkoppelte und die Pfad-gekoppelte Signalisierung, dargestellt. Mit den Architekturen und Protokollen, die im Umfeld der IETF MIDCOM- und NSIS-Arbeitsgruppen entworfen und standardisiert wurden, wird je eine konkrete Ausprägung dieser Grundverfahren vorgestellt.

Kapitel 5 beschreibt die prototypische Implementierung des SIMCO-Protokolls und den Aufbau einer Testumgebung, mit der das Zusammenspiel von SIP-basierter IP-Telefonie und Pfad-entkoppelter Firewall-Steuerung untersucht und die Machbarkeit des Ansatzes gezeigt wird. Ergänzend werden einige Messergebnisse zur Leistungsfähigkeit der Paketfilterung mit dem Betriebssystem Linux präsentiert.

Die dynamische Konfiguration der Firewalls trägt zu den Verzögerungen beim Aufbau einer Multimedia-Sitzung bei, die von den Teilnehmern als störend empfunden werden. Um diesen Beitrag zu verringern, muss neben der Verarbeitung der Signalisiernachrichten in den Netzknoten auch ihr Transport über das IP-Netz optimiert werden. In Kapitel 6 werden daher zunächst verschiedene Transportschichtprotokolle und Konfigurationsmöglichkeiten für den Transport von Signalisiernachrichten über IP vorgestellt. Am Beispiel von SIMCO wird untersucht, welche Anpassungen an einem Signalisierprotokoll dafür ggf. notwendig sind, ein auf dem Stream Control Transmission Protocol (SCTP) basierender, optimierter Transport wird spezifiziert. Anschließend werden Verzögerungen beim Nachrichtentransport durch Messungen am „SIMCO over SCTP“-Prototypen und analytische Modelle quantifiziert. Dabei wird insbesondere auf den Zusammenhang zwischen einer – evtl. gar nicht benötigten – Reihenfolgesicherung in der Transportschicht und dem verzögernden Effekt des Head-Of-Line Blocking eingegangen. Dieser tritt auf, wenn infolge eines Paketverlustes Nachrichten erneut übertragen werden und darauffolgende Nachrichten zur Reihenfolgesicherung empfängerseitig gepuffert werden müssen. Die hier gewonnenen Resultate sind auch auf andere Signalisierprotokolle übertragbar.

Basierend auf den Untersuchungen einzelner Mechanismen in den vorangegangenen Kapiteln werden in Kapitel 7 die beiden grundsätzlichen Signalisierverfahren miteinander verglichen, wobei von einem Szenario ausgegangen wird, in dem die IP-Telefonie-Plattformen mehrerer Betreiber zusammengeschaltet wurden. Der Vergleich erfolgt sowohl bezüglich funktionaler und sicherheitsrelevanter Aspekte, als auch quantitativ bezüglich der Auswirkungen auf die Verzögerungen beim Sitzungsaufbau. Kapitel 8 schließt diese Arbeit mit einer Zusammenfassung der Ergebnisse ab.

2 Grundlagen

IP-Telefonie-Plattformen sind auf dem *Internet Protocol* (IP) basierende Kommunikationsnetze, über die Multimedia-Kommunikation – insbesondere Telefonie – abgewickelt werden kann. Zur Erhöhung der Netzsicherheit, sowie ggf. aus weiteren Gründen (z. B. wirtschaftlicher, politischer oder regulatorischer Natur) sind solche Netze in Bereiche eingeteilt. Netzverkehr, der entsprechende Bereichsgrenzen überschreitet, wird dort Zugriffskontrollen unterworfen. Ziel dieser Arbeit ist die Bewertung von Architekturen zur Steuerung von Netzelementen, die solche Zugriffskontrollen auf Medienströme durchführen. Bevor auf die Architekturen solcher IP-Telefonie-Plattformen näher eingegangen wird, sollen in diesem Kapitel daher zunächst die Grundlagen der IP-Telefonie sowie der Netzsicherheit – insbesondere der Zugriffskontrolle am Netzübergang mit Hilfe so genannter *Firewalls* – erläutert werden.

2.1 VoIP und IP-Telefonie

In diesem Abschnitt soll zunächst IP-Telefonie gegen andere Ausprägungen von IP-basierter Multimedia-Kommunikation abgegrenzt werden. Anschließend wird näher auf die grundsätzlichen Funktionen der Multimedia-Kommunikation eingegangen, insbesondere auf das Session Initiation Protocol (SIP), welches den de-facto Standard für die Signalisierung solcher Anwendungen darstellt.

2.1.1 Begriffsdefinitionen, Einordnung und Abgrenzung

Das betrachtete Problemfeld befindet sich im Umfeld der IP-basierten Multimedia-Kommunikation. Die hierfür üblicherweise verwendeten Protokolle, welche in den folgenden Abschnitten vorgestellt werden, weisen eine hohe Flexibilität auf und können daher für eine Vielzahl von Anwendungen und Szenarien verwendet werden, die zumindest aus funktionaler Sicht recht ähnlich sind. Teilweise werden die Begriffe *Voice over IP (VoIP)*, *Internet-Telefonie*, *IP-Telefonie* und *Multimedia-Kommunikation über IP* praktisch synonym verwendet [12]. Betrachtet man hingegen die nichtfunktionalen Eigenschaften der vorgeschlagenen Lösungen, neben der Dienstgüte insbesondere die Netzsicherheit, so können diese Begriffe gegeneinander abgegrenzt werden. Sie sollen in dieser Arbeit wie folgt verwendet werden:

Als *Multimedia-Kommunikation* wird der Transport von Ton- und Video-Daten in Quasi-Echtzeit oder Echtzeit bezeichnet. Eine wichtige Kenngröße ist dabei die Verzögerung, die die Multimedia-Daten durch die Analog-Digital-Wandlung im Sender, die Codierung, das Aufteilen in

IP-Pakete, den Transport im Netz und die entsprechende Wiedergabe im Empfänger erfahren. Während für *unidirektionale* Übertragung, wie z. B. IP-basierte Fernsehausstrahlung (IPTV), eine Verzögerung im Bereich von Sekundenbruchteilen bis u. U. wenigen Minuten noch akzeptabel erscheint (Quasi-Echtzeit), so ist eine sinnvolle *bidirektionale* Kommunikation (z. B. Telefonie) nur unter Echtzeit-Bedingungen möglich. Laut [G.114] soll die Verzögerung im Telefon-Netz nicht über 150 ms liegen (gemessen „Mund-zu-Ohr“ in eine Richtung). Nicht in die Kategorie der Multimedia-Kommunikation fallen Anwendungen, bei denen Multimedia-Daten in einem Rechner zunächst als Datei (z. B. im MP3-Format [13]) aufgezeichnet werden und dann mit beliebig großem Zeitversatz mit Protokollen zur Dateiübertragung (z. B. HTTP, div. „filesharing tools“, etc.) transportiert werden.

Für die bidirektionale Echtzeit-Multimedia-Kommunikation über IP-basierte Netze wird der Sammelbegriff *Voice over IP (VoIP)* verwendet. Darunter fällt unter anderem die IP-Telefonie. Ein wichtiges Merkmal der *IP-Telefonie* ist, dass sie eine Bedienerführung besitzt, die der der herkömmlichen Telefonnetze ähnelt (z. B. Verbindungsaufbau wird unilateral vom rufenden Teilnehmer initiiert, der gerufene Teilnehmer wird durch Klingeln o. ä. über die eingehende Verbindungsanforderung informiert, etc.). Ein Spezialfall der IP-Telefonie ist die *Internet-Telefonie*. Hierbei erfolgt die Übertragung über das öffentliche Internet, was Auswirkungen auf die Dienstgüte und die Sicherheit haben kann.

Alternativ kann IP-Telefonie auch in *IP-Telefonie-Plattformen* durchgeführt werden. Dabei handelt es sich um Netze, die zwar das IP-Protokoll verwenden, jedoch vom öffentlichen Internet vollständig getrennt bzw. nur über Gateways erreichbar sind. Die Errichtung solcher getrennten Netze kann wirtschaftliche Gründe haben, oder zur Erhöhung der Sicherheit oder Dienstgüte erfolgen. Eine Übersicht über solche IP-Telefonie-Plattformen wird in Abschnitt 3.3.2 gegeben.

2.1.2 Sprach-/Ton-Übertragung über IP-basierte Netze

Grundsätzlich müssen bei der IP-Telefonie zwei Arten von Daten über das Kommunikationsnetz übertragen werden: die eigentlichen *Nutzdaten* und die *Signalisiernachrichten*.

Der prinzipielle Ablauf der Übertragung von Audio-Daten ist in Abbildung 2.1 [14] dargestellt. Senderseitig wird das analoge Signal zunächst periodisch abgetastet und in ein digitales Signal

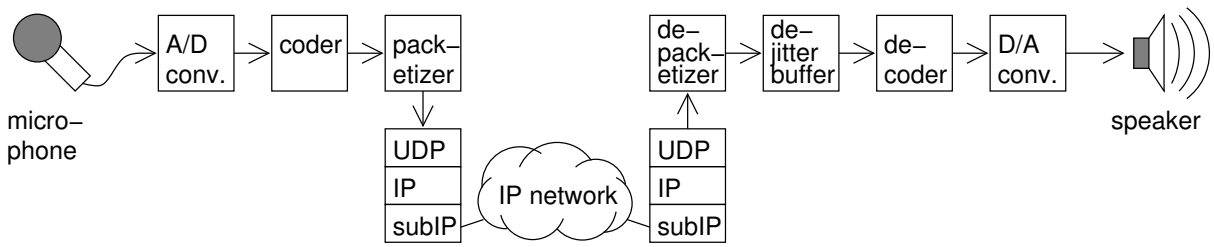


Abbildung 2.1: Sprachübertragung über IP-basierte Netze (Prinzip)

Anmerkung: Da die meisten Textelemente in den Abbildungen englischsprachige Abkürzungen, Fachbegriffe und Namen sind, wurden – zugunsten eines einheitlichen Erscheinungsbildes – auch die wenigen „normalen“ Wörter in englischer Sprache dargestellt.

gewandelt. Anschließend wird dieses Signal in einen Byte-Strom kodiert. Dabei können verschiedene Algorithmen angewendet werden, von der einfachen *Puls-Code-Modulation* (PCM) [G.711], die jedem Abtastwert ein Codewort zuordnet, bis hin zu komplexen Verfahren (z. B. [G.729, 15]), die Sprechpausen erkennen oder die Komplexität des Sprachsignals entsprechend psychoakustischer Modelle reduzieren, um die Netto-Datenrate hinter dem Kodierer zu reduzieren. Der so entstandene, kontinuierliche Byte-Strom wird in Datenblöcke sinnvoller Größe aufgeteilt, die ggf. zusammen mit Zeitstempeln zur synchronen Wiedergabe sowie ggf. weiteren Steuerinformationen als Nutzlast von IP-Paketen übertragen werden. Hierfür wird häufig das *Real-time Transport Protocol* (RTP, s. u.) verwendet.

Bei der empfängerseitigen Wiedergabe werden prinzipiell diese Verarbeitungsschritte rückgängig gemacht. Zusätzlich ist ein so genannter *De-Jitter Buffer* notwendig. Dieser gleicht Unterschiede in der Laufzeit der einzelnen IP-Pakete durch das Netz aus, die in einem paketvermittelnden Netz prinzipbedingt auftreten können.

Bei der beschriebenen Kette von Verarbeitungsschritten gibt es eine ganze Reihe von Parametern, die die von den Teilnehmern wahrgenommene Sprachqualität beeinflussen. Darunter gehören Parameter, die nicht beeinflusst werden können, z. B. die Verzögerung aufgrund des geographischen Abstandes der Teilnehmer, und solche, die zumindest teilweise beeinflusst werden können. Dazu zählen unter anderem die Dimensionierung des zu Grunde liegenden IP-Netzes, ggf. das Vorhandensein von Mechanismen zur Priorisierung des Echtzeit-Verkehrs vor anderen Verkehren, sowie die Wahl des Kodierungsverfahrens und der Zeitabstände, in denen der Paketisierer ein Paket abschickt. Die Erforschung dieser Freiheitsgrade sowie die Weiterentwicklung von Verfahren zur Sprachübertragung über IP ist Gegenstand vieler wissenschaftlicher Arbeiten (siehe z. B. [16, 17]) und Aktivitäten von Standardisierungsgremien; in dieser Arbeit sollen solche Fragestellungen hingegen nicht weiter vertieft werden.

2.1.3 Signalisierung

Signalisierung, auch als *Zeichengabe* bekannt, ist die „Übermittlung von Information zu Steuerungszwecken. Im Bereich der Telekommunikation werden diese Nachrichten meist in codierter Form übermittelt. Zweck der Zeichengabe ist die Verständigung von Teilnehmern des Netzes mit dem Netz bzw. mit Einrichtungen des Netzes und umgekehrt“ [18].

Die wichtigste Aufgabe der Signalisierung bei der Telefonie ist die *Verbindungssteuerung*. Sie sorgt während der *Verbindungsaufbau-Phase* dafür, dass in den Endsystemen (Telefonapparat o. ä. beim Teilnehmer), sowie – je nach verwendeter Technologie – ggf. auch in den vermittelnden Netzknoten dazwischen, Zustandsübergänge ausgelöst werden, die während der *Nutzdatenaustausch-Phase* die Übertragung der eigentlichen Audio-Daten (siehe Abschnitt 2.1.2) erst ermöglichen. Auch während der Nutzdatenaustausch-Phase kann es zum Austausch von Signalisiernachrichten kommen, genauso wie bei der anschließenden *Verbindungsabbau-Phase*.

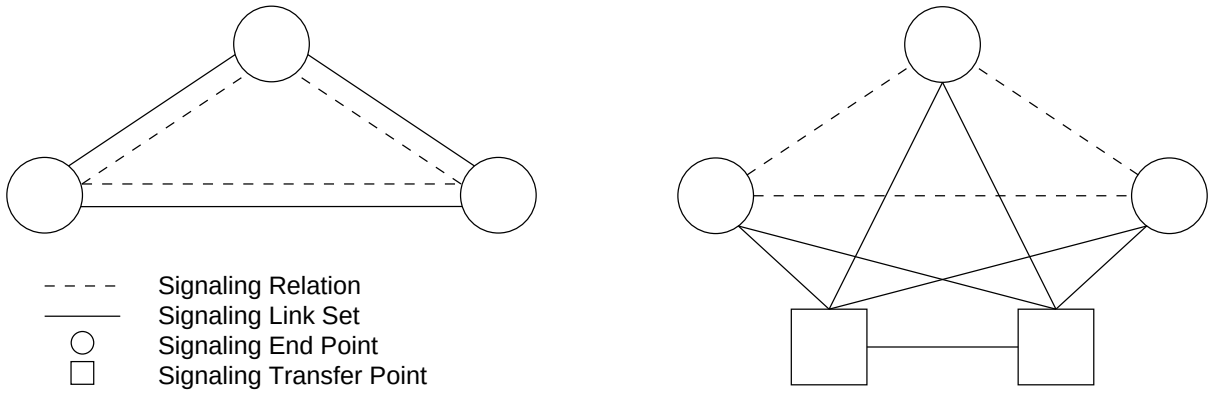
Zu den grundlegenden Aufgaben beim Verbindungsaufbau können unter anderem gehören (je nach verwendeter Technologie): die Alarmierung des gerufenen Teilnehmers, eine Abfrage, ob dieser das Gespräch annehmen will, das Aushandeln von Parametern für die Sprachübertragung (z. B. Kodierungsverfahren), das Finden eines Weges vom rufenden zum gerufenen Teilnehmer, sowie das Durchschalten von Sprachkanälen oder die Reservierung von Ressourcen für die

Übertragung. Daneben können weitere Funktionen im Umfeld der Verbindungssteuerung benötigt werden, z. B.: Rufnummernportierung, Mobilitätsunterstützung, Anrufweiterschaltung, Prüfen vorausbezahlter Guthaben (engl. *Pre-paid Accounts*), etc. Viele dieser Funktionen werden unter dem Stichwort „Mehrwertdienste“ zusammengefasst. Desweiteren wird Signalisierung für eine Reihe von Aufgaben in Telefon-Netzen verwendet, die nicht in unmittelbarem Zusammenhang mit der Verbindungssteuerung stehen, z. B. das Einbringen neuer Teilnehmer (engl. *Provisioning*), Verwaltungs- und Fehlerbehebungsmassnahmen (engl. *Operation, Administration, and Maintenance*, OAM), sowie die Entgelterfassung.

2.1.4 Innenband- und Außenband-Signalisierung

Ein wichtiges Unterscheidungsmerkmal verschiedener Protokoll-Architekturen ist, wie die Signalisier- und Nutzdaten transportiert werden. Werden sie im selben *Kanal* übertragen, so spricht man von Innenband-Signalisierung (engl. *In-band Signaling*). Erfolgt die Übertragung hingegen in getrennten Kanälen, so spricht man von Außenband-Signalisierung (engl. *Out-of-band Signaling*) [14]. Ein typisches Beispiel aus dem Bereich der konventionellen Telefon-Netze ist die Innenband-Signalisierung auf der analogen Teilnehmerschnittstelle mit DTMF-Tonwahl [Q.23], Hörtönen („Freiton“, „Besetztton“, etc.) und Ansagen (z. B. „Kein Anschluss unter dieser Nummer“), sowie die Außenband-Signalisierung zwischen den Vermittlungsstellen mit Hilfe des *Zeichengabesystem Nr. 7* (ZGS Nr. 7, engl. *Signaling System No. 7* (SS7)[Q.700]).

Bei Verwendung von Außenband-Signalisierung werden insbesondere in den Fernvermittlungsstellen im Kernnetz weniger Betriebsmittel benötigt, da die signalisierungsbezogenen Komponenten für viele Nutzkanäle gemeinsam benutzt werden können und nicht für jeden Nutzkanal eines jeden Bündels entsprechende eigene Leitungsendeinrichtungen vorgehalten werden müssen. Vorteilhaft ist auch die bessere Trennung zwischen Nutzdaten und Signalisiernachrichten. So war es zu Zeiten der analogen Innenband-Signalisierung zwischen Fernvermittlungsstellen für die Teilnehmer unter Umständen möglich, mit Hilfe selbst gebauter Tongeneratoren (so genannte „blue box“ u. ä.) diese Zwischenamts-Signalisierung zu imitieren. Somit konn-



(a) Prinzip eines rein assoziierten Zeichengabenetzes

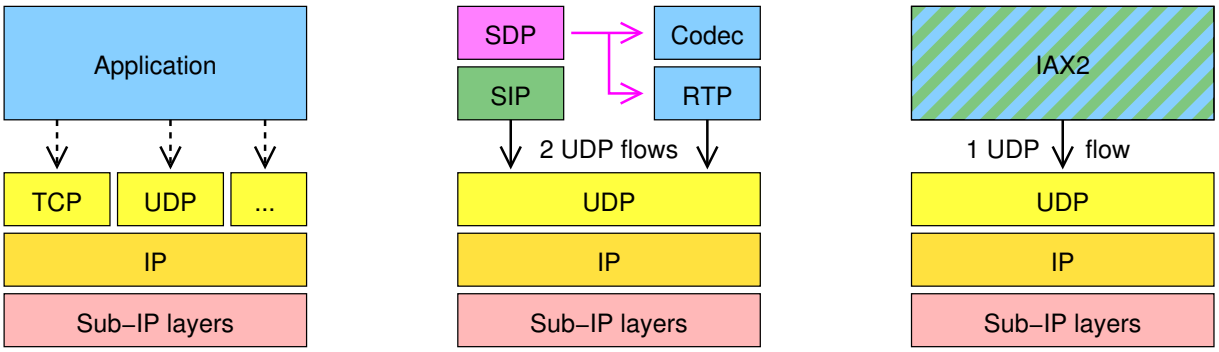
(b) Prinzip eines rein quasiassoziierten Zeichengabenetzes

Abbildung 2.2: Prinzip der assoziierten und quasiassoziierten Signalisierung im ZGS Nr. 7

ten die Vermittlungsstellen im Netz in einen inkonsistenten Zustand gebracht werden und z. B. Nutzkanäle belegt und Gespräche geführt werden, ohne die entsprechende Entgelterfassung zu aktivieren [19]. Ein weiterer wesentlicher Vorteil dieses Verfahrens ist, dass Nutzdaten und Signalisiernachrichten über räumlich getrennte Kanäle und unterschiedliche Netzelemente geführt werden können (siehe Abbildung 2.2, nach [18]). Dies kann u. a. dazu verwendet werden, bestimmte Steuerungsfunktionen wie z. B. Mobilitätsunterstützung im Netz zu zentralisieren, die echtzeitkritischen Multimedia-Datenströme hingegen auf dem kürzest möglichen Pfad zwischen den beiden Teilnehmern durch das Netz zu führen. Zur Erbringung bestimmter Dienste wie z. B. Kurznachrichten (engl. *Short Message Service*, SMS) oder in Situationen, bei denen schon in frühen Phasen der Signalisierung festgestellt wird, dass die Verbindung nicht zu Stande kommen kann (z. B. gerufener Teilnehmer besetzt) muss u. U. gar kein TDM-basierter Nutzkanal durchgeschaltet werden.

Um die Unterscheidung zwischen Innenband- und Außenband-Signalisierung im Umfeld IP-basierter Netze anwenden zu können, muss zunächst eine sinnvolle Definition des *Kanals* gefunden werden. Die betrachteten Protokolle für VoIP-Medientransport und/oder -Signalisierung sind in der Anwendungsschicht des vierschichtigen *ARPANET Reference Model* [RFC 871] einzuordnen. Zwischen Instanzen dieser Protokolle können Daten mit Hilfe der darunterliegenden Transportschicht verbindungsorientiert (TCP, SCTP) oder verbindungslos (UDP) transportiert werden. Verbindungen des *Transmission Control Protocol* (TCP) [RFC 793], Assoziationen des *Stream Control Transmission Protocol* (SCTP) [RFC 2960] bzw. Datagramm-Flüsse des *User Datagram Protocol* (UDP) [RFC 768] zwischen 2 Protokollinstanzen der Anwendungsschicht werden jeweils durch ein 5-Tupel (Quell-IP-Adresse, Ziel-IP-Adresse, Transportprotokoll-Bezeichner, Quell-Portnummer, Ziel-Portnummer) eindeutig identifiziert.

In dieser Arbeit soll daher von IP-basierter Innenband-Signalisierung gesprochen werden, wenn Nutzdaten und Signalisierung über eine gemeinsame TCP-Verbindung, SCTP-Assoziation, oder einen UDP-Flow transportiert werden, welche(r) durch ein solches 5-Tupel gekennzeichnet ist. Werden hingegen getrennte Transportschicht-Verbindungen/Assoziationen/Flows verwendet, so soll dies als Außenband-Signalisierung bezeichnet werden. Dies wird in Abbildung 2.3 verdeutlicht.



a) ARPANET reference model

b) out–of–band signaling with SIP

c) in–band signaling with IAX2

Abbildung 2.3: ARPANET Referenz-Modell, Außenband- und Innenband-Signalisierung

2.1.5 Session Initiation Protocol (SIP)

2.1.5.1 Historische Entwicklung des Protokolls

Das *Session Initiation Protocol* (SIP) wurde ursprünglich im Umfeld der *Multiparty Multimedia Session Control* (mmusic)-Arbeitsgruppe der *Internet Engineering Task Force* (IETF) entwickelt. Diese beschäftigte sich Mitte der 1990er Jahre mit Multimedia-Kommunikation, die auf *IP-Multicasting* [RFC 3170] basiert, z. B. Videokonferenzen mit virtuellem Whiteboard. Für die Ankündigung solcher Multimedia-Sitzungen (engl. *Sessions*) sollte ein Signalisierprotokoll geschaffen werden. Die Telefonie mit nur zwei Teilnehmern und IP-Unicast-Transport (siehe Abbildung 2.4), heute eine der wichtigsten Anwendungen von SIP, ist also nur ein Spezialfall des Anwendungsspektrums.

Henning Schulzrinne präsentierte im März 1996 in [20] das „Multimedia Conferencing System“ (MUCS) samt zugehörigem MUCS-Signalisierprotokoll. Dieses besitzt schon wesentliche Eigenschaften des späteren SIP, verwendet teilweise aber noch andere Terminologie. Am 22. März 1996 wurden zwei Internet-Drafts veröffentlicht, mit den Titeln „Simple Conference Invitation Protocol“ (SCIP) [21] von Henning Schulzrinne bzw. „Session Invitation Protocol“ [22] (man beachte das „Invitation“!) von Mark Handley und Eve Schooler, die in ihrer Arbeit Schulzrinnes MUCS als Basis referenzieren. Beide Dokumente beschreiben Prinzipien und Mechanismen, beinhalten aber noch keine vollständige Protokollspezifikation. Viele Konzepte wurden dabei von SMTP [RFC 821] und HTTP 1.1 [RFC 2616] übernommen. Im Dezember 1996 wurde von den drei Autoren gemeinsam eine verfeinerte Version der SIP-Spezifikation veröffentlicht [23]. Seit diesem IETF Draft steht SIP für „Session Initiation Protocol“.

Am 2. Februar 1999 wurde die bis dahin stark weiterentwickelte Spezifikation als „proposed standard“ bestätigt und am 17. März 1999 als [RFC 2543] veröffentlicht. Nach abermaliger gründlicher Überarbeitung im Umfeld der eigens gegründeten SIP-Arbeitsgruppe wurde am 3. Juli 2002 dieses Dokument durch die RFC 3261 bis 3266 [RFC 3261, ff.] ersetzt, welche bis heute (Ende 2006) die „Kernspezifikation“ von SIP darstellen. Seither wurde aber eine Vielzahl von optionalen Protokollerweiterungen entwickelt. In [24] wird ein Überblick über die wichtigsten IETF-Standards bzgl. SIP gegeben; dabei werden über einhundert andere Dokumente referenziert. SIP ist damit das komplexeste Protokoll, das je von der IETF standardisiert wurde.

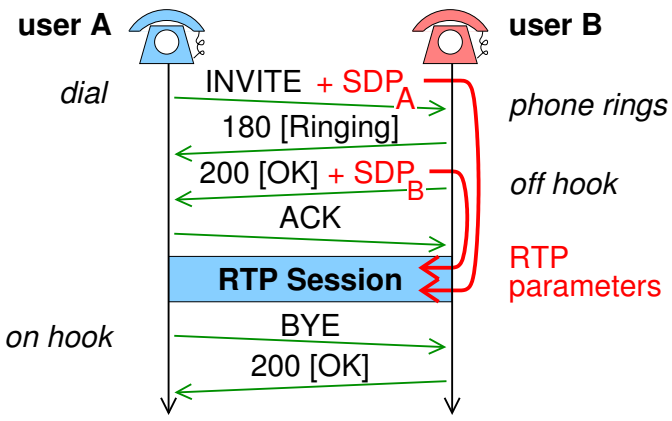


Abbildung 2.4: Einfacher Sitzungsaufbau mit SIP/SDP und RTP

Im Jahr 1998 wurde das *Third Generation Partnership Project* (3GPP) [25] als Kooperation verschiedener regionaler Standardisierungsorganisationen im Bereich der Telekommunikation gegründet, darunter u. a. das *European Telecommunications Standards Institute* (ETSI). Ziel von 3GPP ist die Weiterentwicklung des GSM zu einem Mobilkommunikationssystem der dritten Generation. Neben der Weiterentwicklung der drahtlosen Übertragungstechnik gehört dazu auch die Umstellung von kanalorientierter Vermittlung zur Paketvermittlung. Den Kern dieser Architektur bildet das *IP Multimedia Subsystem* (IMS, siehe Abschnitt 3.4). Schon frühzeitig wurde die Wichtigkeit einer Interoperabilität mit dem Internet erkannt und beschlossen, nach Möglichkeit die Internet-Standards der IETF unverändert zu verwenden. Benötigte Funktionalitäten, die von vorhandenen IETF-Standards nicht abgedeckt werden, sollen in den IETF-Arbeitsgruppen diskutiert werden und zur Ergänzung der IETF-Standards führen. Die Zusammenarbeit zwischen 3GPP und IETF wird in [RFC 3113] beschrieben. Für die Sitzungssignalisierung wurde von 3GPP das zu diesem Zeitpunkt in [RFC 2543] spezifizierte SIP ausgewählt.

Da von 3GPP und anderen Organisationen, die ein Interesse an der Verwendung von SIP gefunden hatten, eine Vielzahl von Wünschen bzgl. der Erweiterung von SIP an die IETF herangetragen werden, wurde dort die *Session Initiation Proposal Investigation* (SIPPING)-Arbeitsgruppe gegründet. Diese hat die Aufgabe, spezifische Anwendungsszenarien von SIP zu untersuchen und ggf. Anwendungs-Richtlinien herauszugeben. Ferner sollen in dieser Arbeitsgruppe Erweiterungswünsche bzgl. SIP gesammelt, klassifiziert und priorisiert an die IETF SIP-Arbeitsgruppe zur Spezifikation einer entsprechenden Erweiterung weitergeleitet werden. Speziell für die Weiterentwicklung von SIP wurde ein Prozess definiert [RFC 3427], der die üblichen Vorgehensweisen der IETF ergänzt und stärker formalisiert.

Im Jahr 2004 wurde von ETSI das *Telecoms and Internet converged Services and Protocols for Advanced Networks*-Komitee (TISPAN) [26] samt zugehörigen Arbeitsgruppen gegründet. Ziel dieser Organisation ist es, unter Verwendung der 3GPP IMS-Standards eine „Next Generation Network“ (NGN)-Architektur zu definieren, die sowohl drahtlose, als auch drahtgebundene Breitband-Netzzugänge (z. B. ADSL) unterstützt. Da die TISPAN-Architektur das IMS als ein zentrales Subsystem verwendet, spielt auch hier SIP eine zentrale Rolle.

2.1.5.2 Einordnung von SIP in den IP-Protokollstapel

SIP ist ein transaktionsbasiertes Protokoll in der Anwendungsschicht des IP-Protokollstapels, das zur Sitzungssteuerung (bzw. Verbindungssteuerung) verwendet wird. Dazu gehört insbesondere, dass der rufende Teilnehmer seinen Wunsch zum Aufbau einer neuen Multimedia-Sitzung dem gerufenen Teilnehmer mitteilen und dieser diese „Einladung“ (engl. *Invitation*) annehmen oder ablehnen kann. Die dazu benötigten Signalisiernachrichten können entweder direkt zwischen den beiden SIP-Protokollinstanzen der Teilnehmer ausgetauscht oder über so genannte *SIP-Proxies* (s. u.) im Netz weitergeleitet werden, z. B. zur Mobilitätsunterstützung. Ebenfalls zur Verbindungssteuerung gehört die Signalisierung am Ende der Sitzung, sowie so genannte Mehrwertdienste, z. B. das Weiterleiten neu ankommender Rufe oder auch bestehender Verbindungen.

Nicht Aufgabe von SIP im engeren Sinne ist die Aushandlung der Parametrisierung und Codierung der Medienströme. Diese hängt von den durch die Endgeräte unterstützten Formaten (engl. *Terminal Capabilities*) und ggf. den Präferenzen der Nutzer ab. Für diese Aushandlung muss

ein anderes Protokoll verwendet werden, welches von SIP als Anhang (engl. *Attachment*) der SIP-Nachrichten transportiert werden kann. In aller Regel wird für diesen Zweck das *Session Description Protocol* (SDP) [RFC 4566] verwendet. Bei diesem Protokoll, das noch aus den Anfangszeiten der SIP-Entwicklung stammt und sich mehr an den Anforderungen der IP-Multicast-basierten Gruppenkommunikation orientiert, wurden gewisse Defizite bei der Aushandlung von Terminal-Eigenschaften und Medienstrom-Codierungen identifiziert [27]. Obwohl seit geraumer Zeit an einer Weiterentwicklung gearbeitet wird [28], ist SDP immer noch so vorherrschend, dass, wenn von „SIP“ geredet wird, fast immer eigentlich SIP und SDP gemeint sind. Die Aushandlung der zu verwendenden Medienströme über SIP und SDP wird für das in dieser Arbeit betrachtete Problemfeld eine entscheidende Rolle spielen; die identifizierten Probleme und die vorgeschlagenen Lösungen sind jedoch von so grundsätzlicher Natur, dass sie auch gültig bleiben werden, wenn SDP durch ein Nachfolgeprotokoll wie z. B. [28] ersetzt wird.

Da es sich bei SIP um ein Signalisierprotokoll zur Außenband-Signalisierung handelt, wird ein eigenes Protokoll für den Transport der eigentlichen Multimedia-Datenströme benötigt. Hierfür wird in der Regel das *Real-time Transport Protocol* (RTP) [RFC 3550, RFC 3551] verwendet. RTP wird durch das *RTP Control Protocol* (RTCP) ergänzt, welches zur Überwachung der Übertragung (z. B. Latenzen beim Pakettransport) verwendet werden kann. Da RTCP sowohl technisch (z. B. Aspekte von Adressierung und Nachrichtentransport) als auch organisatorisch (z. B. Standardisierung) sehr eng an RTP gekoppelt ist, soll im Folgenden – soweit nicht explizit anders beschrieben – „RTP“ als Sammelbezeichnung für die Kombination aus RTP und RTCP verwendet werden.

Eine Übersicht über SIP und SDP, RTP und RTCP, dem Codec zum Kodieren der analogen Mediendaten (der i. d. R. kein Protokoll bzw. Protokoll-Instanz im eigentlichen Sinne darstellt), sowie den darunterliegenden Schichten des IP-Protokollstapels wird in Abbildung 2.5 gegeben.

SIP-Nachrichten können laut SIP-Kernspezifikation sowohl verbindungslos über UDP, als auch verbindungsorientiert über TCP übertragen werden. Es sind aber auch alternative Transportprotokolle möglich, z. B. wird in [RFC 4168] der optionale, SCTP-basierte Transport spezifiziert. Da die Nachrichtenübertragung über UDP keinen Schutz vor Nachrichtenverlust bzw. -duplizierung und Reihenfolgefehler bietet, besitzt SIP selbst entsprechende Protokollmechanismen, z. B. Sequenznummern, Bestätigungs-Nachrichten und wiederholte Übertragung nach Zeitüberschreitung. Viele der in Kapitel 6 dargestellten Ergebnisse über die Wahl des Transportprotokolls für die Firewall-Steuerung können auch auf SIP übertragen werden.

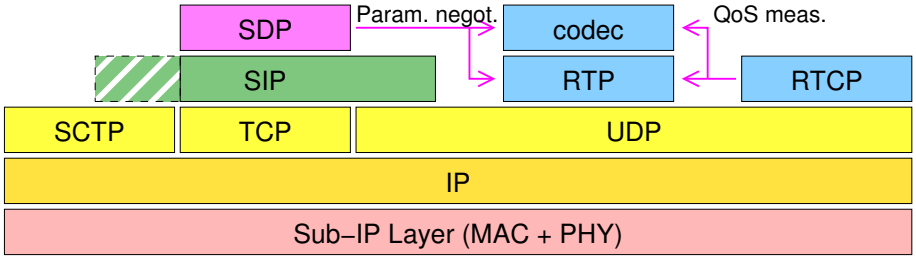


Abbildung 2.5: Protokollstapel mit SIP/SDP und RTP

2.1.5.3 Protokollinstanzen

Die SIP-Spezifikation [RFC 3261] definiert verschiedene Typen von Protokollinstanzen. Der so genannte *SIP User Agent* (UA) ist die SIP-Protokollinstanz im Endsystem beim Teilnehmer. Zusammen mit den Protokollinstanzen für Medienkodierung und -transport (Codec und z. B. RTP) wird sie i. d. R. als Anwendungs-Software für Computer („Softphone“) oder auf einem Mikroprozessor in spezieller Telefon-Hardware („Hardphone“) implementiert. Der UA zerfällt logisch in zwei Komponenten, den *User Agent Client* (UAC) und den *User Agent Server* (UAS). Der UAC initiiert Transaktionen, indem er so genannte *SIP Requests* an die Gegenstelle versendet und auf die entsprechenden Antworten wartet. Der UAS ist für die Bearbeitung eingehender Requests verantwortlich. Jede SIP-UA-Protokollinstanz muss sowohl einen UAC, als auch einen UAS implementieren, da die Rollen im Verlauf einer Sitzung wechseln können. So ist z. B. bei einem Verbindungsaufbau der UA des rufenden Teilnehmers in der Rolle des UAC, der einen *INVITE-Request* an den UAS des gerufenen Teilnehmers sendet. Legt am Gesprächsende der gerufene Teilnehmer zuerst auf, so ist seine SIP-Protokollinstanz jetzt in der Lage des UAC, der einen *BYE-Request* an den UAS des rufenden Teilnehmers sendet.

SIP-Proxies sind Protokollinstanzen auf dem Weg zwischen SIP User Agents, die SIP-Nachrichten ggf. bearbeiten und weiterleiten. Die wichtigste Anwendung solcher Proxies ist die Verkehrslenkung von SIP-Nachrichten oberhalb der Verkehrslenkung der IP-Schicht, z. B. zur Mobilitätsunterstützung oder zum Weiterleiten von Rufen. Sie können aber auch andere Funktionen haben, z. B. die Authentisierung des rufenden Teilnehmers oder die Autorisierung einer Verbindungsaufbau-Anforderung. SIP-Proxies können entweder zustandslos (engl. *stateless*) oder zustandsbehaftet (engl. *stateful*) implementiert werden. Im ersten Fall werden Anforderungen bzw. die entsprechenden Antworten individuell weitergeleitet; im zweiten Fall wird Zustandsinformation über begonnene, aber noch nicht abgeschlossene SIP-Transaktionen gehalten. Da die SIP-Transaktion zum Aufbau einer Multimedia-Sitzung beendet ist, sobald die Sitzung begonnen hat, hält ein SIP-Proxy keine Zustandsinformationen über laufende Sitzungen.

So genannte *SIP Back-to-Back User Agents* (B2BUA) verhalten sich wie zwei „Rücken an Rücken gekoppelte User Agents“. Sie haben ähnliche Funktionen wie Proxies, halten aber Zustandsinformationen bzgl. der laufenden Sitzungen. Dies bezieht sich aber nur auf Aspekte der Signalisierung; die RTP-Medienströme laufen in der in [RFC 3261] beschriebenen Architektur immer direkt zwischen den Endsystemen, d. h. an Proxies und B2BUAs vorbei. Anders als Proxies, welche von User Agents gesendete Requests nur verändern oder umleiten können, können B2BUAs auch selbst Requests erzeugen. So kann z. B. eine laufende Multimedia-Sitzung vom B2BUA beendet werden, indem ein *BYE*-Request an alle beteiligten User Agents gesendet wird.

Ein *SIP Registrar* ist eine Protokollinstanz, die *REGISTER-Requests* bearbeitet. Diese werden für Adressabbildungen, insbesondere für die Mobilitätsunterstützung benötigt (siehe Abschnitt 2.1.5.9). Ebenfalls in diesem Kontext werden so genannte *Redirect Server* verwendet, die eingehende *INVITE*-Anforderungen zwar zurückweisen, dabei aber SIP-Adressen mitgeben, unter der der gewünschte Teilnehmer derzeit erreichbar ist.

Im Zuge wissenschaftlicher Veröffentlichungen, Aktivitäten anderer Standardisierungsorganisationen, oder Werbekampagnen der Hersteller wurden noch weitere Typen von Protokollinstanzen definiert (z. B. „Session Border Controller“, siehe Abschnitt 4.1.1), diese sind jedoch nicht Bestandteil der SIP-Kernspezifikation nach [RFC 3261].

2.1.5.4 Transaktionen und Nachrichtentypen

SIP ist ein transaktionsorientiertes Protokoll. Eine *SIP-Transaktion* besteht aus einer Anforderung (*Request*), die vom UAC zum UAS gesendet wird, sowie aus einer beliebigen Anzahl von vorläufigen Antworten (*Provisional Response*) und genau einer endgültigen Antwort (*Final Response*), die in Gegenrichtung gesendet werden.

SIP Requests transportieren in ihrer ersten Zeile den Namen einer Methode (*Method*), die auf dem Server ausgelöst werden soll. Das SIP-Basisprotokoll nach [RFC 3261] kennt nur sechs Methoden: *INVITE*, um den Aufbau einer neuen Sitzung anzufordern und *ACK* als Bestätigungs-Nachricht im Zuge des Aufbau-Vorgangs. *BYE* dient zum regulären Abbau einer Sitzung, wohingegen *CANCEL* dazu verwendet wird, eine begonnene, aber noch nicht erfolgreich abgeschlossene Transaktion (z. B. *INVITE*) abzubrechen. Die *REGISTER*-Methode dient zur Signalisierung von Adress-Abbildungen (siehe Abschnitt 2.1.5.9). Mit der *OPTION*-Methode kann schließlich geprüft werden, ob die Gegenstelle optionale Protokollerweiterungen unterstützt.

Eine solche optionale Methode ist beispielsweise die *INFO*-Methode [RFC 2976], mit der während einer laufenden Multimedia-Sitzung sitzungsbezogene Steuerinformationen übertragen werden können, z. B. vom Teilnehmer gesendete DTMF-Ziffern zu Fernsteuerung eines Anrufbeantworters, oder Informationen zur Anzeige der bereits angefallenen Gesprächs-Entgelte. Mit der *UPDATE*-Methode [RFC 3311] kann ein Endpunkt die Parameter der Multimedia-Sitzung (z. B. verwendete Codecs, Status ggf. dafür durchgeführter QoS-Ressourcen-Reservierungen, etc.) ändern, ohne eine zweite *INVITE*-Nachricht, ein so genanntes *Re-INVITE* zu senden, welches weitere Auswirkungen auf den Zustand der Sitzung haben könnte. Mit der ebenfalls optionalen *SUBSCRIBE*-Methode kann eine SIP-Instanz einer anderen SIP-Instanz signalisieren, dass sie asynchron über bestimmte Ereignisse informiert werden möchte, die je nach Anwendungszweck in so genannten „event packages“ zusammengefasst sind [RFC 3265]. Dies kann z. B. für „Presence“-Anwendungen [RFC 3856] verwendet werden, mit denen Teilnehmer explizit bekanntgeben können, dass sie derzeit in der Lage und willens sind, an Multimedia-Sitzungen teilzunehmen.

SIP responses beinhalten als wichtigstes Feld einen numerischen, dreistelligen Statuscode, der durch eine Klartext-Meldung ergänzt wird. Ähnlich wie bei SMTP oder HTTP ist die Kodierung dabei so gewählt, dass die Antwort schon anhand der ersten Ziffer einer Klasse zugeordnet werden kann. Somit kann ein UAC auch auf Antwort-Codes sinnvoll reagieren, die in einer Protokoll-Erweiterung spezifiziert sind, und deren genaue Bedeutung dem Client nicht bekannt sind. Die Response Codes 100 . . . 199 gelten als vorläufige Antworten, die den Empfang eines Requests und ggf. einen gewissen Fortschritt bei der Bearbeitung bestätigen; das endgültige Ergebnis steht jedoch noch nicht fest und wird später mit einer weiteren Antwort mitgeteilt. Die 2xx-Klasse beschreibt den erfolgreichen Abschluss einer Transaktion; derzeit ist in dieser Klasse nur *200 OK* spezifiziert. Alle folgenden Antwort-Klassen signalisieren Fehler: Antworten der 3xx-Klasse weisen einen Request (insbesondere *INVITE*) zurück, geben aber alternative Adressen an, unter der der Teilnehmer erreicht werden kann. 4xx-Antworten bedeuten, dass ein bestimmter Server den jeweiligen Request so nicht bearbeiten kann. Der UAC hat jedoch die Möglichkeit, den gleichen Request an einen anderen Server zu senden, oder einen modifizierten Request (z. B. nach Hinzufügen von Authentisierungs-Informationen) erneut zu senden. Die

5xx-Klasse beschreibt interne Fehler in einem SIP-Server, die 6xx-Klasse solche, die so global sind, dass der Request von keiner SIP-Instanz im Netz erfolgreich bearbeitet werden kann.

Grundsätzlich werden für einen Sitzungsaufbau mit SIP mindestens drei Nachrichten benötigt: die *INVITE*-Nachricht, die dazugehörige *200*-Antwort, die nach der Annahme des Rufes durch den gerufenen Teilnehmer gesendet wird, und die *ACK*-Nachricht, die den Empfang der *200*-Nachricht quittiert (siehe Abbildung 2.4). Die *INVITE*- und die *200*-Nachricht enthalten i. d. R. je einen SDP-Block, mit dem die jeweilige Instanz bekannt gibt, auf welcher IP-Adresse und Transportschicht-Portnummer auf ankommende Medienströme gewartet wird und welche Codecs beim Empfang unterstützt werden. Der Endpunkt des rufenden Teilnehmers muss darauf vorbereitet sein, unmittelbar nach dem Versenden der *INVITE*-Nachricht, noch vor Ende der Verbindungsaufbau-Signalisierung, schon Medienströme zu empfangen. Diese so genannten *Early Media* können für Ansagen genutzt werden, z. B. des verbleibenden Restguthabens im Voraus bezahlter Telefonkarten (engl. *Prepaid Calling Card*). Ein weiteres wichtiges Anwendungsgebiet ist die Weiterleitung von Hörtönen und Ansagen, falls sich der gerufene Teilnehmer hinter einem Gateway in einem analogen Telefonnetz mit Innenbandsignalisierung befindet.

Die in Abbildung 2.4 zusätzlich dargestellte, provisorische *180*-Antwort ist optional; ihr Empfang wird normalerweise auch nicht quittiert. Da eine solche Quittierung provisorischer Antworten in einigen Szenarien, insbesondere bei der Zusammenschaltung mit ISDN-Netzen, notwendig ist, wurde hierzu die optionale *PRACK*-Methode definiert [RFC 3262], deren Verwendung analog zur *ACK*-Methode für endgültige Antworten funktioniert. Die *PRACK*-Methode kann auch zum Transport weiterer Informationen in dieser Phase des Sitzungsaufbaus verwendet werden.

Neben dem Namen der Methode bei einer Request-Nachricht bzw. dem Status-Code bei einer Response-Nachricht muss noch eine Vielzahl weiterer Parameter übertragen werden können, um den Aufbau einer neuen Multimedia-Sitzung signalisieren zu können. Das Nachrichtenformat wird im folgenden Abschnitt beschrieben.

2.1.5.5 Nachrichten-Format und -Codierung

Wie bei vielen anderen Anwendungsschichtprotokollen des IP-Protokollstapels (z. B. SMTP, HTTP) auch, werden SIP-Nachrichten textbasiert übertragen. Dafür wird der UTF-8 Zeichensatz [RFC 2279] verwendet. Ein Beispiel für eine SIP-Nachricht ist in Abbildung 2.6 dargestellt.

Eine SIP-Nachricht ist grundsätzlich in drei Teile gegliedert: Die erste Zeile der Nachricht („start-line“) ist je nach Nachrichten-Typ entweder eine „Request-Line“ oder eine „Status-Line“. Erstere transportiert den Methoden-Namen, eine *Request-URI* (s. u.) und die SIP-Protokoll- und Versionskennung, zweitere die SIP-Protokoll- und Versionskennung, den Status-Code und eine textuelle Erklärung des Status.

Es folgt der Nachrichten-Kopf (*Message Header*), der aus einer beliebigen Anzahl von *Header Fields* besteht, die jeweils aus einer Zeile der Form „`header-name : header-value`“ (Schlüsselwort : Wert) bestehen. Je nach Nachrichten-Typ müssen bestimmte Felder in der Nachricht vorkommen, andere sind optional; manche Felder dürfen auch mehrfach auftreten.

Durch eine Leerzeile getrennt folgt der optionale Nachrichten-Körper (*Message Body*). Dieser kann, auf die selbe Weise wie E-Mail-Anhänge realisiert werden (siehe [RFC 2045] ff.), prinzipiell beliebige Daten transportierten, die für das SIP-Protokoll selbst ohne direkte Bedeutung sind. Eine wichtige Anwendung dieses Mechanismus ist der Transport von SDP-Nachrichten, mit der die Parameter der zur Sitzung gehörenden Multimedia-Datenströme angekündigt bzw. ausgehandelt werden (siehe Abschnitt 2.1.5.2).

Laut dem ersten RFC zu SIP ermöglicht diese textbasierte Codierung der SIP-Nachrichten – verglichen mit anderen üblichen Formaten, wie z. B. TLV (Type-Length-Value), ASN.1 [X.680] oder XML [29]) – eine einfache Implementierbarkeit der Protokollinstanzen, eine einfache Fehlersuche und viele Erweiterungsmöglichkeiten für das Basis-Protokoll [RFC 2543, Sec. 1.5.3]. Durch die vielen Freiheitsgrade bei der Gestalt einer Nachricht (z. B. Groß-/Kleinschreibung, Anzahl der Leerzeichen, optionale Zeilenumbrüche, etc.) ergibt sich allerdings zur Laufzeit ein gewisser Aufwand (und damit Verzögerung) für das Parsen einer empfangenen Nachricht in den SIP-Protokollinstanzen [30]. Im aktualisierten Nachfolge-RFC [RFC 3261] wird nicht mehr auf die Vor- und Nachteile der gewählten Codierung eingegangen.

2.1.5.6 Kryptographischer Schutz der Nachrichten

Die SIP-Spezifikation nach [RFC 3261] sieht drei Mechanismen zum Schutz der Nachrichten vor Angriffen (vgl. Abschnitt 2.2) vor. Die von HTTP übernommene *Digest Authentication* [RFC 2617] bietet eine einfache, einseitige Authentisierung auf Basis eines geteilten Geheimnises (z. B. Passwort in Verbindung mit der Nutzer-Kennung). Eine größere Schutzhöhe

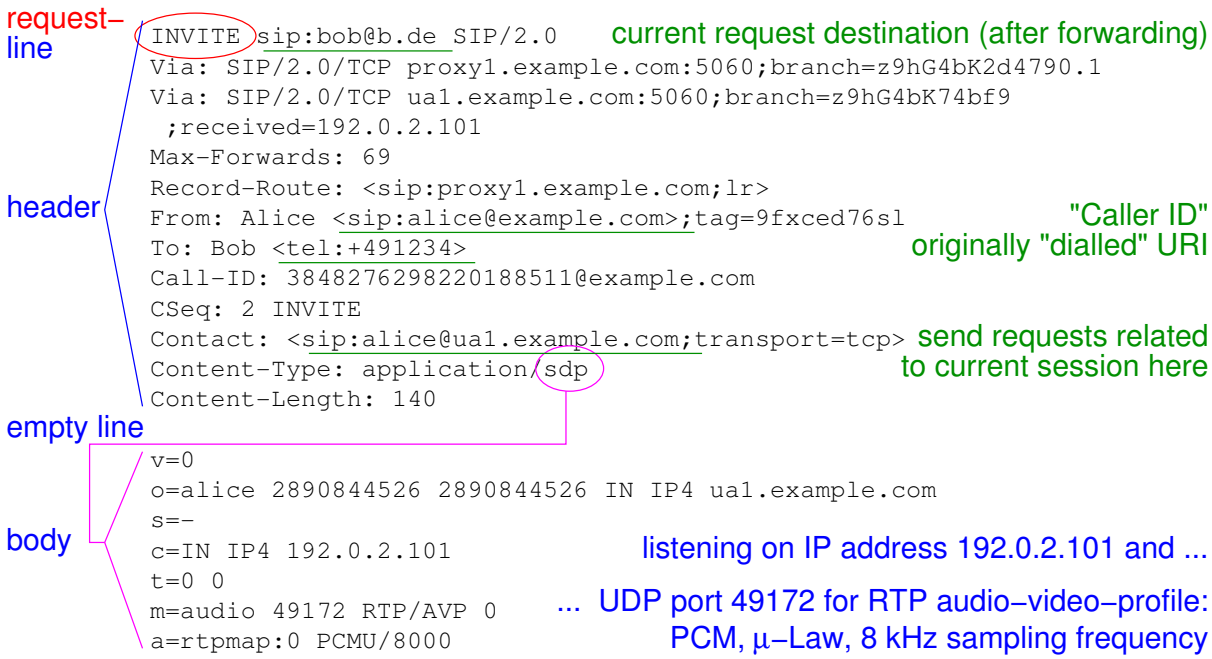


Abbildung 2.6: SIP INVITE-Nachricht zwischen Proxy 1 und Proxy 2 in Abbildung 2.7

wird mit den beiden anderen Verfahren erreicht, die beidseitige Authentisierung sowie Schutz der Integrität und Vertraulichkeit bieten können, aber das Vorhandensein von Mechanismen zum Verteilen von kryptographischen Schlüsseln voraussetzen. Falls SIP-Nachrichten über eine Kette von SIP-Instanzen (z. B. Proxies) transportiert werden, bietet *Transport Layer Security* (TLS) [RFC 4346] jeweils einen abschnittsweisen Schutz beim Transport zwischen benachbarten Instanzen. Die *Secure/Multipurpose Internet Mail Extensions* (S/MIME) [RFC 3850, RFC 3851] hingegen können zum Ende-zu-Ende-Schutz der Nachrichten verwendet werden. Da bestimmte Felder einer Nachricht von Proxies gelesen bzw. modifiziert werden müssen, können allerdings nicht alle Nachrichtenfelder signiert bzw. verschlüsselt werden. Welches dieser Verfahren in einer gegebenen Netz-Topologie sinnvoll anwendbar ist, hängt entschieden davon ab, wo Angreifer angenommen werden und welche SIP-Instanz welche anderen SIP-Instanzen für vertrauenswürdig einstuft.

Alle oben genannten Verfahren schützen ausschließlich die Signalisiernachrichten, nicht jedoch die eigentlichen Nutzdaten, d. h. den RTP-Medienstrom. Für die kryptographisch gesicherte Übertragung von Multimedia-Strömen kann das *Secure Real-time Transport Protocol* (SRTP) [RFC 3711] verwendet werden; die dafür benötigten Sitzungsschlüssel können über einen mit S/MIME oder TLS gesicherten Signalisierkanal zwischen den Multimedia-Endgeräten ausgetauscht werden.

2.1.5.7 Adressierung

Sowohl Teilnehmer, als auch Endgeräte (i. d. R. User Agents) werden bei SIP mit Hilfe eines *Uniform Resource Identifier* (URI) [RFC 3986] identifiziert. Das generische Format einer solchen Adresse ist `sip:user:password@host:port;uri-parameters?headers`; sehr oft wird davon nur `sip:user@host` verwendet. Dabei bezeichnet „host“ den Namen bzw. die IP-Adresse eines Rechners, oder einen Domain-Namen, der mit Hilfe des *Domain Name System* (DNS) [RFC 1034, RFC 1035] abgebildet wird [RFC 3263]. Das Feld „user“ beinhaltet einen Bezeichner, der im Kontext von „host“ beliebig gewählt werden kann, z. B. ein Account-Namen auf einem Mehrbenutzersystem, oder auch eine ISDN-Rufnummer [E.164]. Dieses Feld wird entsprechend einer lokalen Richtlinie des „host“ interpretiert. Beginnt die URI mit `sips:` statt `sip:`, so soll die entsprechend Ressource „sicher“ kontaktiert werden, indem die Signalisiernachrichten über *Transport Layer Security* (TLS) [RFC 4346] transportiert werden [31]; dies macht jedoch keine Aussage über einen evtl. gewünschten Schutz der Medienströme. In [RFC 2806] werden die optionalen URI-Schemata `tel:`, `fax:` und `modem:` spezifiziert, denen Telefonnummern gemäß ITU-T Empfehlung E.164 [E.164] folgen. Anders als bei der Verwendung als „user“-Bezeichner einer `sip:`-URI, ist bei der `tel:`-URI definiert, dass die folgende Nummer immer als E.164-Nummer zu interpretieren ist. Die SIP-Kernspezifikation sieht vor, dass alle SIP-Instanzen die `sip:`- und `sips:`-Schemata unterstützen müssen. Andere Schemata sind hingegen optional; Proxies dürfen optionale Schemata während der Weiterleitung auf `sip:` bzw. `sips:` abbilden.

SIP unterstützt verschiedene Formen der Mobilität sowie eine an Bedingungen knüpfbare Anrufweiterschaltung. Dazu sind in vielen SIP-Nachrichten, insbesondere der *INVITE*-Nachricht, mehrere Adressfelder vorhanden. In der `To:`-Kopfzeile wird die vom rufenden Teilnehmer „gewählte“ URI transportiert. Falls er nicht anonym bleiben will, enthält die `From:`-Kopfzeile

seine Adresse, die ggf. vom UA des gerufenen Teilnehmers während des Klingelns angezeigt wird. Beide Adressen sind also für die Teilnehmer direkt sichtbar. Diese so genannten *Address of Record* (AOR) sind i. d. R. langfristig gültig und können z. B. in Telefonbüchern veröffentlicht werden. Im Gegensatz dazu ist die so genannte *Request-URI*, die in der ersten Zeile einer Request-Nachricht auf den Methoden-Namen folgt, i. d. R. für die Teilnehmer nicht sichtbar. Dieses Adress-Feld kann auf dem Weg einer Nachricht durch das Netz von SIP-Proxies geändert werden und gibt das jeweils aktuelle Ziel nach einer ggf. erfolgten Weiterleitung an. Für diesen Zweck können u. U. auch URIs verwendet werden, die zeitlich nur sehr begrenzt gültig sind, z. B. von User Agents, deren IP-Adresse vom Zugangsnetz dynamisch zugewiesen wurde. Entsprechend wird in der `Contact`:-Kopfzeile eine ggf. nur kurzfristig gültige Absender-Adresse angegeben, an die von der Gegenstelle initiierte Transaktionen zu adressieren sind, z. B. wenn eine Sitzung durch den gerufenen Teilnehmer beendet wird. Die Weiterleitung von Rufen mit Hilfe von Proxies, die dazu die *Request-URI* einer Nachricht ändern, ist in Abschnitt 2.1.5.9 beschrieben.

2.1.5.8 Adressabbildungen und Weiterleitung von SIP-Nachrichten

Die Abbildung einer SIP-URI auf letztendlich die IP-Adresse eines Servers, der Requests bzgl. dieser URI verarbeiten oder weiterleiten kann, ist in [RFC 3263] beschrieben. Diese Abbildung erfolgt in drei Stufen: zunächst wird durch Abfragen von so genannten *NAPTR-Records* im DNS ermittelt, welche Transport-Modi die Ziel-Domäne für eingehende SIP-Transaktionen unterstützt (z. B. SIP über TLS über TCP). Der Absender kann einen solchen Transportmodus auswählen und über *SRV-Records* eine Liste von Namen der Server erhalten, die diesen Modus unterstützen. Einer dieser Namen muss durch Abfrage von *A-Records* auf eine IP-Adresse abgebildet werden, um eine Transportschicht-Verbindung zu diesem Server aufbauen zu können bzw. verbindungslos zu transportierende Nachrichten an ihn adressieren zu können.

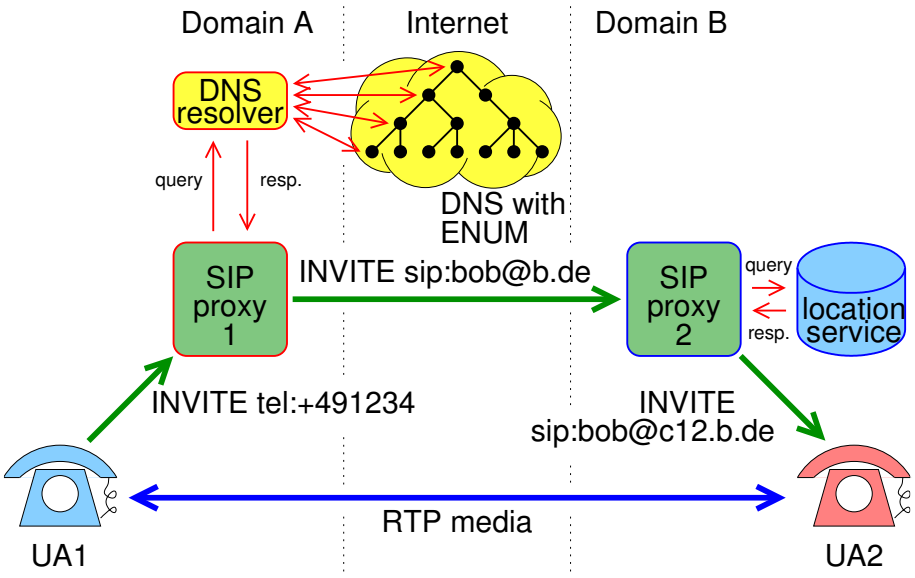


Abbildung 2.7: SIP Trapezoid mit DNS/ENUM

ENUM (tElephone NUmber Mapping) [RFC 3761, ff.] ist ein von der IETF entworfenes Verfahren, mit dessen Hilfe E.164-Nummern mit Hilfe des DNS auf andere Nummern- bzw. Namensräume abgebildet werden können, u. a. auf `sip:`-URIs. Zusammen mit den `tel:`-URIs ermöglicht ENUM, die vom ISDN bekannten Rufnummern auch mit SIP weiter zu benutzen. Bei der Zusammenschaltung SIP-basierter Netze mit ISDN/PSTN-Netzen ist dies sogar zwingend erforderlich, da an diesen Endgeräten i. d. R. nur Ziffern gewählt werden können. In Szenarien, in denen Ende-zu-Ende-Konnektivität auf der IP-Schicht angenommen werden kann (z. B. Internet-Telefonie), wird die Zuordnung von E.164-Nummern zu `sip:`-URIs im DNS öffentlich zugänglich hinterlegt. Dies wird als *User ENUM* bezeichnet. Eine eingegebene E.164-Nummer kann somit in einem ersten Schritt sofort auf eine `sip:`-URI abgebildet werden; anhand dieser kann der Sitzungsaufbau dann wie oben beschreiben fortgesetzt werden. Liegt die Kontrolle über das Ändern und Abfragen der ENUM-Datensätze nicht letztendlich beim Teilnehmer, sondern bei seinem Netzbetreiber, wird dies als *Carrier ENUM* oder *Infrastructure ENUM* [32] bezeichnet.

In Abbildung 2.7 ist das schon in der SIP-Kernspezifikation [RFC 3261] eingeführte „SIP Trapezoid“ dargestellt, ergänzt um DNS- und ENUM-Mechanismen, die in [RFC 3263] bzw. [RFC 3761, ff.] spezifiziert werden. Hierbei handelt es sich um ein typisches SIP-Anwendungsszenario in einem Netz, welches Ende-zu-Ende-Konnektivität auf der IP-Schicht bietet (z. B. Internet). Der Name „Trapezoid“ rührt von der Tatsache her, dass in dieser Konfiguration die RTP-Medienströme direkt zwischen den Endsystemen laufen, die SIP-Signalisierung jedoch über 2 SIP-Proxyserver geführt wird, die bei der Wegesuche zum gerufenen Teilnehmer behilflich sind.

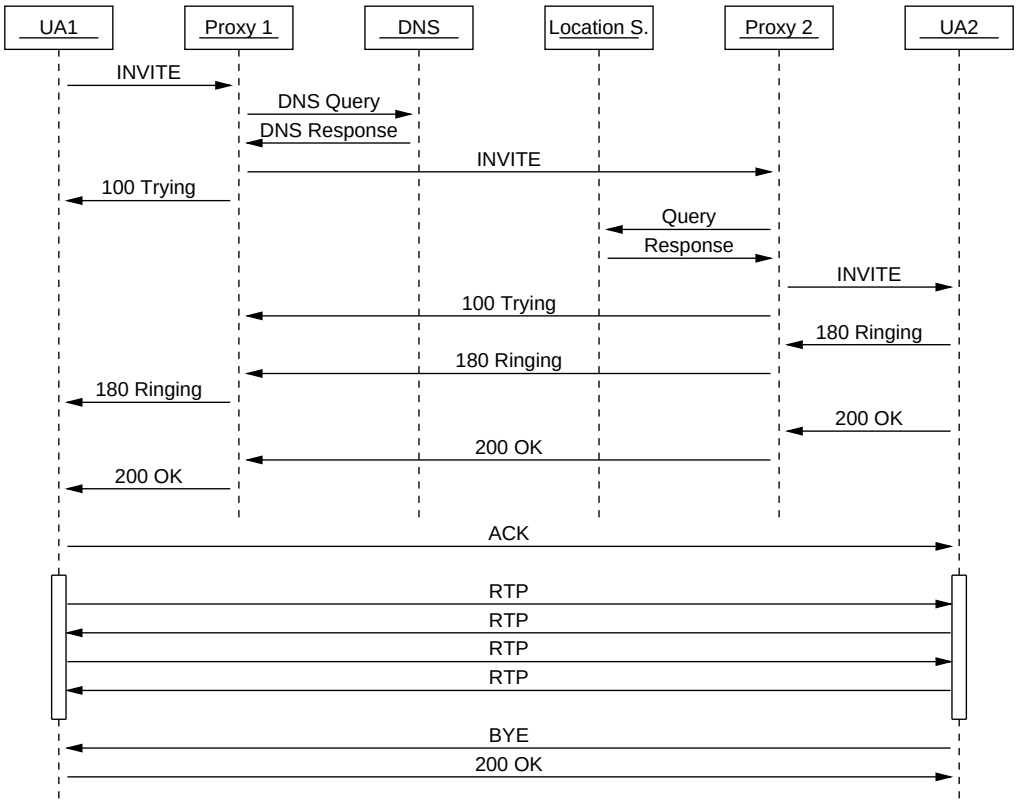
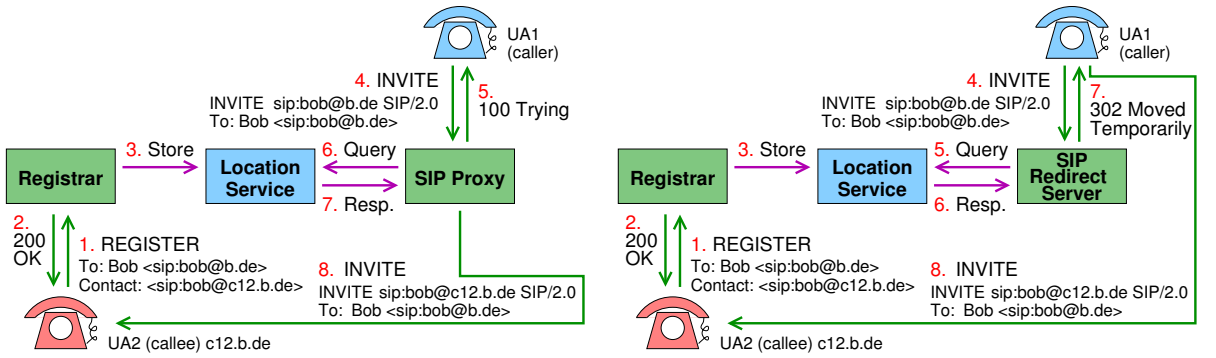


Abbildung 2.8: Nachrichten-Sequenzdiagramm zu Abbildung 2.7

Abbildung 2.8 zeigt ein Nachrichtensequenzdiagramm für einen Sitzungsaufbau in der in Abbildung 2.7 dargestellten Netztopologie. Es wird dabei davon ausgegangen, dass der rufende Teilnehmer seinen SIP User Agent „UA1“ dazu benutzen möchte, eine Verbindung zum gerufenen Teilnehmer herzustellen, von dem die (fiktive) E.164-Nummer +491234 bekannt ist. Der gerufene Teilnehmer benutzt einen SIP User Agent „UA2“, dem temporär eine IP-Adresse zugewiesen wurde (z. B. mit DHCP [RFC 2131]), was dem rufenden Teilnehmer aber nicht bekannt sein muss. Die umfangreichen Adressabbildungen können insbesondere mobile User Agents deutlich belasten bzw. überfordern. Deshalb sendet „UA1“ eine *INVITE*-Nachricht mit der Request-URI `tel:+491234` an den „SIP proxy 1“. Dieser Proxy, der z. B. vom *Internet Service Provider* (ISP) des rufenden Teilnehmers betrieben werden könnte, ist in „UA1“ als so genannter *Outbound Proxy* konfiguriert. Mit Hilfe von ENUM bildet „SIP proxy 1“ die E.164-Nummer auf die beispielhafte URI `sip:bob@b.de` ab und ermittelt mit Hilfe des DNS (s. o.) den so genannten *Inbound Proxy* der Domäne B, der der gerufene Teilnehmer angehört. Dieser „SIP proxy 2“ bildet mit Hilfe eines Verzeichnisdienstes die längerfristig gültige *Address of Record* `sip:bob@b.de` auf die nur temporär zugewiesene Adresse des „UA2“ `sip:bob@c12.b.de` und die dazugehörige IP-Adresse ab und leitet die *INVITE*-Nachricht mit abermals geänderter Request-URI an den „UA2“ weiter.

2.1.5.9 Mobilitätsunterstützung

SIP unterstützt *Nutzermobilität* (engl. *Personal Mobility*) [33], d. h. ein Teilnehmer kann immer unter der selben für die Kommunikationspartner sichtbaren Adresse erreichbar sein, auch wenn er eine andere Teilnehmerendeinrichtung verwendet oder seinen Standort im Netz wechselt. Die dazu verwendeten Adressabbildungen auf temporäre, standort- bzw. gerätebezogene SIP-Adressen kann, wie schon in Abschnitt 2.1.5.8 beschrieben, mit Proxies erfolgen, die die entsprechend umgeschriebene SIP-Nachricht in Richtung des Ziels weiterleiten (siehe auch Abbildung 2.9 (a)). Anstelle des Inbound Proxy kann aber auch ein Redirect Server verwendet werden, der eingehende SIP-Requests mit einer Antwort der 3xx-Klasse zurückweist, die SIP-URIs enthält, unter denen der Teilnehmer derzeit oder dauerhaft erreichbar ist (siehe Abbildung 2.9 (b)). Das zweite Verfahren erzeugt weniger Last in zentralen Netzelementen, da dort weniger Nachrichten verarbeitet und keine transaktions- oder sitzungsbezogenen Zustandsinfor-



(a) Weiterleitung der Signalisierung mit SIP-Proxy

(b) Zurückweisen und Verweis auf derzeitige Adresse mit SIP-Redirect-Server

Abbildung 2.9: Weiterleitung von SIP-INVITE-Nachrichten an mobile Nutzer

mationen gehalten werden müssen. Andererseits werden dem rufenden Teilnehmer (bzw. seinem Outbound Proxy) Adress-Informationen signalisiert, über die evtl. Rückschlüsse auf den Standort des gerufenen Teilnehmers gezogen werden können, was aus Datenschutz-Gründen u. U. bedenklich ist.

Die zum Weiter- bzw. Umleiten von SIP-Nachrichten benötigten Zuordnungen (engl. *Address bindings*) zwischen lang- und kurzfristig gültigen Adressen können mit Hilfe der SIP *REGISTER*-Methode an SIP Registrar Server signalisiert werden. Auch eine Registrierung durch Dritte ist möglich. Nicht Teil der SIP-Spezifikation ist hingegen die Ausgestaltung des Systems, in dem die Zuordnungen gespeichert werden, und die Schnittstellen zum Eintragen bzw. Abfragen. Deshalb sind in Abbildung 2.9 dafür nur sehr generische Bezeichnungen verwendet.

2.1.5.10 Signalisieren von Vorbedingungen

Für den Aufbau einer Multimedia-Sitzung werden prinzipiell nur drei SIP-Nachrichten benötigt (*INVITE*, *200* und *ACK*; siehe Abbildung 2.4). Dabei wird der gerufene Teilnehmer schon unmittelbar nach dem Empfang der *INVITE*-Nachricht durch sein Endgerät über den eingehenden Ruf informiert, z. B. durch ein Klingel-Signal. Dies ist in manchen Szenarien problematisch, da in dieser Phase des Verbindungsaufbaus die Aushandlung der Parameter für die Medienströme noch nicht abgeschlossen ist. Diese Parameter werden für bestimmte Hilfs-Funktionen benötigt, z. B. für die Reservierung von Ressourcen zur Dienstgüteunterstützung, die somit erst zu einem späteren Zeitpunkt ausgeführt werden können. Schlagen diese Funktionen fehl, z. B. weil die angeforderten Ressourcen nicht zur Verfügung stehen, so kann die Multimedia-Sitzung nicht zu Stande kommen, obwohl der gerufene Teilnehmer bereits alarmiert wurde.

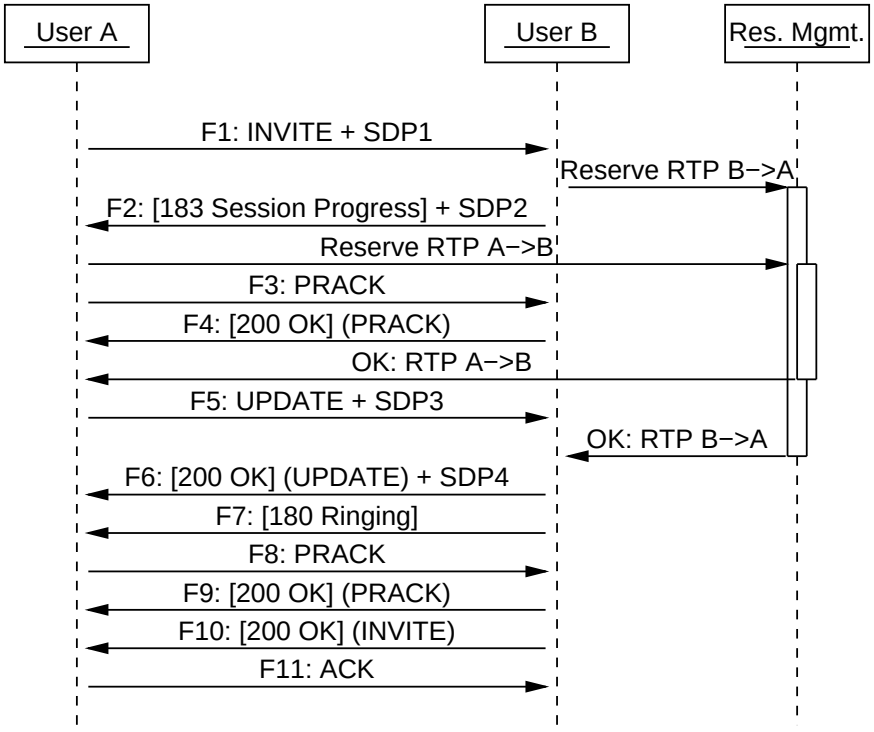


Abbildung 2.10: Sitzungsaufbau mit Vorbedingungen nach [RFC 3312]

Zur Vermeidung dieser für den gerufenen Teilnehmer lästigen, so genannten *Ghost Rings* wird in [RFC 3312] eine kombinierte Erweiterung von SIP und SDP spezifiziert, mit der Vorbedingungen (engl. *Preconditions*) signalisiert werden können (siehe Abbildung 2.10). Dazu werden SIP-Nachrichten mit SDP-Anhängen ausgetauscht, die neben der jeweiligen Zieladresse und der Liste unterstützter Codecs für die Medienströme auch den gewünschten und den bereits erreichten Status der QoS-Reservierung enthalten. Hierbei kann für beide Richtungen getrennt angegeben werden, ob eine QoS-Reservierung Ende-zu-Ende oder nur in den Zugangsnetzen geschehen soll bzw. bereits erfolgt ist. Der Empfang einer *INVITE*-Nachricht mit einem SDP-Anhang, der noch unerfüllte Vorbedingungen enthält, löst keine Alarmierung des gerufenen Teilnehmers aus, stattdessen wird ggf. mit der Reservierung für Medienströme in Rückwärtsrichtung begonnen. Die Reservierung in Vorwärtsrichtung geht vom User Agent des rufenden Teilnehmers aus, sobald diesem die dafür benötigten Parameter bekannt sind. Die Gegenstelle wird mit Hilfe aktualisierter SDP-Nachrichten über erfolgte Reservierungen informiert. Zum Transport dieser SDP-Anhänge kommt u. a. die optionale SIP *UPDATE*-Methode [RFC 3311] zum Einsatz. Sind alle Vorbedingungen erfüllt, wird der gerufene Teilnehmer alarmiert und der Verbindungsaufbau wird wie gewohnt fortgesetzt.

2.1.5.11 Interworking mit anderen Protokollfamilien

Falls SIP-basierte Netze mit Telefonnetzen zusammengeschaltet werden sollen, die andere Signalisierprotokolle verwenden, so muss das Zusammenspiel (engl. *Interworking*) der Protokolle spezifiziert werden, z. B. für H.323 (siehe Abschnitt 2.1.6.1) in [RFC 4123].

Das Zusammenspiel von SIP mit dem im *Integrated Services Digital Network* (ISDN) für die Zwischenamts-Signalisierung verwendeten *ISDN User Part* (ISUP) wird in [RFC 3398] bzw. in ähnlichen Dokumenten der ITU-T spezifiziert. Abbildung 2.11 zeigt ein typisches Szenario für die Zusammenschaltung eines SIP/RTP-basierten Netzes mit einem ISDN-Netz. Die SIP-Signalisierung wird von den SIP-basierten Teilnehmern über Proxies zum so genannten *Media Gateway Controller* (MGC) weitergeleitet. Dort werden die Zustandsinformationen für die Verbindungssteuerung gehalten und es erfolgt die Umsetzung auf das Nachrichtenformat des ISUP. Die ISUP-Nachrichten werden allerdings zunächst mit den Protokollen der *Signaling Transport* (SIGTRAN)-Familie über IP bis zum *Signaling Gateway* (SGW) transportiert, wo die Umsetzung auf die physikalische Schicht des ISDN erfolgt. Der MGC steuert über das

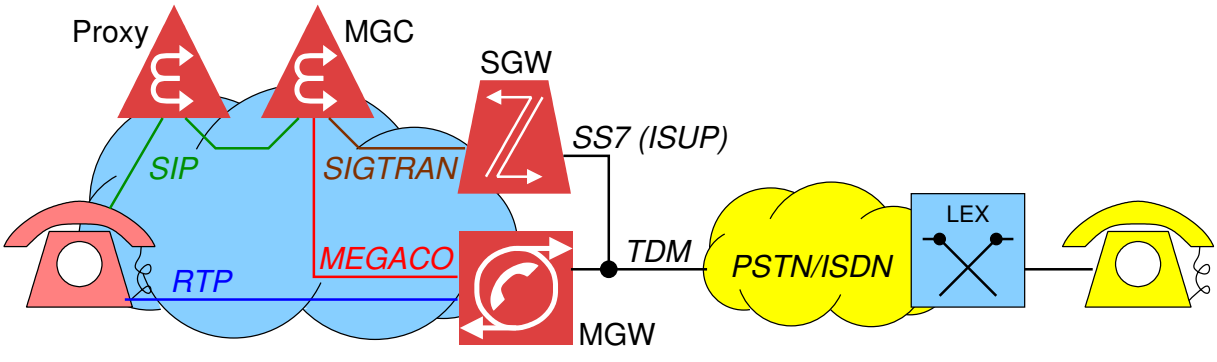


Abbildung 2.11: Zusammenschaltung von SIP/RTP-basierten Netzen mit ISDN-Netzen

MEGACO-Protokoll auch das *Media Gateway* (MGW, auch: MG), welches RTP-Medienströme auf einen Zeitschlitz in einem *Time Division Multiplex* (TDM)-System umsetzt.

2.1.5.12 Private Erweiterungen und SIP-Profile

SIP erfreut sich großer Aufmerksamkeit und Beliebtheit bei vielen verschiedenen Interessengruppen, die das Protokoll für teilweise stark unterschiedliche Anwendungen in verschiedenen Szenarien anwenden möchten. Dementsprechend gibt es viele Wünsche bzgl. der Erweiterung des Protokolls, um den jeweiligen Anforderungen genüge zu tragen. In [RFC 3427] wird ein Prozess zur Weiterentwicklung der Protokoll-Spezifikation durch die IETF SIP-Arbeitsgruppe definiert. Spezielle Interessen können teilweise auch ohne breiten Konsens in der Arbeitsgruppe durch Spezifikation und Registrierung eines *P-Headers* abgedeckt werden, sofern dieser den offiziellen Standards nicht widerspricht. Das „P“ als Präfix für solche Nachrichtenfelder steht für „preliminary“, „private“ oder „proprietary“, da solche Protokollerweiterungen i. d. R. deshalb nicht auf normalem Wege standardisiert werden, weil ihre Funktion und ggf. Seiteneffekte auf andere Mechanismen noch nicht vollständig verstanden wurde, weil sie in nur sehr speziellen Szenarien außerhalb des Internets eingesetzt werden kann, oder weil ein sehr spezieller, evtl. patentierter Mechanismus zum Einsatz kommt.

Die große Zahl optionaler Protokollerweiterungen führt zwangsläufig zu Interoperabilitätsproblemen zwischen den Implementierungen verschiedener Hersteller. Daher werden von verschiedenen Hersteller- oder Betreiber-Konsortien so genannte *SIP Profiles* definiert. Dabei handelt es sich um Meta-Standards, die die Verwendung einer bestimmten Kombination von Parametern und optionalen Erweiterungen für eine bestimmte Anwendung in einem bestimmten Netz-Szenario spezifizieren. Für das Zusammenschalten von Netzen, in denen verschiedene SIP Profiles zum Einsatz kommen, werden u. U. entsprechende Gateways benötigt [34].

2.1.6 Alternative Signalisierprotokolle für IP-Telefonie

Für die IP-Telefonie stehen eine ganze Reihe von Signalisierprotokollen zur Auswahl, die von verschiedenen Gremien bzw. Herstellern entwickelt und standardisiert wurden. In den so genannten IP-Telefonie-Plattformen (siehe Abschnitt 3.3.2), welche nach den Plänen der etablierten Telefonnetzbetreiber das ISDN ablösen sollen, wird die Verbindungssteuerung nach derzeitigem Planungsstand überwiegend auf SIP basieren. Dennoch soll im Folgenden ein kurzer Überblick über alternative Signalisierprotokolle und -architekturen gegeben werden.

2.1.6.1 H.323

Die ITU-T Empfehlung H.323 [H.323] spezifiziert eine Familie von Signalisierprotokollen, die Konzepte der Signalisierung auf der ISDN-Teilnehmerschnittstelle [Q.931] übernehmen. Diese Protokolle fanden zeitweise eine recht große Verbreitung, um private Nebenstellenanlagen durch IP-Telefonie in lokalen Netzen (LAN) zu ersetzen [12]. In jüngerer Vergangenheit (Stand:

2006) verliert dieser Standard jedoch zu Gunsten von SIP an Bedeutung; im Umfeld der in Abschnitt 3.3.2 diskutierten IP-Telefonie-Plattformen wird die Verwendung von H.323 nicht in Erwägung gezogen.

2.1.6.2 MEGACO/H.248

Das *Media Gateway Control Protocol*, welches von der IETF und der ITU-T gemeinsam spezifiziert und als MEGACO [RFC 3525] bzw. Empfehlung H.248 [H.248.1 v3] standardisiert wurde, ist primär für die Verwendung im Umfeld von Netzübergängen in Netze mit anderen Technologien, insbesondere ISDN, vorgesehen. *Media Gateway Controller* (MGC) steuern damit *Media Gateways* (MGW, auch: MG) am Netzübergang, welche jeweils die RTP-Medienströme vieler simultaner Gespräche umsetzen, beispielsweise auf ein TDM-System eines ISDN-Netzes. In diesem Sinne ist MEGACO eher als Ergänzung, weniger als Konkurrenz zu SIP einzustufen.

Es gibt jedoch auch Ansätze (z. B. [35]), die dieses Protokoll zur Teilnehmersignalisierung verwenden. Die Verbindungssteuerung ist in diesem Fall sehr viel zentralisierter als bei der Verwendung von SIP oder H.323, d. h. es befindet sich weniger verbindungsbezogene Zustandsinformationen im VoIP-Client, dafür mehr in einer zentralen Steuerinstanz (z. B. Softswitch).

2.1.6.3 IAX2

Das *Inter-Asterisk eXchange (Version 2) Protocol* (IAX2), wurde als zunächst proprietäre Lösung im Umfeld der Open-Source Software „Asterisk“ entwickelt, welche zunächst als kostenlose Alternative zu kommerziellen Lösungen für private Nebenstellenanlagen entwickelt wurde. Mittlerweile erfreut sich Asterisk recht großer Beliebtheit und wird auch in professionellen Umgebungen eingesetzt [36].

IAX2, dessen Protokollspezifikation mittlerweile auch als IETF Draft veröffentlicht wurde [37], unterscheidet sich von den andern in diesem Abschnitt vorgestellten Protokollen dahingehend, dass es das einzige Protokoll ist, das Signalisierung und Mediendaten gemeinsam überträgt, also ein Protokoll mit Innenband-Signalisierung. Obwohl IAX2 ursprünglich für den Transport von Multimedia-Nutzdaten und Signalisierung zwischen den Asterisk-Servern (so genanntes *Trunking*) entwickelt wurde, existieren mittlerweile auch VoIP-Clients, die mit IAX2 an einen Asterisk-Server angebunden werden können, z. B. [38].

2.1.6.4 P2PSIP

Schon beim „klassischen“ SIP werden die meisten Funktionen in den Endsystemen platziert. Die einzigen Funktionen oberhalb der IP-Schicht, die nicht dort erbracht werden, sind die des *Domain Name Systems* (DNS) und des SIP-Registrars, welche benötigt werden, wenn Teilnehmer oder Endgeräte mobil sind, wenn IP-Adressen den Endsystemen nur temporär zugewiesen werden oder wenn eine teilnehmerbezogene statt endgerätebezogene Adressierung gewünscht wird.

Ziel der in der akademischen Welt viel betrachteten *Peer-to-Peer-SIP* (P2PSIP)-Arbeitsgruppe der IETF ist es, auch diese Funktionen dezentral und selbstorganisierend mit Hilfe einer *Distributed Hash Table* (DHT), zum Beispiel auf Basis von „Chord“ [39] zu erbringen, die die Abbildung von SIP-URIs auf IP-Adressen speichert. Somit soll ein offener Standard als Gegenstück zu den sehr erfolgreichen, proprietären Protokollen von „Skype“ (siehe Abschnitt 2.1.6.5) geschaffen werden. Derzeit (Ende 2006) tut sich diese Arbeitsgruppe allerdings noch sehr schwer mit der Definition eines Anwendungs-Szenarios, das nicht sehr einfach mit zentralen Servern implementiert werden könnte und bei dem die Sicherheitsanforderungen nicht doch wieder die Einführung zentraler Komponenten (z. B. zur Registrierung neuer Teilnehmer, wie bei Skype) erforderlich machen.

2.1.6.5 Proprietäre Protokolle, insbesondere Skype

Neben den oben genannten Protokollen existieren noch diverse andere, überwiegend herstellerbzw. betreiberspezifische Protokolle, z. B. das *Skinny Client Control Protocol* (SCCP), welches in IP-basierten privaten Nebenstellenanlagen des Herstellers *Cisco Systems* zum Einsatz kommt.

Vor allem bei privaten Nutzern erfreut sich derzeit (2006) die Internet-Telefonie-Lösung *Skype* [40] großer Beliebtheit. Skype verwendet so genannte *Peer-to-Peer*-Technik, um die Funktionen für eine Multimedia-Sitzung zwischen zwei Skype-Teilnehmern ohne zentrale Infrastruktur zu erbringen. Dazu wird u. a. das Verzeichnis aller Skype-Teilnehmer verteilt auf so genannten *Super Nodes* gespeichert. Jede „normale“ Skype-Instanz auf dem Endgerät beim Teilnehmer kann, ohne Zutun des Teilnehmers zum Super Node werden, falls automatische Messungen ergeben, dass das Endgerät über eine leistungsfähige CPU, sowie über eine breitbandige Internet-Anbindung ohne restriktive Firewall verfügt. Ein solcher Skype-Client unterhält viele Verbindungen zu anderen Clients und tauscht mit diesen auch dann Nachrichten aus, wenn sein jeweiliger Nutzer gerade nicht telefoniert, um so beim Verbindungsaufbau zwischen anderen Teilnehmern zu helfen. Dies kann ein erhebliches Transfervolumen auf dem Internet-Zugang verursachen. Für einige andere Funktionen des Skype-Dienstes wie das Anmelden eines Teilnehmers oder für Gespräche in das ISDN/PSTN muss jedoch vom Skype-Betreiber eine entsprechende Infrastruktur („login server“ bzw. Gateways) vorgehalten werden. Skype-Instanzen kommunizieren untereinander über ein proprietäres Protokoll, dessen Spezifikation vom Hersteller bewusst nicht veröffentlicht wird, und welches mit verschiedenen Verschlüsselungs- und Verschleierungs-Maßnahmen gegen Analyse und *Reverse Engineering* geschützt ist [41]. Ein Faktor, der wesentlich zum Erfolg von Skype beigetragen hat ist, dass es Techniken zum Überwinden restriktiv konfigurierter Firewalls oder Adressumsetzer enthält. Dazu wird u. a. systematisch nach Regeln in der Firewall gesucht, die eigentlich zur Freigabe bekannter anderer Anwendungen dienen sollen [42]. Somit kann Skype auch in vielen Umgebungen (z. B. Firmennetzen) verwendet werden, in denen die Nutzung nicht explizit durch den Sicherheits-Verantwortlichen freigegeben wurde. Dies kann – je nach Betrachterstandpunkt – als praktische Eigenschaft oder als Bedrohung gesehen werden.

2.2 Sicherheit in Kommunikationsnetzen

Der Schwerpunkt dieses Teilkapitels liegt in der Einführung von Grundbegriffen der Netzsicherheit. Darauf aufbauend werden im nächsten Abschnitt die Grundkonzepte so genannter *Firewalls* vorgestellt, welche einen wesentlichen Beitrag zur Absicherung von Kommunikationsnetzen leisten können.

2.2.1 Grundbegriffe der Netzsicherheit

Ausgangspunkt von Analysen zur Sicherheit von Kommunikationsnetzen sind i. d. R. die so genannten *Sicherheitsanforderungen*. Werden diese abstrakten Anforderungen mit einem schützenswerten Gut, einem *Wert* verknüpft, so spricht man von einem *Schutzziel*. *Netzsicherheit* beschreibt die Eigenschaft eines Kommunikationsnetzes und der damit in Verbindung stehenden Endsysteme, eine wohldefinierte Menge von Schutzzielen zu erfüllen.

Eine Menge von grundsätzlichen Basisanforderungen, aus denen sich alle weiteren Sicherheitsanforderungen zusammensetzen lassen, wurde anwendungsnah in [43] definiert. Grundsätzliche Sicherheitsanforderungen, die die *Verlässlichkeit*, d. h. die Sicherheit eines Systems während der Diensterbringung betreffen, sind demzufolge die *Vertraulichkeit*, die *Integrität* und die *Verfügbarkeit* eines schützenswerten Gutes. Wird Kommunikation, die bisher auf „herkömmliche Weise“ (z. B. direkte Sprach-Kommunikation zwischen Menschen, Briefe, etc.) durchgeführt wurde, durch Telekommunikation ersetzt, so ergeben sich weitere Sicherheitsanforderungen bzgl. der *Beherrschbarkeit* dieser neuartigen Kommunikationsweise, nämlich die *Zurechenbarkeit* und die *Rechtsverbindlichkeit*.

Verschiedene Betroffene können verschiedene, u. U. sogar widersprüchliche Schutzziele haben. So kann z. B. ein Teilnehmer den Wunsch nach anonymer Kommunikation haben (Schutzziel: Vertraulichkeit der eigenen Identität gegenüber dem gerufenen Teilnehmer und/oder dem Netzbetreiber), während z. B. der Netzbetreiber sicherstellen will, dass er für seine Leistungen auch die ihm zustehenden Entgelte erhält (Schutzziele: Zurechenbarkeit von Rufen und der dabei in Anspruch genommenen Ressourcen zu einem Teilnehmer sowie Rechtsverbindlichkeit der erstellten Abrechnungen). Dementsprechend kann ein Kommunikationsnetz auch nicht global als *sicher* oder komplementär als *nicht sicher* erklärt werden. Ein System kann stattdessen nur als *sicher* aus dem Bezugspunkt eines Betroffenen erklärt werden, wenn es alle Schutzziele dieses Betroffenen erfüllt. Das Finden von Lösungen, die die Schutzziele aller Betroffenen erfüllen bzw. das Aushandeln von fairen Kompromissen, bei denen einzelne Betroffene einige ihrer jeweiligen Schutzziele bewusst und gezielt aufgeben, um ein Zustandekommen der Kommunikation nicht zu verhindern, wird unter dem Begriff *mehrseitige Sicherheit* [44] zusammengefasst.

Als *Angriff* wird eine unautorisierte Handlung eines *Angreifers*, d. h. einer Instanz, i. d. R. einer Person, bezeichnet, die im Erfolgsfall zur Verletzung eines oder mehrerer Schutzziele anderer Betroffener führt. Angriffe können sich nicht nur auf die eigentlichen *Nutzdaten*, d. h. die kommunizierten bzw. verarbeiteten Informationen selbst beziehen. Auch beispielsweise allein die Tatsache, dass Kommunikation stattgefunden hat und welche Parteien daran beteiligt waren, kann Schutzzielen unterliegen (z. B. Vertraulichkeit gegenüber Dritten, Integrität der für die Abrechnung verwendeten Kommunikationsdatensätze). Angriffe gegen die Kommunikations-

Infrastruktur gefährden nicht nur die Verfügbarkeit der Dienste (so genannte *Denial-of-Service*-Attacken, DoS). So kann die Kompromittierung der funktionellen Integrität eines Schutzmechanismus (z. B. Deaktivierung einer Zugriffskontrolle durch Löschen der Konfigurationsdateien) ein Einfallstor für weitere Angriffe gegen andere Schutzziele öffnen. Ein *Angreifermodell* beschreibt die maximal unterstellte Stärke angenommener Angreifer, indem es charakteristische Attribute wie z. B. die *Motivation* für den Angriff, *Zugangsmöglichkeiten* zum System, das *Systemwissen* über das anzugreifende System, sowie die technischen, finanziellen und zeitlichen *Ressourcen* des Angreifers benennt. Es bildet somit die Basis für die Planung von Sicherheitsmechanismen bzw. für die Bewertung von Sicherheitseigenschaften eines Systems.

Die so genannten *Sicherheitsmechanismen* beschreiben technische und organisatorische Verfahren, die auf Werte angewendet werden, um bestimmte Schutzziele zu erreichen [43]. Ein fiktiver, omnipotenter Angreifer, der über beliebig viele Ressourcen und Wissen verfügt, und der physischen Zugang zu allen Systemen besitzt, könnte beliebig aufwändige Sicherheitsmechanismen brechen. Ferner ist es aus wirtschaftlichen Gründen nicht sinnvoll, in den Schutz eines Gutes mehr Ressourcen und damit letztendlich Geld zu investieren, als dem Schaden entspricht, der bei einem erfolgreichen Angriff auftritt. Aus der im Angreifermodell (s. o.) spezifizierten maximalen angenommenen Stärke der Angreifer leitet daher sich die (minimal) benötigte *Schutzhöhe* ab, d. h. die Widerstandsfähigkeit der Maßnahmen, die die Angriffe abwehren sollen.

Bei der Verwendung technischer Schutzmechanismen muss festgelegt werden, wo diese implementiert werden sollen. Diese so genannte *Allokation* von Sicherheitsmechanismen umfasst sowohl einen *horizontalen Freiheitsgrad*, der die räumliche Anordnung beschreibt (z. B. Integration einer Sicherheitsfunktion in ein vorhandenes Netzelement, oder „Einschleifen“ eines eigenen Elements in den Übertragungspfad) und einen *vertikalen Freiheitsgrad*, der die logische Platzierung (z. B. relativ zu den Schichten des Protokollstapels) beschreibt. Hierbei muss auch berücksichtigt werden, dass es zu Abhängigkeiten und Wechselwirkungen zwischen den Basis-Sicherheitsanforderungen kommen kann. So könnte z. B. ein Angreifer die für die Verkehrslenkung im Netz verwendeten Tabellen manipulieren um zu erreichen, dass eine vertrauliche Nachricht nicht den direkten, für den Angreifer nicht zugänglichen Weg zwischen zwei anderen Teilnehmern nimmt, sondern einen Umweg nimmt, an dem sie vom Angreifer abgehört werden kann. Die Verletzung eines (möglicherweise zweitrangigen) Schutzziels (Integrität der Routing-Tabelle) führt in diesem Fall zur Verletzung eines ganz anderen Schutzziels (Vertraulichkeit der Nachricht).

2.2.2 Separation und Mediation

Zwei wichtige Grundmechanismen zum Schutz verschiedener Informationen bzw. Informationsflüsse, die in einem System verarbeitet bzw. übertragen werden, sind die Separation und die Mediation [45]. Die *Separation* zielt dabei zunächst auf die Trennung von Nachrichten bzw. von Betriebsmitteln, die zu Nachrichtenverarbeitung und -transport verwendet werden, sofern die Nachrichten verschiedenen Schutzzielen unterliegen bzw. verschiedenen Betroffenen zuzuordnen sind. Bei *physikalischer Separation* erfolgt die Trennung z. B. über räumliche Distanz, über das Verwenden jeweils dedizierter Leitungen, etc. Verwandt ist die *temporäre Separation*, bei der Nachrichten zeitlich versetzt mit dem selben System übertragen bzw. verarbeitet werden. Zwischenzeitig wird das System in einen wohldefinierten Ausgangszustand zurückgesetzt,

um ungewollte „Informations-Leckagen“ zwischen den Bereichen zu vermeiden. Diese beiden Verfahren stehen in einem gewissen Widerspruch zu dem Ziel, durch gemeinsame Nutzung von Ressourcen einen *Bündelungsgewinn* und damit eine effizientere Ressourcennutzung zu erzielen. Bei der *logischen Separation* können Ressourcen hingegen gemeinsam benutzt werden, die Kennzeichnung der Zugehörigkeit einer Nachricht zu einem bestimmten Sicherheits-Kontext erfolgt durch explizite oder implizite Bezeichner (z. B. IP-Adressen, verschiedene Wellenlängen auf einer Glasfaser, etc.). Sämtliche Zugriffe nicht voll vertrauenswürdiger Instanzen auf diese Informationen müssen bei diesem Schutzkonzept von einem vertrauenswürdigen Referenz-Monitor (s. u.) überwacht und autorisiert werden. Bei der *kryptographischen Separation* werden Nachrichten mit Hilfe kryptographischer Algorithmen, die durch so genannte *kryptographische Schlüssel* parametrisiert werden, zu Chiffraten transformiert. Falls die benötigten Schlüssel nur berechtigten Instanzen bekannt sind, können die Schutzziele „Integrität und Vertraulichkeit übertragener Nachrichten“ selbst dann erreicht werden, wenn Angreifer Zugriff auf die Chiffrate haben. Ohne Kenntnis der Schlüssel können Angreifer die *verschlüsselten* Nachrichten nicht interpretieren (Schutz der Vertraulichkeit) bzw. sie können keine *digitalen Signaturen* für unautorisiert erzeugte bzw. veränderte Nachrichten erzeugen (Schutz der Integrität).

Zwischen den so separierten Bereichen realisiert die *Mediation* eine Vermittlung, unter Berücksichtigung der jeweiligen Schutzziele. Ein wichtiger Aspekt ist die Zugriffskontrolle, d. h. die Prüfung von Informationen, ob sie in einen Bereich hineinfließen bzw. ihn verlassen dürfen. Es kann aber auch zu den Aufgaben des Mediators gehören, vertrauliche Daten zu verschlüsseln, bevor sie durch einen Transit-Bereich geleitet werden, in dem Angreifer vermutet werden [46].

2.2.3 Zugriffskontrolle

Das grundsätzliche Verfahren zur *Zugriffskontrolle* mit Zugriffskontrolllisten (engl. *Access Control List*, ACL) ist in Abbildung 2.12 schematisch dargestellt. Eine Instanz (engl. *Principal*, z. B. Person, Software-Prozess, etc.) stellt eine Anfrage (engl. *Request*, z. B. lesen, erstellen, modifizieren, löschen, etc.) bezüglich eines Objektes (engl. *Object*, z. B. Datei, Netz- oder CPU-Ressource, Adresse, etc.). Zunächst erfolgt eine *Authentisierung* (engl. *Authentication*), bei der die anfordernde Instanz identifiziert wird oder die von ihr behauptete Identität überprüft wird. Dies kann auf Basis von Eigenschaften oder Dingen geschehen, die die Instanz hat (z. B. Dokument einer vertrauenswürdigen dritten Partei, das die Identität bestätigt), weiß (z. B. Passwort), oder tun kann (z. B. Berechnung von Antworten auf bestimmte Fragen). Nachdem so die Identität festgestellt wurde, wird geprüft, ob die Instanz zu der gewünschten Operation bzgl. der gewünschten Objekte berechtigt ist. Dieser Schritt wird *Autorisierung* (engl. *Authorization*) genannt. Dazu wird in einer Zugriffskontrollliste nachgeschaut, die Tupeln aus (Instanz, Operation, Objekt) die Werte *erlaubt* oder *nicht erlaubt* zuordnet.

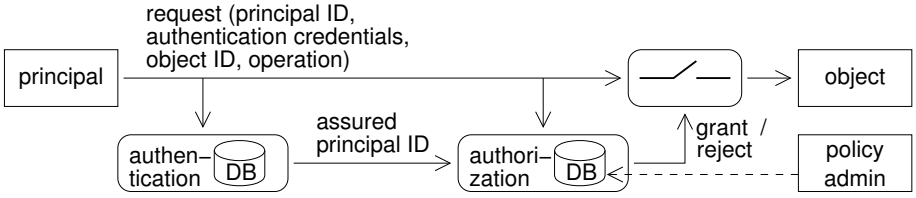


Abbildung 2.12: ACL-basierte Zugriffskontrolle (schematisch)

Ein alternatives Verfahren, bei dem die Identität der anfragenden Instanz dem Mediator nicht bekannt sein muss, ist in Abbildung 2.13 dargestellt. Die anfragende Instanz legt dem Mediator zusammen mit der Anfrage ein Dokument vor, welches für den Mediator (z. B. mit Hilfe digitaler Signaturen) nachprüfbar von einer vertrauenswürdigen Dritten Instanz stammt, und welches aussagt, dass die Instanz, die dieses Dokument vorlegt, zu der Operation berechtigt ist.

2.2.4 Vertrauensdomänen und Platzierung von Zugriffskontroll-Mechanismen

Neben der Spezifikation von Schutzzielen und Angreifermodell und dem Ableiten der benötigten Schutzhöhe (s. o.) ist es für eine realistische und effiziente Planung *skalierbarer Schutzmechanismen* notwendig, das Gesamt-System in *separat sicherbare Bereiche* [47] zu untergliedern, z. B. auf Basis technischer und organisatorischer Gegebenheiten, den Schutzzielen, etc. Ein Bereich, der mehrere Subsysteme mit gleichen oder ähnlichen Gegebenheiten und Bedrohungspotentialen umfasst, kann dann als Ganzes abgesichert werden. So genannte *Vertrauensdomänen* sind Bereiche, die als frei von Angreifern (bzgl. bestimmter Schutzziele) angenommen werden. Diese Annahme kann auf verschiedenartigen Grundlagen basieren, z. B. der technisch/wissenschaftlichen Annahme, dass kein effizientes Verfahren bekannt ist, einen bestimmten kryptographischen Algorithmus zu brechen. Auch organisatorische Argumente können eine Rolle spielen, z. B. dass sich das betrachtete System in einem physikalisch zugangsgeschützten Gebäude befindet, zu dem nur Mitarbeiter Zugang haben, denen per Dienstanweisung bestimmte Verhaltensweisen untersagt sind.

Die Mediation, d. h. die Vermittlung zwischen separierten Bereichen, die sich gegenseitig nicht (voll) vertrauen, muss auf eine Weise durchgeführt werden, dass alle Bereiche darauf vertrauen können, dass ihre jeweiligen Schutzziele gewahrt werden. Für die Platzierung entsprechender Mechanismen existieren verschiedene Möglichkeiten, mit unterschiedlichen Auswirkungen darauf, wie vielen externen Instanzen vertraut werden muss. Mehrbenutzer-Betriebssystemen (z. B. UNIX) sind ein typisches Beispiel für Konfigurationen, bei denen sich nicht vertrauende Bereiche (hier: Prozesse), über einen zentralen, gemeinsam vertrauten Bereich (das Betriebssystem) kommunizieren. So genannte Referenz-Monitore im Betriebssystem führen Zugriffskontollen auf Interprozess-Kommunikation, Datei-Zugriffe, etc. durch (siehe Abbildung 2.14 a)). Bei der Zusammenschaltung IP-basierter Netze ist es hingegen üblich, dass jeder Bereich seine jeweiligen Schutzziele durch Maßnahmen am Rande des eigenen Bereichs (z. B. Firewalls) gewährleistet. (siehe Abbildung 2.14 b)). Eine Nachricht, die eine Bereichsgrenze überschreitet, wird so zweimal (den einen Bereich verlassend und den anderen Bereich betretend) geprüft; somit wird kein gemeinsam vertrauter Bereich benötigt. Insbesondere bei kryptographischen Verfahren ist oft weniger der eigentliche Mechanismus zur Verschlüsselung bzw. Signatur der

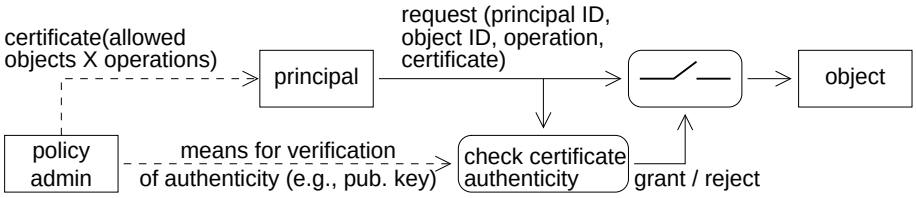


Abbildung 2.13: Zertifikat-basierte Zugriffskontrolle (schematisch)

Daten ein Problem, sondern der skalierbare Aufbau von Vertrauens-Verhältnissen und der dazugehörige Austausch der kryptographischen Schlüssel. Dieser kann mit Hilfe einer *Public Key Infrastructure* (PKI) zentralisiert werden, wohingegen die Verschlüsselung dezentral bleibt (siehe Abbildung 2.14 c)). Schutzmechanismen können von Bereichen zum eigenen Schutz oder auch kooperativ zum Erreichen von Schutzzielen anderer Bereiche implementiert werden. Ein Beispiel hierfür ist das Sicherheitskonzept lokaler Netze (z. B. Ethernet). Die Vertraulichkeit von Nachrichten, die zwischen zwei Stationen ausgetauscht werden, ist nur gewährleistet, wenn die Netzwerkkarten bzw. Betriebssysteme aller anderen an das Netz angeschlossenen Stationen die entsprechenden Frames nicht an möglicherweise boshafte lokale Prozesse ausliefern (siehe Abbildung 2.14 d)). Möglichkeiten zum Erreichen eigener bzw. fremder Schutzziele durch Zugriffskontrolle am Rand eines Bereiches sind in Abbildung 2.15 nach [48] zusammengefasst.

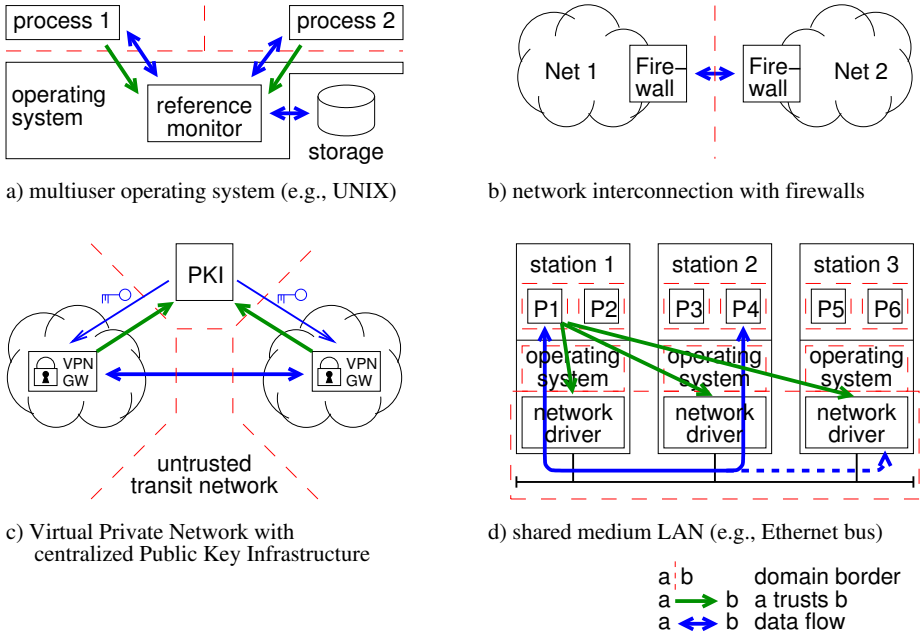
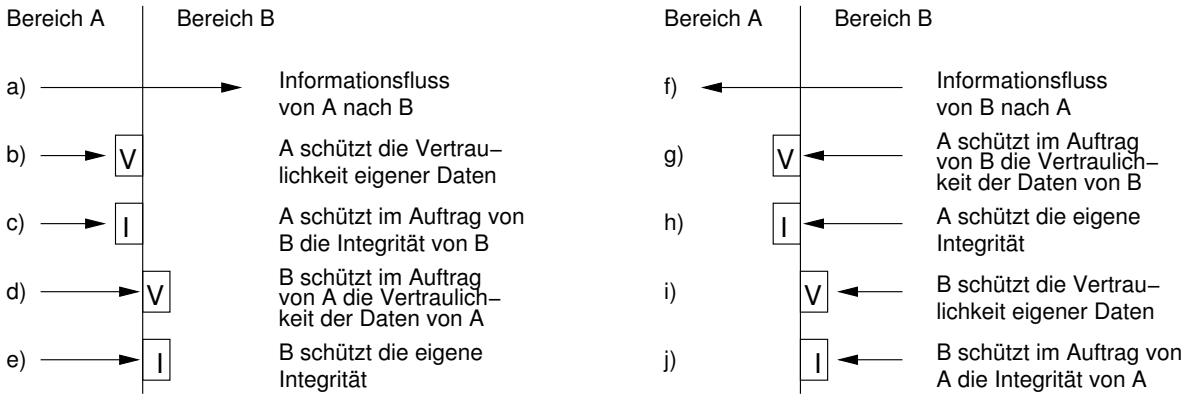


Abbildung 2.14: Beispielhafte Konfigurationen bei der Mediation zwischen Bereichen



(a) Schutz bzgl. von Informationsflüssen von A

(b) Schutz bzgl. von Informationsflüssen an A

Abbildung 2.15: Veranschaulichung von Selbstschutz und delegiertem Schutz

2.3 Firewalls

2.3.1 Begriffsdefinition und grundsätzliche Aufgaben

Firewalls sind ein in der Praxis allgemein akzeptiertes und vielfach angewendetes Konzept zum Schutz von Netzübergängen, insbesondere von IP-basierten Kommunikationsnetzen. Dennoch existiert in der Literatur keine allgemein anerkannte Definition dieses Begriffs. Dementsprechend wird er sehr uneinheitlich verwendet, vom Synonym für Paketfilter (s. u.) bis hin zur Bezeichnung für die Gesamtheit aller Sicherheitsmechanismen am Netzübergang. Unstrittig ist, dass zu den Hauptaufgaben einer Firewall gehört, *Zugriffskontrollen auf den eine Bereichsgrenze überschreitenden Netzverkehr* durchzuführen. Bei einer Firewall handelt es sich also prinzipiell um ein System mit Gateway-Funktionalität, das Verkehr aber nur selektiv, entsprechend der jeweiligen Sicherheitsrichtlinie weiterleitet.

Firewalls werden sehr oft für die Anbindung IP-basierter lokaler Netze an das Internet verwendet. Ein wesentliches Konzept hierbei ist die Unterscheidung zwischen dem vertrauenswürdigen lokalen Netz (Bereich) *innerhalb* der Firewall und dem nicht vertrauenswürdigen Internet *außerhalb*. Durch das konsequente Hindurchleiten sämtlichen Verkehrs durch ein (oder wenige) Firewall-Systeme und entsprechende Zugriffskontrollen dort können u. U. die Anforderungen an die Schutzmechanismen der Systeme innerhalb des vertrauenswürdigen Bereiches etwas gelockert werden [RFC 1244, Sec. 3.9.1].

Neben der Zugriffskontrolle gehört zu den weiteren, optionalen Aufgaben einer Firewall, die *Modifikation* von Nachrichten (z. B. Entfernen von als schädlich identifizierten Nachrichten-Teilen, z. B. Viren) und die *Auditierung*, d. h. das Protokollieren abgewiesener Nachrichten. *Adressumsetzungen* werden in Firewalls durchgeführt, um Rückschlüsse auf die Identität einzelner Teilnehmer oder die Netz-Topologie eines Bereiches zu verhindern. In IP-Routern an der Grenze zwischen lokalen Netzen und dem Internet findet oft eine so genannte *Network Address and Port Translation* (NAPT)[RFC 2663, RFC 3022] statt. Dies geschieht meistens primär aus einem nicht unmittelbar sicherheitsrelevanten Grund, der Einsparung global eindeutiger IPv4-Adressen. Dass dabei i. d. R. der Aufbau neuer TCP-Verbindungen bzw. UDP-Flows nur in einer Richtung (vom Client im LAN zum Server im Internet) möglich ist, wird jedoch von vielen Systemverantwortlichen als ein die Sicherheit erhöhender Nebeneffekt gesehen [RFC 4864].

Ein thematisch verwandter Begriff ist der der so genannten *Middlebox*, welche in [RFC 3234] definiert wird als *„any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host“*, d. h. als ein Transit-Netzelement auf dem Pfad zwischen zwei Endsystemen, das Funktionen ausführt, die über die eines „normalen“ IP-Routers hinausgehen. Diese recht breite Definition, die keine Aussage über den Zweck eines Netzelements (z. B. Sicherheitsmechanismus) macht, umfasst praktisch alle Netzelemente, die als Firewall eingesetzt werden können.

Vor diesem Hintergrund sollen in dieser Arbeit die Begriffe wie folgt verwendet werden:

Als *Firewall* soll das Gesamtsystem bezeichnet werden, das eingesetzt wird, um Bereiche mit verschiedenen Sicherheitanforderungen und -niveaus abzusichern, wofür primär Zugriffskontrollen auf den einen Netzübergang überquerenden Verkehr und ggf. weitere, flankierende Maß-

nahmen angewendet werden. Ein einzelnes Netzelement, welches entweder alleine oder im Verbund mit anderen Elementen zum Aufbau der Firewall verwendet wird, soll allgemein als *Firewall-Element* bzw. *Middlebox*, oder spezifischer als *Paketfilter*, *Proxy*, etc. (Klassifizierung siehe unten) bezeichnet werden.

2.3.2 Prinzipieller Aufbau eines Firewall-Elements

Abbildung 2.16 zeigt schematisch den grundsätzlichen Aufbau eines Firewall-Elements, d. h. einer Middlebox mit Firewall-Funktionalität (Abbildung nach [49, 50], verfeinert). Es handelt sich dabei um eine mögliche Implementierung des in Abbildung 2.12 dargestellten Grundverfahrens zur Zugriffskontrolle. Der Fluss der Nutzdaten durch das Netzelement ist in der Abbildung durch dicke Pfeile dargestellt; dünne Pfeile stellen Steuerinformationen dar.

Prinzipiell handelt es sich bei einem solchen System um eine Netzkoppeleinheit (engl. *Interworking Unit*), die um Funktionen zur Zugriffskontrolle erweitert wurde. Die Vermittlungsfunktion wird von der *Interworking Function* sowie den darunterliegenden Schichten des Protokollstapels erbracht. Dazu gehören insbesondere Funktionen zur Verkehrslenkung (engl. *Routing*), da ein Firewall aufgrund seiner Position auf einer Bereichsgrenze prinzipbedingt mindestens zwei (ggf. logische) Schnittstellen hat und Verkehr zwischen den Bereichen entsprechend weiterlei-

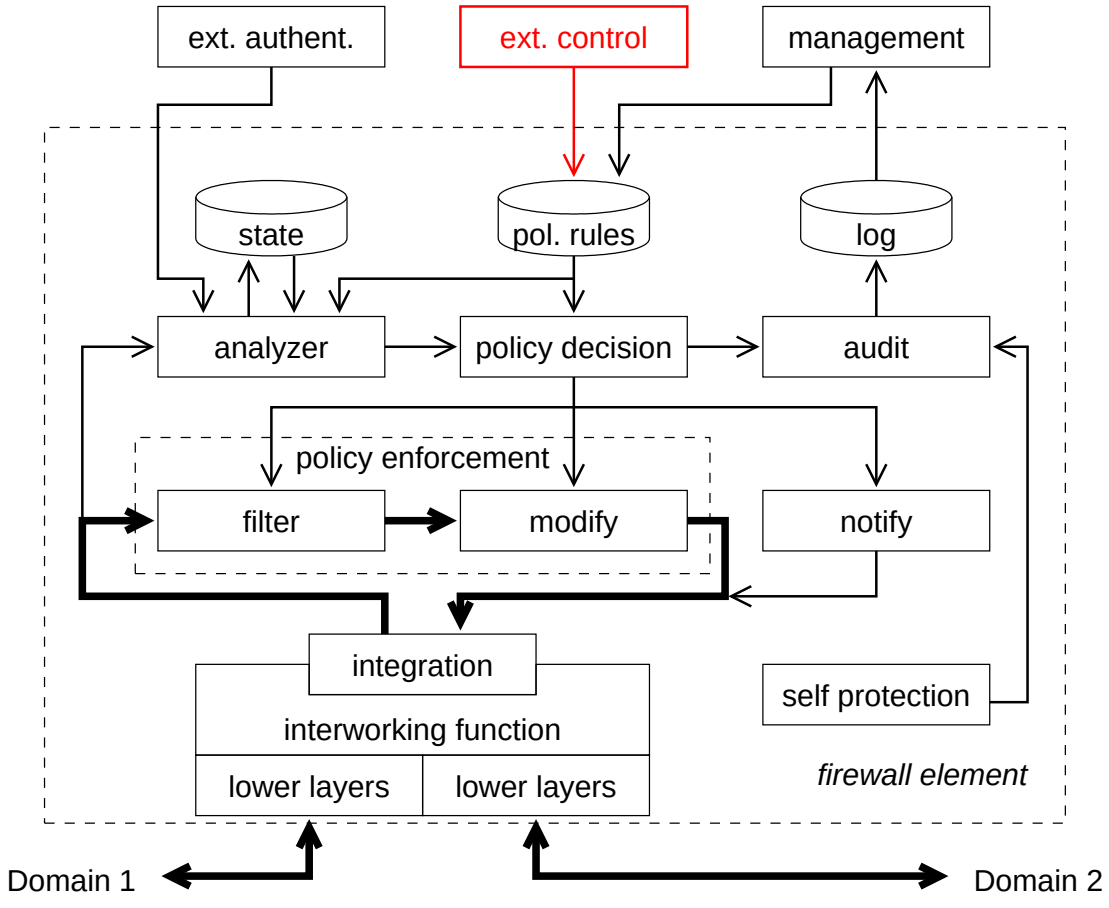


Abbildung 2.16: Blockschaltbild eines Firewall-Elements

ten muss. Die Protokollschicht, auf welcher die Interworking Function arbeitet, ist ein wesentliches Merkmal zur Klassifizierung von Middleboxes (siehe Abschnitt 2.3.3.2).

Die Module zur Erbringung der Sicherheitsfunktionen werden mit Hilfe des Einbindungsmoduls (engl. *Integration Module*) in den Datenpfad eingeschleift. Bei der Implementierung dieses Moduls ist besonders darauf zu achten, dass keine Daten an den Sicherheitsmodulen vorbeifließen können [49].

Die durch das Firewall-Element fließenden Datenströme werden im Analyse-Modul (engl. *Analyzer Module*) untersucht. Die dazu verwendeten Parameter und Methoden, die u. U. auch ein Halten von Zustandsinformationen (engl. *State*) oder eine Abfrage externer Authentisierungs-Systeme erfordern (siehe Abschnitt 2.3.3.3), sind ein weiteres wichtiges Kriterium zur Klassifizierung. Das Analyse-Ergebnis wird an das Entscheider-Modul (engl. *Policy Decision Module*) weitergeleitet, welches darüber entscheidet, wie mit den analysierten Daten zu verfahren ist. Dazu werden Zugriffskontrolllisten (engl. *Access Control Lists*) konsultiert. Diese enthalten Regeln (engl. *Policy Rules*), die einer Reihe von Bedingungen (engl. *Conditions*) eine Aktion (engl. *Action*) zuordnen. Diese wird im Umsetzer-Modul (engl. *Policy Enforcement Module*) durchgeführt, welches z. B. aus je einem Unter-Modul zum Verwerfen (engl. *Filter*) von ganzen Nachrichten oder Teilen davon oder zum modifizieren (engl. *Modify*) dieser bestehen kann. Zusätzlich können weitere Aktionen ausgelöst werden, z. B. das Protokollieren (engl. *Audit*) eines sicherheitsrelevanten Ereignisses, oder z. B. das Benachrichtigen (engl. *Notify*) des Absenders, dass die Nachricht wegen Verstoß gegen eine Sicherheitsrichtlinie verworfen wurde.

Das Firewallschutzmodul (engl. *Self Protection Module*) implementiert Schutzfunktionen, die eine Kompromittierung des Firewalls selbst und damit eine Beeinträchtigung seiner Schutzfunktion verhindern sollen. Auch hier können Protokoll-Daten zu sicherheitsrelevanten Ereignissen anfallen.

2.3.3 Klassifikation von Firewall-Elementen

2.3.3.1 Vertikaler Freiheitsgrad der Allokation

Das abstrakte Konzept „Firewall“ kann in IP-Netzen durch sehr verschiedene technische Maßnahmen realisiert werden. Oft werden Firewalls auch durch eine Kombination mehrerer Middleboxes implementiert. Ein wesentliches Merkmal bei der Klassifikation solcher Firewall-Elemente ist die vertikale Allokation der Sicherheitsfunktionen, d. h. auf welcher Schicht des Protokollstapels das Netzelement arbeitet. Hierbei ist zu unterscheiden zwischen der Schicht, auf der die Nachrichten vermittelt werden („Interworking Layer“) und den Schichten, die bei der Zugriffskontroll-Entscheidung berücksichtigt werden.

2.3.3.2 Kopplungs-Schicht

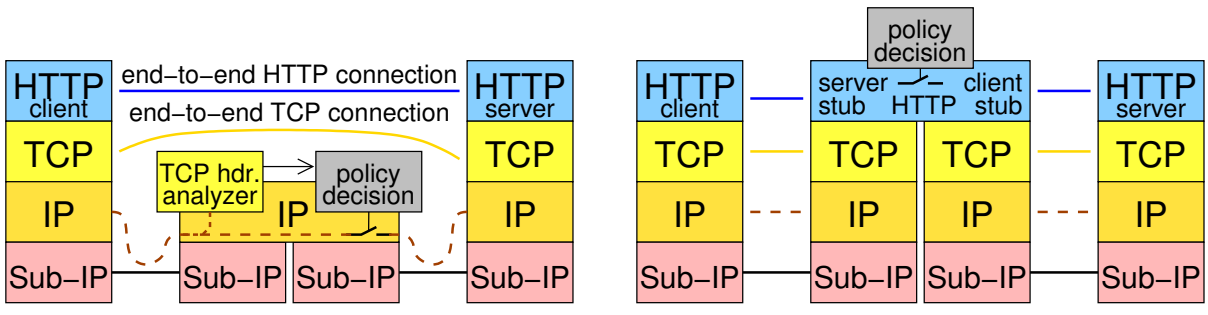
Die Schicht, auf der die Weiterleitung von Nachrichten durchgeführt wird, hat einen großen Einfluss darauf, wie die Middlebox in das umgebende Netz eingebunden wird. Sie ist auch ausschlaggebend für die spezifischere Bezeichnung der Middlebox.

So genannte *Paketfilter* (engl. *Packet Filter*) sind Router, die um Funktionen zur Zugriffskontrolle ergänzt wurden. Sie arbeiten auf der IP-Schicht und leiten einzelne IP-Pakete weiter, nachdem jeweils individuell eine Zugriffskontroll-Entscheidung getroffen wurde. *Application Layer Gateways*, auch *Proxy* genannt, terminieren hingegen die Transportschichtverbindung vollständig, interpretieren das Anwendungsschicht-Protokoll und setzen dies vollkommen neu auf (siehe Abbildung 2.17).

Paketfilter und Proxies sind die beiden wichtigsten Grundtypen von Middlebox, die für Firewalls verwendet werden, aber auch in den anderen Schichten ist Vermittlung mit Zugriffskontrolle möglich, z. B. SOCKS-Proxies in der Transportschicht oder Brücken (engl. *Bridges*) mit Zugriffskontrollfunktionen im lokalen Netz. Diese sind jedoch weniger gebräuchlich [49].

2.3.3.3 Analysierte Schichten

Für die Zugriffskontroll-Entscheidung können nicht nur Parameter der Protokollschicht herangezogen werden, auf der die Netzkopplung stattfindet, sondern auch solche von darunterliegenden und darüberliegenden Schichten. Ein wichtiger Freiheitsgrad beim Entwurf von Firewall-Systemen ist, inwieweit die oberen Protokollschichten analysiert werden sollen. Systeme, die nur die unteren Protokollschichten betrachten, zeichnen sich durch eine vergleichsweise einfache Implementierbarkeit, eine universelle Anwendbarkeit unabhängig von der Menge der verwendeten Anwendungen und durch vergleichsweise hohe Skalierbarkeit aus, da keine oder nur wenig verbindungsbezogene Zustandsinformationen gehalten werden müssen. Sollen hingegen auch höhere Protokollschichten in die Entscheidung mit einbezogen werden, so steigt der Aufwand, sowohl bei der Implementierung als auch zur Laufzeit, da mehr Parser für verschiedene Protokolle implementiert werden müssen und da die Analyse höherer Schichten i. d. R. Zustandsinformationen zwischen den einzelnen Nachrichten gehalten werden müssen. Andererseits ist so eine besonders feingranulare Zugriffskontrolle möglich, die z. B. einzelne Methoden des Anwendungsprotokolls unterscheiden kann.



(a) Paketfilter (b) Proxy

Abbildung 2.17: Paketfilter und Application Layer Gateway (Proxy) (schematisch):
Die logischen Verbindungen der Transport- und Anwendungsschicht laufen Ende-zu-Ende durch den Paketfilter hindurch; für jedes einzelne IP-Paket wird dort eine Zugriffskontrollentscheidung getroffen. Beim Proxy werden hingegen die logischen Verbindungen aller Protokollschichten bis einschließlich der Anwendungsschicht terminiert und neu aufgesetzt.

Paketfilter führen Zugriffskontrollen auf IP-Pakete durch, d. h. für jedes einzelne IP-Paket wird vom Paketfilter entschieden, ob es weitergeleitet oder verworfen werden soll. Für diese Entscheidung werden i. d. R. nicht nur Informationen aus dem IP-Header, sondern auch aus der Transportschicht (z. B. TCP/UDP-Portnummern), sowie ggf. aus der MAC-Schicht (insbes. MAC-Adressen) und der Bitübertragungsschicht (insbes. Kennung der physikalischen Schnittstelle) berücksichtigt. Informationen aus Protokollschichten oberhalb der Transportschicht (z. B. Anwendungsschicht), werden i. d. R. nicht berücksichtigt, eine von der Transportschicht ggf. durchgeführte Segmentierung von Nachrichten wird nur in seltenen Fällen rückgängig gemacht (z. B. so genanntes *TCP Stream Reassembly*). Eine solche, so genannte *Deep Packet Inspection*, würde die Vorteile der Paketfilter gegenüber anderen Firewall-Element-Typen bezüglich Implementierungsaufwand, Vielseitigkeit und Skalierbarkeit u. U. stark relativieren.

Application Layer Gateways müssen, schon alleine um ihrer Netzkopplungs-Funktionalität nachkommen zu können, die Protokolle sämtlicher Schichten bis einschließlich der Anwendungsschicht terminieren und neu aufsetzen. Es ist somit naheliegend, die Parameter und die gehaltenen Zustandsinformationen all dieser Schichten bei der Zugriffskontrollentscheidung miteinzubeziehen. Dies erlaubt Zugriffskontrollen mit feinerer Granularität. So kann z. B. mit einem HTTP-Proxy gezielt der Zugriff auf einzelne URLs auf einem Server gesperrt werden, wohingegen mit einem Paketfilter nur der Zugriff auf den HTTP-Dienst eines Endsystems vollständig gesperrt werden kann (siehe Abbildung 2.17).

2.3.3.4 Halten von Zustandsinformationen

Eng verwandt mit der Frage, welche Protokollschichten bei der Zugriffskontroll-Entscheidung berücksichtigt werden, ist die Frage, wieviele verbindungs- bzw. transaktionsbezogene Zustandsinformationen dabei gehalten werden. Prinzipiell sind hier unabhängig von der analysierten Protokollschicht verschiedene Vorgehensweisen möglich, von einem einfachen Vergleich bestimmter Protokoll-Kopffelder mit im Firewall-Element statisch hinterlegten Mustern, über das Nachvollziehen der Zustände des Protokolls bis hin zum Abfragen externer Informationsquellen.

Ein häufiges Ziel bei der Konfiguration von Paketfiltern ist beispielsweise, den Aufbau neuer TCP-Verbindungen von der „sicheren“ zur „unsicheren“ Seite hin zu erlauben, nicht jedoch in der Gegenrichtung. Dies kann ohne Speichern von Zustandsinformationen in einem so genannten *Stateless Packet Filter* implementiert werden, indem lediglich „erste“ Pakete in der verbotenen Richtung verworfen werden, d. h. IP-Pakete, die eine neue TCP-Verbindung aufbauen wollen und die an der charakteristischen Kombination von Bitschaltern (engl. *Flags*) im TCP-Protokollkopf (*SYN*=True, *ACK*=False) erkennbar sind. Folge-Pakete, d. h. Pakete, die keine neue Verbindung aufbauen wollen, werden einfach in beiden Richtungen durchgelassen. Auch wenn das Ziel so auf einfache Weise erreicht wird, hat dieses Verfahren gewisse Nachteile. Ein Angreifer kann bei dieser Firewall-Konfiguration Pakete in das zu schützende Netz senden, die aussehen, als ob sie zu einer bestehenden, von der „sicheren“ Seite aus aufgebauten Verbindung gehören, obwohl ein entsprechender Verbindungsaufbau nie stattgefunden hat.

Für einen Angreifer kann es verschiedene Gründe geben, eine Nachricht eines bestimmten Typs (z. B. Folge-Paket) an eine Protokollinstanz zu senden, obwohl diese laut Protokollspezifikation im aktuellen Zustand der Protokollinstanz (z. B. Verbindung geschlossen) gar nicht auftreten

dürfte. Eine unsachgemäße Implementierung dieser Protokollinstanz, z. B. fehlende Fehlerbehandlungsroutinen für diese Fälle, kann die Verfügbarkeit der Protokollinstanz gefährden [51] und u. U. bis hin zur Kompromittierung des Systems führen. Antwortet das System auf unerwartete Nachrichten-Typen hingegen mit einer Fehler-Nachricht, und kann der Angreifer die unerwarteten Nachrichten mit einer Quell-Adresse versehen, die nicht seiner eigenen Adresse entspricht (*Spoofing*), so wird die Fehler-Nachricht an einen unbeteiligten Dritten versendet, dem diese Adresse eigentlich gehört. Sowohl bei dieser so genannten *Reflektor-Attacke*, als auch beim direkten Senden unerwarteter Pakete an ein System kann der Ressourcenverbrauch für den Transport dieser Pakete die Verfügbarkeit des zu schützenden Bereichs gefährden oder zumindest die Dienstgüte senken.

Ein *Stateful Packet Filter* verwendet daher ein *Connection Tracking Module*, um die Zustandsübergänge der TCP-Protokollinstanzen in den Endpunkten nachzuvollziehen. TCP-Folgepakete, die Nutzdaten tragen, werden von so einem Firewall-Element nur dann durchgelassen, wenn die entsprechende Verbindungsaufbau-Sequenz (engl. *Handshake*) beobachtet wurde. Dazu müssen in diesem Netzelement Zustandsinformationen über die offenen Verbindungen gehalten werden, was für zusätzlichen Speicherbedarf und erhöhten Aufwand beim Weiterleiten von Paketen sorgt, da für jedes Paket erst geprüft werden muss, ob es zu einer bekannten Verbindung gehört. Ferner muss dafür gesorgt werden, dass die Zustandsinformationen entfernt werden, falls die Verbindung über einen längeren Zeitraum inaktiv war, z. B. wegen eines nicht ordnungsgemäßen Verbindungsabbruchs durch eine „abgestürzte“ Protokollinstanz. Problematisch ist eine solche zustandsbehaftete Zugriffskontrolle im Zusammenhang mit dynamischer Verkehrslenkung, wie dies insbesondere in der IP-Schicht weit verbreitet ist. Werden IP-Pakete, die zu einer bestehenden TCP-Verbindung gehören, z. B. infolge eines Leitungs-Ausfalls plötzlich über ein anderes Firewall-Element geleitet, so sind dort evtl. die entsprechenden Zustandsinformationen nicht vorhanden. Die Pakete werden dann dort verworfen, was zum Abbruch der TCP-Verbindung führt, obwohl die Konnektivität auf der IP-Schicht dank des dynamischen Routings wieder hergestellt wurde.

Auch in höheren Protokollschichten kann, je nach Anwendungszweck, sowohl zustandslose als auch zustandsbehaftete Filterung zum Einsatz kommen. Beispielsweise kann zum Schutz vor anonymen Anrufen ein SIP-Proxy so konfiguriert werden, dass er *INVITE*-Nachrichten mit einem Status-Code *433 – Anonymity Disallowed* [52] zurückweist, wenn der *From*-Header auf *Anonymous* gesetzt ist. Dazu muss in dem SIP-Proxy keine Zustandsinformation gehalten werden. Die Entscheidung, ob eine Rufaufbau-Anforderung tatsächlich zum gerufenen Teilnehmer weitergeleitet wird, kann auch von sehr viel komplexeren Prüfungen abhängen, die das Halten von Zustandsinformationen erfordern. Beispielsweise haben so genannte *Touring Tests* [53] das Ziel, erwünschte menschliche Anrufer von automatisierten Systemen zu unterscheiden, die anrufen, um unerwünschte Werbe-Botschaften abzuspielen.

2.3.3.5 Einbringen der Zugriffskontrollregeln

Ein weiteres wichtiges Unterscheidungsmerkmal für Firewall-Systeme ist, wie und wann die Regeln, die die Sicherheitsrichtlinie implementieren, in das Netzelement eingebracht werden. Bei sehr vielen Firewalls, die lokale Netze gegen das Internet abschirmen und dazu überwiegend Nicht-Echtzeit-Datenverkehr filtern, sind die Firewall-Regeln der statischen Firewall-Konfigu-

ration zuzuordnen. Das Ändern des Regelsatzes wird von der Systemverwaltung in vergleichsweise großen Zeitabständen und unabhängig von den aktuell durch die Firewall fließenden Datenströmen durchgeführt [54]. Dazu kommen Werkzeuge zum Einsatz, die von einfachen Kommandozeilen-Befehlen des jeweiligen Betriebssystems bis hin zu recht komplexen Management-Systemen (z. B. [55]) reichen. Neuere Entwicklungen, sowohl im Bereich der Anwendungen, als auch im Bereich der Netzsicherheit machen es in Zukunft jedoch erforderlich, dass Firewall-Regelsätze von externen Steuer-Systemen dynamisch modifiziert werden können, z. B. in Abhängigkeit von der Sitzungssignalisierung oder von automatisch erkannten Angriffen auf den zu schützenden Bereich. Architekturen und Protokolle, die zur Signalisierung dieser Regeländerungen verwendet werden können, werden in den folgenden Kapiteln dieser Arbeit ausführlich untersucht und bewertet.

2.3.4 Mehrstufige Firewall-Systeme

Um die jeweiligen Vorteile von Paketfiltern und Proxies zu vereinigen, werden Firewall-Systeme oft als Kombination mehrerer Firewall-Elemente aufgebaut. Abbildung 2.18 zeigt eine mögliche Anordnung zur sicheren Anbindung eines lokalen Netzes (z. B. einer Firma) an das Internet. Die feingranulare und detaillierte Filterung des grenzüberschreitenden Verkehrs erfolgt in diesem Szenario auf der Anwendungsschicht. Dazu wird für jedes erwünschte Anwendungsschicht-Protokoll ein Proxy benötigt. Diese können (wie im Bild dargestellt) in jeweils einem eigenen Netzelement untergebracht werden; aus Gründen der Ressourcen-Effizienz können aber auch mehrere Proxy-Prozesse auf einem Netzelement installiert werden.

Aufgrund ihrer vergleichsweise komplexen Implementierung und der i. d. R. großen Menge gehaltener Zustandsinformationen können solche Proxies leicht selbst Ziel von Angriffen werden [54] – sowohl solche, die die Verfügbarkeit der Systeme gefährden (Denial-of-Service Attack), als auch solche, die sie kompromittieren können. Ein vorgeschalteter Paketfilter (Nr. 1) soll davor einen gewissen Schutz bilden, indem er schon auf Paket-Ebene all die Protokolle blockiert, die die Bereichsgrenze überhaupt nicht überqueren sollen, sofern diese an Parametern der IP- und Transportschicht (z. B. Portnummern) identifiziert werden können. Ferner können hier

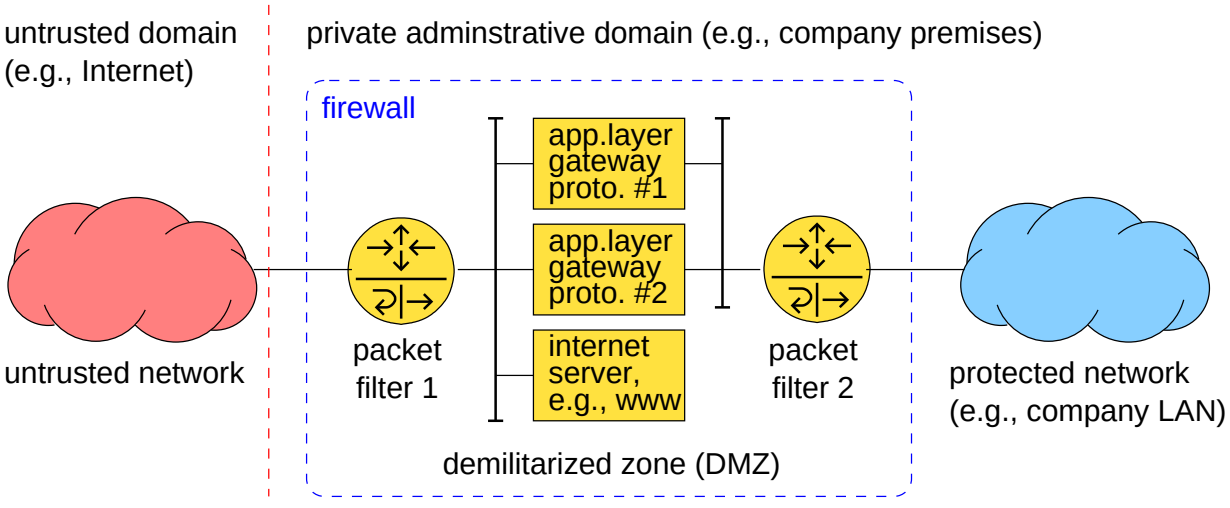


Abbildung 2.18: Mehrstufiges Firewall-System mit „demilitarized zone“ (DMZ)

schon in einem gewissen Umfang IP-Pakete mit illegalen IP-Optionen, gefälschten Absenderadressen (sog. *Spoofing*) und DoS-Attacken durch Paketfluten abgewehrt werden.

Die Kompromittierung eines Proxies kann dennoch nicht ausgeschlossen werden. Ein solcher Proxy wäre für Angreifer ein idealer Brückenkopf für Angriffe gegen die Rechner im zu schützenden Netz. Deshalb wird ein zweiter Paketfilter (Nr. 2) zwischen die Proxies und das zu schützende Netz platziert, um solche Angriffsmöglichkeiten so weit wie möglich zu beschränken. Oft wird für diesen Paketfilter bewusst die Hard- und Software anderer Hersteller ausgewählt, so dass eine evtl. bekannt gewordene Schwachstelle in der Implementierung eines Herstellers nur eine der beiden Schutz-Schichten verwundbar macht.

Server, die mit einer großen, ggf. anonymen Menge von Partnern im unsicheren Bereich kommunizieren müssen, z. B. der „öffentliche“ WWW-Server einer Firma, sind, ähnlich wie die Proxies, ebenfalls einem erhöhten Risiko der Kompromittierung ausgesetzt. Deshalb bietet es sich an, diese Systeme ebenfalls in dem Bereich zwischen den beiden Paketfiltern zu platzieren. Dieser Bereich, der sich sowohl bezüglich der Netztopologie, als auch bezüglich des Sicherheits-Niveaus „zwischen“ dem unsicheren und dem zu schützenden Bereich liegt, wird oft als *Demilitarized Zone* (DMZ) oder auch als *Screened Subnet* bezeichnet. Die in Abbildung 2.18 dargestellte Anordnung ist nur eine von vielen Möglichkeiten zur Anordnung mehrstufiger Firewall-Systeme; in [49] werden diverse Platzierungs-Alternativen vorgestellt und untersucht.

2.3.5 Spezifikation der Zugriffskontroll-Regeln für Firewalls

2.3.5.1 Grundprinzipien

Das zentrale Konzept von Firewalls ist – wie bei allen anderen regelbasierten Zugriffskontrollsystemen auch – die Regel (engl. *Policy Rule*), welche eine Menge von Aktionen an eine Menge von Bedingungen knüpft [RFC 3198]. Bei Paketfiltern werden durch die Bedingungen bestimmte Paket-Flüsse (engl. *Flows*) beschrieben, indem für jede Dimension des charakteristischen Fünf-Tupels (Quell-IP-Adresse, Ziel-IP-Adresse, Transportschichtprotokollkennung, Quell-Portnummer, Ziel-Portnummer) ein Wert bzw. Bereich angegeben wird. Weitere Bedingungen können dieser Menge hinzugefügt werden, z. B. bestimmte Kombinationen von *TCP Flags*. Die Menge der Aktionen beschreibt, wie mit einem Paket zu verfahren ist, z. B. Weiterleitung in Richtung des Ziels entsprechend der normalen Funktion eines IP-Routers, Verwerfen des Pakets mit oder ohne Benachrichtigung des Absenders (über eine ICMP-Nachricht [RFC 792]), ggf. zusätzliche Protokollierung des Ereignisses, etc. Eine Regel, die einen bestimmten Flow durch einen ansonsten weitgehend „geschlossenen“ Paketfilter (d. h. Whitelist-Ansatz, s. u.) passieren lässt, wird im Englischen auch als *Pinhole* bezeichnet [RFC 3303].

Viele Firewalls werden mit statischen Regelsätzen betrieben, d. h. das Ändern des Regelsatzes wird von der Systemverwaltung in vergleichsweise großen Zeitabständen und unabhängig von den aktuell durch die Firewall fließenden Datenströmen durchgeführt. Handelt es sich dem eingesetzten Firewall-Element um einen Paketfilter, beruht die Filter-Entscheidung, ob ein Paket weitergeleitet oder zurückgewiesen bzw. verworfen werden soll, somit ausschließlich auf den in den IP-Paketen enthaltenen Parametern, den quasi-statischen Regeln im Paketfilter, sowie im Falle von zustandsbehafteter Filterung auf Zustandsinformationen, die dynamisch im

Paketfilter erzeugt und gehalten werden. Diese Zustandsinformationen beziehen sich i. d. R. ausschließlich auf den Kontext einer TCP-Verbindung bzw. auf einen UDP-Flow, eine Korrelation verschiedener TCP-Verbindungen untereinander findet normalerweise nicht statt. Dies ist für eine Zugriffskontrolle bei den "klassischen" Internet-Protokollen der TCP/IP-Protokollsuite (z. B. HTTP, SMTP) auch nicht erforderlich.

2.3.5.2 Prozess zur Regel-Spezifikation

Die Regel-Liste für einen Firewall wird von der Systemverwaltung häufig in einem dreistufigen Prozess erstellt [54]:

In einem ersten Schritt erfolgt die Spezifikation einer vergleichsweise abstrakten Sicherheitsrichtline, unter Berücksichtigung der definierten Sicherheits-Domänen, sowie von Hostnamen, Benutzern, Nutzdatenflüssen, etc. Ziel einer solchen Richtlinie ist es, die Verwendung bestimmter (Anwendungsschicht-)Dienste zwischen Protokollinstanzen in verschiedenen Bereichen des Netzes zu reglementieren. Je nach gefordertem Schutzniveau kann dabei entweder das restriktivere *Whitelist*-Prinzip verwendet werden, bei dem die Verwendung sämtlicher Dienste bis auf explizit angegebene Ausnahmen unterbunden wird, oder das weniger restriktive *Blacklist*-Prinzip, bei dem alle Dienste zunächst erlaubt sind, und die zu unterbindenden Ausnahmen einzeln angegeben werden. Bei Diensten, die nach dem Client-Server-Prinzip erbracht werden, kann dabei auch berücksichtigt werden, welche Instanz die Rolle des Clients bzw. Servers übernimmt. Ein Beispiel für eine Zugriffskontrollregel auf diesem Abstraktionsniveau wäre, allen Arbeitsplatz-Rechnern in einem zu schützenden LAN zu erlauben, als Client HTTP-Verbindungen zu Servern im (als unsicher angenommenen) Internet aufzubauen, um den Benutzern Zugriff auf das World Wide Web zu erlauben.

In einem zweiten Schritt wird diese Sicherheitsrichtline verfeinert und an den Typ des verwendeten Firewall-Elements angepasst. Hierbei müssen alle Schichten des Protokollstapels sowie die Adressen der Protokollinstanzen berücksichtigt werden. Wurde beispielsweise im ersten Schritt spezifiziert, dass ein ausschließlich unidirektionaler Nutzdatenfluss zwischen zwei Protokollinstanzen in verschiedenen Bereichen zu erlauben ist, so muss u. a. das Transportschichtprotokoll betrachtet werden. Wird hierfür ein Protokoll mit Mechanismen zur Fehlersicherung (z. B. TCP) verwendet, wird es aufgrund der dazu benötigten Quittierungen (engl. *Acknowledgment*) in den darunterliegenden Protokollschichten zu einem bidirektionalen Nachrichtenfluss kommen. Soll zur Durchsetzung der Sicherheitsrichtlinie ein Paketfilter eingesetzt werden, so muss dementsprechend eine Regel vorgesehen werden, die diese IP-Pakete in Gegenrichtung akzeptiert.

Im dritten Schritt wird der Regelsatz schließlich in einer Syntax formuliert, die in so in das zu verwendende Firewall-Element eingebracht werden kann. Diese Syntax ist von der Implementierung des Firewall-Elements abhängig (i. d. R. herstellerspezifisch).

2.3.5.3 Einfluss von Adressabbildungen

Im zweiten Schritt des oben beschriebenen Prozesses ist eine detaillierte Kenntnis der für Nutzer, Prozesse, Hosts, etc. vergebenen Bezeichner und Adressen, sowie der Adressabbildungen zwischen den einzelnen Protokollen notwendig. Bei der Konfiguration eines Paketfilters muss bekannt sein, wie die im ersten Schritt verwendeten Adress- und Protokollkennungen der Anwendungsschicht auf entsprechende Kennungen der Netzwerk- und Transportschicht abgebildet werden, da ein Paketfilter nur die Informationen dieser unteren Schichten für die Filterentscheidung berücksichtigt. Als Beispiel für Adressabbildungen über mehrere Protokollschichten hinweg ist in Abbildung 2.19 schematisch dargestellt, wie ein Client des HTTP-Protokolls (d. h. ein „WWW-Browser“ und das zugrundeliegende Betriebssystem) aus einer Anwendungsschicht-URI (Uniform Ressource Identifier) letztendlich die MAC-Adresse für die Kommunikation über Ethernet ermittelt, und welche Adress-Informationen in den Protokoll-Köpfen der einzelnen Protokollschichten übertragen werden.

In [56] werden verschiedene Verfahren zur Adressabbildung klassifiziert: *statische* Verfahren verwenden Algorithmen (z. B. Abschneiden oder Aneinanderhängen von Adresskomponenten) oder sehr langlebige Zuordnungs-Tabellen (z. B. *Well-Known Port Numbers*, s. u.). *Dynamische* Verfahren hingegen verwenden Abfragen an Verzeichnisdienste, die zentralisiert (z. B. SUNRPC „portmapper“) oder verteilt (z. B. *Domain Name System*, DNS) implementiert sein können. Alternativ können Anfragen auch im Rundsende-Verfahren (engl. *Broadcast*) an alle möglichen Informationensquellen gesendet werden (z. B. *Address Resolution Protocol*, ARP).

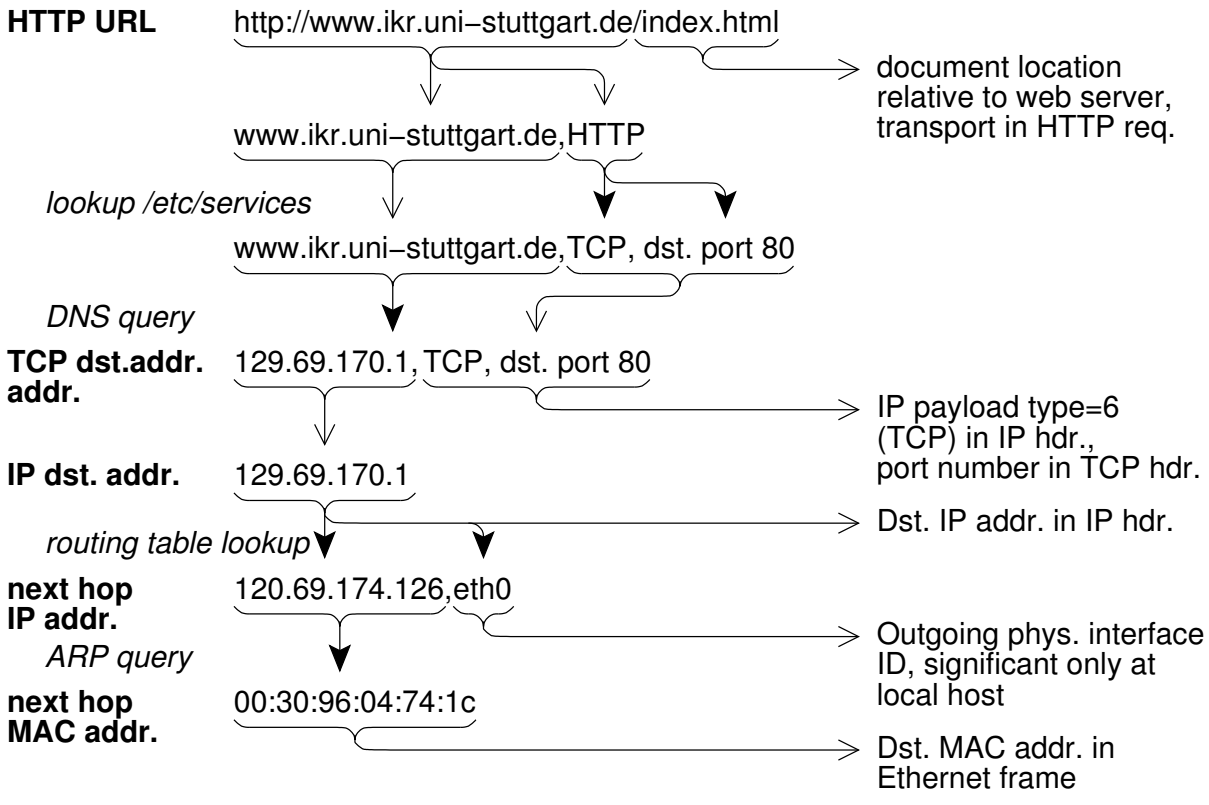


Abbildung 2.19: Adressumsetzungen im TCP/IP-Protokollstapel (Beispiel: HTTP)

2.3.5.4 Berücksichtigung statischer Adressabbildungen

Erfolgt die Abbildung zwischen zwei Adress-Familien in einem statischen Verfahren, so können Filter-Bedingungen, die bezüglich Adressen der Urmenge formuliert sind, statisch auf Bedingungen bezüglich Adressen der Bildmenge transformiert werden. Dies gelingt jedoch nur, wenn die Adressabbildung keine $m:1$-Abbildungen durchführt, bei der verschiedene Adressen der Urmenge, die von widersprüchlichen Regeln erfasst werden, auf eine gemeinsame Adresse in der Bildmenge abgebildet werden.

Eine statische Adressabbildung, deren Eigenschaften bei der Konfiguration von Firewalls – insbesondere von Paketfiltern – sehr oft ausgenutzt werden, ist das so genannte *Well-Known Port Numbers*-Konzept. Dieses ordnet sehr vielen Anwendungsschichtprotokollen der TCP/IP-Protokollfamilie ein Transportschichtprotokoll und eine (bei Client/Server-Protokollen serverseitige) Portnummer zu. Diese Zuordnung wird von der *Internet Assigned Numbers Authority* (IANA) verwaltet [57]. Eine Kopie dieser Datei (bzw. eine auf die tatsächlich genutzten Dienste verkürzte Version davon) wird von den Betriebssystemen der am Internet angeschlossenen Hosts für die oben beschriebene Abbildung verwendet; auf UNIX-Systemen ist sie i. d. R. unter dem Namen `/etc/services` im Dateisystem gespeichert. Die Aktualisierung dieser Datei erfolgt bei Bedarf durch den Systemadministrator. Ein Mechanismus zur Benachrichtigung oder automatischen Aktualisierung bei Änderungen ist nicht etabliert. Dies ist vergleichsweise unproblematisch, da i. d. R. nur neue Einträge in die Tabelle hinzugefügt werden.

Die oben als Beispiel genannte Regel, die den Rechnern im zu schützenden Bereich den Zugriff auf HTTP-Server im nicht vertrauenswürdigen Bereich erlaubt, kann mit diesem Wissen als Regeln für einen Paketfilter formuliert werden. So müssen IP-Pakete vom zu schützenden Bereich zum nicht vertrauenswürdigen Bereich erlaubt werden, sofern diese TCP-Segmente tragen, die an den Ziel-Port 80 adressiert sind. Ferner müssen die entsprechenden Antwortpakete (d. h. Quell-Port 80) in Gegenrichtung erlaubt werden. Unter Berücksichtigung der TCP Flags oder mit Hilfe zustandsbehafteter Filterung (siehe Abschnitt 2.3.3.4) kann die Menge der erlaubten Pakete weiter eingeschränkt und somit die Schutzhöhe vergrößert werden.

Problematisch bei der Konfiguration von Paketfiltern unter Berücksichtigung der Well-Known Port Numbers ist, dass es sich bei dieser Tabelle um eine von einer zentralen Registrierungsstelle gemachte Empfehlung handelt – es ist weder technisch schwierig, noch illegal, einen anderen Dienst als den offiziell registrierten auf einer bestimmten Portnummer zu betreiben. Insbesondere Angreifer, die Endsysteme auf beiden Seiten eines solchen Paketfilters unter ihrer administrativen Kontrolle haben, können ein vom Firewall-Administrator verbotenes Anwendungsschicht-Protokoll durch den Paketfilter hindurch nutzen, indem sie die entsprechende (serverseitige) Protokollinstanz an die Well-Known Port Number eines anderen Dienstes binden, die vom Paketfilter erlaubt wird.

2.3.5.5 Berücksichtigung dynamischer Adressabbildungen

Die Abbildung zwischen verschiedenen Adress-Familien kann auch durch die Abfrage eines externen Verzeichnisdienstes geschehen. Ein typisches Beispiel ist die Abbildung von Hostnamen auf IP-Adressen mit Hilfe des Domain Name System (DNS) [RFC 1034, RFC 1035],

einer hierarchischen, verteilten Namens-Datenbank. In diesem Szenario ist eine einfache, statische Transformation von Filter-Bedingungen im allgemeinen Fall nicht möglich. Änderungen an den Adresszuordnungen im DNS würden – zumindest theoretisch – auch eine Anpassung von Paketfilterregeln erforderlich machen.

Prinzipiell wäre es möglich, dass das Firewall-Element ebenfalls Abfragen an den Namensdienst sendet, um die dynamischen Adressabbildungen nachzuvollziehen. Im oben angegebenen Beispiel könnte dies bedeuten, einen DNS-Client in den Paketfilter zu integrieren. Dieses Vorgehen hat jedoch eine Reihe von Nachteilen: Ein Angreifer, der den bzw. Teile des Namensdienstes unter seiner Kontrolle hat, könnte dafür sorgen, dass der Namensdienst unterschiedliche Antworten liefert, je nachdem, ob er von einem „normalen“ Client oder der Firewall abgefragt wird. Somit könnten Firewall-Regeln unterlaufen werden. Desweiteren würde dieses Vorgehen einen Hauptvorteil der Paketfilter relativieren: die einfache Implementierbarkeit und den vergleichsweise ressourceneffizienten Betrieb. Ferner würden die DNS-Abfragen durch die Firewall für zusätzliche Latenzen sorgen und ein Ausfall des Namensdienstes würde u. U. die Funktion der Firewall beeinträchtigen. Bei Internet-Firewalls, die als Paketfilter implementiert sind, ist es aufgrund dieser Nachteile nicht üblich, solche Abfragen externer Namensdienste zu integrieren. Falls tatsächlich eine Filterung auf Basis individueller Hostnamen stattfinden soll, wird eher zu Anwendungsschicht-Proxies gegriffen, da deren Zugriffskontrolllisten Hostnamen direkt unterstützen.

Eine Transformation der abstrakten Richtlinien in einen statischen Paketfilter-Regelsatz kann jedoch in bestimmten Szenarien trotz Verwendung des DNS möglich sein. In der Praxis soll ein Paketfilter häufig dazu verwendet werden, ein privates Netz gegen ein öffentliches Netz (z. B. das Internet) zu schützen. Hierbei sind i. d. R. die im lokalen Netz verwendeten Adressen – sowohl Hostnamen als auch IP-Adressen – bekannt und die administrative Kontrolle über den entsprechenden Teil des DNS-Namensraumes liegt in der selben Hand wie die Kontrolle über die Firewall. Dies bedeutet, dass die Abbildung der lokalen Adressen trotz Einsatz des DNS als quasi-statisch angenommen werden kann. Falls die abstrakte Sicherheitsrichtlinie nicht zwischen einzelnen Hosts im Internet unterscheidet, so kann diese Menge „alle Hosts im Internet“ formuliert werden als „alle Hosts, deren IP-Adresse nicht in dem bekannten, lokalen IP-Adressraum liegt“. Weiter vereinfacht wird das Szenario, wenn die abstrakte Sicherheitsrichtlinie nur zwischen „internen“ Systemen (d. h. Systemen innerhalb des zu schützenden Netzes mit bekanntem IP-Adressraum) und „äußeren“ Systemen unterschiedet. Dann ist es ausreichend, mit dem Paketfilter zu prüfen, ob eine Quell- bzw. Ziel-IP-Adresse innerhalb des IP-Adressraumes des zu schützenden, internen Netzes liegt. Dass solche Szenarien, bei denen die durch das DNS prinzipiell mögliche Dynamik bei der Adressabbildung keinen Einfluss auf die Paketfilterregeln hat, in der Praxis recht häufig anzutreffen sind, hat die weite Verbreitung der vergleichsweise einfach zu implementierenden Paketfilter erst ermöglicht.

2.4 Zusammenfassung

In diesem Kapitel wurden Grundkonzepte der SIP-basierten IP-Telefonie, sowie der Netzsicherheit und Firewalls zunächst unabhängig voneinander dargestellt. Im Folgenden soll darauf eingegangen werden, wie Firewalls die Sicherheit der IP-Telefonie erhöhen können, aber auch welche Probleme dabei auftreten können und welche Lösungsansätze existieren.

3 Schutz von IP-Telefonie durch Zugriffskontrolle am Netzübergang

In diesem Kapitel sollen Bedrohungsszenarien für VoIP aufgezeigt werden und untersucht werden, wie Firewalls zum Erreichen bestimmter Schutzziele beitragen können.

3.1 Bedrohungsszenarien für IP-Telefonie

3.1.1 Allgemeine Gefährdungen für VoIP

Ausgehend von den in Abschnitt 2.2.1 eingeführten grundsätzlichen Sicherheitsanforderungen analysieren eine Studie [58] des deutschen *Bundesamt für Sicherheit in der Informationstechnik* (BSI) und ein ähnliches Dokument [59] des US-amerikanischen *National Institute of Standards and Technology* (NIST) Bedrohungsszenarien für VoIP-Lösungen. Sie konzentrieren sich auf Szenarien, bei denen eine klassische private Nebenstellenanlage durch VoIP in einem privaten, Ethernet- bzw. WLAN-basierten lokalen Netz (LAN) ersetzt wird.

Eine wichtige Rolle spielen Nachrichten, die in das für VoIP genutzte LAN eingeschleust werden, da sie die Sicherheit auf mehrere Weisen gefährden können: Die Verfügbarkeit des VoIP-Systems kann durch DoS-Attacken bedroht werden. Dabei werden Komponenten mit einer Flut von Nachrichten überlastet, die nicht von legitimen Nachrichten unterschieden werden können und deren Bearbeitung jeweils deutlich mehr Ressourcen in Anspruch nimmt als die Erzeugung beim Angreifer. Oft enthalten sie Absenderadressen, die nicht der Adresse des Angreifers entsprechen, um die Rückverfolgung zu erschweren [11]. Die Vertraulichkeit und Integrität/Authentizität der Kommunikationsinhalte kann mittelbar gefährdet werden, z. B. durch Zugriff auf unzureichend geschützte Konfigurations-Schnittstellen (Kompromittierung der funktionellen Integrität der Protokollinstanzen) oder durch Manipulation von Adressabbildungen oder Verkehrslenkung. Dies kann auf praktisch allen Schichten des Protokollstapels geschehen, z. B. durch Einschleusen manipulierter ARP-, OSPF-, DNS- oder SIP *REGISTER*-Nachrichten.

Ein weiterer Aspekt, der ausführlich diskutiert wird, ist die – verglichen mit analogen bzw. ISDN-basierten Endgeräten – sehr hohe Komplexität der Software auf den VoIP-Endsystemen. Dies erhöht die Wahrscheinlichkeit, dass eine Protokollinstanz Implementierungs-Schwachstellen enthält, die von einem Angreifer durch Senden entsprechender Nachrichten ausgenutzt werden können, z. B. mangelnde Validierung der Eingabedaten oder so genannte *Buffer Overflows* [60]. Ein kompromittiertes VoIP-Endsystem kann diverse Schutzziele gefährden, z. B.

indem Gesprächsinhalte aufgezeichnet werden oder ferngesteuert teure Mehrwertdienste genutzt werden. Desweiteren kann es als Brückenkopf für weitere Angriffe gegen andere Instanzen verwendet werden, z. B. weil diese Ziele für den Angreifer nicht direkt erreichbar sind, oder um seine wahre Herkunft zu verschleiern.

Diesen Befunden entsprechend lautet eine zentrale Empfehlung beider Studien, innerhalb einer Firma die Sprachkommunikation von der sonstigen Datenkommunikation im LAN zu separieren, und zwar zumindest logisch (z. B. über getrennte IP-Adressräume [59, Seite 5] oder *Tagged VLANs* [58, Abb. 6.4], bei erhöhten Sicherheitsanforderungen auch physikalisch, d. h. über getrennte Leitungen [58, Abb. 6.6]. Der Zugriff auf diese separaten Bereiche, insbesondere auf die Konfigurations-Schnittstellen der IP-Telefone, soll mit Hilfe von Firewalls auf das Nötigste beschränkt werden. Beide Studien weisen darauf hin, dass die Filterung von VoIP-Protokollen mit den bei der Erstellung der Studien verfügbaren Firewall-Systemen nicht problemlos möglich ist, und skizzieren eine Teilmenge der in der hier vorliegenden Arbeit untersuchten Lösungsansätze.

3.1.2 Die SPIT-Problematik

Die auf SMTP [RFC 821] basierende *Electronic Mail* ist eine der ältesten und erfolgreichsten Kommunikationsformen im Internet. Ein bedeutsames Problem dieses Mediums ist der massenhafte Verstand unerwünschter Werbebotschaften, die als *Unsolicited Commercial Email* (UCE) oder als *Spam* bezeichnet werden [RFC 2635]. Verschiedenen Studien (z. B. [61, 62]) zufolge trugen sie im Jahr 2006 mit ca. 80% zum E-Mail-Gesamtaufkommen im Internet bei. Dies kann zum Absinken der Verfügbarkeit des Dienstes führen: durch Mail-Server, die durch die schiere Zahl der oft in Schüben gesendeten Nachrichten überlastet werden, durch Empfänger, die viel Zeit zum Aussortieren benötigen und dabei evtl. sogar versehentlich erwünschte Nachrichten übersehen, oder durch Fehlentscheidungen (*False Positives*) automatisierter Mail-Filtersysteme (z. B. [63]), die mittlerweile zur Standardausrüstung von E-Mail-Dienstanbietern gehören.

Es ist zu befürchten, dass auch jede Form von „öffentlichem“ VoIP, d. h. außerhalb kleiner Vertrauensdomänen wie z. B. firmeninterner Nebenstellenanlagen, mit steigender Verbreitung immer mehr von dieser Problematik betroffen sein wird. Für die Verbreitung unerwünschter (Werbe-)Botschaften über IP-Telefonie-Anwendungen wurde der Begriff *Spam over IP Telephony* (SPIT) geprägt. Die synchrone Natur der Telefonie verschärft die Probleme gegenüber der asynchronen Kommunikation per E-Mail mehrfach: Zum Zeitpunkt des Verbindungsaufbaus kann eine Filter-Entscheidung nur auf Metainformationen aus der Signalisierung (z. B. Absender- und Empfängeradressen) basieren. Die Alarmierung des gerufenen Teilnehmers stellt einen viel aufdringlicheren Eingriff in seine aktuelle Tätigkeit dar, als z. B. der Empfang einer E-Mail. Deshalb ist es auch nicht sinnvoll, die Verbindung automatisch zu beenden, selbst wenn diese durch eine nachträgliche Prüfung der eigentlichen Inhalte (z. B. mittels Spracherkennung und Listen SPIT-typischer Wörter) als SPIT erkannt wurde. Deshalb können die bekannten Lösungen für E-Mail-Spam nicht ohne Weiteres auf SPIT übertragen werden. Dennoch existiert eine ganze Reihe von Vorschlägen zur Milderung der SPIT-Problematik [64, 65].

Viele dieser Vorschläge basieren auf einer Prüfung der Adresse des rufenden Teilnehmers durch zentralisierte oder teilnehmerspezifische Blacklists bzw. Whitelists oder mit Hilfe von Reputations-Systemen [66], die die Einschätzung bestimmter Nutzer bzgl. der Vertrauenswürdigkeit anderer Nutzer sammeln [67] und automatisch bewerten. Voraussetzung dafür sind authentische

Absenderadressen; dies soll i. d. R. mit Hilfe kryptographischer Verfahren und einer zentralen *Public Key Infrastructure* (PKI) bzw. einem dezentralen *Web of Trust* zur Schlüsselverwaltung und -verteilung sichergestellt werden. Problematisch ist dabei, wie mit neuen Teilnehmern umgegangen werden soll: Ist die Registrierung einer neuen SIP-Adresse (z. B. bei einer PKI) zu aufwändig, kann dies die Nutzbarkeit und Akzeptanz des Systems beeinträchtigen. Ist der Aufwand hingegen sehr gering, können SPIT-Versender Eintragungen in Blacklists bzw. negative Bewertungen in Reputations-Systemen unterlaufen, indem sie für jeden SPIT-Anruf eine neue Absender-Adresse verwenden. Dies wird dazu führen, dass bisher unbekannten Teilnehmern bei ihrem erstmaligen Kontaktversuch ein gewisses Misstrauen entgegen gebracht wird [68].

Andere Vorschläge kommen auch ohne fälschungssichere Absenderadressen aus, z. B. indem der rufende Teilnehmer vor Verbindungsaufbau mit Hilfe eines *Micro-payment Systems* einen kleinen Geldbetrag an einen Treuhänder überweist, der zurückerstattet wird, falls der gerufene Teilnehmer bestätigt, dass es sich bei der Verbindung nicht um SPIT gehandelt hat. Mit Hilfe so genannter *Turing Tests* [53] sollen Menschen von Tonbandansagen unterschieden werden könnnen, z. B. indem vor dem Durchschalten der Verbindung an den gerufenen Teilnehmer eine sehr einfache Rechenaufgabe vorgelesen wird, deren Ergebnis über die Zifferntastatur des Telefons einzugeben ist.

Mit Ausnahme der Turing Tests beziehen sich alle genannten Mechanismen ausschließlich auf die Signalisierung, d. h. nachdem die Anti-SPIT-Systeme und der gerufene Teilnehmer die INVITE-Transaktion akzeptiert haben, können die Medienströme prinzipiell Ende-zu-Ende zwischen den beiden VoIP-Endgeräten fließen. Viele Implementierungen solcher Endsysteme warten auf vorhersehbaren UDP-Portnummern auf ankommende RTP-Ströme und spielen sie auch dann ab, wenn sie von einer IP-Adresse ausgehen, die nicht bei der SIP-Signalisierung als Gegenstelle aufgetreten ist [60]. In einigen Szenarien (z. B. Konferenz-Schaltungen) ist eine Beschränkung auf die „richtige(n)“ Gegenstelle(n) auch gar nicht ohne weiteres möglich, da dem Endsystem zu wenige Informationen übermittelt werden. Ein SPIT-Versender kann dies ausnutzen, um RTP-Ströme mit Werbebotschaften an geratene Zieladressen zu senden, in der Hoffnung, dass diese anstelle des legitimen Medienstroms einer bestehenden Verbindung abgespielt werden bzw. dass beide Ströme als Konferenz gemischt werden. Abhängig von der Netztopologie können Firewalls, die nicht zuvor per SIP signalisierte Medienströme blockieren, zur Lösung dieses Problems beitragen.

Ein alternativer Ansatz, der von den Interconnection-Strukturen der etablierten ISDN-Netze geprägt ist, sieht vor, die Signalisierung aller IP-Telefonie-Rufe über wenige zentrale SIP-Provider abzuwickeln, die mit ihren jeweiligen Nutzern juristische Verträge abgeschlossen haben. Diese beinhalten Nutzungsrichtlinien (z. B. Verbot des SPIT-Versands) für die Nutzer und verpflichten den Anbieter, auf evtl. dennoch auftretende Beschwerden angemessen zu reagieren. Mehrere solche SIP-Anbieter können sich zu einem Konsortium (sog. *Circle of Trust*) zusammenschließen, dessen Interconnection-Verträge die SPIT-Freiheit der vermittelten Rufe garantieren, sowie ggf. Standards für die Authentisierung von Teilnehmern, die Vorfallsbehandlung, evtl. auch Schadensersatzansprüche bei dennoch auftretendem SPIT festlegen [69]. Eine solche Struktur kann prinzipiell als kryptographisch gesichertes, virtuelles *Overlay-Netz* über dem ungesicherten Internet etabliert werden; alternativ können die dem Konsortium angehörigen Anbieter auch eigene, IP-basierte, aber vom Internet getrennte Netze aufbauen (siehe Abschnitt 3.3.2), um so z. B. auch gleichzeitig eine garantierte Dienstgüte anbieten zu können.

3.1.3 Besondere Anforderungen in öffentlichen IP-Telefonie-Netzen

Falls IP-Telefonie von einem öffentlichen Netzbetreiber als kommerzieller Dienst – ähnlich zur Telefonie über die konventionellen PSTN/ISDN-Netze – angeboten werden soll, ergibt sich daraus eine Reihe von weiteren Anforderungen mit Bezug zur Sicherheit, teilweise aus kommerziellen Interessen dieses Anbieters, teilweise aus gesetzlichen bzw. regulatorischen Anforderungen des jeweiligen Landes.

Ein wichtiges Anliegen vieler kommerzieller Netzbetreiber ist, Informationen, aus denen Rückschlüsse auf die Topologie und Leistungsfähigkeit des eigenen Netzes, die Zahl der Teilnehmer (Kunden), etc. gezogen werden könnten, sowohl an der Teilnehmerschnittstelle, als auch am Netzübergang zu Mitbewerbern herauszufiltern (engl. *Topology Hiding*). Eng damit verwandt ist die evtl. erwünschte Möglichkeit, den Teilnehmern auf Wunsch *anonyme Anrufe* zu ermöglichen [RFC 3323]. Diese Ziele können primär durch Modifikation der Signalisiernachrichten erreicht werden, z. B. durch Entfernen von *Via*-Zeilen oder indem die Absenderadresse auf *„Anonymous“* gesetzt wird. Es kann aber auch notwendig sein, zusätzlich die IP-Adressen in den RTP-Medienströmen mit Hilfe von *Network Address and Port Translation* (NAPT) oder RTP-Proxies umzusetzen, da auch hieraus prinzipiell Rückschlüsse auf die netzinterne IP-Adressvergabe und -Zuordnung gezogen werden können.

In vielen Ländern müssen Anbieter von Telekommunikationsdiensten *Abhörschnittstellen* zur Verfügung stellen, über die staatliche Stellen die Kommunikationsdatensätze oder die Inhalte der Kommunikation bestimmter verdächtiger Personen abfragen können [TS101671]. Falls der Zugriff auf die Kommunikationsinhalte nicht auf IP-Paketebene unmittelbar an der Teilnehmerschnittstelle geschehen soll, müssen die Medienströme trotz ggf. vorhandenem dynamischen Routings „im Netz“ gefunden werden. Alternativ können Zwangspunkte im RTP-Medienpfad geschaffen werden (z. B. RTP-Proxies). Die dabei auftretenden Probleme sind teilweise mit denen der Firewalls vergleichbar, hinzu kommt allerdings, dass die Abhörmaßnahme für die betroffenen Teilnehmer nicht erkennbar sein darf.

Falls Entgelte für den Telefonie-Dienst nicht über eine monatliche Grundgebühr (engl. *Flat Rate*) erhoben werden sollen, sondern abhängig von der Anzahl oder der Dauer der Gespräche, so muss verhindert werden, dass die Teilnehmer Signalisierung und Medienströme an den für die Erfassung von Kommunikationsdatensätzen verwendeten SIP-Servern vorbei, direkt zwischen den Endsystemen austauschen. Dies kann mit Hilfe von Firewalls erreicht werden, die nur solche Medienströme passieren lassen, die vorher über die „offiziellen“ SIP-Server signalisiert wurden. Ob solche nutzungsabhängigen Tarife angesichts der weiten Verbreitung von Flat Rates oder kostenlosen (i. d. R. werbefinanzierten) Angeboten eine nennenswerte Akzeptanz finden werden, bleibt abzuwarten; evtl. kann die Attraktivität durch Zusatzleistungen wie bessere Dienstgüte (QoS) oder besseren SPIT-Schutz erhöht werden.

Grundsätzlich gilt in öffentlichen Netzen, dass die Betreiber bei der Planung von Sicherheitsfunktionen keinerlei Annahmen über die Eigenschaften, die Vertrauenswürdigkeit oder den Kooperationswillen der Protokollinstanzen beim Teilnehmer machen können. Dementsprechend kommt Mechanismen zur Zugriffskontrolle nicht im Endsystem, sondern in Transitsystemen an der Teilnehmerschnittstelle bzw. am Netzübergang, d. h. Firewalls und ähnliche Systemen, eine große Bedeutung zu.

3.2 Horizontaler Freiheitsgrad der Allokation von Sicherheitsfunktionen

Der so genannte „Horizontale Freiheitsgrad der Allokation von Sicherheitsfunktionen“ bezeichnet die verschiedenen Möglichkeiten beim Entwurf eines Kommunikationssystems bezüglich der räumlichen Anordnung von Schutzmechanismen auf dem Pfad zwischen Nachrichten-Quelle und -Senke [70].

So können beispielsweise die Schutzziele *Vertraulichkeit* und *Integrität* durch kryptographische Verfahren erreicht werden, indem die Nachrichten von einer Protokollinstanz verschlüsselt bzw. signiert werden, während die andere Instanz sie entschlüsselt bzw. die Signatur prüft. Die Sicherheits-Assoziation zwischen diesen beiden Protokollinstanzen kann Teil-Pfade bzw. Bereiche mit verschieden großen Ausdehnungen überspannen: Befinden sich die Sicherheitsfunktionen auf zwei benachbarten Netzelementen, die über einen physikalischen Link direkt miteinander verbunden sind, spricht man von einer Link-zu-Link-Funktion (Beispiel: WEP-Verschlüsselung im WLAN [71]). Überspannt die Sicherheits-Assoziation einen Teilabschnitt des Pfades, auf dem sich weitere Netzknoten mit Vermittlungsfunktion befinden, so wird dies als Knoten-zu-Knoten-Funktion (Beispiel: IPsec [RFC 4301] zwischen zwei zum Security-Gateway erweiterten Routern) bezeichnet. Eine Ende-zu-Ende-Funktion wird in den Endpunkten der Kommunikationsbeziehung implementiert und schützt somit die Nachrichten auf dem kompletten Pfad (Beispiel: TLS [RFC 4346]). Auch Mechanismen zur Zugriffskontrolle können sowohl in den Endsystemen, als auch in Transit-Netzelementen implementiert werden; letzteres wird im Umfeld IP-basierter Netze als Firewall bezeichnet.

Das Konzept der Firewalls steht in einem gewissen Widerspruch zu einem der wichtigsten Entwurfs-Prinzipien des Internets, dem so genannten „End-to-End Argument“ [72]. Dieses besagt, dass viele der in einem Kommunikationsnetz benötigten Funktionen besser in den höheren Schichten des Protokollstapels und in bzw. zwischen den Endsystemen als „im Netz“ (d. h. in bzw. zwischen den Transit-Knoten) erbracht werden können. Ausnahmen zu dieser Regel seien aber u. U. zur Steigerung der Leistungsfähigkeit sinnvoll. Das klassische Beispiel dafür ist der Schutz gegen Bitfehler beim Nachrichtentransport: Da solche Bitfehler nicht nur während der Übertragung auf Leitungen auftreten können, sondern auch in Folge defekter Netzknoten, ist eine abschnittsweise Sicherung in tieferen Schichten alleine nicht ausreichend. Eine Ende-zu-Ende-Sicherung (z. B. mit TCP) schützt hingegen gegen beide Fehlerarten und macht daher die Sicherung auf der tiefen Schicht aus funktionaler Sicht überflüssig. Um die Latenzen und den zusätzlichen Ressourcenverbrauch, die bei einer erneuten Ende-zu-Ende-Übertragung entstehen, zu vermeiden, kann eine zusätzliche, abschnittsweise Sicherung von Segmenten mit besonders hoher Fehlerwahrscheinlichkeit (z. B. im Funkzugangsnetz) dennoch sinnvoll sein.

Auch im Bereich der Sicherheit ist das „End-to-End Argument“ von Bedeutung. Zwei sehr wichtige Schutzziele, der Schutz von Vertraulichkeit und Integrität übertragener Nachrichten, lassen sich besonders wirkungsvoll mit Hilfe kryptographischer Maßnahmen in den oberen Protokollschichten zwischen den Endsystemen implementieren. Eine alternative Implementierung in den tieferen Schichten, z. B. durch abschnittsweise Verschlüsselung in der Sicherungsschicht zwischen den Routern, bietet im Vergleich eine geringere Angriffshöhe, da sich Anzahl der Netzelemente, in denen die Nachricht im Klartext vorliegt und denen deshalb vertraut werden muss, erhöht. Eine Verschlüsselung, die in den unteren Schichten zusätzlich zu einer in den oberen Schichten angewendet wird, trägt i. d. R. nicht zu dem aus Teilnehmer-Sicht so wichtigen

Ende-zu-Ende-Schutz bei; sie kann aber zur Erreichung anderer Schutzziele sinnvoll sein (z. B. Schutz eines drahtlosen Zugangsnetzes). Für Authentisierung und Zugriffskontrolle gilt, dass diese in den oberen Protokollschichten und im Endsystem besonders feingranular durchgeführt werden können, da nur dort die Semantik des Anwendungsprotokolls, sowie Kennungen wie z. B. Nutzerkennungen und Prozessbezeichner bekannt sind.

Ein weiteres prinzipielles Problem von Firewalls ist, dass Sicherheitsrichtlinien, die die Nutzung bestimmter Dienste einschränken sollen, zumindest von solchen Angreifern, die Systeme auf beiden Seiten der Firewalls unter ihrer Kontrolle haben, leicht umgangen werden können, sobald wenigstens ein Protokoll von der Firewall erlaubt wird. Im einfachsten Fall können – wie in Abschnitt 2.3.5.4 beschrieben – Paketfilter unterlaufen werden, indem Anwendungsschicht-Protokolle auf einer anderen als der offiziellen Well-Known Port Number betrieben werden. Aber auch kompliziertere Firewall-Implementierungen können überwunden werden, indem das gewünschte Protokoll in die Nachrichten des von der Firewall erlaubten Protokolls eingebettet wird. Da nahezu jedes Protokoll zumindest für die Übertragung eines einfachen Morse-Codes genutzt werden kann, ist ein solches „Tunneln“ durch die Firewall praktisch immer möglich, auch wenn dies zum Teil extrem ineffizient ist (z. B. Voice-over-DNS [73]).

Es gibt aber auch Gründe, warum es nicht ausreichend sein kann, Schutzmechanismen nur in den Endsystemen zu platzieren. Insbesondere bei größeren lokalen Netzen mit einer entsprechend großen Zahl von Endsystemen kann das Erstellen und Aufrechterhalten konsistenter Sicherheitsrichtlinien mühselig und fehleranfällig sein. Desweiteren kann es vorkommen, dass Schutzmechanismen im Endsystem durch uneinsichtige Nutzer oder Computer-Viren, Trojanische Pferde, etc. deaktiviert werden, oder dass ungeschützte Systeme (z. B. Laptops) in das Netz eingebracht werden. Es erscheint daher vernünftig, noch mindestens eine weitere Verteidigungslinie vor den Endsystemen zu haben, an der Zugriffskontroll-Richtlinien durchgesetzt werden. Während die bisher genannten Gründe für den Einsatz von Firewalls eher von praktischer Natur sind, gibt es auch grundsätzliche, architekturelle Argumente. So kann sich ein Endsystem nicht selbst gegen Denial-of-Service-Angriffe schützen, die versuchen, seine Internet-Anbindung zu unterbrechen, indem sie diese Leitung mit unnützen IP-Paketen fluten. Eine Zugriffskontrolle, die die angreifenden Pakete erst dann abweist, wenn sie die überlastete Leitung bereits überquert und verstopft haben, setzt zu spät an und ist daher wirkungslos. Ein Schutzmechanismus gegen solche Angriffe muss vielmehr vor dem Bitraten-Flaschenhals installiert werden, z. B. beim Internet Service Provider, am Rande des Kernnetzes.

Es existiert eine ganze Reihe wissenschaftlicher Veröffentlichungen, die sich mit der Abwehr von DoS-Attacken im Internet beschäftigen. Viele beschränken ihren Fokus auf den Schutz bestimmter Rechner, die nur mit einer relativ kleinen Anzahl anderer, bekannter Rechner kommunizieren (z. B. [74]). In anderen Beiträgen werden wesentliche Änderungen an der Internet-Architektur oder am IP-Protokollstapel vorgeschlagen, z. B. die Einführung neuer Felder im IP-Paketformat, neuer Signalisierprotokolle oder neuer zustandsbehafteter Netzelemente im Kernnetz. In [75] wird ein Überblick über mehrere solche Ansätze gegeben und ein weiterer vorgeschlagen, bei dem SIP-basierte Signalisierung dazu verwendet wird, im Netz verteilte Firewalls über neue Verbindungen bzw. Flows zu informieren.

Es erscheint zweifelhaft, ob solche weitreichenden Änderungen der Internet-Architektur, insbesondere eine Abkehr vom Paradigma des „zustandslosen Kernnetzes“ im Internet kurz- bis mittelfristig eingeführt werden [76]. Eine andere Situation könnte sich allerdings in Netzen er-

geben, die zwar die IP-Protokolle nutzen, aber nicht direkt mit dem Internet verbunden sind. Darunter fallen insbesondere so genannte IP-Telefonie-Plattformen. Bei diesen sind einerseits die Sicherheitsanforderungen höher sind als im Internet, andererseits können die für die Steuerung der Firewalls benötigten Informationen vergleichsweise einfach aus der sowieso vorhandenen, SIP-basierten Control-Plane abgeleitet werden. Auf die grundsätzlichen Unterschiede bzgl. der zugrundeliegenden Sicherheitsphilosophie und Netzarchitektur soll im Folgenden eingegangen werden.

3.3 Netzarchitekturen SIP-basierter Netze

Der in Abschnitt 2.1.5 gegebene Überblick über das Session Initiation Protocol konzentriert sich auf die Protokollmechanismen und -eigenschaften von SIP und seiner begleitenden Protokolle. Auf eine detaillierte Einordnung von SIP in eine *Netzarchitektur* wurde dort bewusst verzichtet – dies entspricht dem Credo der für die SIP-Standardisierung maßgeblich verantwortlichen *Internet Engineering Task Force* (IETF), die sich auf die Standardisierung von Protokollen konzentriert. Dabei soll insbesondere keine Position zu den für einen kommerziellen Netzbetrieb benötigten Geschäftsmodellen bezogen werden. Falls SIP nicht „nur“ dazu verwendet werden soll, eine private Nebenstellenanlage durch LAN-basiertes VoIP zu ersetzen, sondern das Ziel die Umstellung der öffentlichen Telefonnetze (PSTN) von TDM/SS7 auf RTP/SIP ist, müssen Lösungen gefunden werden, die über reine Protokollmechanismen weit hinausgehen. Eine zentrale Fragestellung dabei ist, wie Domänen zusammengeschaltet werden können, die verschiedener administrativer Kontrolle und Verantwortung unterliegen.

Im folgenden Abschnitt soll auf Unterschiede zwischen der „offenen“ Internet-Architektur und der „geschlossenen“ Architektur der etablierten Telefon-Netze eingegangen werden. Die Architekturkonzepte dieser „geschlossenen“ Netze lassen sich auch auf die SIP/IP-Protokollfamilie übertragen; Abschnitt 3.4 gibt daher einen Überblick über die 3GPP IMS- und ETSI TISPAN-Architekturen als eine mögliche Ausprägung dieser so genannten *SIP-Plattformen*. Dabei wird besonders auf Funktionen am Netzübergang eingegangen. Da bei Überlegungen zu Verzögerungs-Effekten in solchen Netzen die Zahl der durchquerten Transit-Domänen und somit die Zahl der zu überwindenden Netzübergänge auf dem Pfad zwischen den Teilnehmern eine Rolle spielt, werden in Abschnitt 3.5 Konzepte zur Netzzusammenschaltung vorgestellt, die derzeit entworfen und diskutiert werden.

3.3.1 Die „offene“, Internet-basierte Architektur

Den Basis-Standards zu SIP (neben [RFC 3261] ist in diesem Zusammenhang insbesondere [RFC 3263] besonders wichtig) liegen Annahmen über die grundsätzliche Ausgestaltung SIP-basierter Multimedia-Kommunikation zu Grunde, auch wenn diese Annahmen dort nicht explizit dokumentiert sind. Diese entsprechen weitgehend dem Modell, wie E-Mails im Internet weitergeleitet werden, und können wie folgt charakterisiert werden [77]:

Eine zentrale Annahme ist, dass alle SIP-Instanzen auf Netzelementen platziert werden, die mit dem „offenen“ Internet verbunden sind, d. h. die auf der IP-Schicht uneingeschränkt Pakete Ende-zu-Ende übertragen können. Fragen, von wem und wie diese IP-Konnektivität zur Verfügung

gestellt wird, wie für sie ggf. Nutzungsentgelte erhoben und abgerechnet werden, wie sie gegen Angriffe auf den unteren Schichten des Protokollstapels geschützt wird, usw., sollen vollkommen unabhängig von SIP gelöst werden. Es erfolgt in diesem Szenario auch keine grundsätzliche Unterscheidung zwischen Nutzern und Betreibern der Dienste, die sich direkt auf die Netz- und Protokoll-Architektur abbildet. Bei den etablierten ISDN-Netzen werden für die Signalisierung auf der Teilnehmerschnittstelle (engl. *User-to-Network Interface*, UNI) andere Protokolle (z. B. „D-Kanal-Protokoll“ [Q.931]) verwendet als für die Zwischenamtssignalisierung (engl. *Network-to-Network Interface*, NNI) zum Einsatz kommen (z. B. SS7/ISUP [Q.761]). IP-basierte Netze sollen hingegen SIP auf allen Schnittstellen verwenden. Da eine Protokollumsetzung am Netz-Rand (z. B. in der OVSt) somit nicht zwingend erforderlich ist, muss bei solchen Netzen damit gerechnet werden, dass auch im Kernnetz Signalisiernachrichten von nicht vertrauenswürdigen Quellen auftreten können, die syntaktisch oder (z. B. aufgrund umgangener Berechtigungsprüfungen) semantisch nicht korrekt sind.

In dem so genannten *SIP Trapezoid* nach [RFC 3263] (siehe Abbildung 2.7) kommt den beiden Proxies eine vergleichsweise geringe Bedeutung zu; sie sind nur für die Wegesuche und Weiterleitung der initialen *INVITE*-Nachricht zuständig. Alle anderen für die Multimedia-Kommunikation benötigten Funktionen, u. a. der Medientransport, werden in den Endpunkten bzw. durch direkte Kommunikation über IP zwischen diesen Endpunkten erbracht. Dazu gehört z. B. auch die Prüfung der Identität eines Anrufers, z. B. mit Hilfe von digitalen Signaturen, das Zurückweisen unerwünschter Anrufe und ggf. die Reservierung von QoS-Ressourcen.

Das Routing der *INVITE*-Nachricht basiert auf der der Domänen-Komponente der Ziel-URI. Daraus wird senderseitig mit Hilfe von DNS-Abfragen nach [RFC 3263] die Liste möglicher Inbound Proxies der Ziel-Domäne bestimmt. Das öffentliche DNS im Internet liefert, unabhängig von der Identität des Anfragenden, immer die selben Antworten. Somit kontaktieren alle Quell-Domänen die selben Eingangs-Punkte eine bestimmten Ziel-Domäne. Diese müssen daher öffentlich erreichbar sein, um universale Konnektivität sicherstellen zu können. Eine Weiterleitung der *INVITE*-Nachricht über mehrere Proxies (z. B. in Transit-Domänen) wird von diesen Standards nicht vorgesehen. Dies bedeutet, dass es aus Sicht der SIP-Instanzen eine logische Vollvermaschung aller Domänen gibt. Die Abbildung auf die nicht vollvermaschte physikalische Netztopologie erfolgt mit den Routing-Verfahren der IP-Schicht.

In [77] wird aufgezeigt, das dieses auf [RFC 3263] basierende Verfahren zur domänenübergreifenden Weiterleitung von SIP-basierten Rufen nur schwer mit den für ISDN/PSTN-Netzbetreiber üblichen Geschäftsmodellen vereinbar ist. Die Furcht vieler (potenzieller) Netzbetreiber, dass bei einer solch „offenen“ Interconnection-Struktur die Probleme durch unerwünschte (Werbe-)Anrufe überhand nehmen könnten, wird dort als weiterer Grund benannt, warum Anfang 2007 nur ein sehr kleiner Anteil der SIP-Nutzer tatsächlich aus dem offenen Internet erreicht werden könne. Die Mehrzahl dieser Teilnehmer befinde sich hingegen in SIP/IP-„Inseln“, die nur mit Umweg über das ISDN/PSTN oder ggf. manuell eingerichtete, private Netzübergänge verbunden sind.

Auch die Zahl der registrierten ENUM-Domänen liegt hinter den Erwartungen zurück. So waren in Deutschland im März 2007, über ein Jahr nach Abschluss des ENUM-Pilotbetriebs und Aufnahme des Wirkbetriebes erst ca. 7300 Domänen registriert, seither ist die Zahl sogar wieder leicht rückläufig (Juni 2007: 6600) [78]. Für den Nordamerikanischen Nummernraum ist zum selben Zeitpunkt noch kein User-ENUM im Wirkbetrieb.

3.3.2 IP-Telefonie-Plattformen

Alternativ zu dem im vorangegangenen Abschnitt vorgestellten Konzept kann IP-Telefonie auch in „geschlossenen“ *IP-Telefonie-Plattformen* durchgeführt werden. Darunter sollen im Folgenden Netze verstanden werden, die zwar die Protokolle der TCP/IP-Protokollfamilie und insbesondere auch SIP verwenden, deren Netzarchitektur sich aber mehr oder weniger deutlich vom „offenen“ Internet unterscheidet. Im Umfeld der ITU-T werden solche Überlegungen unter dem Stichwort *Next Generation Network* (NGN) [Y.2011] zusammengefasst. Verschiedene Ausprägungen solcher Plattformen verwenden Strukturierungsprinzipien, die vom PSTN bekannt, in der Internet-Welt bisher aber eher unüblich sind. Dazu gehören z. B. die Unterscheidung zwischen *Control Plane* und *Data Plane* (siehe Abbildung 3.1) oder die Unterscheidung zwischen Teilnehmer- und Zwischenamtssignalisierung. Im Gegensatz zum „offenen“ Internet sind solche Netze sehr viel stärker in Domänen mit ggf. unterschiedlichen Sicherheitsrichtlinien eingeteilt. Diese werden insbesondere an den Domänengrenzen durch Protokollfilterung mit Hilfe von Firewalls bzw. Gateways durchgesetzt.

Beim Entwurf solcher Plattformen kann und muss bestimmt werden, wie restriktiv diese Sicherheitsrichtlinien sein sollen, da dies ggf. einen Einfluss auf die weitere Ausgestaltung der Netzarchitektur hat. Hier ergibt sich ein recht großer Freiheitsgrad. So ist es einerseits denkbar, eine im Prinzip „offene“, Internet-artige Erreichbarkeit auf der IP-Schicht vorzusehen und die Control Plane nur optional zu verwenden, um für die so signalisierten Rufe bestimmte zusätzliche Eigenschaften sicherzustellen, z. B. eine höhere Dienstgüte oder verifizierte Absenderadressen. Ein anderer Ansatz ist, die Architektur des ISDN mit seinen engen Schnittstellen und wohldefinierten Diensten beizubehalten und lediglich die TDM-basierte Übertragungstechnik durch IP-basierte Protokolle zu ersetzen. In diesem Szenario wäre eine direkte Kommunikation zwischen Teilnehmern auf der IP-Schicht komplett unmöglich gemacht.

Eine mögliche Ausprägung dieses Grundgedankens sind die 3GPP IMS- und ETSI TISPAN-Architekturen, es gibt aber auch ähnliche Überlegungen von anderen Konsortien, z. B. [35].

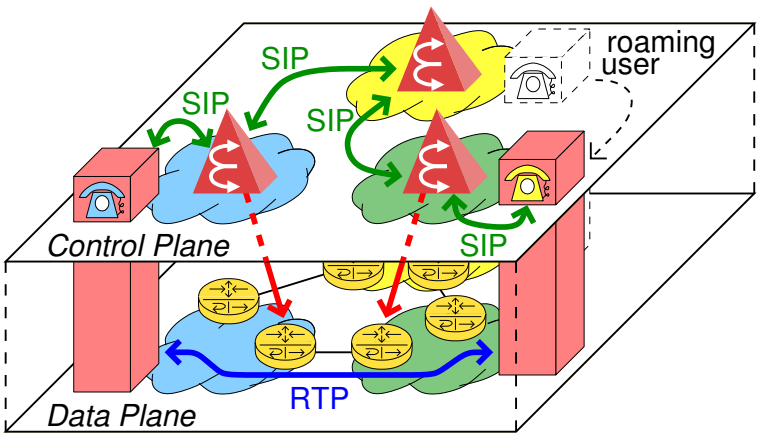


Abbildung 3.1: Control Plane und Data Plane in IP-Telefonie-Plattformen

3.4 Grundzüge der IMS- und TISPAN-Architektur

Das *IP Multimedia Subsystem* (IMS) wurde von 3GPP als Architektur für die Erbringung von Multimedia-Kommunikationsdiensten in IP-basierten Mobilfunknetzen der dritten Generation spezifiziert und standardisiert. Die dabei verwendeten Konzepte sind weitgehend unabhängig von dem zugrundeliegenden IP-Netz, lediglich an einigen Stellen (z. B. Dienstgüte-Unterstützung) konnten die Architektur vereinfachende Annahmen gemacht werden, da nur der GPRS-basierte Mobilfunkzugang unterstützt werden muss. Die ETSI TISPAN-Arbeitsgruppe hat das IMS als einen wesentlichen Kern ihrer Architektur übernommen und weitere Module und Schnittstellen hinzugefügt, um auch andere Zugangsnetz-Techniken unterstützen und festnetzspezifische Anforderungen erfüllen zu können. Desweiteren unterstützt die TISPAN-Architektur auch Dienste, die nicht auf SIP basieren, z. B. WWW-Zugang über HTTP, E-Mail oder „Peer-to-Peer Filesharing“.

In der IMS-Spezifikation werden Funktionsblöcke (engl. *Functions*), sowie die dazwischenliegenden Schnittstellen definiert. Hierbei handelt es sich ausschließlich um logische Funktionen. Wie diese auf den physikalischen Netzelementen platziert werden, ist eine Entwurfs-Entscheidung des Herstellers bzw. Netzbetreibers. Sie beeinflusst u. a., ob Schnittstellen durch Kommunikation über das Netz oder durch Host-interne Kommunikation mit deutlich geringeren Latenzen implementiert werden können. Da diese Architektur eine große Komplexität und eine Vielzahl von Funktionsblöcken aufweist, werden in Abbildung 3.2 nur die für die Betrachtung des Netzübergangs wichtigsten Funktionen und Schnittstellen schematisch dargestellt. Eine vollständige Liste aller Schnittstellen befindet sich in [TS 23.002].

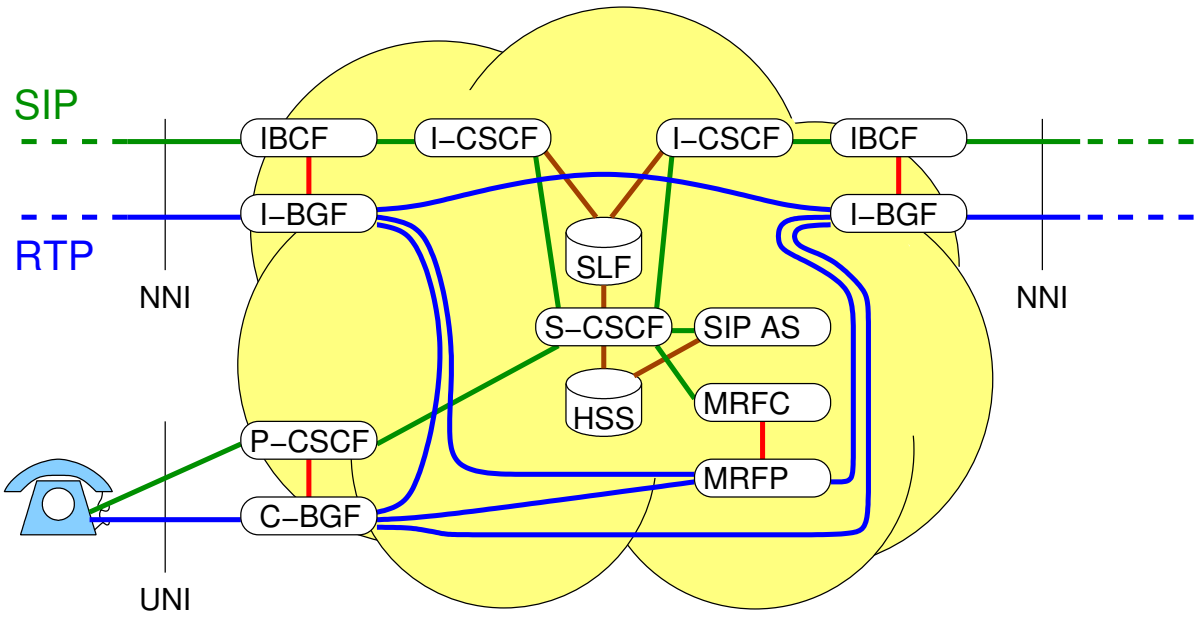


Abbildung 3.2: Überblick über die grundsätzliche Architektur und die wichtigsten Funktionsblöcke von 3GPP IMS und ETSI TISPAN (schematisch), mit besonderem Augenmerk auf die für Netzübergänge relevanten Funktionsblöcke.

3.4.1 Die Control Plane im IMS

Die Sitzungssignalisierung erfolgt im IMS mit Hilfe von SIP. Verglichen mit den Internet-Szenarien der IETF wird ein viel größerer Anteil der Sitzungssteuerung und Dienstelogik nicht in den Endgeräten, sondern in SIP-Servern „im Netz“ platziert. Diese Server erbringen drei Gruppen von Funktionen, die zusammenfassend als *Call/Session Control Function* (CSCF, in älteren Dokumenten auch *Call State Control Function*) bezeichnet werden.

Die zentrale Komponente der Control Plane ist die so genannte *Serving-CSCF* (S-CSCF). Alle Signalisiernachrichten, die von bzw. zu einem Teilnehmer gesendet werden, werden von der S-CSCF bearbeitet. Zu den Funktionen der S-CSCF gehören Adressabbildungen und die Wegesuche für SIP-Nachrichten. Falls der Teilnehmer einen Mehrwertdienst gewählt hat, werden von der S-CSCF die entsprechenden *Application Servers* (AS) mit in die Sitzung einbezogen, die die jeweilige signalisierungsbezogene Dienstelogik implementieren. Falls auch medienbezogene Dienste (z. B. Sprach-Ansagen) benötigt werden, wird zusätzlich ein *Media Resource Function Processor* (MRFP) hinzugezogen, welcher von einem *Media Resource Function Controller* (MRFC) gesteuert wird. Vor der Weiterleitung einer Nachricht wird von der S-CSCF geprüft, ob der jeweilige Teilnehmer überhaupt zum Anruf dieses Ziels bzw. zur Nutzung dieses Dienstes berechtigt ist. Hierfür und auch für andere Zwecke (z. B. Einbuchen eines Teilnehmers in das Netz, d. h. SIP Registrar-Funktionalität) steht die S-CSCF in Kontakt mit dem *Home Subscriber Server* (HSS), einer Datenbank, die eine Weiterentwicklung des aus dem GSM bekannten *Home Location Register* (HLR) ist.

Zur Steigerung der Kapazität können in einem Netz auch mehrere S-CSCF installiert werden. Jedem Teilnehmer wird dann eine S-CSCF fest zugeordnet. Diese Zuordnung wird in einer weiteren Datenbank, der *Subscription Locator Function* (SLF), abgelegt, so dass andere Instanzen der Plattform die für einen bestimmten Teilnehmer „zuständige“ S-CSCF ermitteln können. Um die Verfügbarkeit zu erhöhen, können die jeweiligen Netzelemente und Protokollinstanzen redundant ausgelegt werden, dies zählt logisch jedoch als ein Knoten.

Die *Proxy-CSCF* (P-CSCF) stellt für die SIP-Signalisierung den Kontakt zwischen den Teilnehmern und der Control Plane her, d. h. sie implementiert den signalisierungsbezogenen Teil des UNI. Viele der Aufgaben der P-CSCF haben Bezug zur Netzsicherheit. So wird die Signalisierung durch kryptographische IPsec-Tunnel zwischen dem Endgerät und der P-CSCF geschützt. Im Zuge des Aufbaus dieser Tunnel wird die Identität des Teilnehmers von der P-CSCF festgestellt und anderen Funktionsblöcken in der selben Vertrauensdomäne (i. d. R. die Control Plane des eigenen Netzes) zur Verfügung gestellt. Dazu wird eine entsprechende Zeile (*P-Asserted-Identity* [RFC 3325]) in alle vom Teilnehmer empfangenen und in das Netz weitergeleiteten SIP-Nachrichten eingefügt. Ferner werden vom Teilnehmer kommende SIP-Nachrichten hier auf syntaktische Korrektheit geprüft. An die P-CSCF kann – aus Sicht der 3GPP IMS-Standards optional – eine Funktion gekoppelt werden, die für Autorisierung oder QoS-Management von Medienströmen zuständig ist.

Die *Interrogating-CSCF* (I-CSCF) befindet sich an der Grenze zwischen administrativen Domänen und stellt dementsprechend den signalisierungsbezogenen Teil des NNI dar. Eine der Hauptaufgaben der I-CSCF ist, aus anderen Domänen ankommende Signalisiernachrichten an die S-CSCF weiterzuleiten, die dem gerufenen Teilnehmer zugeordnet ist. Dazu wird die SLF (s. o.) abgefragt. Bei ausgehendem Verkehr kann die I-CSCF einzelne Felder einer Nachricht entfer-

nen oder verschlüsseln, um Informationen über die eigene Domäne gegenüber anderen Domänen zu verschleiern. Diese Funktion wird als *Topology Hiding Internetwork Gateway* (THIG) bezeichnet. Prinzipiell können die I-CSCF einer Domäne im öffentlichen DNS bekannt gemacht werden, so dass SIP-Instanzen im Internet Nachrichten an diese Domäne entsprechend [RFC 3263] senden können. Für die Weiterleitung von Transitverkehr durch die Domäne hindurch definiert die Spezifikation [TS 23.228] mehrere Optionen, mit oder ohne Einbeziehung der S-CSCF.

Die bisher benannten Funktionen der IMS-Architektur beziehen sich alle auf die Control Plane. Mit diesen Funktionen ist es auch prinzipiell möglich, eine komplexe Dienstelogik oberhalb eines „offenen, Internet-artigen“ IP-Netzes (siehe Abschnitt 3.3) zu betreiben [79, S. 112].

3.4.2 Die TISPAN-Funktionen am Netzübergang

Neben dem IMS als Kern für SIP-basierte Dienste sind zwei weitere Subsysteme der TISPAN-Architektur im Zusammenhang mit dieser Arbeit von besonderer Bedeutung [ES282001]: Das *Network Attachment Subsystem* (NAS) authentisiert und autorisiert den Teilnehmer zur prinzipiellen Nutzung des IP-Zugangsnetzes und nimmt ggf. eine Konfiguration seines Endgerätes auf der IP-Schicht vor (z. B. Zuweisung einer IP-Adresse). Das *Resource and Admission Control Subsystem* (RACS) dient zur Steuerung von Paketfiltern (in der ETSI-Terminologie als *Gate Control* bezeichnet [ES282001, Abschnitt 5.2.1]), von Network Address and Port Translators (NAPT), sowie zur Prüfung, ob die von einem Teilnehmer gesendeten Datenströme in Abhängigkeit des Teilnehmer-Profils und der freien Netz-Ressourcen akzeptiert werden können.

Auf der IP-Schicht werden die oben genannten Funktionen am Netzübergang durch die so genannte *Border Gateway Function* (BGF) erbracht. Es kann dabei weiter zwischen der *Core BGF* (C-BGF) an der Grenze zwischen Zugangs- und Kernnetz (UNI) und der *Interconnection BGF* (I-BGF) am Netzübergang zu anderen Betreibern (NNI) unterschieden werden. Eine der Funktionen der BGF ist die Filterung von IP-Paketen auf Basis von IP-Adressen und Transportschicht-Portnummern, d. h. Paketfilter-Funktionalität.

Die *Interconnection Border Control Function* (IBCF) ist eine SIP-Protokollinstanz zur Absicherung des (logischen) Netzübergangs in der Control Plane. Sie kann Funktionen zur Zugriffskontrolle und zum Verstecken der Netztopologie implementieren, die über die Funktionen des I-CSCF (siehe Abschnitt 3.4.1) hinausgehen. Desweiteren kann sie mit dem RACS interagieren, zur Steuerung von Firewalls (d. h. Steuerung der I-BGF), NAPT und zur Reservierung von QoS-Ressourcen. Die Steuerung der C-BGF kann entsprechend durch den P-CSCF erfolgen.

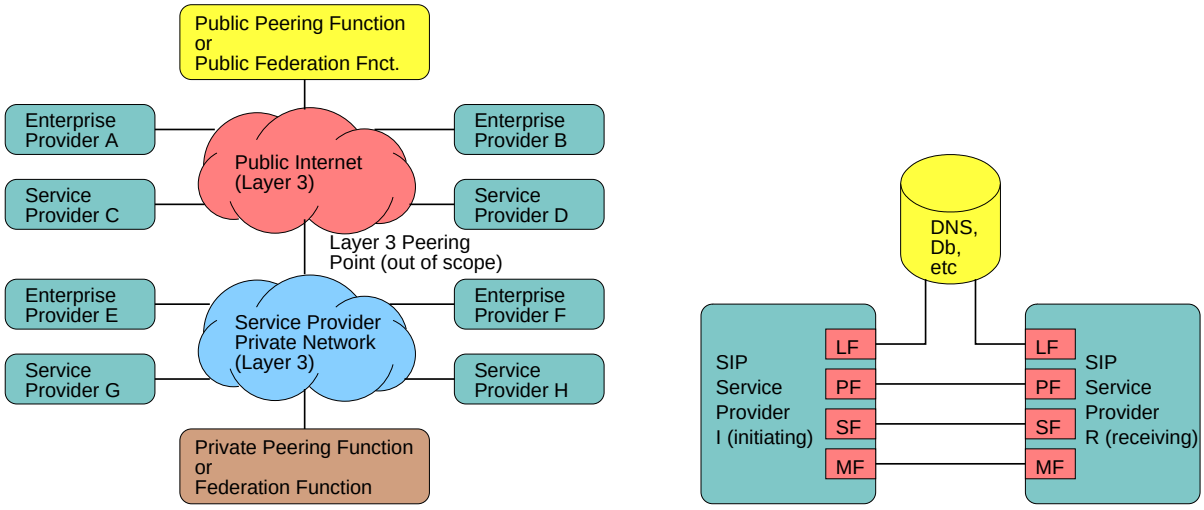
Auch bei den hier beschriebenen Funktionen handelt es sich um logische Funktionen, für die es verschiedene Optionen bzgl. der Platzierung auf physikalischen Netzelementen gibt. Dementsprechend können die Schnittstellen zwischen IBCF und I-BGF bzw. zwischen P-CSCF und C-BGF entweder interne Schnittstellen innerhalb eines Netzelements sein, oder auch externe Schnittstellen, für die ein Kommunikationsprotokoll benötigt wird.

3.5 Konzepte für die Netzzusammenschaltung

Bei den in Abschnitt 3.4 beispielhaft vorgestellten IMS/TISPAN-Architekturen und ähnlichen Ansätzen wird davon ausgegangen, dass sich eine solche IP-Telefonie-Plattform unter der administrativen Kontrolle eines einzigen Netzbetreibers befindet. Aufgrund der deregulierten nationalen Märkte und um internationale Verbindungen zu ermöglichen, werden Konzepte zur effizienten und sicheren Zusammenschaltung von Netzen benötigt. Die selben Sicherheitsüberlegungen, die zur Abschottung des *User-to-Network Interface* dieser Plattformen geführt haben, verbieten, dass diese ohne Zugriffskontrollen am Netzübergang mit anderen, nicht vertrauenswürdigen Netzen der Mitbewerber oder dem Internet zusammengeschaltet werden. Dementsprechend wurden mit der *IBCF* und *I-BGF* (bzw. äquivalenten Funktionen in anderen Architekturen) Zugriffskontrollmechanismen für das *Network-to-Network Interface* spezifiziert.

Wie die einzelnen IP-Telefonie-Plattformen, d. h. ihre jeweiligen Netzelemente am Rand zusammengeschaltet werden sollen, wird derzeit in verschiedenen Standardisierungsgremien und Herstellerkonsortien diskutiert. Eine solche Netzzusammenschaltung hat i. d. R. sowohl wirtschaftliche und rechtliche, als auch technische Aspekte. Gegenstand entsprechender juristischer Verträge kann neben Fragen der Abrechnung z. B. auch sein, ob Absenderadressen vom Netzbetreiber verifiziert werden müssen oder inwieweit ein Netzbetreiber garantiert, dass es sich bei den aus seinem Netz stammenden Rufen nicht um unerwünschte Werbung (SPIT) handelt, bzw. wie auf entsprechende Beschwerden reagiert werden muss. Solche Verträge können entweder bilateral zwischen einzelnen Betreibern geschlossen werden, oder mit einer zentralen Stelle. Diese legt Richtlinien bzgl. dieser Fragestellungen fest, die für alle Mitglieder verbindlich sind. Dies wird oft als *Circle of Trust* oder als *Föderation* bezeichnet.

Bei den technischen Fragestellungen geht es primär darum, wie die Wegesuche für Signalisierung und Medienströme implementiert wird. Die einfachste Realisierungsmöglichkeit ist eine logische (d. h. auf SIP-Ebene) Vollvermaschung der SIP-Plattformen aller Betreiber. Bis zu ei-



(a) Network Context (b) Reference Architecture

Abbildung 3.3: Von der IETF SPEERMINT Arbeitsgruppe betrachtete Szenarien der Netzzusammenschaltung („Network Context“) und dafür vorgesehene Referenz-Architektur nach [80]

ner gewissen Größe der Föderation ist dies durchaus denkbar, da hierfür keine Vollvermaschung auf der physikalischen Schicht mit Leitungen benötigt wird. Stattdessen kann ein IP-Netz als Transfernetz zum Einsatz kommen, welches „normales“ IP-Routing verwendet. Prinzipiell ist dafür sogar die Verwendung des Internets denkbar, ggf. unter Zuhilfenahme von z. B. mit IPsec kryptographisch gesicherten Tunneln (siehe Abbildung 3.3(a)). Der Zugang zu diesem (ggf. virtuellen) Transfernetz darf nur solchen Plattform-Betreibern gestattet werden, die sich vertraglich zur Einhaltung der jeweiligen Sicherheitsrichtlinien verpflichtet haben. Für diese Art der Netzzusammenschaltung reichen die Mechanismen des „SIP Trapezoids“ [RFC 3263] und ENUM [RFC 3761] prinzipiell aus, da die Weiterleitung der Signalisiernachrichten immer noch direkt zwischen der Domäne des rufenden Teilnehmers zur Domäne des gerufenen Teilnehmers erfolgt, ohne auf SIP-Ebene durch eine Transit-Domäne zu gehen. Die IETF *Session PEERing for Multimedia INTerconnect* (SPEERMINT)-Arbeitsgruppe arbeitet derzeit überwiegend technische, aber auch die Sicherheitsanforderungen betreffende Empfehlungen für dieses Modell aus. Alternativ zu DNS/ENUM sollen hier auch andere, zentrale Datenbanken untersucht werden, in denen die Zuordnung der Teilnehmer zu ihren Heimat-Netzbetreibern gespeichert ist. Ferner werden Erweiterungen des SIP-Protokolls diskutiert, mit denen einzelne Plattformbetreiber signalisieren können, dass einzelne ihrer Richtlinien von denen für die Föderation global festgelegten Richtlinien abweichen. Letzteres wird in der SPEERMINT-Terminologie als *Policy Function* (PF) bezeichnet, Sie ergänzt die *Signaling Function* (SF) und die *Media Function* (MF) für die Übertragung von Signalisierung bzw. Mediendaten, sowie die *Location Function* (LF) zur Abfrage der zentralen Teilnehmer-Datenbank (siehe Abbildung 3.3(b), schematisch nach [80]).

Ähnliche Konzepte werden auch von der *GSM Association* (GSMA), einer Vereinigung von Mobilfunk-Betreibern und -Herstellern untersucht. Neben einer logischen Vollvermaschung von IMS-Plattformen mit Hilfe einer vom öffentlichen Internet komplett separaten DNS/ENUM-Hierarchie werden auch Szenarien betrachtet, in denen SIP-Server, so genannte *IPX Proxies*, die Bildung sternförmiger Strukturen ermöglichen, um die Konfiguration der Randknoten für die Netzbetreiber zu vereinfachen (siehe Abbildung 3.4). Die Proxies können auch das so genannte *Protokoll-Interworking* erleichtern, wenn verschiedene Netzbetreiber zueinander inkompatible Protokollversionen verwenden (z. B. SIP-Erweiterungen oder auch IPv4/IPv6). Die Medienströ-

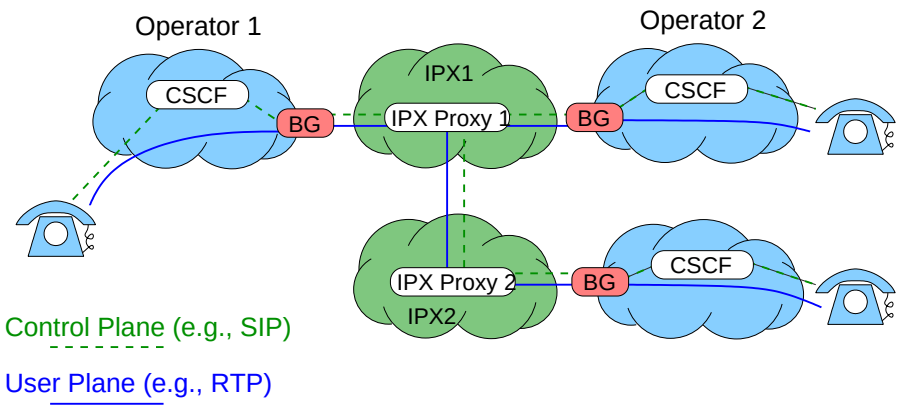


Abbildung 3.4: Netzzusammenschaltung auf Session-Ebene über Zwischennetze, die IP-Exchange-Proxies (IPX-Proxies) enthalten

me können entweder direkt fließen oder ebenfalls über Proxies geführt werden, z. B. zur Transcodierung, falls Ursprungs- und Ziel-Netz unterschiedliche Sprach-Codecs verwenden [81, 82].

Die Konzepte beider Gruppen gehen bei ihrem derzeit (Anfang 2007) dokumentierten Entwicklungsstand von der logischen Vollvermaschung über ein gemeinsames Transfernetz mit voller IP-Konnektivität zwischen allen Plattformen aus. Dies setzt voraus, dass sich die Betreiber der angeschlossenen Netze auf einen gemeinsamen Satz von Richtlinien einigen und einen vertrauenswürdigen Betreiber für das Transfernetz finden können. Längerfristig erscheint daher die Erforschung von Konzepten sinnvoll, die eine Weiterleitung von Signalisiernachrichten auf SIP-Ebene durch eine Kette mehrerer Transit-Domänen ermöglichen. Ansätze für ein entsprechendes Routing-Protokoll sind bisher aber nur sehr rudimentär vorhanden (siehe z. B. [RFC 2871]).

Eine Studie [83], die sich mehr auf die wirtschaftlichen und regulatorischen als um die technischen Aspekte der Netzzusammenschaltung konzentriert, untersucht u. a. die derzeitige Situation im PSTN/ISDN und folgert, dass sich zukünftig eine dreistufige Hierarchie von NGN-Netzen etablieren könnte: Internationale Transit-Netze, große nationale Netze (z. B. der Ex-Monopolisten) und kleine, regionale Anbieter (siehe Abbildung 3.5). In einem solchen Szenario könnten sich zwischen zwei Teilnehmern bis zu sechs Domänen und somit bis zu 14 Firewalls auf dem Medienpfad befinden, falls alle Domänengrenzen, inklusive der lokalen Netze der Teilnehmer, mit Firewalls abgesichert werden.

3.6 Zusammenfassung und Fazit

Nach einem Überblick über Bedrohungsszenarien für VoIP wurden in diesem Kapitel Möglichkeiten, aber auch Grenzen beim Einsatz von Firewalls aufgezeigt. IP-Telefonie-Plattformen sind Netze, die zwar IP-basierte Protokolle verwenden, deren Sicherheitsrichtlinien i. d. R. aber deutlich strikter als die des Internets sind. Anders als bei einem typischen „Internet-Szenario“ sind bei der Zusammenschaltung solcher Plattformen Szenarien denkbar, bei denen eine zweistellige Anzahl von Firewalls auf dem Medienpfad zwischen zwei Teilnehmern liegt. Darum ist es notwendig, den Einfluss der Firewalls auf die Dienstgüte zu untersuchen. Neben der Bearbeitung der einzelnen RTP-Pakete, die einen Einfluss auf die „Mund-zu-Ohr-Verzögerung“ hat, muss dabei auch betrachtet werden, wie die Konfiguration der Firewalls den Verbindungsaufbau vor Gesprächsbeginn verzögert.

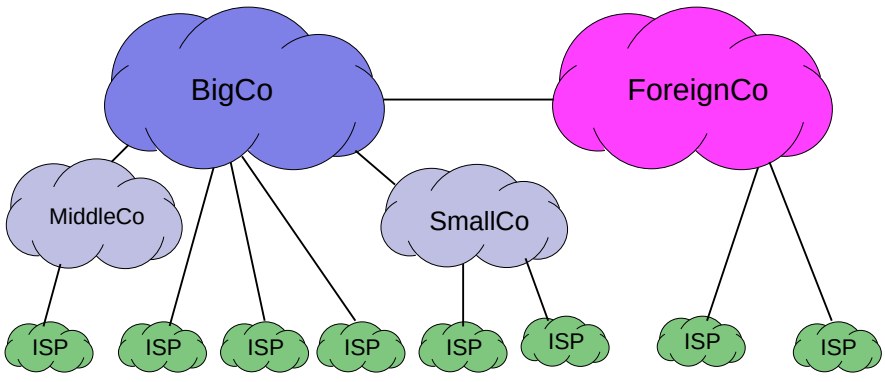


Abbildung 3.5: Hypothetisches Interconnection-Szenario, nach [83]

4 Architekturen verteilter Firewalls

In den vorangegangenen Kapiteln wurde dargestellt, dass SIP/RTP-basierte Multimedia-Anwendungen – anders als die meisten „klassischen“ Internet-Anwendungen (z. B. E-Mail, WWW) – auf Außenbandsignalisierung, d. h. auf getrennten Flows für Signalisierung und Nutzdatentransport beruhen, und dass dies Probleme beim Überqueren von Firewalls bereitet. Aufgrund der Tatsache, dass für RTP anstelle einer *Well-Known Port Number* eine mittels SIP/SDP dynamisch ausgehandelte Portnummer verwendet wird, ist ein RTP-Medienstrom für einen einfachen Paketfilter schwierig als solcher zu erkennen. Aus dieser und aus anderen Überlegungen (z. B. SPIT-Abwehr) resultiert die Notwendigkeit, bei Zugriffskontrollmechanismen am Netzübergang (Firewalls) die Signalisierung und die Medienströme gemeinsam zu betrachten. Verschiedene Architekturvarianten für dieses Ziel sollen in diesem Kapitel vorgestellt werden.

4.1 Klassifikation grundsätzlicher Architekturen

Abbildung 4.1 gibt eine Übersicht über Architekturvarianten für Firewalls, die SIP und RTP gemeinsam betrachten. Ein wichtiges Kriterium zur Klassifikation ist, ob sich die logischen

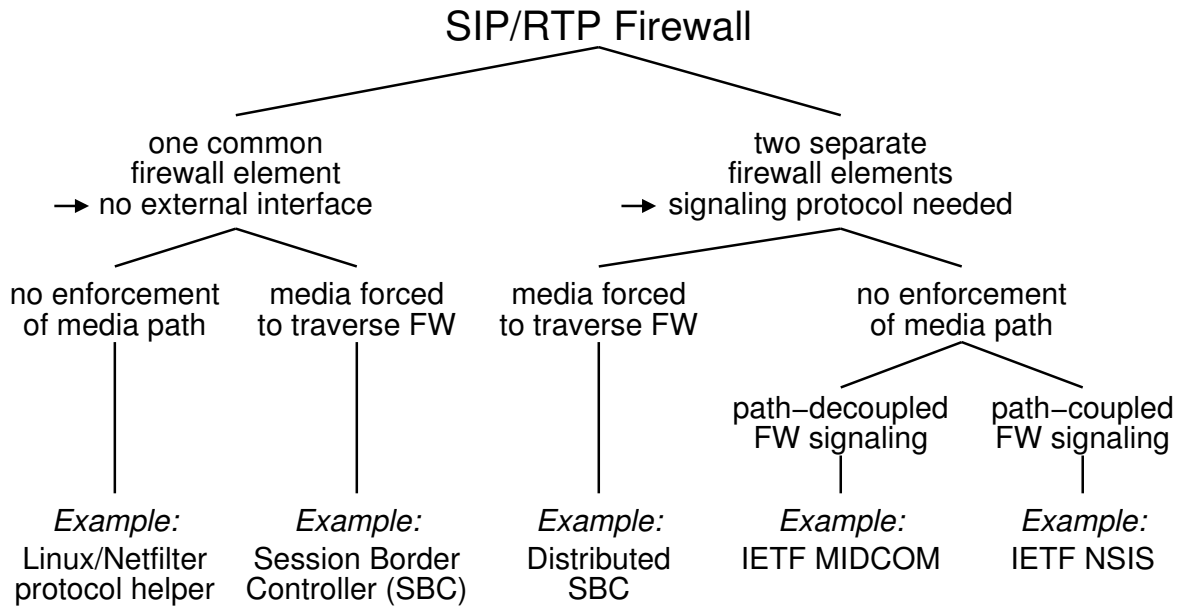


Abbildung 4.1: Klassifikation von Firewalls für SIP/RTP

Instanzen, die SIP bzw. RTP bearbeiten (im Folgenden als *Signalisierkomponente* bzw. als *Medienkomponente* bezeichnet) in einem physikalischen Netzelement befinden, oder ob sie getrennt in verschiedenen Netzelementen platziert sind.

4.1.1 Signalisier- und Medienkomponente in einem Netzelement

Ein wesentlicher Vorteil der Unterbringung von Signalisier- und Medienkomponente in einem gemeinsamen Netzelement ist, dass die Schnittstelle zwischen diesen beiden Komponenten dann eine geräteinterne Schnittstelle ist. Somit kann vom jeweiligen Hersteller problemlos eine proprietäre Lösung gewählt werden (siehe Abbildung 4.2). Vor dem Hintergrund, dass die Standardisierung solcher Schnittstellen für den Einsatz zwischen räumlich getrennten Komponenten (siehe Abschnitte 4.4 und 4.6) noch nicht abgeschlossen ist, ist dies ein wichtiges Argument der Firewall-Hersteller. Dass Signalisierdaten und Medienströme durch einen gemeinsamen Zwangspunkt geführt werden müssen, stellt jedoch auch einen wesentlichen Nachteil dieser Lösung dar. Es ist so nicht möglich, wie in Abbildung 3.1 schematisch dargestellt, die Signalisierung auf Umwegen z. B. durch das Heimatnetz eines mobilen Nutzers (Roaming) oder an einem Mehrwertdiensteserver vorbei zu führen, gleichzeitig aber die Medienströme auf dem direkten Pfad zwischen den Teilnehmern mit der minimalen Verzögerung zu transportieren.

Bei dieser Klasse von Firewall-Architekturen kann weiter unterschieden werden, ob das Hindurchleiten von Signalisierung und Medienströmen durch ein gemeinsames Netzelement als gegeben angenommen wird, oder ob dies von der Firewall erzwungen wird. In die erste Kategorie fallen Lösungen, die die SIP-Signalisierung nur passiv analysieren, ohne die Signalisiernachrichten zu ändern. Dazu ist u. U. keine vollständige SIP-Protokollinstanz (Proxy, B2BUA, etc.) notwendig; u. U. genügt ein Parser für SIP/SDP Nachrichten um zu ermitteln, welche *Pinholes* für die RTP-Medienströme geöffnet werden müssen. Ob diese Medienströme dann auch tatsächlich durch das Firewall-Element fließen, kann von dieser Architektur nicht beeinflusst werden. Nachteilig bei der rein passiven Analyse der SIP-Signalisierung ist, dass die *INVITE*-Nachricht einen anderen Pfad durch das Netz (z. B. über Proxies) nehmen kann als die *BYE*-Nachricht (u. U. direkt zwischen den beiden User Agents). Somit kann es vorkommen, dass die Firewall zwar den Gesprächsbeginn feststellen kann, aber nicht das Ende der Verbindung. Somit wäre unklar, wann die Pinholes für die Medienströme wieder zu schließen sind. Aufgrund dieser Einschränkungen eignet sich dieser Ansatz praktisch nur für solche Szenarien, bei denen

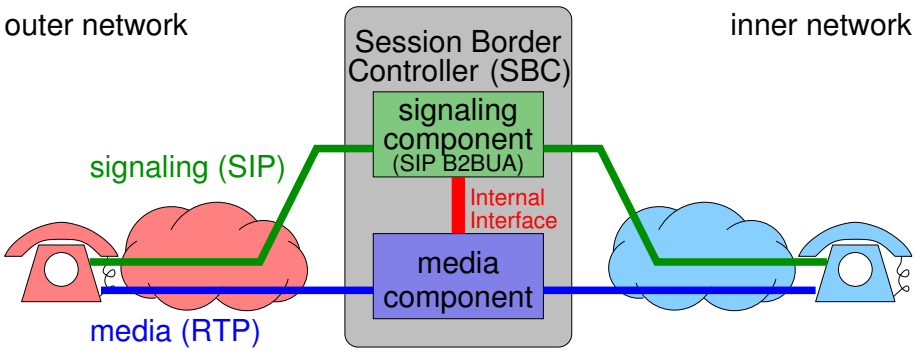


Abbildung 4.2: Architektur eines “Session Border Controllers” (SBC) nach [84]

durch die Netztopologie vorgegeben ist, dass alle Signalisiernachrichten und Medienströme durch dieses Firewall-Element fließen müssen, z. B. Absicherung eines privaten lokalen Netzes mit nur einem Übergang zum Internet. Ein Beispiel für diesen Architektur-Ansatz sind die „SIP conntrack helper“ für das Betriebssystem „Linux“ (in [85] und [86] unabhängig voneinander implementiert).

Falls die Medienströme dazu gezwungen werden sollen, eine bestimmte Medienkomponente zu durchlaufen, so wird dies i. d. R. dadurch erreicht, dass anstelle eines transparenten Paketfilters ein RTP-Proxy verwendet wird, der auf IP-Schicht adressierbar ist. Durch eine Modifikation der Signalisiernachrichten (Umschreiben der IP-Adressen in den SDP-Nachrichten mit Hilfe eines SIP *Back-to-Back User Agents*, B2BUA) werden die Endsysteme dazu gebracht, die Medienströme nicht direkt an die Gegenstelle, sondern an den RTP-Proxy zu senden.

Die Kombination aus B2BUA zur Bearbeitung der Signalisierung und einer entsprechenden Medienkomponente in einem Netzelement wird von den Herstellern unter dem Namen *Session Border Controller* (SBC) vermarktet. Es sei angemerkt, dass diese Bezeichnung in den offiziellen SIP-Standards der IETF [RFC 3261, ff.] nicht erwähnt oder spezifiziert wird. Die grundsätzliche Architektur eines SBC (siehe Abbildung 4.2) weißt deutliche Ähnlichkeiten mit der IETF MIDCOM-Architektur (siehe Abschnitt 4.4 und Abbildung 4.6) auf, mit dem Unterschied, dass es sich bei MIDCOM um eine verteilte Architektur mit Signalisier- und Medienkomponente in getrennten Netzelementen handelt. In gewisser Weise kann ein *Session Border Controller* daher als „MIDCOM in a box“ betrachtet werden. Allerdings gibt es deutliche Unterschiede bzgl. der Funktion der Signalisierkomponente: Der B2BUA wird bei MIDCOM „nur“ dazu benötigt, um durch Analyse der SIP/SDP-Signalisierung die Parameter für das Öffnen der Pinholes für die Medienströme zu ermitteln. Die SIP-Signalisierung soll dabei so wenig wie möglich beeinflusst werden. Hingegen ist bei einem SBC die Steuerung der Medienkomponente nur ein Teilaspekt der Signalisierkomponente. Aus anderen Gründen, z. B. zum Verschleiern der Netztopologie, zur Anonymisierung von Teilnehmern, zum Unterdrücken bestimmter SIP-Dienstemerkmale oder zur Ermöglichung der Zusammenarbeit inkompatibler SIP-Erweiterungen, etc. wird hier bewusst massiv in die SIP-Signalisierung eingegriffen. Die Auswirkungen solcher Netzelemente auf die Sicherheit, Interoperabilität, Skalierbarkeit, Dienstgüte, usw. von SIP-basierten Netzen wird noch untersucht [84].

4.1.2 Signalisier- und Medienkomponente in getrennten Netzelementen

Falls die Signalisier- und die Medienkomponente der IP-Telefonie-Firewall in zwei getrennten physischen Netzelementen platziert werden sollen, so wird ein Kommunikationsprotokoll für den Austausch von Steuerinformationen zwischen diesen beiden Komponenten benötigt. Nachdem die Signalisierkomponente im Zusammenspiel mit anderen Instanzen der Control Plane einen bestimmten Ruf autorisiert hat, werden mit Hilfe dieses Signalisierprotokolls die entsprechenden Regeländerungen in die Medienkomponente eingebracht.

Wie in Abbildung 4.1 dargestellt, kann auch hier zunächst unterschieden werden, ob durch Modifikation der SIP/SDP-Signalisierung in der Signalisierkomponente erzwungen wird, dass die RTP-Medienströme eine bestimmte Medienkomponente durchqueren, oder ob die Filterung der Medienströme transparent erfolgt. In letzterem Fall erfolgt der Transport der Medienströme Ende-zu-Ende zwischen den Multimedia-Endgeräten. Die Wahl des Pfades hängt ausschließlich

von der Verkehrslenkung auf der IP-Schicht ab, welche in großen Netzen i. d. R. von einem dynamischen Routing-Protokoll gesteuert wird. Die Firewall-Steuerung muss daher dafür sorgen, dass die Pinholes zur Freigabe der Medienströme in diejenige Medienkomponente eingetragen werden, die sich tatsächlich auf dem Pfad der Medienströme befindet. Es kann hier weiter zwischen zwei Varianten unterschieden werden, je nachdem, von welcher Komponente die Signalisierung initiiert wird und wie die Signalisiernachrichten durch das Netz gesendet werden. Diese beiden Grundverfahren, die *Pfad-entkoppelte Firewall-Signalisierung* und die *Pfad-gekoppelte Firewall-Signalisierung*, sowie die jeweils zugehörigen, von der IETF standardisierten Protokollarchitekturen werden in den folgenden Abschnitten beschrieben.

4.2 Pfad-entkoppelte Firewall-Signalisierung

Bei der *Pfad-entkoppelten Firewall-Signalisierung* (siehe Abbildung 4.3(a)) kommen i. d. R. vergleichsweise einfache Client-Server-Protokolle zum Einsatz. Die Server-Instanz befindet sich auf der Medienkomponente und wartet auf Steuerbefehle, die das Öffnen von Pinholes für Medienströme anfordern. Die Client-Instanz des Firewall-Signalisierprotokolls wird in eine an der Sitzungssignalisierung beteiligte Instanz (z. B. SIP B2BUA) integriert, die die für die Spezifikation der Pinholes benötigten Parameter aus der Sitzungssignalisierung ermitteln kann. Falls die Nachrichten der Sitzungssignalisierung (z. B. SIP) und die Medienströme (z. B. RTP) auf verschiedenen Pfaden durch das Netz transportiert werden, läuft die Firewall-Signalisierung „quer“ dazu, zwischen Netzelementen auf diesen beiden Pfaden. Dass die Nachrichten der Firewall-Signalisierung somit nicht notwendigerweise dem Medienpfad folgen, begründet den Namen dieses Verfahrens. Damit dies funktionieren kann, muss der Client-Instanz bekannt sein, welche zu steuernden Server-Instanzen tatsächlich auf dem Pfad eines gegebenen Medienstroms liegen. Die Steuerkommandos werden dann explizit an diese Instanzen adressiert.

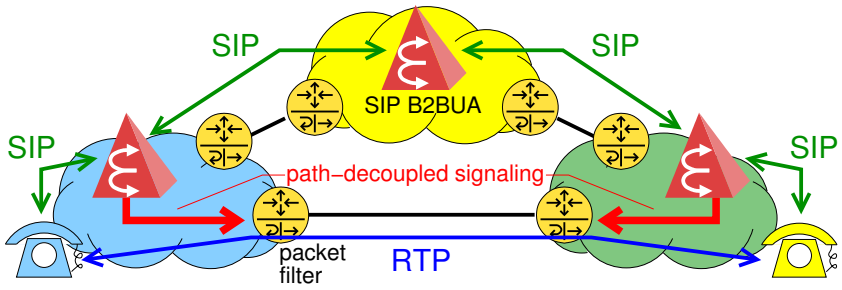
Es gibt eine ganze Reihe von Protokollen, die eine Pfad-entkoppelte Firewall-Signalisierung (engl. *path decoupled firewall signaling*, in [87] als *path targeted signaling* bezeichnet) ermöglichen. Einer der ersten Lösungsansätze für das *SIP/RTP Firewall Traversal Problem* war das im Umfeld des „SIP Express Router“ (SER) [88] entstandene, experimentelle „Firewall Control Protocol“ (FCP) [89, 90]. Auch kommerzielle, proprietäre Lösungen wie z. B. [91] waren recht früh verfügbar. Nach einer Reihe von Diskussionen im Umfeld anderer Arbeitsgruppen und nach Vorplanungstreffen [92, 93] wurde Anfang 2001 die IETF MIDCOM-Arbeitsgruppe gegründet, um eine entsprechende Architektur und dazugehörige Protokolle zu standardisieren. Auf die im Umfeld dieser Arbeitsgruppe entstandenen Ergebnisse wird in den Abschnitten 4.4 und 4.5 näher eingegangen. Für den Einsatz in Next Generation Networks wurde in einer ITU-T Empfehlung [H.248.37] eine Erweiterung des *Media Gateway Control Protocol* [H.248.1 v3] zur Steuerung von Paketfiltern und Adressumsetzern (NAPT) spezifiziert. Auch bei *Universal Plug and Play* (UPnP), einer Architektur zur automatischen Konfiguration von Netzelementen in kleinen lokalen Netzen, sind Mechanismen zur Steuerung des Firewalls im Internet-Zugangsrouter vorgesehen [94], die in die Kategorie der Medienpfad-entkoppelten Signalisierung fallen.

4.3 Pfad-gekoppelte Firewall-Signalisierung

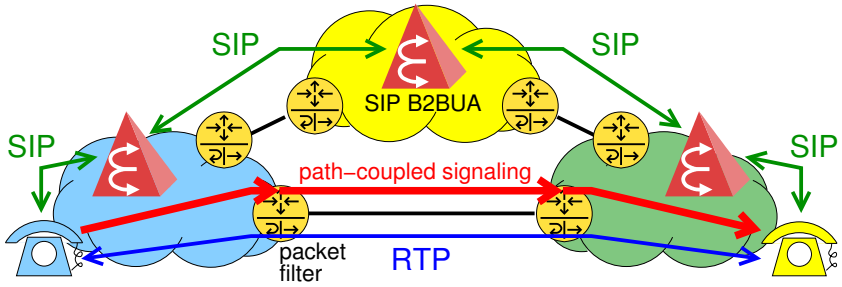
Ein alternatives Konzept ist die *Pfad-gekoppelte Firewall-Signalisierung*. Hierbei werden Signalisiernachrichten entlang des zukünftigen Medienpfades gesendet, die einen Medienstrom ankündigen und auf dem Pfad vorhandene Firewalls entsprechend konfigurieren (siehe Abbildung 4.3(b)). Die Signalisiernachrichten müssen dabei nicht notwendigerweise Ende-zu-Ende den vollständigen Medienpfad entlanglaufen (wie in Abbildung 4.4(a) dargestellt). Auch Instanzen, von denen bekannt ist, dass sie auf dem Medienpfad liegen, kommen prinzipiell als Endpunkt der Pfad-gekoppelten Signalisierung in Frage. Dies können z. B. *Session Border Controller* (SBC) sein, die durch Eingreifen in die SIP/SDP-Sitzungssignalisierung dafür sorgen, dass sie mit ihrer Medienkomponente immer auf dem Medienpfad liegen (siehe Abbildung 4.4(b)).

Bei der Pfad-gekoppelten Signalisierung werden die Signalisiernachrichten nicht an die einzelnen Firewall-Elemente auf dem Pfad adressiert. Als Zieladresse wird stattdessen die Adresse des Endpunkts der Firewall-Signalisierung angegeben. Firewall-Elemente, die dieses Signalisierverfahren unterstützen, müssen durch geeignete Mechanismen in der Lage sein, solche Signalisiernachrichten zu erkennen und entsprechende Konfigurationsänderungen (i. d. R. Öffnen von Pinholes) vorzunehmen. Die Anwendung dieses Signalisierverfahrens ist nicht auf das Steuern von Firewalls und Adressumsetzern beschränkt. Tatsächlich wurden schon in der Vergangenheit Protokolle vorgeschlagen, implementiert und standardisiert, die diesem Prinzip folgen. Darunter fällt das *Resource ReServation Protocol* (RSVP) [RFC 2205], mit dem Reservierungen von Ressourcen zur Dienstgüteunterstützung signalisiert werden können.

Schon recht früh existierten Vorschläge, RSVP zur Steuerung von Firewalls für Multimedia-Anwendungen mit Außenbandsignalisierung (zum damaligen Zeitpunkt noch auf H.323 statt



(a) Medienpfad-entkoppelte Firewall-Signalisierung (z. B. IETF MIDCOM)

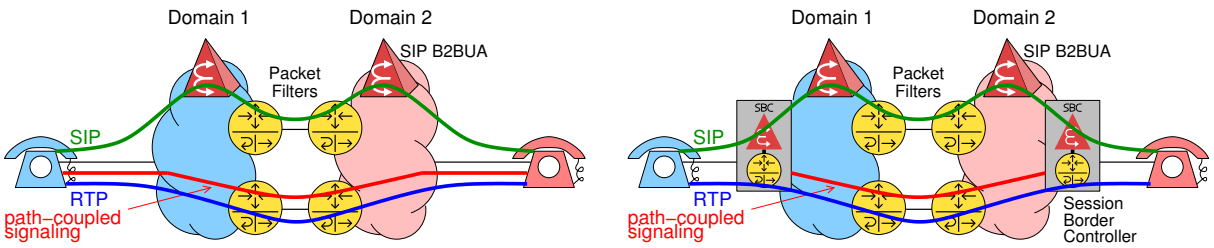


(b) Medienpfad-gekoppelte Firewall-Signalisierung (z. B. IETF NSIS)

Abbildung 4.3: Medienpfad-entkoppelte und Medienpfad-gekoppelte Firewall-Signalisierung

SIP basierend) zu verwenden [95, 50]. Dies ist auch naheliegend, da unter syntaktischen Gesichtspunkten das Anfordern von Ressourcen für einen Flow sehr ähnlich ist wie das Öffnen eines Pinholes in einem Paketfilter – in gewisser Weise kann ein Paketfilter als QoS-Vorrichtung mit nur zwei möglichen QoS-Stufen für einen Flow – „Bitrate Null“ und „Best Effort“ – gesehen werden. Bezüglich der Anforderungen an die Authentisierung und Autorisierung von Signalisiernachrichten können sich diese Anwendungsfälle jedoch erheblich unterscheiden. Ein mögliches Anwendungsszenario für Protokolle zur QoS-Ressourcen-Reservierung ist der Einsatz innerhalb einer einzelnen administrativen Domäne mit kooperativen Nutzern, die ein gemeinsames Verständnis haben, welche Anwendungen eine höhere Dienstgüte erfahren sollen. In diesem Szenario kann der Autorisierung u. U. eine nur untergeordnete Bedeutung zugemessen werden; die beteiligten Router können dann selbständig, basierend auf der Verfügbarkeit von Ressourcen entscheiden, ob eine Reservierung akzeptiert wird. In anderen Szenarien mit weniger kooperativen Nutzern muss hingegen verhindert werden, dass diese alle ihre Datenströme als höchstprior deklarieren, z. B. über Nutzungsentgelte. Vor einem großflächigen Einsatz im Internet sind dazu noch etliche eher administrative als technische Fragen zu lösen [96]. Anders als QoS-Mechanismen werden Firewalls prinzipbedingt immer in Mehr-Domänen-Szenarien eingesetzt, bei welchen sich nicht alle Betroffenen voll vertrauen. Eine Firewall wäre somit komplett nutzlos, wenn jedermann – Angreifer eingeschlossen – einfach das Öffnen beliebiger neuer Pinholes für evtl. bösartige Flows anfordern könnte. Deshalb müssen Signalisiernachrichten zur Pfad-gekoppelten Firewall-Steuerung durch das zu steuernde Firewall-Element sicher authentisiert und autorisiert werden können.

In den oben genannten Veröffentlichungen zur RSVP-basierten Firewall-Steuerung wird von einem „Internet-Szenario“ ausgegangen, d. h. die beiden Multimedia-Endgeräte befinden sich in je einem Zugangsnetz, die über das Internet verbunden sind (siehe Abbildung 4.5(a)). Auf dem Medienpfad befinden sich zwei als Paketfilter realisierte Firewalls, die das jeweilige Zugangsnetz schützen sollen. Die Steuerung dieser Paketfilter erfolgt durch Multimedia-Endgeräte mittels Ende-zu-Ende RSVP-Signalisierung. Dies setzt voraus, dass ein Vertrauensverhältnis zwischen dem Paketfilter am Rand eines Zugangsnetzes und den darin befindlichen Multimedia-Endgeräten besteht. Da die zum Öffnen von Pinholes benötigten Parameter sowohl in der RSVP *PATH*-Nachricht, als auch in der dazugehörigen, in Gegenrichtung gesendeten *RESV*-Nachricht enthalten sind, muss ein Paketfilter nur auf solche Nachrichten reagieren, die aus dem von ihm vertrauten Bereich stammen. In Abbildung 4.5(a) ist dies schematisch am Beispiel der Signalisierung dargestellt, die zum Öffnen eines Pinholes für einen Medienstrom von „UA2“ zu „UA1“ benötigt wird. Dazu sendet „UA1“ eine *PATH*-Nachricht in Richtung von „UA2“. Diese löst das Öffnen eines Pinholes im Paketfilter „PF1“ aus, welcher „UA1“ vertraut. Der Paketfil-



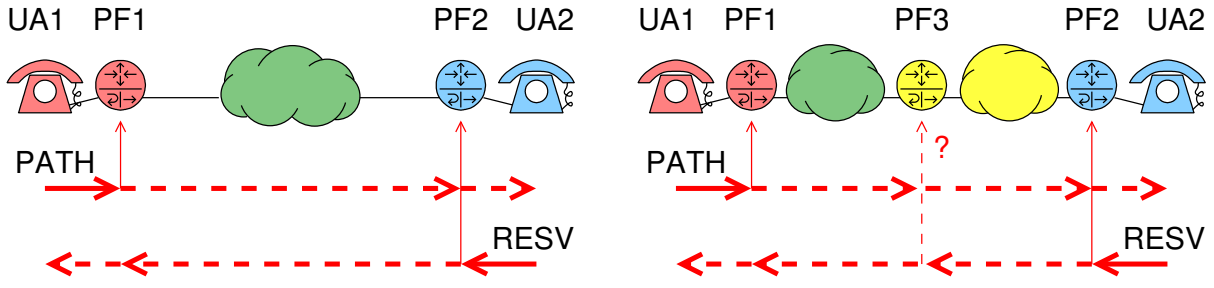
(a) Ende-zu-Ende (b) abschnittsweise

Abbildung 4.4: Medienpfad-gekoppelte Signalisierung: Ende-zu-Ende oder abschnittsweise

ter „PF2“ hingegen ignoriert diese Nachricht. Dort wird ein Pinhole erst geöffnet, nachdem der aus Sicht von „PF2“ vertrauenswürdige „UA2“ die *RESV*-Nachricht mit den selben Parametern zurückgesendet hat.

Da die Steuerung der Paketfilter durch die Endsysteme erfolgt, kommt dieses vergleichsweise einfache Schema prinzipiell ohne Signalisier-Komponente in der Firewall (H.323-Gatekeeper, SIP-Proxies, etc.) aus. Für die Firewalls ist es somit jedoch nicht möglich, die angeforderten Pinholes den per H.323 bzw. SIP signalisierten Multimedia-Sitzungen zuzuordnen, wie dies zur Erreichung bestimmter Schutzziele in IP-Telefonie-Plattformen notwendig ist (siehe Abschnitt 3.1). Desweiteren ist die Authentisierung von Signalisiernachrichten problematisch, wenn weitere Firewalls auf dem Medienpfad liegen, die keinem der beiden Endsysteme vertrauen (siehe Abbildung 4.5(b)). Wie die bei RSVP prinzipiell vorhandenen Mechanismen zur kryptographischen Authentisierung in solchen Szenarien konkret eingesetzt werden können, ist nicht im Detail beschrieben.

In der Dissertation von Cédric Aoun [87] werden die grundsätzlichen Ansätze *Pfad-entkoppelte Signalisierung* (dort als *Path-Targeted Signaling* bezeichnet) und *Pfad-gekoppelte Signalisierung* (dort als *Path-Directed Signaling* bezeichnet) im Zusammenhang mit Firewall- und NAPT-Steuerung unterschieden. Es wird dort ein wesentlicher Vorteil der Pfad-gekoppelten Signalisierung stark betont, nämlich dass bei diesem Verfahren bei der Instanz, die das Öffnen von Pinholes anfordert, keine Kenntnisse über die Netztopologie, über den aktuellen Zustand eines evtl. vorhandenen dynamischen Routingprotokolls oder über die Auswirkungen kaskadierter Adressumsetzer (NAPT) benötigt würden. Darauf folgend wird eine aus zwei Protokollschichten bestehende Architektur für die Pfad-gekoppelte Steuerung von Paketfiltern und Adressumsetzern vorgeschlagen, die in Bezug zu damals diskutierten Vorschlägen bzgl. eines RSVP-Nachfolgeprotokolls [97] steht. Es werden auch Mechanismen zur kryptographischen Authentisierung von Nachrichten diskutiert; auf eine Kopplung an die Sitzungssignalisierung wird aber auch hier nicht eingegangen. Diese Arbeiten haben die Arbeit der IETF Arbeitsgruppe *Next Steps In Signaling* (NSIS) beeinflusst, deren Ziel es ist, eine solche Architektur zu standardisieren. Bei den Beratungen zur Planung dieser Architektur, die in Abschnitt 4.6 vorgestellt wird, wurden die Begriffe *Path-coupled Signaling* und *Path-decoupled Signaling* geprägt.



(a) „Internet-Szenario“ (b) Szenario mit mehreren Transitdomänen

Abbildung 4.5: Authentisierung RSVP-basierter Firewall-Signalisierung

4.4 Die IETF MIDCOM-Architektur

Die Anfang 2001 ins Leben gerufene MIDCOM-Arbeitsgruppe (MIDdlebox COMmunications) der IETF hat das Ziel, eine Architektur und ggf. Protokolle zur Pfad-entkoppelten Steuerung von Middleboxes durch externe Prozesse zu entwerfen.

Von dem recht breit definierten Begriff Middlebox (siehe Abschnitt 2.3.1) werden derzeit nur Paketfilter und Network Address Translators (NAT) konkret betrachtet, es sind jedoch auch andere Typen von Middleboxes denkbar, z. B. Paket-Analysesysteme als Teil eines Network Intrusion Detection System (NIDS), sofern deren Regelsätze ebenfalls im Wesentlichen aus dem 5-Tupel zur Charakterisierung eines Flows bestehen.

4.4.1 Vorgehensweise der Arbeitsgruppe

In [RFC 3303] wurde zunächst eine Terminologie und eine abstrakte Architektur zur Steuerung von Middleboxes definiert. Obwohl diese Architektur klar dem Prinzip des *Path-decoupled Signaling* folgt, kommt dieser Begriff in den MIDCOM-Dokumenten nicht vor, da diese Klassifikation erst später etabliert wurde. In einem begleitenden Dokument [RFC 3304] werden – ebenfalls noch sehr abstrakt – Anforderungen an das MIDCOM-Signalisierprotokoll benannt. Teilweise basierend auf Erfahrungen mit experimentellen Vorläuferprotokollen bzw. proprietären Eigenentwicklungen wurde in [RFC 3989] die Semantik eines Client-/Server-Protokolls beschrieben, das diesen Anforderungen entspricht. Dieses Dokument enthält jedoch noch keine Angaben zur Codierung der Nachrichten, so dass es nicht als alleinige Basis für eine Implementierung dienen kann.

Den Vorgaben der Internet Engineering Steering Group (IESG) entsprechend, wurde nicht sofort mit der Spezifikation eines eigenständigen MIDCOM-Protokolls begonnen. Stattdessen wurde untersucht, ob die benötigte Funktionalität auf Basis eines bereits standardisierten Protokolls erbracht werden kann. In [RFC 4097] findet sich eine Bewertung der Protokolle SNMP [RFC 3411, u. a.], RSIP [RFC 3102, ff.] Megaco [RFC 3015], Diameter [RFC 3588] und COPS [RFC 3084] bzgl. ihrer Eignung für die MIDCOM-Architektur. Noch vor dem Abschluss dieser Vergleichsuntersuchungen wurde – teilweise aus politischen Gründen – das *Simple Network Management Protocol* (SNMP) Version 3 (SNMPv3) als Basis für das offizielle MIDCOM-Protokoll ausgewählt. Die MIDCOM-Protokollsemantik soll über eine so genannte *Management Information Base* (MIB) auf SNMP abgebildet werden. Somit soll ein Protokoll, das bisher in der Management-Plane angesiedelt war, jetzt für Signalisierungszwecke verwendet werden. Die Standardisierung dieser MIB ist bis heute (Stand: Mai 2007) noch nicht abgeschlossen; die Spezifikation liegt lediglich als Arbeitsdokument (Internet Draft) vor [98].

Der Entwurf eines eigenständigen MIDCOM-Protokolls war von der IESG nur für den nicht eingetretenen Fall vorgesehen, dass die Bewertung alle oben genannten Protokolle für ungeeignet befände. Dennoch wurde von einigen Mitgliedern der Arbeitsgruppe mit „SImple Middlebox COntrol Protocol“ (SIMCO) eine vergleichsweise einfache, direkte Binärcodierung der MIDCOM-Protokollsemantik spezifiziert und als experimenteller RFC veröffentlicht [RFC 4540].

4.4.2 Protokollinstanzen

In [RFC 3303] werden folgende an der MIDCOM-Architektur beteiligten Instanzen definiert:

End-Host Kommunikationsendpunkt der eigentlichen Nutzer-Anwendung, z. B. SIP-Telefone, FTP-Client/-Server, etc.

Middlebox Ein Netzelement, das auf dem Pfad zwischen den End-Hosts liegt und einen oder mehrere so genannte *Middlebox Services* bereitstellt. Zu diesen Diensten, die auf der Netzwerk- bzw. Transportschicht erbracht werden, können neben Paketfilterung und Adressumsetzungen (NAPT) auch z. B. *Intrusion Detection*, *Load Balancing*, oder *Tunneling* von IP-Paketen gehören.

MIDCOM Agent Ein vertrauenswürdiger Prozess, der am Anwendungsprotokoll beteiligt ist, Entscheidungen über die Konfiguration von Middlebox Services trifft und diese über das MIDCOM-Protokoll an die Middlebox signalisiert.

MIDCOM Agents können sowohl in Endpunkte (z. B. SIP User Agent) als auch in Transitsysteme (z. B. SIP B2BUA) des Anwendungsprotokolls integriert werden, sofern ein Vertrauensverhältnis zwischen diesem System und der gesteuerten Middlebox besteht (z. B. gleiche administrative Kontrolle).

MIDCOM PDP Von den MIDCOM Agents an eine Middlebox signalisierte Regel-Änderungen können von dieser ggf. zur Autorisierung an den optionalen MIDCOM Policy Decision Point (PDP) weitergeleitet werden. Das dazu zu verwendende Protokoll ist derzeit nicht spezifiziert.

Eine mögliche Anordnung dieser Instanzen in einem einfachen Szenario mit SIP-basierter Multimedia-Kommunikation wird in [RFC 3303] beschrieben (siehe Abbildung 4.6).

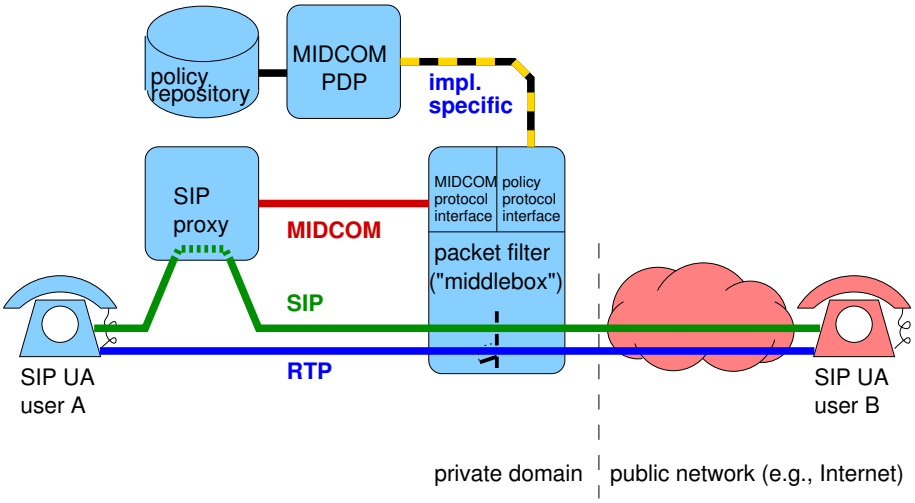


Abbildung 4.6: Die IETF MIDCOM-Architektur mit SIP nach [RFC 3303].
Der MIDCOM-Agent ist in eine SIP-Instanz integriert, die dort fälschlicherweise als SIP Proxy gekennzeichnet ist – korrekt wäre ein B2BUA (siehe Abschnitt 4.4.4).

4.4.3 Protokollsemantik

Die Grundprinzipien des MIDCOM-Protokolls sind in [RFC 3989] bzw. in dem derzeit in Arbeit befindlichen Nachfolgedokument [99] beschrieben. Demnach handelt es sich bei diesem Protokoll um ein transaktionsbasiertes Client-Server-Protokoll. Persistente Signalisier-Assoziationen, so genannte MIDCOM-Sessions, sollen beim Systemstart von den MIDCOM Agents hin zu den Middleboxes aufgebaut, authentisiert und dann ständig aufrechterhalten werden. So sollen unnötige Verzögerungen durch zusätzliche Handshakes beim Eintragen oder Löschen von Policy Rules vermieden werden. Die Architektur sieht eine $m : n$-Beziehung zwischen steuernden Agents und gesteuerten Middleboxes vor. Dies kann z. B. dazu verwendet werden, um alle Medien-Elemente am Rand eines Netzes durch einen zentralen SIP-Server, der oft als *Softswitch* (SSw) bezeichnet wird, zu steuern (siehe Abbildung 4.7).

Nachdem eine MIDCOM-Session mit Hilfe der dafür vorgesehenen Nachrichten aufgebaut und authentisiert wurde (siehe Abbildung 4.8(a)), kann diese für die Übertragung von Transaktionen und asynchronen Benachrichtigungen verwendet werden. Eine Transaktion besteht aus einer Anforderung, die vom MIDCOM Agent zur Middlebox gesendet wird, sowie aus einer positiven oder negativen Antwort. Die Zuordnung von Antworten zu Anforderungen erfolgt über Transaktions-Bezeichner (engl. *Transaction Identifier*, TID), die vom Agent eindeutig vergeben werden. Es wird zwischen *Configuration Transactions* und *Monitoring Transactions* unterschieden. Während mit ersteren Zustandsänderungen in der Middlebox angefordert werden können, dienen zweitere zur Abfrage dieser Zustände, ohne sie zu ändern. Eine Middlebox kann Agents, die eine MIDCOM-Session zu ihr unterhalten, auch unaufgefordert über Zustandsänderungen informieren, z. B. nach Ablauf eines Timers in der Middlebox. Dazu werden asynchrone Benachrichtigungen (engl. *Asynchronous Notifications*) versendet. Eine Übersicht über alle Anforderungs-Nachrichten, zu denen es i. d. R. jeweils eine positive und eine negative Bestätigungsnachricht gibt, sowie über die asynchronen Benachrichtigungen wird in Tabelle 4.1 gegeben.

Die *Configuration Transactions* dienen dazu, *Policy Rules* (siehe Abschnitt 2.3.2) zu reservieren, zu aktivieren, zu deaktivieren und wieder zu löschen (siehe Abbildung 4.8(b)). Deren Bedingungen beschreiben mit Hilfe von Parametern der IP- und der Transportschicht die Flows, auf die der Middlebox Service angewendet werden soll. Die genaue Bedeutung dieses Services hängt vom Typ der Middlebox ab; dieser wird beim Aufbau einer MIDCOM-Session von

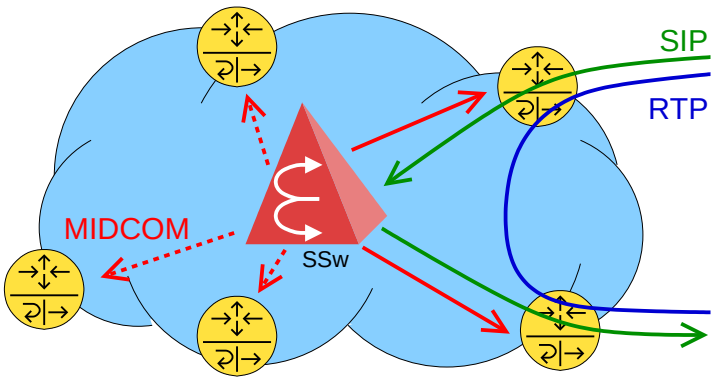


Abbildung 4.7: Zentraler Softswitch (SSw) steuert verteilte Paketfilter, hier: Transitnetz

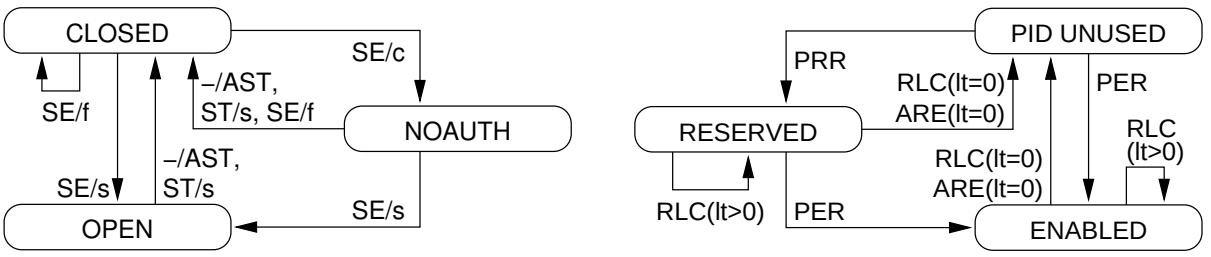
der Middlebox an den Agent signalisiert. Derzeit sind die Middlebox-Typen „Firewall“ (d. h. Paketfilter) und „Network Address and Port Translator“ (NAPT) spezifiziert.

Beim Anlegen einer Policy Rule wird dieser durch die Middlebox ein Bezeichner (*Policy Rule Identifier*, PID) zugewiesen. In der zur PER- oder PRR-Transaktion gehörenden Antwort wird dieser PID-Wert dem Agent mitgeteilt. Er muss vom Agent bei allen weiteren Anforderungen mitgesendet werden, die sich auf diese Regel beziehen, z. B. zum Löschen der Regel (siehe Abbildung 4.9).

Policy Rules werden in der Middlebox grundsätzlich durch einen so genannten *Soft State* repräsentiert, d. h. sie besitzen eine Gültigkeitsdauer (engl. *Lifetime*). Wird diese nicht rechtzeitig vor Ablauf vom besitzenden Agent durch eine entsprechende Transaktion verlängert, so wird die Policy Rule aus der Middlebox entfernt. Somit soll verhindert werden, dass z. B. Pinholes in einem Paketfilter unbegrenzt lange geöffnet bleiben, falls der MIDCOM-Agent unkontrolliert terminiert oder die MIDCOM-Session unterbrochen wird. Eine Policy Rule kann vom Agent gelöscht werden, indem er die Gültigkeitsdauer auf Null setzt.

4.4.4 Grundsätzliches Zusammenspiel mit SIP

Das grundsätzliche Zusammenspiel mit SIP wird bereits in der MIDCOM-Architekturbeschreibung [RFC 3303] skizziert. In dem dort dargestellten, einfachen Szenario (siehe Abbildung 4.6) dient als MIDCOM-Agent ein SIP-Server. Dieser bestimmt aus der SIP/SDP-Sitzungssignalisierung die Parameter der RTP-Medienströme und konfiguriert über das MIDCOM-Protokoll auf einem Paketfilter entsprechende Regeln, die die Medienströme passieren lassen. Dieser SIP-Server wird in [RFC 3303] fälschlicherweise als SIP Proxy dargestellt. In dem dargestellten Szenario kann es u. U. jedoch erforderlich sein, in die SIP-Signalisierung einzugreifen, z. B. zur Modifikation der SDP-Nachrichten, falls es sich bei der Middlebox um einen Adressumsetzer (NAPT) handelt, oder um die Multimedia-Sitzung abzubrechen, falls die benötigten Pinholes nicht geöffnet werden können (siehe Abschnitt 5.4). Diese Eingriffe in die SIP/SDP-Signalisierung überschreiten die in [RFC 3261] definierten Funktionen eines SIP Proxy. In einem jüngeren Dokument [99] wird deshalb für dieses Szenario die Verwendung eines SIP *Back-to-Back User Agent* (B2BUA) vorgesehen.



(a) Zustandsautomat einer MIDCOM-Session

(b) Zustandsautomat einer Policy Rule (nur erfolgreiche Transaktionen dargestellt)

Abbildung 4.8: Zustandsautomaten von MIDCOM-Session und Policy Rule

In Abbildung 4.10 wird ein zu dem Szenario in Abbildung 4.6 passendes Nachrichtensequenz-Diagramm dargestellt. Es kommt hier SIP in seiner Grundversion nach [RFC 3261] zum Einsatz und es wird der Gut-Fall dargestellt, d. h. der Verbindungsaufbau gelingt ohne Fehler. Sobald der B2BUA vom rufenden Teilnehmer die *INVITE*-Nachricht mit der angehängten SDP-Nachricht erhält, ermittelt er daraus die Parameter des Medienstroms in Rückwärtsrichtung, d. h. vom gerufenen zum rufenden Teilnehmer. Über eine *PER*-Transaktion wird sofort ein entsprechendes Pinhole im Paketfilter geöffnet, noch bevor die *INVITE*-Nachricht weitergeleitet wird. So wird sichergestellt, dass der rufende Teilnehmer evtl. auftretende *Early Media* empfangen kann, die noch vor dem Abschluss des Verbindungsaufbaus gesendet werden (z. B. Ansagetexte, Hörtöne, etc.). Die SIP-Signalisierung zum Rufaufbau wird „normal“ (vgl. Abschnitt 2.1.5.4) fortgesetzt, bis die *200 OK*-Nachricht vom gerufenen Teilnehmer empfangen wird, nachdem dieser den Ruf entgegengenommen hat. Mit einer zweiten *PER*-Transaktion wird ein Pinhole für den Medienstrom in Vorwärtsrichtung geöffnet; der Sprachkanal ist somit vollduplex durch die Firewall durchgeschaltet.

Das Verhalten des Systems im Fehlerfall, z. B. wenn ein Pinhole aufgrund von Ressourcenmangel, Kommunikationsproblemen der MIDCOM-Session oder Verstößen gegen Richtlinien des MIDCOM PDP nicht geöffnet werden kann, ist in [RFC 3303] nicht beschrieben. Diese Fälle werden in Abschnitt 5.4 näher untersucht.

Abweichend von der üblichen Darstellungsweise für Nachrichtensequenz-Diagramme [Z.120] sind die Pfeile in Abbildung 4.10 nach unten geneigt. Dadurch soll die zeitliche Verzögerung angedeutet werden, die durch den Transport der Nachrichten zwischen den Protokollinstanzen entsteht. Zusammen mit der lokalen Bearbeitungszeit δ in der Middlebox trägt die Verzögerung beim Transport der MIDCOM-Nachrichten zur Dauer R einer *PER*-Transaktion bei. Die Auswirkungen verschiedener Konfigurationen auf die Verzögerungen beim Aufbau einer neuen Multimedia-Sitzung, welche von den Teilnehmern als störend empfunden werden, sowie Möglichkeiten zur Verringerung werden in den folgenden Kapiteln untersucht.

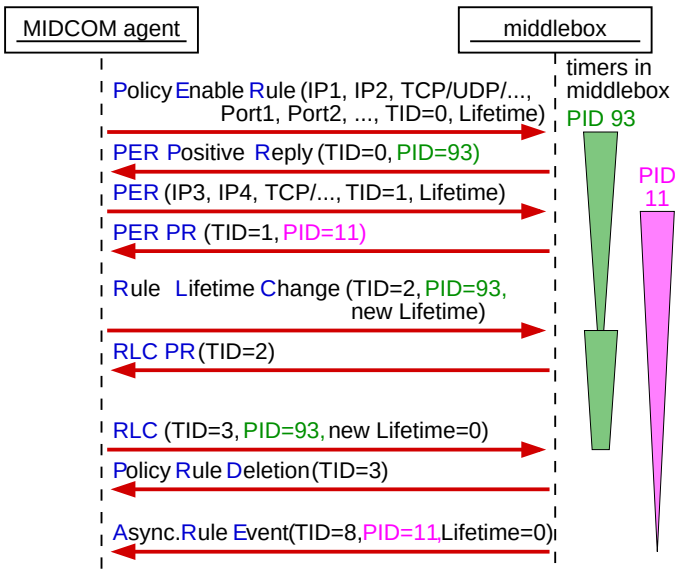


Abbildung 4.9: MIDCOM-Transaktionen erzeugen, verändern und löschen Policy Rules in der Middlebox

Tabelle 4.1: Nachrichten des MIDCOM-Protokolls nach [RFC 3989] bzw. SIMCO [RFC 4540]

MIDCOM	SIMCO	Bezeichnung	Funktion
Sitzungssteuerung			
SE	SE	Session Establishment	Aufbau einer neuen MIDCOM-Session
-	SA	Session Authentication	Beantwortung der Authentisierungs-Frage der Middlebox durch den Agent (Zustandsübergang NOAUTH–OPEN). Entspricht SE mit Parameter bei MIDCOM.
ST	ST	Session Termination	Beendigung einer MIDCOM-Session
AST	AST	Asynchronous Session Termination	Beendigung einer MIDCOM-Session durch die Middlebox
Regel-Verwaltung			
PRR	PRR	Policy Reserve Rule	Reservieren einer Adresse bei Adressumsetzern („NAPT address binding“)
PER	PER	Policy Enable Rule	Eintragen und Aktivieren einer Regel oder (nur MIDCOM) Aktivieren einer reservierten Regel
-	PEA	PER After Reservation	Aktivieren einer reservierten Regel
RLC	PLC	Policy Rule Lifetime Change	Festlegen einer neuen Gültigkeitsdauer für eine Regel; Löschen einer Regel durch neue Gültigkeitsdauer Null
PRL	PRL	Policy Rule List	Abfrage der IDs aller Regeln in der Middlebox (z. B. nach Umschaltung auf redundanten Agent im Fehlerfall)
PRS	PRS	Policy Rule Status	Abfrage von Informationen über eine bestimmte Regel
ARE	ARE	Asynchronous Policy Rule Event	Middlebox benachrichtigt Agent über Löschung einer Regel durch anderen Agent, nach Timer-Ablauf oder wegen Konflikt mit PDR
Sperren von Regeln (MIDCOM: nicht vorgesehen – SIMCO: optional)			
-	PDR	Policy Disable Rule	Eintragen und Aktivieren (!) einer „speziellen“ Regel, die das Eintragen „normaler“ Regeln (Pinholes) mit überlappenden Adress-Parametern blockiert bzw. schon vorhandene löscht
Gruppen-Verwaltung (MIDCOM: optional – SIMCO: nicht vorgesehen)			
GLC	-	Group Lifetime Change	Gültigkeitsdauer für alle Regeln einer Gruppe neu festlegen Zuordung einer Regel zu einer Gruppe erfolgt beim Anlegen der Regel (PER)
GL	-	Group List	Abfrage der IDs aller Gruppen
GS	-	Group Status	Abfrage von Informationen über eine bestimmte Gruppe

4.5 SIMCO

Das „Simple Middlebox Configuration Protocol“ (SIMCO) [RFC 4540] ist ein vergleichsweise einfaches Protokoll zur Pfad-entkoppelten Steuerung von Middleboxes. Es orientiert sich sehr eng an der abstrakten MIDCOM-Protokoll-Semantik nach [RFC 3989] und definiert eine binäre Codierung dieser Nachrichten. Aufgrund der historischen Entwicklung der jeweiligen IETF-Dokumente und dem Hinzufügen eines optionalen Protokollmechanismus unterscheiden sich die Nachrichten des abstrakten MIDCOM-Protokolls bzw. von SIMCO leicht; eine Übersicht mit Vergleich wird in Tabelle 4.1 gegeben. Da die Bedeutung der Nachrichten identisch ist, sind die in Abbildung 4.8 dargestellten Zustandsautmaten bis auf die Namen der Nachrichten für beide Protokolle gültig.

Die zu einer SIMCO-Session gehörenden Nachrichten werden über eine TCP- bzw SCTP-Verbindung transportiert, die vom Agent zur Middlebox aufgebaut wird. Sie kann mit TLS oder IPsec kryptographisch abgesichert werden. Abbildung 4.11 zeigt die prinzipielle Struktur einer SIMCO-Nachricht. Hierbei kommt so genanntes Type-Length-Value-Encoding (TLV) zum Einsatz. Jede SIMCO-Nachricht beginnt mit einem immer 4 Oktetts langen Nachrichtenkopf (engl. *header*). Die ersten beiden Oktetts kennzeichnen den Typ (*Type*) der Nachricht. Dieses Feld kann weiter unterteilt werden: Der *Basic Type* kann nur einen von vier definierten Codes annehmen, die den Bedeutungen *Request*, *Positive Reply*, *Negative Reply* oder *Asynchronous Notification* entsprechen. Entsprechend gibt der *Sub-Type* an, um welches Kommando bzw. um welche Antwort darauf es sich handelt. Es folgen das 2 Oktetts lange Längen-Feld (*Length*) und der einzige in jeder SIMCO-Nachricht vorhandene Wert (*Value*), der 4 Oktetts lange Transaktions-Bezeichner (TID). Das Längen-Feld gibt die Länge der auf den SIMCO-Header folgenden, optionalen Werte an. Je nach Typ der Nachricht ist die Angabe verschiedener Parameter vor-

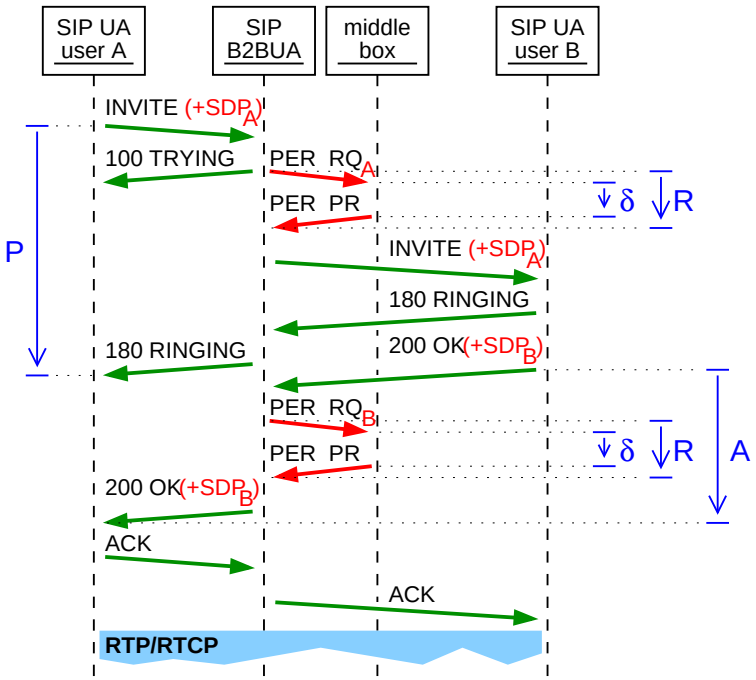


Abbildung 4.10: Zusammenspiel des IETF MIDCOM-Protokolls mit SIP: Gut-Fall

geschrieben oder optional. Diese so genannten *Attribute* sind ihrerseits wieder TLV-kodiert. In sehr frühen Entwürfen der SIMCO-Protokollspezifikation war die Verwendung eines textbasierten Nachrichtenformats vorgesehen, ähnlich wie es z. B. auch bei SIP verwendet wird. Dieses wurde zu Gunsten der TLV-Codierung aufgegeben, da diese ein eindeutiges Nachrichtenformat definiert, bei dem es keine Variationen z. B. bzgl. Groß/Kleinschreibung, Zahl von Leerzeichen, etc. geben kann. Dadurch sinkt der Aufwand beim Einlesen einer Nachricht (vgl. [30]).

Gegenüber dem abstrakten MIDCOM-Protokoll definiert SIMCO ein zusätzliches, optionales Konzept, das Sperren von Regeln mit Hilfe der *PDR*-Transaktion. Dies ist nicht zu verwechseln mit dem Löschen einer zuvor angeforderten Policy Rule, wofür auch bei SIMCO die *PLC*-Transaktion verwendet wird, um die neue Gültigkeitsdauer auf Null zu setzen. Hingegen werden mit Hilfe der *PDR*-Transaktion neue Zustandsinformationen in die Middlebox eingebracht, gewissermaßen eine „negative Regel“, die das Anlegen neuer Policy Rules mit *PER* verhindert, wenn diese überlappende Adressparameter haben. Bereits vorhandene Policy Rules, die in Konflikt mit einer neuen *PDR*-Transaktion stehen, werden von der Middlebox gelöscht; die betroffenen Agents werden darüber mit einer asynchronen Benachrichtigung informiert. Ein Anwendungsfall für diesen Mechanismus sind Systeme zur Erkennung von Angriffen (engl. *Network Intrusion Detection System*, NIDS). Solche Systeme analysieren den Verkehr in einem lokalen Netz, z. B. bzgl. Quell- und Zieladressen, Datenraten, oder verdächtigen Signaturen in der Nutzlast der Pakete (z. B. Viren). Als potenziell gefährlich eingestufte Datenströme, die von einem mutmaßlichen Angreifer von außerhalb in das lokale Netz gesendet werden, können vom NIDS mit Hilfe der *PDR*-Transaktion am Paketfilter blockiert werden. Die Prioritäten, die die von verschiedenen Quellen über verschiedene Mechanismen in den Paketfilter eingebrachten Regeln haben, werden in Tabelle 4.2 zusammenfassend dargestellt. Falls es zu Widersprüchen zwischen solchen Regeln kommt, entscheidet letztendlich die Regel mit der höchsten Priorität, wie mit einem Paket zu verfahren ist.

Tabelle 4.2: Prioritäten von SIMCO-Regeln verschiedenen Ursprungs

Priorität	Quelle	Konfiguration	Erlauben/Verbieten
hoch	Firewall-Administrator	statisch	beides möglich
...	SIMCO-Agent	dynamisch via *PDR*	Verbieten
...	SIMCO-Agent	dynamisch via *PER*	Erlauben
niedrig	Firewall-Administrator	statisch (*default policy*)	i. d. R. Verbieten

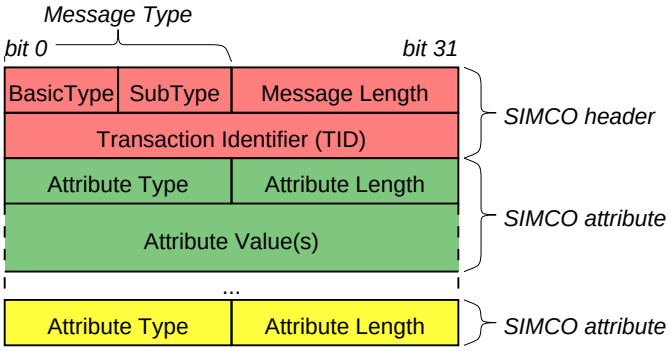


Abbildung 4.11: Struktur einer SIMCO-Nachricht: Type-Length-Value-Codierung

4.6 Die IETF NSIS-Architektur

Die IETF Arbeitsgruppe *Next Steps In Signaling* (NSIS) wurde im November 2001 gegründet, um eine aus zwei Schichten bestehende Architektur und dazugehörige Protokolle für Signalisier-Anwendungen zu spezifizieren. Das erste konkret betrachtete Anwendungsszenario ist die Reservierung von QoS-Ressourcen. Soweit dies sinnvoll erscheint, sollen Konzepte und Mechanismen von RSVP übernommen werden. NSIS kann somit als Weiterentwicklung von RSVP betrachtet werden und folgt grundsätzlich dem Pfad-gekoppelten Modell.

4.6.1 Grundsätzliche Architektur

Die NSIS-Architektur [RFC 4080] wurde erweiterbar geplant, so dass prinzipiell beliebige Zustandsänderungen mit Bezug zu einem Flow an Protokollinstanzen signalisiert werden können, die sich auf dem Pfad befinden, den dieser Flow durch das Netz nimmt. Unter einem Flow wird auch hier eine Menge von IP-Paketen bezeichnet, die zwischen zwei Endpunkten durch das Netz transportiert werden. Sie werden i. d. R. über ein 5-Tupel (Quell-IP-Adresse, Ziel-IP-Adresse, Transportschichtprotokoll-Bezeichner, Quell-Portnummer, Ziel-Portnummer) identifiziert; allerdings kann sich dieses 5-Tupel auf dem Pfad zwischen seinen Endpunkten auch ändern, z. B. wegen Adressumsetzern (NAPT). Die NSIS-Protokollarchitektur besteht aus zwei Protokollschichten (siehe Abbildung 4.12):

- einem gemeinsamen *NSIS Transport Layer Protocol* (NTLP) als Basis, der auch als „Messaging Layer“ bezeichnet wird, und
- mehreren, anwendungsspezifischen *NSIS Signaling Layer Protocols* (NSLP), z. B. zur
  - Dienstgüteunterstützung [100]
  - Steuerung von Paketfiltern und Adressumsetzern (NAPT) [101]
  - Konfiguration von Messgeräten zur Abrechnung bzw. Messung der Dienstgüte [102]

4.6.2 Protokollinstanzen

Folgende Instanzen sind an der NSIS-Signalisierung beteiligt:

Data Sender (DS) Sender des (unidirektionalen) Nutzdaten-Flows, für den entsprechende Zustände in den Middleboxes (z. B. Öffnen von Pinholes) erzeugt werden sollen.

Data Receiver (DR) Empfänger des (unidirektionalen) Nutzdaten-Flows, für den entsprechende Zustände in den Middleboxes (z. B. Öffnen von Pinholes) erzeugt werden sollen.

NSIS Initiator (NI) NSIS-Instanz, von der die NSIS-Kommandos ausgehen. Sie muss das NTLP und das gewünschte NSLP (z. B. NAT/FW NSLP) implementieren. Diese kann sich im selben Netzelement wie der Data Sender bzw. Data Receiver befinden (vgl. Abbildung 4.4(a)) oder auf einem anderen Netzelement, von dem bekannt ist, dass es auf dem Pfad des Flows liegt (vgl. Abbildung 4.4(b))

NSIS Responder (NR) NSIS-Instanz, die NSIS-Antwortnachrichten versendet. Sie muss das NTLP und das gewünschte NSLP (z. B. NAT/FW NSLP) implementieren. Sie kann sich bei dem Data Receiver, beim Data Sender oder in einem eigenen Netzelement befinden.

NSIS Forwarder (NF) NSIS-Instanz, die sich in der Kette der Middleboxes entlang des Pfades zwischen NSIS Initiator und NSIS Receiver befindet. Sie muss auf jeden Fall das NTLP implementieren. Wird auch das gewünschte NSLP (z. B. NAT/FW NSLP) implementiert, so können dessen Signalisiernachrichten bearbeitet werden und ggf. entsprechende Zustandsänderungen (z. B. Öffnen von Pinholes) gemacht werden; anderenfalls werden die Nachrichten transparent weitergeleitet.

4.6.3 Der NSIS „Messaging Layer“ GIST

Die NSIS-Transportschicht (NTLP) ist die gemeinsame Basis für alle NSIS-Anwendungen. Es handelt sich dabei um einen abstrakten Architektur-Block. Dieser kann durch das *General Internet Signaling Transport* (GIST)-Protokoll [103] implementiert werden, das in früheren, experimentellen Versionen als *General Internet Messaging Protocol for Signaling* (GIMPS) bezeichnet wurde. Die prinzipielle Aufgabe des NTLP bzw. GIST ist der effiziente und sichere Transport von Signalisiernachrichten zwischen Netzelementen auf dem Pfad eines Flows, die NSIS unterstützen. Dazu müssen Mechanismen bereit gestellt werden, mit denen der jeweilige NSIS-Nachbarknoten u. a. auf Basis der für die Verkehrslenkung verwendeten Routingtabellen ermittelt werden kann. Mit dem so genannten *Session Identifier* werden Bezeichner bereitgestellt, die auch dann Gültigkeit behalten, wenn der *Flow Identifier*, d. h. das den Flow identifizierende 5-Tupel geändert wird, z. B. durch Network Address and Port Translation (NAPT), oder Gateways zwischen IPv4- und IPv6-basierten Netzen. Durch Multiplexing-Mechanismen können die Nachrichten verschiedener NSLP über eine GIST-Assoziation transportiert werden. Falls ein Netzelement zwar NSIS und somit GIST, aber nicht den gewünschten NSLP unterstützt, so ist die GIST-Instanz für das transparente Weiterleiten dieser Nachrichten verantwortlich.

Der Transport der Signalisiernachrichten kann verbindungslos im so genannten *Datagram Mode* (D-Mode) oder verbindungsorientiert im *Connection Mode* (C-Mode) erfolgen. Je nach Signalisierlast zwischen zwei benachbarten Knoten und anderen Anforderungen können verschiedene Transportschichtprotokolle für den Transport von GIST verwendet werden, z. B. TCP, UDP, SCTP oder DCCP (vgl. Kapitel 6). Ein abschnittsweiser kryptographischer Schutz von NSIS-Nachrichten beim Transport zwischen Nachbarknoten kann mit IPsec oder TLS erfolgen.

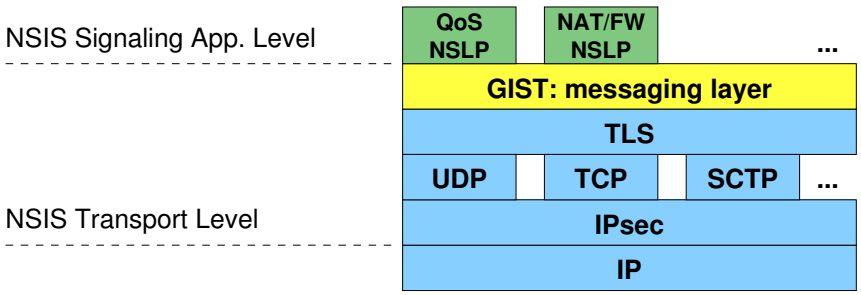


Abbildung 4.12: Die IETF NSIS-Architektur: Protokollstapel

Ebenfalls zum Umfang des NSLP bzw. GIST gehören Mechanismen zur Unterstützung von *Rerouting* (d. h. Änderung der Tabellen zur Verkehrslenkung), so genanntem asymmetrischen Routing (d. h. Datagramme laufen in der „Gegenrichtung“ zwischen den beiden Endpunkten über einen anderen Pfad) und von Endpunkt-Mobilität (z. B. im Zusammenspiel mit MobileIP [RFC 3344]). Die bei RSVP vorhandene Unterstützung für IP Multicast wurde hingegen entfernt, da sie nur sehr selten zum Einsatz kommt und erheblich zur Komplexität des Protokolls beiträgt [104]. Mechanismen, die es ermöglichen, dass die NSIS-Signalisiernachrichten den Pfad abschnittsweise verlassen und durch Protokollinstanzen geleitet werden, die nicht auf dem Pfad des gesteuerten Flows liegen („off-path NSIS“), wurden diskutiert [105], gehören derzeit (2006) aber nicht zum Standard.

4.6.4 Steuerung von Paketfiltern mit NSIS

Die Aufgabe des *NAT/Firewall NSLP* [101, 106] ist die dynamische Steuerung von Network Address (and Port) Translators (NAT/NAPT) und Paketfiltern (im Umfeld dieser Arbeitsgruppe als „Firewalls“ bezeichnet). Für das Ermitteln der Nachbarknoten entlang des Pfades und den Transport der Signalisiernachrichten wird GIST (s. o.) verwendet. Sowohl das NAT/FW NSLP als auch GIST verwenden Type-Length-Value-Encoding (TLV), die der Struktur von SIMCO-Nachrichten (siehe Abbildung 4.11) ähnelt. Mit Hilfe des NAT/FW NSLP werden Pinholes in Paketfiltern geöffnet oder *Address Bindings* bei NAPT angefordert. Ähnlich wie bei der MIDCOM-Architektur handelt es sich dabei um Soft States, d. h. diese Regeln werden nach Ablauf ihrer Gültigkeitsdauer entfernt, sofern diese nicht rechtzeitig verlängert wurde. Anders als bei MIDCOM können an Paketfilter sowohl erlaubende, als auch verbietende Regeln signalisiert werden, d. h. es sind Whitelist- und Blacklist-Konfigurationen möglich.

Das Protokoll definiert vier Nachrichten-Grundtypen: Mit der *CREATE*-Nachricht, die vom *NSIS Initiator* zum *NSIS Responder* gesendet wird, werden Regeln vorgemerkt, die einen Paket-Flow vom *Data Sender* zum *Data Receiver* ermöglichen. Der *NSIS Initiator* muss sich daher im selben Netzelement wie der *Data Sender* oder zumindest auf der selben Seite der zu überquerenden Middleboxes befinden. Die *CREATE*-Nachricht wird vom *NSIS Responder* mit einer positiven oder negativen *RESPONSE*-Nachricht beantwortet; bei einer positiven Antwort werden die vorgemerkten Regeln tatsächlich aktiviert, anderenfalls werden sie gelöscht. Mit Hilfe von Zustandsinformationen in den *NSIS Forwardern* wird die *RESPONSE*-Nachricht durch die selbe Kette von NSIS-Knoten zurückgeschickt, den die auslösende *CREATE*-Nachricht genommen hatte. Die notwendigen Regeln für einen Nutzdatenfluss in Gegenrichtung werden hingegen vollkommen unabhängig mit einem zweiten Paar von *CREATE*- und *RESPONSE*-Nachrichten signalisiert, um asymmetrisches Routing im Netz zu unterstützen. Ebenfalls mit der *CREATE*-Nachricht kann die Gültigkeitsdauer einer Regel verlängert oder eine Regel gelöscht werden, indem die Gültigkeitsdauer auf Null gesetzt wird.

In manchen Konfigurationen kann es notwendig sein, dass die NSIS-Signalisierung zunächst von der NSIS-Instanz ausgeht, die sich auf der Seite des *Data Receivers* befindet. Dies ist insbesondere dann der Fall, wenn dieser sich „hinter“ einem Adressumsetzer (NAPT) in einem Bereich mit nur lokal gültigen IP-Adressen [RFC 1918] befindet. Mit Hilfe einer *EXT*-Nachricht (external) und der dazugehörigen *RESPONSE*-Nachricht kann die Adresszuordnung (*Address Binding*) im Adressumsetzer angefordert werden, die benötigt wird, dass die von der Gegen-

seite gesandte *CREATE*-Nachricht und der darauffolgende Nutzdaten-Flow den *Data Receiver* überhaupt erreichen können. Die *EXT*-Nachricht kann vom Empfänger eines Flows auch dazu genutzt werden, diesen von einem vorgelagerten Paketfilter verwerfen zu lassen, z. B. wenn der Empfänger Ziel eines *Denial-of-Service-Angriffs* ist.

NOTIFY-Nachrichten können asynchron von jeder NAT/FW-Instanz erzeugt und versendet werden, z. B. wenn eine Regel nach Ablauf ihrer Gültigkeitsdauer gelöscht wird oder wenn eine Änderung in der Verkehrslenkung (dynamisches Routing) festgestellt wurde, die ein Signalisieren von Regeln an die jetzt neu auf dem Pfad liegenden Middleboxes erforderlich macht.

In Abbildung 4.13 ist das Öffnen eines Pinholes in einem Paketfilter in einem einfachen Szenario dargestellt. Über ein Protokoll zur Sitzungssignalisierung (z. B. SIP/SDP) handeln die beiden Endpunkte zunächst unabhängig von der späteren NSIS-basierten Firewall-Signalisierung die Parameter des Nutzdatenstroms (z. B. RTP) aus. Dieses Protokoll zur Sitzungssignalisierung kann unabhängig vom Pfad des späteren Datenstroms geführt werden, z. B. über Proxies. Prinzipiell wäre es auch denkbar, auf ein solches Protokoll gänzlich zu verzichten und die Endpunkte statisch zu konfigurieren; in diesem Fall würde es sich aber auch anbieten, den Paketfilter entsprechend statisch zu konfigurieren und somit auf die Firewall-Signalisierung zu verzichten. Nach dem Abschluss der Sitzungssignalisierung öffnet „User A“ mit Hilfe einer *CREATE*-Nachricht ein Pinhole, welches den darauf folgenden unidirektionalen Datenstrom durch den Paketfilter passieren lässt. Falls Daten bidirektional übertragen werden sollten, müsste mit einem zweiten Paar von *CREATE*/*RESPONSE*-Nachrichten ein Pinhole in Rückwärtsrichtung geöffnet werden. Hierzu würde „User B“ die Rolle des *NSIS Initiator* übernehmen.

4.6.5 Integration von NSIS- und SIP-Signalisierung

Firewalls an Grenzen von Sicherheitsdomänen wären praktisch wirkungslos, wenn jede Protokollinstanz im Netz das Öffnen beliebiger Pinholes anfordern könnte. Eine sichere Authentisierung und Autorisierung solcher Signalisiernachrichten ist daher unabdingbar. Prinzipiell kann das Zusammenspiel zwischen der SIP-basierten Sitzungssignalisierung und der NSIS-basierten Paketfiltersteuerung erfolgen, wie es im vorherigen Abschnitt beschrieben wurde. Die Kopplung findet in diesem Szenario nur in den Endpunkten statt. Dies setzt voraus, dass diesen im Einflussbereich des Nutzers befindlichen Multimedia-Endgeräten vertraut wird, dass sie keine „gefährlichen“ Pinholes im Firewall anfordern. Das NSIS-Framework sieht optional die Ver-

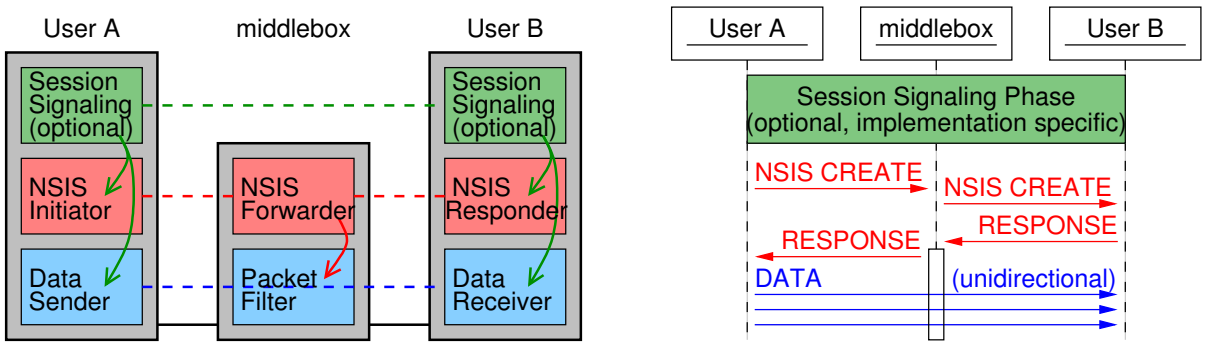


Abbildung 4.13: Steuerung eines Paketfilters mit NSIS in einem einfachen Szenario

wendung von IPsec oder TLS für einen abschnittsweisen kryptographischen Schutz der Nachrichten beim Transport zwischen Nachbarknoten vor (siehe Abschnitt 4.6.3). Insbesondere bei Szenarien mit mehreren Transit-Domänen kann es jedoch vorkommen, dass ein Firewall nicht allen *NSIS Forwardern* auf dem Pfad vertraut, dass sie die Nachrichten nicht verfälschen.

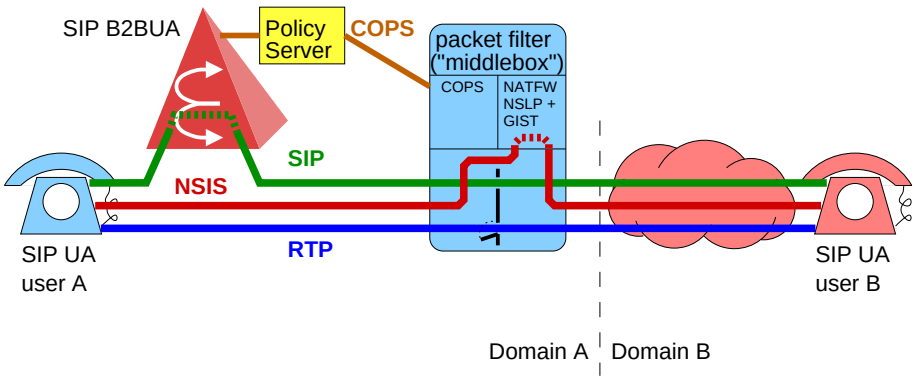
Aus diesen und anderen Gründen (siehe Abschnitt 3.1) kann es notwendig sein, dass ein Paketfilter vor dem Öffnen eines über NSIS angeforderten Pinholes überprüft, ob in den SIP-Servern der Domäne Zustandsinformationen über die fragliche Multimedia-Sitzung vorliegen. Ein Verfahren zur kryptographischen Authentisierung Pfad-gekoppelter Signalisierung und zur Kopplung an die Sitzungssignalisierung ist in [RFC 3521] für die QoS-Signalisierung mit RSVP beschrieben. Dieses Verfahren kann auch im Umfeld von NSIS verwendet werden. Die dazu benötigten zusätzlichen Nachrichtenformate werden derzeit standardisiert [107].

Die Grundidee dieses Verfahrens ist, dass zunächst der Rufaufbau mit SIP signalisiert wird. Dabei werden die SIP-Nachrichten durch mindestens einen SIP-Server (z. B. B2BUA) pro Domäne geleitet, unter Verwendung der Standard-Verfahren zur Verkehrslenkung von SIP-Nachrichten in der Anwendungsschicht (vgl. Abschnitte 2.1.5.8 und 3.5). Diese SIP-Server können die beteiligten Teilnehmer authentisieren und z. B. Zugriffskontrolllisten oder andere Maßnahmen zur Autorisierung des Rufes verwenden. Falls der Ruf nicht autorisiert werden kann, wird dies dem rufenden Teilnehmer über die SIP-Signalisierung mitgeteilt. Andernfalls werden kryptographische Autorisierungs-Token zur Autorisierung der Medienströme erzeugt. Dies kann entweder im SIP-Server selbst geschehen, oder in einem abgesetzten Autorisierungs-Server. Im zweiten Fall kann z. B. das *Common Open Policy Service Protocol* (COPS) [RFC 2748, RFC 3084] für die Kommunikation zwischen diesen beiden Systemen verwendet werden.

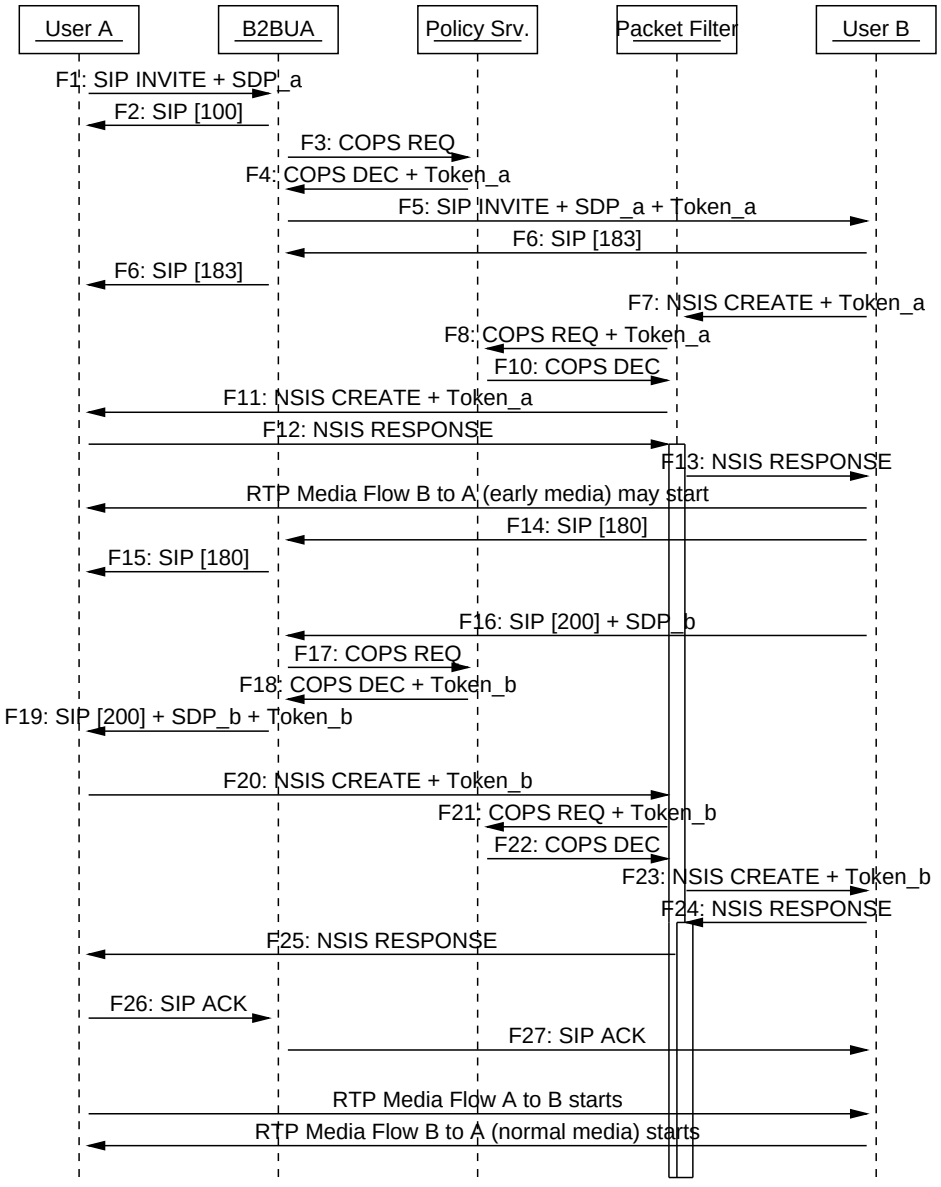
Die Autorisierungs-Token enthalten u. a. die aus dem SDP entnommenen Parameter der jeweils zu erlaubenden Medienströme, in Vorwärts- bzw. Rückwärtsrichtung. Sie werden vom SIP-Server an geeignete SIP-Nachrichten angehängt und erreichen somit die Multimedia-Endgeräte, die die Funktion des *NSIS Initiator* bzw. *NSIS Receiver* enthalten. Diese können die Tokens, ohne sie zu entschlüsseln, an die jeweiligen NSIS-Nachrichten anhängen. Wird eine NSIS-Nachricht, die ein solches Autorisierungs-Token beinhaltet, von einem Paketfilter empfangen, so muss dieser die Authentizität des Tokens prüfen. Prinzipiell kann dies lokal geschehen, indem eine digitale Signatur auf Basis des öffentlichen Schlüssels des Autorisierungs-Servers oder auf Basis eines geteilten Geheimnis geprüft wird. Alternativ kann das Token zur Prüfung an den Autorisierungs-Servers zurückgeschickt werden. Dazu kann z. B. ebenfalls COPS verwendet werden. Ist diese Prüfung erfolgreich, so kann der Paketfilter davon ausgehen, dass die mit NSIS „angekündigten“ Flows Medienströme sind, die zu einem Ruf gehören, der von den SIP-Servern autorisiert wurde. Dieses Verfahren wird anhand der in Abbildung 4.14 dargestellten, einfachen Netztopologie und dem dazugehörigen Nachrichtensequenz-Diagramm illustriert.

4.7 Zusammenfassung und Fazit

In diesem Kapitel wurden Firewall-Architekturen für Multimedia-Anwendungen klassifiziert. Werden Signalisier- und Medienkomponente in getrennten Netzelementen platziert, wird ein Steuerprotokoll benötigt. Die beiden Grundverfahren, die Pfad-entkoppelte und die Pfad-gekoppelte Signalisierung, sowie die entsprechenden IETF-Protokolle wurden vorgestellt.



(a) Szenario



(b) Nachrichtensequenz-Diagramm

Abbildung 4.14: Kopplung von NSIS an die SIP-Signalisierung zur Autorisierung von Pinholes

5 Untersuchung eines Netzübergangs mit Pfad-entkoppelter Firewall-Steuerung

5.1 Kriterien und Methoden zum Vergleich von Firewall-Architekturen

Bei der Bewertung einzelner Komponenten oder ganzer Architekturen von Kommunikationsnetzen wird häufig zwischen *funktionalen* und *nichtfunktionalen Anforderungen* unterschieden. Erstere beziehen sich auf das beim Betrieb des Systems erzielte (bzw. erwartete) Ergebnis, i. d. R. die Ermöglichung einer bestimmten Art der Kommunikation. Nichtfunktionale Aspekte befassen sich hingegen mit Randbedingungen, die eingehalten werden müssen, oder mit Charakteristika des Systems, die über die eigentliche Funktion hinausgehen, z. B. Leistungsfähigkeit, Skalierbarkeit, Sicherheit, Mobilitätsunterstützung, Zusammenarbeit mit anderen Systemen, etc.. Standardisierungsgremien und der Gesetzgeber können weitere Randbedingungen liefern. Die Abgrenzung zwischen diesen Kategorien ist allerdings nicht sehr scharf. Beispielsweise ist die Erhöhung der Netzsicherheit der primäre Zweck einer Firewall. Hierzu gehört auch, dass der Fluss der legitimen Datenströme durch die Firewall auch dann nicht beeinträchtigt wird, wenn die Firewall das maximal angenommene Angriffsverkehrs-„Angebot“ abwehren muss, was eine entsprechende Leistungsfähigkeit der Firewall-Elemente voraussetzt.

Insbesondere die funktionalen Anforderungen haben einen direkten Einfluss auf den Systementwurf. Dieser kann in verschiedenen Darstellungsformaten und Detaillierungsstufen dokumentiert werden, z. B. als textuelle Beschreibung mit Skizzen. Eine Alternative ist die Nutzung einer Spezifikationssprache (z. B. SDL [Z.100]) zur Erstellung eines Systemmodells, das so detailliert ist, dass daraus u. U. automatisiert Programmcode synthetisiert oder mit Hilfe formaler Methoden (z. B. *Model Checking* [108]) die Erfüllung bestimmter Anforderungen gezeigt werden kann, z. B. Unmöglichkeit von Systemverklemmungen. Der Detaillierungsgrad der Beschreibung ist ein kritischer Punkt: ist er zu gering, besteht die Gefahr, dass das mögliche Auftreten bestimmter Situationen, z. B. Fehlerfälle, übersehen wird und das Systemverhalten dort unspezifiziert ist. Die Erstellung eines sehr detaillierten Modells erfordert hingegen genaue Kenntnis der Umgebung, z. B. mögliche Fehlercodes des zugrundeliegenden Betriebssystems. Dadurch kann der Aufwand schnell den einer „richtigen“, ggf. prototypischen Implementierung erreichen. Nur eine solche Implementierung und dazugehörige Tests schafft jedoch letztendliche Gewissheit, dass eine Idee tatsächlich realisierbar ist. Im Umfeld der Standardisierung der Internet-Protokolle ist es üblich, auf textuelle Beschreibungen und sehr einfache Skizzen zurückzugreifen. Durch Interoperabilitätstests an voneinander unabhängig entstandenen Implementierungen wird geprüft, ob die zugrundeliegenden Dokumente realisierbar, vollständig und unmissverständlich sind.

Im Bereich der Netzsicherheit sind formale Methoden besonders im Bereich der Kryptoanalyse und bei der Analyse von Authentisierungsprotokollen verbreitet, z. B. die „BAN-Logic“ [109], eine so genannte *Belief Logic*. Die Komplexität gesamter Netze stellt jedoch eine ernsthafte Herausforderung dar. Zur Analyse realer Systeme wird deshalb sehr häufig eine systematische, aber nicht „beweisende“ Vorgehensweise gewählt, etwa mit Hilfe von Fragenkatalogen und Checklisten wie z. B. [110].

Für die Leistungsbewertung von Kommunikationsnetzen existiert ein breites Methodenspektrum. Messungen können am realen System im Wirkbetrieb durchgeführt werden, sofern ein solches bereits existiert und zugänglich ist. Einzelne Komponenten oder Prototypen, die auf den wesentlichen Funktionsumfang reduziert bzw. um experimentelle Funktionen erweitert wurden, können in einer Laborumgebung aufgebaut werden; für Messungen müssen dort ggf. bestimmte Aspekte des umgebenden Netzes emuliert werden, z. B. das Teilnehmerverhalten mit Hilfe von Lastgeneratoren oder Verzögerungen beim Nachrichtentransport mit einem WAN-Emulator. Es ist auch möglich, Modelle des Untersuchungsgegenstandes zu entwickeln, und diese z. B. mit ereignisgesteuerter Simulation oder analytischen Berechnungen zu untersuchen. Dies gestattet auch die Betrachtung von Systemen, die mit den vorhandenen Techniken und Ressourcen (noch) nicht hergestellt werden können, oder die Untersuchung an Arbeitspunkten, die im Labor-Maßstab nicht realisiert werden können. Ferner ist die Untersuchung von Effekten in sehr großen oder sehr kleine Zeitskalen leichter möglich, sowie die Isolation einzelner Effekte, die am realen System nur von anderen (Stör-)Größen überlagert gemessen werden könnten. Jedoch besteht auch hier mit zunehmendem Abstraktionsgrad der Modelle die Gefahr, dass wesentliche Effekte versehentlich vernachlässigt werden; desweiteren ist es u. U. schwierig, realistische Werte für die Parameter eines Modells zu finden. Die Wahl einer geeigneten Untersuchungsmethode hängt somit stark vom Untersuchungsziel, sowie von der Verfügbarkeit von Implementierungen und Parametern ab.

In dieser Arbeit sollen Architekturen verteilter Firewalls in IP-Telefonie-Plattformen bezüglich zweier großer Themenkomplexe untersucht werden. Die Untersuchungen zu funktionalen und sicherheitsrelevanten Aspekten der Firewall-Architekturen behandeln neben dem Zusammenspiel mit SIP vor allem die Frage, ob auch bei komplexen Netztopologien ein authentisiertes Öffnen und – nach dem Ende der Multimedia-Sitzung – ein zeitnahes Schließen von Pinholes in allen Firewalls auf dem Medienpfad möglich ist. Zur Untersuchung dieser Fragestellungen wurde zunächst eine prototypische Implementierung des SIMCO-Protokolls erstellt, mit der auch an einem Interoperabilitätstest teilgenommen wurde. Die Erweiterung des Betrachtungswinkels auf eine netzweite Sicht und auf andere Architekturen erfolgt mit Hilfe von Fragenkatalogen.

Im Bereich der Leistungsbewertung soll besonders auf den Beitrag der Firewall-Signalisierung zur Verzögerung des Verbindungsaufbaus eingegangen werden, d. h. auf den Rufverzug bzw. Meldeverzug, je nach der genauen Ausgestaltung des Systems. Messungen am SIMCO-Prototypen werden durch Modelle und Analysen wesentlicher Mechanismen und Effekte in der Transportschicht ergänzt; die Ergebnisse werden auf verschiedene Ende-zu-Ende-Szenarien extrapoliert. Nur am Rande soll auf die eigentliche Zugriffskontrolle in der Medienkomponente eingegangen werden, da deren Leistungsfähigkeit von sehr vielen implementierungsspezifischen Parametern abhängt und unabhängig von den betrachteten Steuerarchitekturen ist.

5.2 Implementierung eines SIMCO-Prototypen

Im Rahmen dieser Arbeit wurde eine protypische Implementierung eines Protokolls zur Pfad-entkoppelten Firewall-Steuerung erstellt, um sowohl Tests bzgl. funktionaler Aspekte, als auch Messungen zur Leistungsfähigkeit durchführen zu können. In diesem Abschnitt wird die Architektur der Testumgebung und der Software vorgestellt; für die detaillierte Beschreibung der Implementierung einzelner Komponenten wird auf entsprechende Dokumente verwiesen.

5.2.1 Auswahl von SIMCO

Für den Prototypen wurde das „Simple Middlebox Configuration Protocol“ (SIMCO) ausgewählt, welches ein vergleichsweise einfaches Protokoll ist, das sich sehr eng an der abstrakten Semantik des IETF MIDCOM-Protokolls [RFC 3989] orientiert. Zum Zeitpunkt der Auswahl war die Spezifikation als Arbeitsdokument („Internet Draft“) veröffentlicht. Anders als z. B. bei der MIDCOM MIB [98] war die SIMCO-Spezifikation schon recht weit fortgeschritten und stabil; mittlerweile liegt sie als [RFC 4540] vor.

5.2.2 Anforderungen an den SIMCO-Prototypen und implementierte Module

Ziel der Arbeiten war die Erstellung einer voll funktionsfähigen, prototypischen Implementierung des SIMCO-Protokolls in C bzw. C++, für den Einsatz auf UNIX-Betriebssystemen, insbesondere Linux [111]. Dieses Betriebssystem, das unter der *GNU Public License* (GPL) [112] steht und dessen Quellcode offen liegt und modifiziert werden darf, erfreut sich zusammen mit seinem Paketfilter-Modul *Netfilter* [113] großer Beliebtheit bei der Absicherung der Internet-Zugänge kleinerer und mittlerer lokaler Netze. Dennoch wurde darauf geachtet, so wenig wie möglich Linux-spezifische Systemaufrufe zu verwenden. An Stellen, wo solche Aufrufe nicht zu vermeiden waren (z. B. Einbringen der Paketfilter-Regeln in den Betriebssystem-Kern, siehe Abschnitt 5.2.3.4), wurden diese in einfach abschaltbare bzw. austauschbare Module gekapselt. Somit konnten große Teile des Quellcodes auch auf das kommerzielle Betriebssystem Solaris des Herstellers Sun Microsystems portiert werden.

Da die Protokollspezifikation zum Beginn der Arbeiten zwar in ihren Grundzügen stabil war, aber noch kleine Änderungen, z. B. bei der Kodierung der Nachrichten, abzusehen waren, musste darauf geachtet werden, dass die Implementierung leicht an solche Änderungen des Standards angepasst werden kann. Auch experimentelle Erweiterungen des Protokolls sollen möglichst leicht integrierbar sein. Dies betrifft insbesondere die Schnittstelle zur darunterliegenden Transportschicht (siehe Abschnitt 5.2.3.5). Die Unterteilung des Gesamtsystems in Module wurde derart gewählt, dass jedes dieser Module einzeln in einem eigenen Unter-Projekt implementiert und getestet werden konnte.

Da neben Experimenten zu funktionalen Aspekten auch Messungen zur Leistungsfähigkeit, insbesondere zu den auftretenden Verzögerungen, gemacht werden sollten, musste neben einer „normalen“ Client-Instanz auch ein Lastgenerator entworfen und implementiert werden. Alle relevanten Module wurden mit zuschaltbaren Funktionen zur Ausgabe von Zeitstempeln versehen. Die bereits erwähnte starke Modularisierung kommt auch diesen Messungen zu Gute, da

so einzelne Effekte durch das Abschalten bestimmter Module oder durch das gezielte Durchmessen eines Moduls isoliert werden können.

Die wichtigsten Komponenten der SIMCO-Testumgebung sind ein SIMCO-Server, der auf der Middlebox zum Einsatz kommt, ein SIMCO-Client, der für Untersuchungen zum Zusammenspiel mit SIP in einen SIP Back-to-Back User Agent (B2BUA) integriert wurde, sowie ein SIMCO-Client, der SIMCO-Requests synthetisch erzeugt und somit als Lastgenerator für Messungen verwendet werden kann. Zur Untersuchung der Nachrichten „auf dem Draht“ wurde der Protokollanalysator *Ethereal* [114] (mittlerweile in *Wireshark* umbenannt) um einen so genannten *Dissector* zum Dekodieren und Anzeigen vom SIMCO-Nachrichten erweitert.

5.2.3 SIMCO-Server

5.2.3.1 Architektur

Abbildung 5.1 zeigt die prinzipielle Architektur des SIMCO-Servers. Dieser besteht aus zwei Prozessen im so genannten *User Space*, die untereinander und mit dem Betriebssystem-Kern (*Kernel Space*) kommunizieren. Einer dieser beiden Prozesse, der „Haupt-Prozess“, beinhaltet die SIMCO-Protokollmaschine, kommuniziert mit den SIMCO Agents und verwaltet die von diesen signalisierten Regeln. Dieser wird in den folgenden Abschnitten beschrieben. Der „Hilfs-Prozess“, welcher in Abschnitt 5.2.3.4 beschrieben wird, ist relativ einfach aufgebaut; er dient dazu, die Paketfilterregeln in den Betriebssystem-Kern einzubringen.

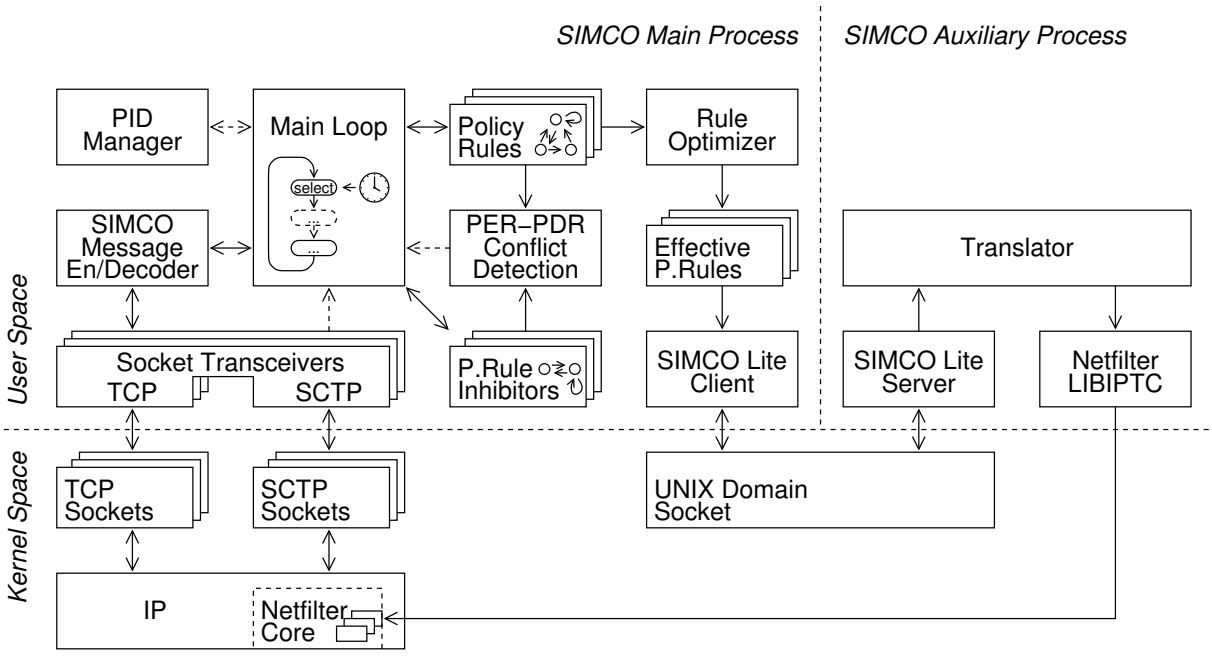


Abbildung 5.1: Architektur des SIMCO-Servers

5.2.3.2 Datenstrukturen

Die wichtigste Datenstruktur des SIMCO-Servers ist die Liste der derzeit gültigen Policy Rules. Diese Liste wird über den Regel-Bezeichner (PID) indiziert. Die Einträge enthalten die Address-Parameter der zugehörigen Pinholes, den Bezeichner des SIMCO Agents, der die Regel eingetragen hat, und den auf die Systemzeit des Servers umgerechneten Zeitpunkt, an dem die Regel gelöscht werden wird, sofern nicht vorher eine Verlängerung oder Verkürzung der Lebensdauer signalisiert wurde. Die in dieser Liste gespeicherten Adressparameter haben keinen unmittelbaren Einfluss auf die Paketfilterung. Dazu müssen sie erst in die Netfilter-Tabellen im Betriebssystem-Kern kopiert werden (siehe Abschnitt 5.2.3.4). Diese redundante Datenhaltung ermöglicht eine einfachere Portierung der Software auf andere Betriebssysteme; ferner können so die tatsächlich zur Filterung eingesetzten Regeln optimiert werden, z. B. durch Umsortieren häufig benutzter Regeln oder durch Zusammenfassen überlappender Regeln.

Die mit Hilfe der *PDR*-Transaktion signalisierten verbotenen Adressbereiche, in denen keine Policy Rules eingetragen werden dürfen, werden in einer entsprechenden *Policy Rule Inhibitors*-Liste gehalten.

Der *PID Manager* verwaltet die Vergabe von Regel-Bezeichnern. Von ihm kann beim Erstellen einer neuen Regel die Zuweisung eines solchen Bezeichners angefordert werden.

Der SIMCO-Server soll mehrere SIMCO-Assoziationen zu verschiedenen SIMCO Agents unterstützen. Es soll dabei einfach möglich sein, neben dem im Standard vorgesehenen TCP auch andere Transportschichtprotokolle zu verwenden (siehe Kapitel 6). In einer Liste werden daher Referenzen auf Instanzen der von einer abstrakten *Socket Transceiver*-Klasse abgeleiteten Klassen gehalten, die den Endpunkt je einer SIMCO-Assoziation repräsentieren (siehe Abschnitt 5.2.3.5).

5.2.3.3 Kontrollfluss

Im SIMCO-Server treten folgende wichtige Aufgaben und Ereignisse auf:

- Auf- bzw. Abbau einer neuen eingehenden Verbindung von einem SIMCO Agent
- Bearbeiten einer empfangenen SIMCO-Nachricht, Senden der Antwort
- Behandlung von Konflikten zwischen *PER*- und *PDR*-Nachrichten
- Prüfen der Gültigkeitsdauern von Regeln
- Optional Optimierung des „effektiven“ Regelsatzes durch Zusammenfassen überlappender Regeln (vgl. [115])
- Einbringen des „effektiven“ Regelsatzes in den Paketfilter im Betriebssystem-Kern

Das Einbringen der Regeln in den Betriebssystem-Kern muss mit so genannten *root*-Rechten geschehen, die anderen Aktionen benötigen diese erhöhten Privilegien nicht. Diese Aufgabe

wurde daher in einen privilegierten „Hilfsprozess“ ausgelagert (siehe Abschnitt 5.2.3.4). Alle anderen Aufgaben können in einem oder mehreren Prozessen mit geringeren Privilegien ausgeführt werden, z. B. unter einer „Pseudo User ID“.

Viele der oben genannten Aufgaben lassen sich im Prinzip parallel abarbeiten, so dass es zunächst sinnvoll erscheint, den „Hauptprozess“ mit vielen nebenläufigen *Threads* zu implementieren, die jeweils einen vergleichsweise einfachen Kontrollfluss hätten. Eine evtl. auftretende Blockierung eines Threads hätte so auch einen geringen Einfluss auf die anderen Aufgaben des Servers. Bei genauerer Analyse zeigt sich jedoch, dass praktisch jede der Aufgaben auf die zentral gehaltenen Listen der Policy Rules bzw. Policy Rule Inhibitors zugreifen muss. Um Inkonsistenzen zu vermeiden, würden bei einer solchen Lösung Mechanismen benötigt, um diese Zugriffe zu serialisieren (z. B. Semaphoren), welche die nebenläufige Ausführung der Aufgaben einschränken würden. Da jede der beschriebenen Aufgaben nur recht wenig Rechenzeit benötigt und ein längeres Blockieren z. B. von Ein-/Ausgabeoperationen recht einfach vermieden werden kann, wurde der SIMCO-Server als ein Prozess mit einer großen „Hauptschleife“ entworfen. Dadurch konnte der Implementierungsaufwand reduziert werden [116]. Der prinzipielle Kontrollfluss des SIMCO-Servers ist in Abbildung 5.2 als SDL-Prozessgraph des „Hauptprozess“ dargestellt.

Eine zentrale Rolle spielt hierbei der Systemaufruf *select()*. Diesem Systemaufruf wird eine Liste von Ein-/Ausgabekanälen übergeben; die Ausführung des Programms wird dann solange unterbrochen, bis Daten aus einem Empfangspuffer gelesen bzw. in einen Sendepuffer geschrieben werden können, oder bis eine maximale Wartezeit (engl. *Timeout*, in Abbildung 5.2 dargestellt als Timer „T1“, der auf den Wert „A“ initialisiert wird) überschritten wurde. Durch den *Timeout*-Mechanismus kann sichergestellt werden, dass auch in Phasen ohne Kommunikation bestimmte Aufgaben regelmäßig erledigt werden, z. B. das Überprüfen der Gültigkeitsdauern von Policy Rules.

Bei Messungen hat es sich herausgestellt, dass das Eintragen einer großen Zahl von Regeln in den Kern des Linux-Betriebssystems recht lange dauern kann (siehe Abschnitt 5.5). Deshalb ist in der Abbildung eine Variante des Programms zu sehen, bei der die Bestätigung zu einer *PER*-Nachricht schon versendet wird, nachdem die Parameter der angeforderten Policy Rule geprüft wurden, aber noch bevor die entsprechende Regel an dem „Hilfsprozess“ zum Eintragen in den Betriebssystem-Kern weitergeleitet wurde. Somit können ggf. auch mehrere in kurzer Folge angeforderte Regeln auf einmal eingetragen werden. Falls es nach einer erfolgreichen Prüfung der Regel im „Hauptprozess“ zu einem Problem beim Eintragen im „Hilfsprozess“ kommt, wird dies an den „Hauptprozess“ zurückgemeldet, welcher den entsprechenden SIMCO-Agent mit Hilfe einer *ARE*-Nachricht informiert. Mit Hilfe der Variablen „T3“ und „T4“ wird im Kontrollfluss nach Abbildung 5.2 eine einfache Regelung zur Überlastabwehr implementiert: Ein Eintragen von Regeln in den Betriebssystem-Kern kann frühestens nach einer Wartedauer erfolgen, die um einen Faktor „C“ mal länger ist als die Dauer des vorangegangenen Eintrage-Vorgangs.

5.2.3.4 Anbindung des Linux Netfilter

Für das Eintragen von Regeln in die jeweiligen Paketfilter-Module existiert bei UNIX-Betriebssystemen keine standardisierte Software-Schnittstelle (API). Während der Hauptteil des SIM-

CO-Servers recht einfach auf verschiedenen UNIX-Varianten installiert werden kann (neben Linux wurde er auch erfolgreich unter Sun Solaris betrieben), wird für jedes Betriebssystem ein spezielles Modul zum Einbringen der Regeln in die Paketfilter-Tabellen benötigt. Das in diesem Abschnitt beschriebene Modul des SIMCO-Servers sowie die in Abschnitt 5.5 beschriebenen Messungen zu seiner Leistungsfähigkeit beziehen sich daher ausschließlich auf das Betriebssystem Linux (Version 2.6) mit dem Paketfilter-Modul „Netfilter“. Für Solaris wurde bisher kein

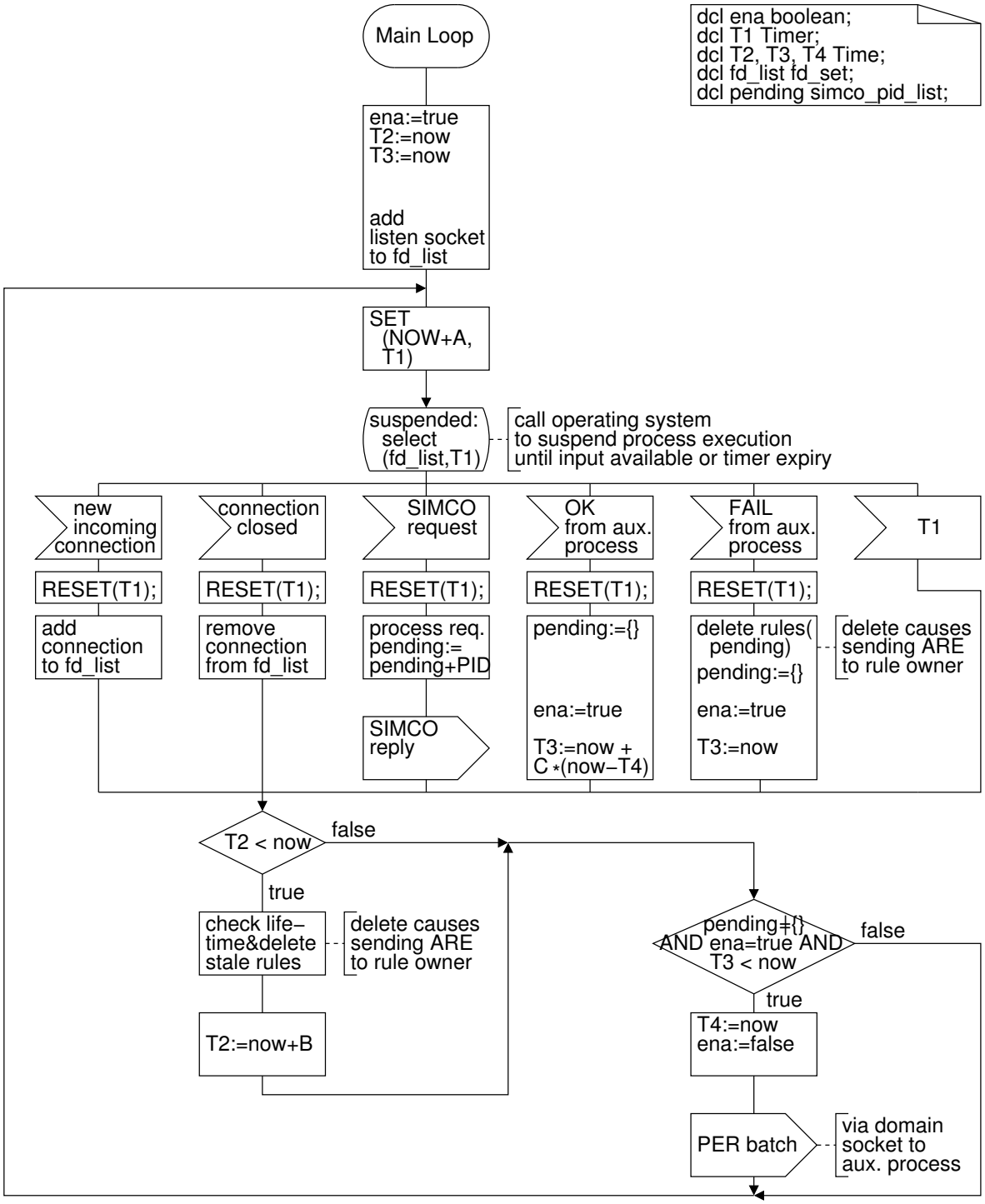


Abbildung 5.2: Kontrollfluss des SIMCO-Servers

solches Modul implementiert, d. h. unter diesem Betriebssystem kann der Server zwar SIMCO-Nachrichten verarbeiten und Regeln verwalten, eine tatsächliche Filterung von Paketen auf Basis dieser Regeln findet jedoch nicht statt.

Um die Netfilter-Tabellen im Linux-Kern ändern zu können, muss ein Prozess mit erhöhten Privilegien (*root*-Rechte) ausgestattet sein. Damit ist der Prozess jedoch auch zu sehr vielen anderen weit gehenden Aktionen berechtigt, z. B. Konfigurationsänderungen an anderen Subsystemen oder Lesen und Modifizieren von Daten anderer Prozesse. Die anderen Funktionen des SIMCO-Servers benötigen hingegen keine erhöhten Privilegien. Trotz sorgfältiger Vorgehensweise bei der Programmierung kann nicht ausgeschlossen werden, dass der SIMCO-Server Implementierungs-Schwachstellen enthält, ähnlich wie dies z. B. über Implementierungen von VoIP-Signalisierprotokollen berichtet wurde [60]. Zum Schutz der Integrität des Firewall-Elements vor Kompromittierung durch Angreifer sollte daher so wenig wie möglich Code des SIMCO-Servers mit erhöhten Privilegien ausgeführt werden. Aus diesem Grund wurde das Eintragen der Regeln in den Betriebssystem-Kern in einen privilegierten „Hilfsprozess“ ausgelagert, während alle anderen Aufgaben vom „Hauptprozess“ übernommen werden, welcher mit den normalen Privilegien eines „Pseudo Users“ ausgeführt wird. Ein weiterer Vorteil dieser Aufgabenteilung ist, dass das Eintragen der Regeln von der Bearbeitung der SIMCO-Nachrichten entkoppelt werden kann.

Vor dem Entwurf dieses „Hilfsprozesses“ wurde die Schnittstelle zum Eintragen von Paketfilter-Regeln in den Linux-Kern untersucht und Messungen zur Verzögerung bei Änderungen am Regelsatz gemacht [117]. Demzufolge hängt die Dauer eines solchen Vorgangs erheblich von der Zahl der Regeln ab, die sich vor bzw. nach der Änderung im Kern befinden. Dagegen ist es nur von untergeordneter Bedeutung, ob eine einzelne Regel hinzugefügt bzw. gelöscht wird, oder ob der komplette Regelsatz durch eine ähnlich große Menge anderer Regeln ersetzt wird. Dieses Verhalten liegt darin begründet, dass bei jedem Änderungsvorgang zunächst die komplette Regel-Liste aus dem Betriebssystem-Kern in den entsprechenden Prozess kopiert wird. Mit Hilfe der Bibliothek *libiptc* wird diese auf schnelle Vergleichsvorgänge beim Filtern von Paketen optimierte Datenstruktur in eine Darstellung transformiert, welche aus verketteten Listen besteht und einfache Änderungen am Regelsatz erlaubt. Nachdem die Änderungen vorgenommen wurden, erfolgt eine entsprechende Rückübersetzung und ein Kopieren in den Linux-Kern. Falls sich schon vor der Änderung eine gewisse Zahl von Regeln in der Liste befunden haben, können diese Kopier- und Übersetzungsvorgänge deutlich mehr Rechenzeit in Anspruch nehmen als z. B. das eigentliche Hinzufügen einer einzelnen Regel.

Vor diesem Hintergrund wurde die Schnittstelle zwischen dem „Hauptprozess“ und dem „Hilfsprozess“ des SIMCO-Servers derart gestaltet, dass jedes mal die komplette Liste aller Pinholes übertragen wird und die so genannte *Chain* im *Netfilter*, in der die Regeln gespeichert werden, komplett überschrieben wird. Verglichen mit einer Lösung, bei der nur die Änderungen übertragen werden, konnte die Schnittstelle so einfacher gehalten werden; da es sich um eine rechnerinterne Kommunikation handelt, sind Leistungsengpässe hier nicht zu befürchten. Diese Übertragung wird vom „Hauptprozess“ ausgelöst, z. B. nachdem über SIMCO neue Pinholes angefordert wurden, oder nachdem die Lebensdauer eines Pinholes abgelaufen ist. Da das Eintragen insbesondere bei einer großen Zahl von Regeln eine gewisse Zeit in Anspruch nimmt, sorgt der in Abschnitt 5.2.3.3 beschriebene Mechanismus dafür, dass dieser Vorgang nicht zu oft ausgelöst wird.

Als Kanal zwischen den beiden Prozessen wird ein *UNIX Domain Socket* verwendet, welcher den verbindungsorientierten, fehlergesicherten Transport eines kontinuierlichen Datenstroms zwischen Prozessen eines UNIX-Systems zur Verfügung stellt. Über diesen wird ein einfaches Protokoll transportiert, welches nur eine Transaktion kennt. Auf eine *BATCH*-Anforderung folgt eine Liste von Pinholes, die die bisher vorhandene Liste im Betriebssystem-Kern ersetzen soll; der „Hilfsprozess“ antwortet mit einer positiven oder negativen Bestätigung. Für die Übertragung der Parameter der Pinholes folgt der *BATCH*-Nachricht ein Bündel (engl. *Batch*) von Nachrichten, die in ihrer Struktur der *PER*-Nachricht von SIMCO entsprechen. Deshalb ist dieses Protokoll in Abbildung 5.1 als „SIMCO Lite“ dargestellt. Gegenüber SIMCO gilt für dieses Protokoll eine ganze Reihe von weiteren, vereinfachenden Annahmen: Der „Hilfsprozess“ muss nur mit einer anderen Instanz kommunizieren. Da sich beide im selben Netzelement befinden, kann auf eine kryptographische Sicherung der Verbindung verzichtet werden. Es sind nur erlaubende Regeln vorhanden, d. h. Konflikte zwischen *PER*- und *PDR*-Nachrichten müssen schon im „Hauptprozess“ aufgelöst werden. Der „Hilfsprozess“ verwaltet keine Regel-Gültigkeitsdauern; dementsprechend muss der „Hauptprozess“ den kompletten Regelsatz neu schreiben, wenn dort das Ablaufen der Gültigkeitsdauer einer Regel festgestellt wurde.

5.2.3.5 Anbindung an die Transportschicht

So genannte *Sockets* repräsentieren in UNIX-Betriebssystemen den *Service Access Point* (SAP) einer Transportschichtverbindung. Die *Socket Transceiver*-Klasse und ihre abgeleiteten Klassen *TCP Transceiver* und *SCTP Transceiver* bilden eine Abstraktionsschicht, die das Senden und Empfangen von Nachrichten (d. h. Datenblöcken) ermöglicht und dabei die Unterschiede zwischen TCP und SCTP verbirgt. Bei SCTP erfolgt beim Senden hier die Auswahl eines *Streams* (siehe Abschnitt 6.4).

Im *TCP Transceiver* wird nach Übergabe einer Nachricht an die TCP-Instanz stets ein sofortiges Leeren des Sendepuffers und Setzen der *PSH-Flags* (push) in den TCP-Segmenten angefordert, um die nachrichtenorientierte Kommunikation von SIMCO auf den bytestromorientierten Transport von TCP abzubilden. Bei TCP kann es – insbesondere unter hoher Last – dennoch dazu kommen, dass die Systemaufrufe zum Senden bzw. Empfangen weniger Daten als angefordert entgegennehmen bzw. liefern. Teile einer noch nicht vollständig gesendeten bzw. empfangenen SIMCO-Nachricht werden in diesem Fall im *TCP Transceiver* zwischengespeichert.

Da es sich bei SIMCO um ein Type-Length-Value-kodiertes Protokoll handelt, werden keine aufwändigen Funktionen zum Erzeugen bzw. Einlesen der Nachrichten benötigt. Stattdessen kann einfach ein Bereich des Hauptspeichers versendet oder eingelesen werden, welcher von einer Datenstruktur (*struct*) belegt wird, die dem jeweiligen Nachrichtenformat entspricht [116].

5.2.4 SIMCO-Client im SIP B2BUA

Um Experimente zu funktionalen Aspekten der MIDCOM-Architektur durchführen zu können, wurde ein SIMCO-Client in den SIP Back-to-Back User Agent (B2BUA) von Vovida [118] integriert [119]. Auf dabei entdeckte Probleme beim Zusammenspiel von SIP und MIDCOM wird, soweit diese von grundsätzlicher Natur sind, in Abschnitt 5.4 eingegangen.

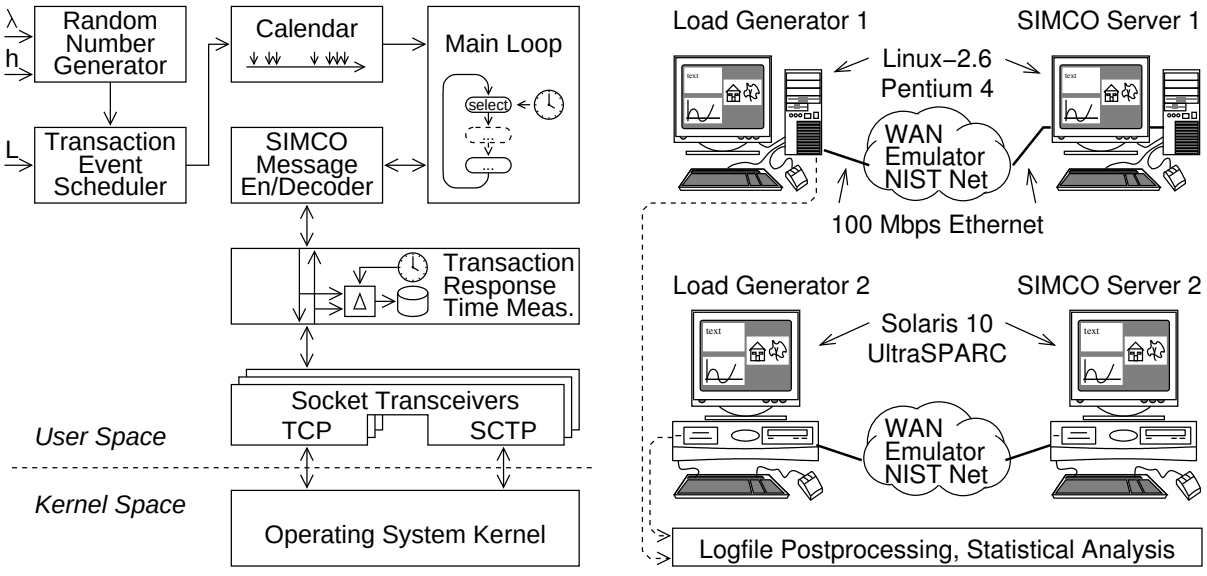
5.2.5 SIMCO-Lastgenerator

Zur Messung von Transaktions-Antwortzeiten des SIMCO-Servers und zur Quantifizierung der Verzögerungen beim Transport von SIMCO-Nachrichten über verschiedene Transportschichtprotokolle wurde ein SIMCO-Lastgenerator entworfen und implementiert. Dieser erzeugt SIMCO-Transaktionen auf Basis eines statistischen Modells, welches den Gesamtverkehr nachbildet, der von einer sehr großen Zahl von Teilnehmern ausgeht.

5.2.5.1 Architektur, Datenstrukturen und Kontrollfluss

Analog zur Struktur des SIMCO-Servers wurde auch der Lastgenerator als ein Prozess mit einer großen Hauptschleife entworfen (siehe Abbildung 5.3(a)) und implementiert [116]. Ein parametrisierbarer Zufallszahlengenerator (siehe Abschnitt 5.2.5.2) bestimmt jeweils die Verbindungsdauer und die Zwischenankunftszeit (engl. *Interarrival Time*, IAT) bis zum nächsten Aufbau einer Verbindung. Aus diesen Werten werden die Zeitpunkte bestimmt, an denen SIMCO-Transaktionen zum Eintragen, Aufrechterhalten und Löschen eines Pinholes benötigt werden. Diese werden im so genannten Kalender, einer nach den Zeitpunkten sortierten Liste gespeichert. Mit Hilfe des *select()*-Systemaufrufs wird in jedem Schleifendurchlauf die Ausführung des Prozesses angehalten, bis der nächste Zeitpunkt im Kalender erreicht ist oder bis eine Antwort-Nachricht vom SIMCO-Server empfangen wird, die verarbeitet werden muss.

Die Antwortzeit der *PER*-Transaktion *R* hat einen negativen Einfluss auf die Verbindungsaufbau-Verzögerung und damit auf die vom Teilnehmer empfundene Dienstgüte. Daher wird sie gemessen und protokolliert als die Zeitdifferenz vom Übergeben der fertig formatierten Anfrage-Nachricht an die Transportschichtprotokollinstanz im Betriebssystem-Kern bis zum Erhalt der entsprechenden Antwort-Nachricht.



(a) SIMCO-Lastgenerator (b) Testumgebung

Abbildung 5.3: Architektur des SIMCO-Lastgenerators und Testumgebung für Messungen

5.2.5.2 *Parametrisierung*

Die von einer gegen unendlich strebenden Zahl von Teilnehmern erzeugten Verbindungsanforderungen können als Poisson-Prozess modelliert werden, sofern das Verhalten einzelner Teilnehmer nicht vom Verhalten anderer Teilnehmer oder dem Zustand des Netzes (z. B. Blockierung) beeinflusst wird [E.731, 120]. Dieser Prozess ist gekennzeichnet durch negativ-exponentiell verteilte Zwischenankunftszeiten mit Mittelwert d_C, was einer mittleren Rate der Verbindungsanforderungen $\lambda_C = d_C^{-1}$ entspricht. Auch die Verbindungsdauer in Telefonnetzen wird häufig als negativ-exponentielle Verteilung mit einem Mittelwert h und einer Verteilungsdichtefunktion $f(T) = \frac{1}{h}\exp(-\frac{T}{h})$ modelliert [14].

Einer solchen Modellierung folgend, fordert der SIMCO-Lastgenerator mit einer konfigurierbaren mittleren Rate λ_u das Öffnen einzelner Pinholes an, welche für eine ebenfalls konfigurierbare mittlere Dauer h geöffnet bleiben sollen. Dabei sind sowohl die Zwischenankunftszeiten als auch die Gültigkeitsdauern der Pinholes negativ-exponentiell verteilt.

Die Zahl der Transaktionen, die zum Öffnen, Aufrechterhalten und Schließen eines Pinholes benötigt werden, hängt von der Dauer ab, die das Pinhole geöffnet bleiben soll. Wie aus Abbildung 4.9 ersichtlich ist, wird ein Pinhole mit einem *PER*-Request geöffnet und mit einem *PLC*-Request mit Parameter 0 wieder geschlossen. Da der entsprechende Zustand im SIMCO-Server mit einer Zeitüberwachung versehen ist (Soft State), sendet der Lastgenerator in Zeitabständen von L periodisch *PLC*-Requests, um die Gültigkeitsdauer rechtzeitig vor Ablauf zu verlängern. Somit werden für ein Pinhole, welches für eine Zeitdauer T geöffnet sein soll, insgesamt

$$n(T) = 2 + \lfloor \tfrac{T}{L} \rfloor \tag{5.1}$$

SIMCO-Transaktionen benötigt. Mit der angenommenen Verteilungsdichtefunktion der Gesprächsdauer (s. o.) kann die mittlere Rate λ_T der SIMCO-Transaktionen berechnet werden:

$$\lambda_T = \lambda_u \cdot \int_0^\infty n(T) \cdot f(T)\, \mathrm{d}T = \lambda_u \left(2 + \frac{1}{e^{L/h} - 1}\right) . \tag{5.2}$$

Werden Pinholes mit einer mittleren Rate λ_u geöffnet und für die mittlere Dauer h offen gehalten, sind im Mittel $U = \lambda_u \cdot h$ Pinholes gleichzeitig geöffnet. Da für eine Multimedia-Sitzung i. d. R. $u = 2$ Pinholes benötigt werden (eines je Richtung), entspricht dies einer Anzahl $m = \frac{U}{u}$ gleichzeitiger Sitzungen, die mit einer mittleren Rate $\lambda_C = \frac{m}{h} = \frac{\lambda_u}{u}$ aufgebaut werden. Daraus ergibt sich folgender Zusammenhang zwischen der Rate neuer Multimedia-Sitzungen und der Zwischenankunftszeit der SIMCO-Transaktionen:

$$d = \frac{1}{\lambda_T} = \frac{1}{u\,\lambda_C} \left(2 + \frac{1}{e^{L/h} - 1}\right)^{-1} . \tag{5.3}$$

Die zu einer Multimedia-Sitzung gehörenden Pinholes werden i. d. R. zu unterschiedlichen Zeitpunkten während des Sitzungsaufbaus geöffnet. Das (bzw. die) Pinhole für Medienströme in Rückwärtsrichtung wird bereits vor Beginn der Rufphase geöffnet, jenes für die Vorwärtsrichtung erst vor Beginn der Gesprächsphase (siehe Abbildung 4.10). Das Schließen erfolgt zeitgleich am Ende der Sitzung. Zur Vereinfachung des Lastgenerators wurde diese Korrelation nicht berücksichtigt; stattdessen werden Pinholes mit einer mittleren Rate von $\lambda_u = u \cdot \lambda_C$ und

voneinander unabhängigen Gültigkeitsdauern angefordert. Pinholes, die u. U. am Anfang einer Verbindung nur kurzzeitig für Early Media geöffnet werden, bleiben unberücksichtigt.

Eine wichtige Größe für die Planung und Dimensionierung von Netzen ist die Anzahl der Teilnehmer, deren Verkehr von einer Firewall mit gegebener Kapazität gefiltert werden kann. Als Anhaltspunkt hierfür kann ein M/G/n/q-Verlustsystem mit endlicher Quellenzahl q betrachtet werden. Dort steht der verarbeitete Verkehr Y mit dem angebotenen Verkehr A in der Beziehung $A = (q-Y)\alpha h$ [14], wobei α die Ankunftsrate einer inaktiven Quelle ist. Diese Rate entspricht hier dem Kehrwert der mittleren Dauer der Pause zwischen zwei Verbindungen eines Teilnehmers, d. h. es wird davon ausgegangen, dass jeder Teilnehmer in der von der Firewall geschützten Domäne einen mittleren Anteil $\rho = \frac{h}{h+\alpha^{-1}}$ der Zeit für Sitzungen mit Teilnehmern in anderen Domänen nutzt. Wenn mit $n \geq q$ genausoviele oder mehr Bedieneinheiten als Quellen zur Verfügung gestellt werden, können keine Verluste von Anforderungen auftreten, d. h. es gilt $Y = A$. Daraus folgt $q = A\frac{1+\alpha h}{\alpha h} = A\frac{1}{\rho}$ und mit $A = \lambda_C h$ weiter $q = \lambda_C h\frac{1}{\rho} = \frac{m}{\rho}$. So kann die Anzahl q der Teilnehmer bestimmt werden, die laut diesem Modell im Mittel m gleichzeitige Multimedia-Sitzungen und die dazugehörige SIMCO-Transaktionsrate λ_T verursachen, sofern das Netz so dimensioniert ist, dass keine Verluste auftreten. Bei der Dimensionierung von Telefonnetzen wird für ρ häufig ein Wert zwischen ca. 0.05 und 0.1 angenommen.

Negativ-exponentielle Verteilungen von Zufallsvariablen, die bei der Modellierung, Berechnung oder Simulation von Kommunikationssystemen häufig für Zwischenankunftszeiten oder Belegungsdauern angenommen werden, beinhalten auch beliebig kleine Zufallswerte. Solche können jedoch nur in Systemmodellen auftreten, bei denen Effekte wie Verarbeitungsdauern, Übertragungsdauern oder der Systemtakt zumindest teilweise vernachlässigt werden; in realen, mikroprozessorgesteuerten Systemen können sie hingegen praktisch nie beobachtet werden. Auch der hier beschriebene SIMCO-Lastgenerator erzeugt an seiner Schnittstelle zum Betriebssystem (oder gar an der Ethernet-Schnittstelle des Rechners) keine genau negativ-exponentielle Verteilung, da mit dem Erzeugen der nächsten SIMCO-Nachricht frühestens begonnen werden kann, nachdem die vorangegangene Nachricht fertig gestellt wurde. Somit kann es zu einem gewissen Verzug zwischen dem vom Zufallszahlengenerator bestimmten und dem tatsächlichen Sende-Zeitpunkt kommen. Auf die Messung der Transaktions-Antwortzeiten hat dies jedoch keinen störenden Einfluss, da dabei die Zeitdifferenz vom tatsächlichen Übergeben der fertig formatierten Anfrage-Nachricht an die Transportschichtprotokollinstanz im Betriebssystem-Kern bis zum Erhalt der entsprechenden Antwort-Nachricht erfasst wird. In die dazugehörigen analytischen Berechnungen geht nur der Mittelwert der Zwischenankunftszeit ein, wohingegen keine Annahmen über die Verteilungsfunktion gemacht wird.

5.2.5.3 Erkennung von Überlast

Wird die vom Lastgenerator zu erzeugende Transaktionsrate zu weit erhöht, kann das Gesamtsystem überlastet werden. Engpässe können dabei im Lastgenerator selbst und im SIMCO-Server entstehen, aber auch beim Nachrichtentransport, z. B. wenn aufgrund einer hohen Paketverlustwahrscheinlichkeit im Netz zu viele Übertragungswiederholungen stattfinden. Die beiden letzten Fälle machen sich aus Sicht des Lastgenerators dadurch bemerkbar, dass sich der Sendepuffer der Transportschichtverbindung füllt und evtl. gewartet werden muss, bevor neue Nachrichten an den Betriebssystem-Kern übergeben werden können. Solange die mittle-

re Bearbeitungs- und Wartezeit geringer ist als die gewünschte mittlere Zwischenankunftszeit, können zeitweilige Verzögerungen in Folge einzelner, sehr geringer Zwischenankunftszeiten (s. o.) wieder ausgeglichen werden. Anderenfalls gerät die Abarbeitung des Kalenders im Vergleich zur Realzeit immer weiter in Verzug; überschreitet die Differenz einen Schwellwert, wird die Messung mit einem entsprechenden Vermerk in der Protokoll-Datei abgebrochen.

5.2.6 Übersicht über die Testumgebung

Die Experimente zu funktionalen Aspekten der MIDCOM-Architektur – insbesondere zum Zusammenspiel mit SIP – wurden in einer Testumgebung durchgeführt, die dem Szenario aus der MIDCOM-Architekturbeschreibung [RFC 3303] entspricht (siehe Abbildung 4.6). Neben dem SIMCO-Server und dem in einen SIP B2BUA integrierten SIMCO-Client (siehe Abschnitt 5.2.4) kamen dabei Linux-basierte SIP-Softphones „KPhone“, sowie Hardphones „Snom 190“ zum Einsatz.

Zur Quantifizierung der durch die Firewall-Steuerung verursachten Verzögerungen des Verbindungsaufbaus wurden mehrere Messungen unternommen, um verschiedene Effekte zu isolieren.

Um den Einfluss des Transportschichtprotokolls und seiner Parametrisierung auf die Verzögerung beim Nachrichtentransport zu bestimmen, wurde mit einem SIMCO-Lastgenerator die Antwortzeit von Transaktionen mit einem SIMCO-Server gemessen. Dabei wurde die Netfilter-Anbindung deaktiviert; d. h. der SIMCO-Server empfängt und verarbeitet die SIMCO-Nachrichten, trägt die entsprechenden Pinholes aber nicht in die Paketfilter-Tabellen im Betriebssystem-Kern ein. Je ein Paar von Lastgenerator und Server wurden auf zwei Rechnern mit Intel Pentium 4-Prozessor, 2.8 GHz Taktfrequenz und dem Betriebssystem Linux-2.6.16, bzw. auf zwei Sun Blade 100 mit UltraSPARC IIe-Prozessor, 500 MHz Taktfrequenz und dem Betriebssystem Solaris 10 installiert (siehe Abbildung 5.3(b)). Diese Rechner sind jeweils über 100 Mbps Fast Ethernet-Segmente und einem Router verbunden, auf dem der WAN-Emulator NISTNet [121] installiert ist. Diese Software emuliert die in Weitverkehrsnetzen auftretenden Verzögerungen und ggf. Paketverluste. Bei allen Messungen waren die übertragenen Datenraten viel geringer als die Übertragungsrate der Ethernet-Segmente, so dass durch diese keine Beeinflussung der Messungen zu befürchten war. Dies wurde durch Kontrollmessungen bestätigt. Eine Dokumentation aller Versionsstände der verwendeten Komponenten kann in [3] gefunden werden. Um eine Beeinflussung der Messungen zum Nachrichtentransport durch das Bearbeiten der SIMCO-Nachrichten und die Verwaltung der Policy Rules im SIMCO-Server zu vermeiden, wurden weitere Messungen mit einem sehr einfachen Test-Protokoll gemacht [2], die wie erwartet keine nennenswerten Abweichungen zeigten.

In einer unabhängigen Versuchsanordnung wurde die Verzögerung beim Einbringen von Paketfilter-Regeln in den Betriebssystem-Kern von Linux untersucht. Der dafür zuständige „Hilfsprozess“ wurde auf einem Rechner mit Intel Pentium 4-Prozessor, 2.8 GHz Taktfrequenz installiert und unabhängig vom „Hauptprozess“ des SIMCO-Servers durchgemessen, indem er direkt an einen lokalen Lastgenerator geschaltet wurde. Ferner wurden in dieser Messreihe die Verzögerungen und Paketverluste beim Filtern von IP-Verkehr durch die Firewall erfasst.

5.3 SIMCO Interop-Event

Es existieren mindestens drei voneinander unabhängig entstandene Implementierungen des SIMCO-Protokolls. Eine davon stammt von den Haupt-Autoren der SIMCO-Spezifikation [122]; die zweite entstand im Zuge dieser Arbeit und wird in dem vorliegenden Dokument beschrieben. Eine dritte SIMCO-Implementierung wird in Firewall-Produkten kommerziell vertrieben [123].

Während eines so genannten *Interop-Events* wurde das korrekte Zusammenspiel von Protokollinstanzen der beiden zuerst genannten Implementierungen getestet [9]. Dieser Test deckte einige kleine Kompatibilitätsprobleme auf, die auf nicht eindeutige Formulierungen in der damals als Arbeitsentwurf vorliegenden SIMCO-Spezifikation zurückzuführen waren. Die betroffenen Textpassagen wurden in neueren Versionen des Dokuments verbessert. Die Implementierungen konnten noch während der Sitzung im Labor entsprechend angepasst werden, so dass die Interoperabilität demonstriert werden konnte.

5.4 Untersuchung des Zusammenspiels von MIDCOM und SIP

Auch wenn es noch einige andere Protokolle zur IP-basierten Außenbandsignalisierung gibt (z. B. RTSP [RFC 2326]), ist SIP-basiertes VoIP das am häufigsten genannte Anwendungsbeispiel für die IETF MIDCOM-Architektur. Dennoch existiert keine detaillierte Spezifikation für das Zusammenspiel zwischen SIP und MIDCOM; nur im Dokument zu den Grundprinzipien der MIDCOM-Architektur [RFC 3303] wird das grundsätzliche Zusammenspiel am Beispiel eines erfolgreichen Verbindungsaufbaus recht abstrakt illustriert (siehe Abschnitt 4.4.4).

Um darüber hinaus gehende praktische Erfahrungen sammeln zu können, wurde das in dem o. g. Dokument beschriebene, einfache Szenario (siehe Abbildung 4.6) mit SIMCO als Signalisierprotokoll in einer Testumgebung aufgebaut. Das Hauptaugenmerk lag dabei auf der Integration des SIMCO-Clients in den *SIP Back-to-Back User Agent* (B2BUA). Hierfür wurde der B2BUA von Vovida [118] ausgewählt. Dabei handelt es sich um eine SIP-Protokollinstanz, die *Call stateful* ist, d. h. anders als bei einem SIP Proxy werden Zustandsinformationen bzgl. laufender Multimedia-Sitzungen gehalten. Die Hauptanwendung, für die dieser B2BUA entwickelt wurde, ist das Bereitstellen von Benutzerkonten oder Telefonkarten mit vorausbezahltem Guthaben (engl. *Prepaid Account* bzw. *Prepaid Calling Card*). Dazu wird der rufende Teilnehmer beim Verbindungsaufbau authentisiert; die Verbindung wird vom B2BUA abgebaut, sobald das vorausbezahlte Guthaben des rufenden Teilnehmers aufgebraucht ist.

Diese Prepaid-Funktionen berücksichtigen ausschließlich die Verbindungsdauer, nicht aber die Art der Medienströme. Daher musste zunächst ein SDP-Parser entworfen werden, um die für die Firewall-Steuerung benötigten Informationen über die Medienströme aus den Signalisiernachrichten gewinnen zu können. Desweiteren musste ein SIMCO-Client entworfen und in den B2BUA integriert werden, der mehrere Pinholes gleichzeitig verwalten kann [119]. Beim Einpassen in den Kontrollfluss des B2BUA musste beachtet werden, dass der SIMCO-Client auch dann Aktionen ausführen muss, wenn gerade keine Verbindungen auf- oder abgebaut werden, z. B. Verlängern von Gültigkeitsdauern aktivierter Pinholes.

Beim Feinentwurf wurden einige Probleme identifiziert, die von grundsätzlicher Bedeutung für das Zusammenspiel von SIP mit MIDCOM und anderen Architekturen zur Firewall-Steuerung sind. Bei der folgenden Diskussion dieser Probleme und möglicher Lösungsansätze soll hingegen nicht auf Aspekte eingegangen werden, die spezifisch für den Vovida B2BUA sind oder die sich aus einer anderen (komplexeren) Netztopologie ergeben würden.

5.4.1 Behandlung von MIDCOM-Fehlerfällen in SIP

Bei der Verwendung von MIDCOM/SIMCO muss jederzeit damit gerechnet werden, dass das Eintragen eines Pinholes fehlschlägt oder dass ein bereits eingetragenes Pinhole gelöscht wird. Ersteres kann z. B. bei Ressourcenknappheit in der Middlebox, bei Abbruch der Verbindung zwischen MIDCOM-Client und -Server oder beim Verstoß gegen eine Richtlinie des MIDCOM PDP (vgl. Abschnitt 4.4.2) auftreten. Die vorzeitige Löschung eines bereits aktivierten Pinholes kann z. B. von einem *Network Intrusion Detection System* mit Hilfe der PDR-Transaktion angefordert werden. Da in diesen Fällen keine Medienströme fließen können, sollte die zugehörige Multimedia-Sitzung „sauber“ beendet bzw. gar nicht erst aufgebaut werden.

5.4.1.1 Fehlerfälle während einer bestehenden Multimedia-Sitzung

Falls die zu einer vollständig aufgebauten Multimedia-Sitzung gehörenden Pinholes vorzeitig geschlossen werden, reißt der Fluss der Medienströme ab. Der B2BUA, welcher über das Schließen der Pinholes z. B. mit einer MIDCOM/SIMCO *ARE*-Nachricht (Asynchronous Rule Event) informiert wird, sollte die Multimedia-Sitzung beenden. Dazu kann der B2BUA je eine SIP *BYE*-Nachricht an die beiden Teilnehmer senden.

Damit die beiden Teilnehmer nicht glauben, der jeweils andere Teilnehmer habe einfach mitten im Gespräch aufgelegt, sollte bei diesem Verbindungsabbruch eine aussagekräftige Fehlermeldung mitgesendet werden, damit die Teilnehmer dieses netzinterne Problem von einem normalen Ende der Verbindung unterscheiden können. Dies ist mit Mechanismen der SIP-Kernspezifikation [RFC 3261] jedoch nicht zu erreichen.

Ein mögliche Lösung ist die Verwendung des *Reason Header Field* [RFC 3326], einer optionalen SIP-Erweiterung. Diese sieht vor, dass auch SIP-Nachrichten, die eine Anforderung enthalten, mit einem numerischen Statuscode und einer textuellen Beschreibung versehen werden können, die normalerweise nur für Antworten vorgesehen sind. Somit kann dem Teilnehmer ein Grund (engl. *Reason*) für das Senden der *BYE*-Nachricht angegeben werden. Von den in [RFC 3261] spezifizierten Statuscodes erscheint *480 Temporarily Unavailable* als am besten geeignet. Allerdings geht aus dieser Nachricht nicht klar hervor, ob es sich um ein netzinternes Problem oder um ein Problem bzw. eine Einstellung beim anderen Teilnehmer (z. B. „Ruhe vor dem Telefon“-Dienst aktiviert) handelt. Besser geeignet, insbesondere auch bei Fehlerfällen noch während des Verbindungsaufbaus (s. u.), erscheint der Code *580 Precondition Failure*, der allerdings im Zuge einer optionalen Protokollerweiterung [RFC 3312] spezifiziert wurde.

5.4.1.2 Fehlerfälle beim Aufbau einer neuen Multimedia-Sitzung

Problematisch beim Zusammenspiel von SIP mit der MIDCOM-Architektur ist, dass die Adressparameter der Medienströme, welche für die Konfiguration des Paketfilters benötigt werden, erst recht spät im Verlauf eines „normalen“ Verbindungsaufbaus übermittelt werden. Wie in Abbildung 2.4 dargestellt, wird die SDP-Nachricht, mit der der gerufene Teilnehmer bekannt gibt, an welche Adresse er RTP-Medienströme in Vorwärtsrichtung erwartet, i. d. R. erst mit der *200 OK*-Nachricht gesendet. Zu diesem Zeitpunkt hat der gerufene Teilnehmer das Gespräch bereits entgegengenommen, d. h. beispielsweise den Hörer abgenommen. Falls die Middlebox z. B. mit einer *Policy Enable Rule Negative Reply*-Nachricht (*PER NR*, siehe Abbildung 5.4) signalisiert, dass das entsprechende Pinhole nicht geöffnet werden kann, muss der Verbindungsaufbau abgebrochen werden. Ein solcher Abbruch in dieser späten Phase des Verbindungsaufbaus, nachdem das Telefon des gerufenen Teilnehmers bereits geklingelt hat, wird als *Ghost Ring* bezeichnet.

Die Signalisierung des Abbruchs vom B2BUA zum User Agent des rufenden Teilnehmers ist vergleichsweise einfach. Da die *INVITE*-Anforderung noch nicht mit einer endgültigen Antwort (*Final Response*, d. h. Antwort mit Statuscode 200 oder größer) beantwortet wurde, kann in diese Richtung eine endgültige Antwort gesendet werden, die einen Fehlercode enthält, z. B. *480 Temporarily Unavailable* oder *580 Precondition Failure* (s. o.).

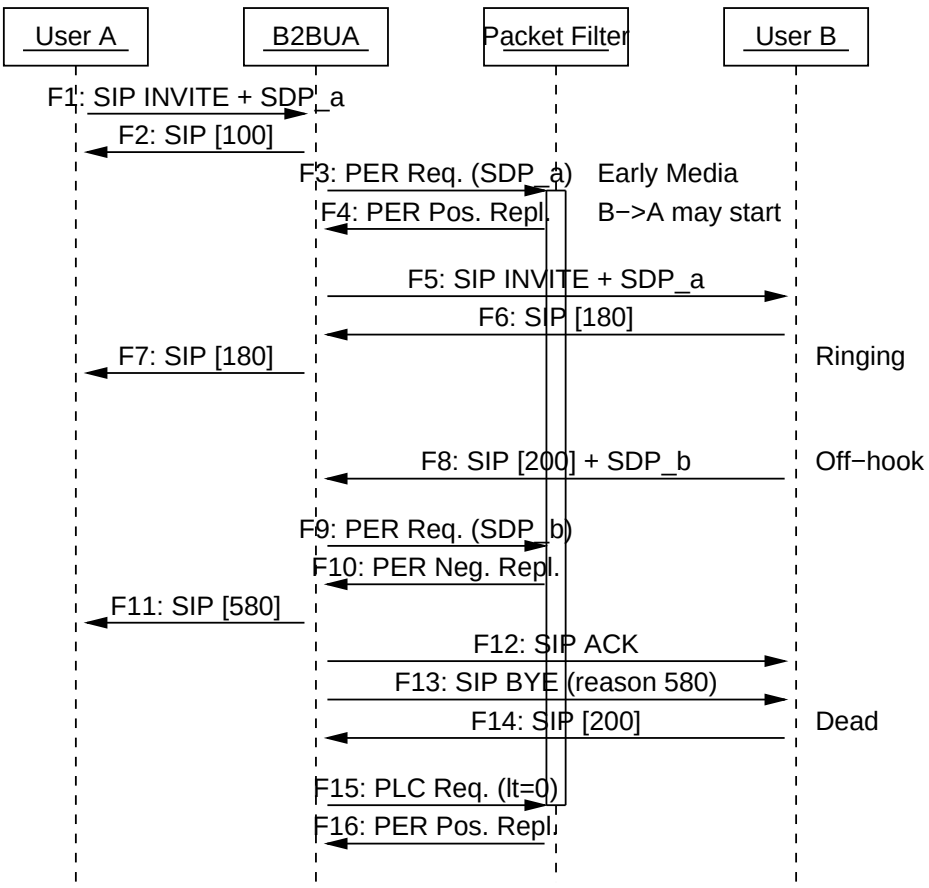


Abbildung 5.4: Zusammenspiel des IETF MIDCOM-Protokolls mit SIP: Fehlerfall.
Da das zweite *Pinhole* nicht geöffnet werden kann, kommt es zum so genannten *Ghost Ring*.

Die zum gerufenen Teilnehmer gesendete *INVITE*-Nachricht kann hingegen zu diesem Zeitpunkt nicht mehr einfach storniert werden (z. B. mit Hilfe einer *CANCEL*-Nachricht), da die *INVITE*-Nachricht bereits mit einer Final Response (*200 OK*) beantwortet wurde und die Transaktion somit abgeschlossen ist. Eine standardkonforme Vorgehensweise für den B2BUA ist, den Verbindungsaufbau mit einer *ACK*-Nachricht abzuschließen, um die Verbindung unmittelbar danach mit einer *BYE*-Nachricht zu beenden (siehe Abbildung 5.4). Auch hier sollte unbedingt ein Reason Header samt entsprechendem Fehler-Code (s. o.) als „Begründung“ mitgesendet werden, damit alle SIP-Instanzen und der gerufene Teilnehmer diesen Fall von einem Abbau der Verbindung durch den rufenden Teilnehmer unterscheiden können. Anderenfalls besteht die Gefahr, dass der gerufene Teilnehmer den rufenden Teilnehmer, dessen Identität i. d. R. schon in der *INVITE*-Nachricht übertragen wird, des „Telefonterrors“ beschuldigt, insbesondere wenn dieser mehrere erfolglose Versuche unternimmt, eine Verbindung herzustellen. Auch Monitoring-Systeme zur Überwachung des Netzbetriebs und Systeme zur Entgelterfassung sollten in der Lage sein, einen solchen Verbindungsabbruch aufgrund netzinterner Probleme von einer sehr kurzen Verbindung zu unterscheiden, um die Korrektheit der Kommunikationsdatensätze zu gewährleisten.

Die hier beschriebene Problematik kann vermieden werden, indem das in Abschnitt 2.1.5.10 beschriebene Verfahren zur Signalisierung von Vorbedingungen verwendet wird. Der gerufene Teilnehmer wird bei Anwendung dieses Verfahrens erst über den eingehenden Ruf informiert, nachdem die Paketfilter erfolgreich konfiguriert und ggf. QoS-Ressourcen reserviert wurden (siehe Abbildung 5.5). Somit können keine *Ghost Rings* auftreten. Ein weiterer Vorteil dieses Verfahrens ist, dass die Zeitdauer für das Konfigurieren der Firewalls nicht zum Meldeverzug, sondern zum Rufverzug beiträgt, welcher von den Teilnehmern i. d. R. als weniger störend empfunden wird. Nachteilig ist, dass bei diesem Verfahren mehr Nachrichten ausgetauscht werden müssen, was den Verbindungsaufbau im fehlerfreien Fall etwas verzögert und zu einer höheren Signalisierlast im Netz führt. Außerdem handelt es sich dabei um eine optionale Protokollerweiterung, die nicht von allen SIP-Implementierungen unterstützt wird.

5.4.2 Behandlung von SIP-Fehlerfällen in MIDCOM

SIP nach [RFC 3261] sieht eine Zeitüberwachung von Zuständen der SIP-Protokollautomaten (Soft State) nur während noch nicht abgeschlossener Transaktionen vor. Ist eine Multimedia-Sitzung erst einmal aufgebaut, wird diese hingegen nicht mit Timern, Nachrichten zur Zustandsauffrischung (engl. *Keepalive Message*), etc. überwacht (Hard State). Für User Agents und Proxies ist dies unproblematisch; erstere können einen unvorhergesehenen Abbruch der Sitzung (z. B. nach „Absturz“ der Software auf der Gegenstelle) recht einfach am Ausbleiben der Medienströme oder am Eingreifen des Nutzers erkennen; zweitere halten i. d. R. keine diesbezüglichen Zustandsinformationen.

Für den SIP B2BUA, welcher Zustandsinformationen über (vermeintlich) aktive Multimedia-Sitzungen hält, ist eine Erkennung eines solchen Fehlers über das Ausbleiben der Medienströme nicht in jedem Fall möglich, da diese RTP-Pakete nicht notwendigerweise über den selben Pfad durch das Netz geführt werden wie die Signalisiernachrichten. Desweiteren sind auch Anwendungen denkbar, bei denen ein zeitweiliges Ausbleiben der RTP-Pakete keinen Fehler darstellt, z. B. bei Kodierungsverfahren, die während Sprechpausen keine Pakete versenden. Dennoch ist

es unabdingbar, dass ein zur Firewall-Steuerung verwendeter B2BUA ein nicht ordnungsgemäß signalisiertes Ende einer Sitzung – sei es durch den „Absturz“ einer anderen Protokollinstanz oder durch gezielte Manipulation – innerhalb angemessener Zeit erkennen kann. Anderenfalls würden die im Paketfilter geöffneten Pinholes beliebig lange offen bleiben und könnten evtl. für den Transport von Angriffs-Verkehr verwendet werden.

Prinzipiell sind sowohl Lösungen denkbar, bei denen der B2BUA Nachrichten zur Abfrage des Sitzungsstatus an beide Endpunkte sendet, als auch solche, bei denen die User Agents in regelmäßigen Abständen Nachrichten versenden, die signalisieren, dass die Sitzung noch aktiv ist. Letzteres ist in [RFC 4028] als optionale Protokollerweiterung standardisiert.

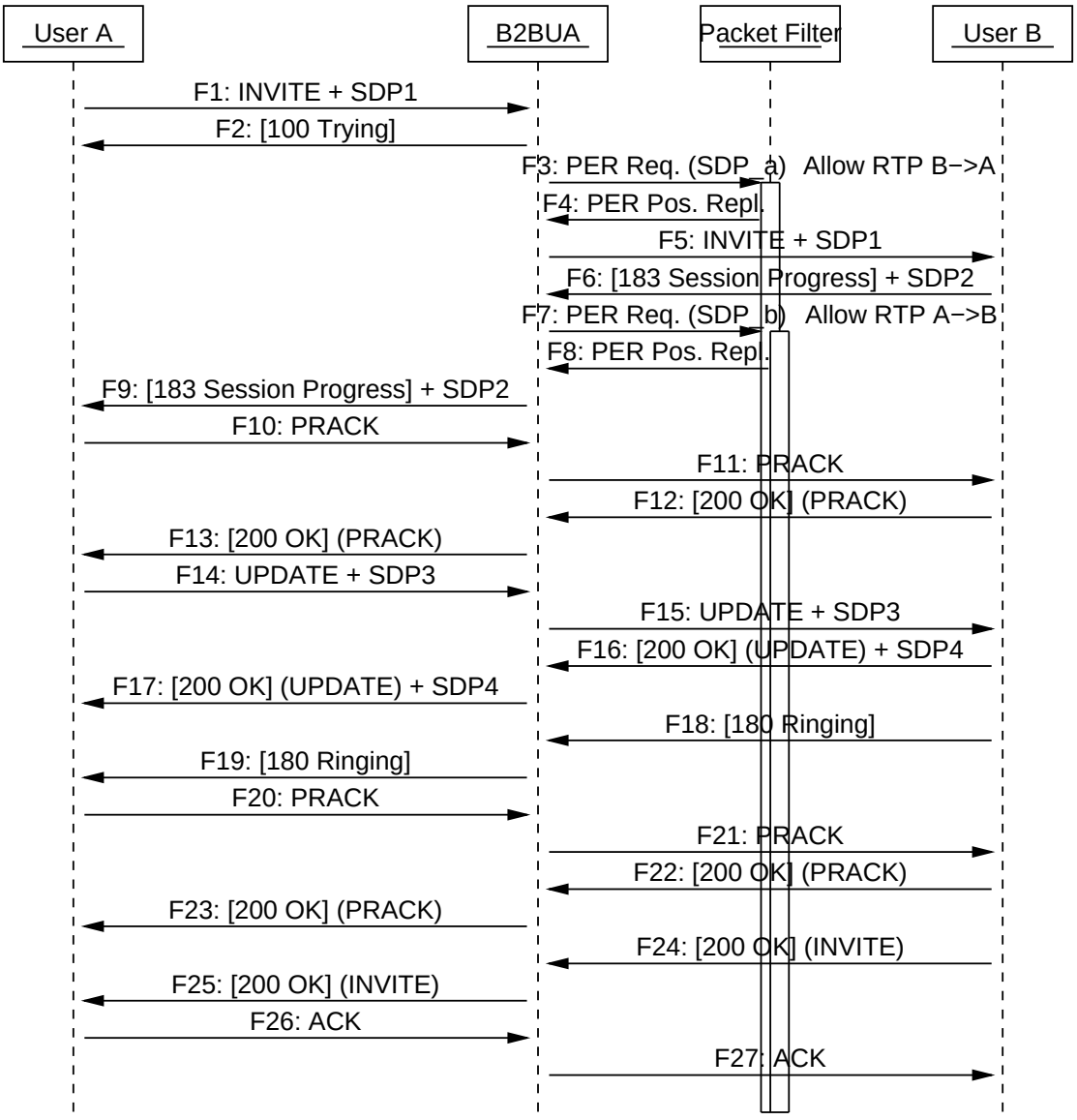


Abbildung 5.5: Zusammenspiel des IETF MIDCOM-Protokolls mit SIP und Vorbedingungen: Gut-Fall

5.5 Bestimmung der Leistungsfähigkeit des Linux Netfilter-Moduls

Es existiert ein breites Spektrum möglicher Architekturen und Implementierungen von Paketfiltern, die als Medienkomponente eingesetzt werden können. Dieses reicht von „Embedded Devices“ (z. B. [124]) über Softwarelösungen auf Standard-Betriebssystemen für Personal Computer (z. B. Linux/Netfilter, s. u.) bis hin zu speziell für diesen Zweck entwickelter Hardware, z. B. [125]. Da die Leistungsfähigkeit eines Paketfilters dementsprechend hochgradig implementierungsspezifisch und zudem weitgehend unabhängig von den Architekturen und Protokollen zur Paketfilter-Steuerung ist, sollen hier nur die wesentlichen Kenngrößen benannt werden. Ergänzend werden die grundsätzlichen Parameter benannt, die – zumindest unter Linux – einen Einfluss auf diese Kenngrößen haben, und die Ergebnisse einiger Messungen an Linux/Netfilter dargestellt, die mit dem in Abschnitt 5.2.3.4 beschriebenen Modul zur Anbindung des SIMCO-Servers gewonnen wurden.

5.5.1 Wesentliche Kenngrößen

Aus Sicht des Teilnehmers sind drei Kenngrößen zur Beschreibung der Leistungsfähigkeit eines Paketfilters relevant:

- Die mittlere Verzögerung δ_F erlaubter Pakete beim Durchlaufen des Paketfilters trägt zur Ende-zu-Ende-Verzögerung der IP-Pakete und somit zur Mund-zu-Ohr-Verzögerung der Sprache bei und sollte daher möglichst gering sein (vgl. Abschnitt 2.1.2).
- Paketverluste vermindern die Übertragungsqualität. Auch wenn die für IP-Telefonie verwendeten Codecs i. d. R. gewisse Verluste tolerieren können, sollte die Wahrscheinlichkeit p_F, dass ein legitimes, eigentlich zu erlaubendes Paket aufgrund von Überlastung des Paketfilters verloren geht, so gering wie möglich sein.
- Die mittlere Dauer zum Hinzufügen einer Regel in den Paketfilter δ_u trägt zusammen mit der anderen Verwaltungsaufgaben, z. B. im SIMCO-Server, und dem Nachrichtentransport (siehe Kapitel 6) zur Antwortzeit R der Firewall-Steuerung und somit zur Verzögerung des Verbindungsaufbaus bei. Auch sie sollte daher möglichst gering sein.

Aus Sicht des Netzbetreibers ist eine wichtige Kenngröße, wieviele gleichzeitige Multimedia-Sitzungen m von einem Paketfilter inspiziert werden können, ohne dass die o. g. Werte ihre jeweiligen Grenzwerte übersteigen. Dabei ist zu beachten, dass ein Medienstrom mehrere Firewalls durchlaufen kann, die jeweils zu diesen Ende-zu-Ende gemessenen Größen beitragen. Desweiteren muss der Paketfilter so dimensioniert werden, dass er genügend Leistungsreserven für die Abwehr des angenommenen maximalen Angriffsverkehrs-„Angebots“ hat, ohne dass darunter die Übertragung der legitimen Datenströme leidet.

5.5.2 Beeinflussende Parameter

Ein Rechner mit Linux-Betriebssystem hat i. d. R. nur einen bzw. wenige Prozessoren (CPU), auf denen mehrere Aufgaben quasi-simultan ausgeführt werden können, indem ihnen die bzw.

eine CPU abwechselnd zugeteilt wird. In dem betrachteten Anwendungsszenario gehören dazu unter anderem folgende Aufgaben:

- der „Hauptprozess“, welcher über SIMCO Steuerbefehle der SIMCO-Agents entgegennimmt, die Policy Rules und ihre Lebensdauern verwaltet, etc.,
- der „Hilfsprozess“, der den effektiven Regelsatz in den Betriebssystem-Kern einbringt,
- die Bearbeitung von Paketen (Empfang, Zugriffskontrolle, Routing, Versand), sowie
- diverse weitere Aufgaben, z. B. Protokollierung von Ereignissen.

Einige diese Aufgaben werden in Prozessen im Userspace durchgeführt, andere in Kernel Tasks, wieder andere in Interrupt-Behandlungsroutinen. Die Wechsel zwischen diesen Kontexten verursachen zusätzliche Latenzen. Pakete können auf ihrem Weg durch den Protokollstapel an diversen Stellen zwischengepuffert werden; insbesondere bei hohen Paketraten kann es vorkommen, dass mehrere aufeinanderfolgende Pakete innerhalb einer Zeitscheibe bearbeitet werden. Es ist daher kaum möglich, die gemessenen Verzögerungen und Stabilitätsgrenzen mit einem einfachen analytischen Modell zu beschreiben. Für das Ändern des Regelsatzes sowie die Paketfilterung ist es jedoch möglich, Parameter zu benennen, von denen die Belastung der CPU und somit letztendlich auch die Verzögerungen abhängen. Den Messungen zufolge sind das die Aufgaben, die am meisten Rechenleistung benötigen.

Geht man zunächst von der mittleren Zahl simultaner Multimedia-Sitzungen m aus, so kann mit der mittleren Sitzungsdauer h die mittlere Rate von Sitzungsaufbauten zu $\lambda_C = \frac{m}{h}$ bestimmt werden. Am Anfang einer Sitzung werden i. d. R. $u = 2$ Pinholes geöffnet (eines je Richtung) und am Ende wieder geschlossen. Daraus ergibt sich eine Regeländerungsrate

$$\lambda_{uc} = 2 \cdot \lambda_C \cdot u = 2 \frac{m \cdot u}{h} \tag{5.4}$$

und eine Anzahl $m \cdot u$ gleichzeitig geöffneter Pinholes für die Medienströme. Zu diesen kommt i. d. R. noch eine kleine, konstante Anzahl c von Regeln zum Erlauben der SIP-Signalisierung und anderer Hilfsprotokolle (z. B. DNS) hinzu. Die Gesamtzahl der Regeln im Paketfilter ist so

$$U = m \cdot u + c \ . \tag{5.5}$$

Der Bedarf an CPU-Ressourcen hängt sowohl von der Anzahl der schon vorhandenen Regeln U, als auch von der Änderungsrate λ_{uc} ab. Die Abhängigkeit von U ergibt sich aus der Tatsache, dass bei jeder Regeländerung der komplette Regelsatz vom Adressraum des Betriebssystem-Kerns in den des „Hilfsprozesses“ kopiert, dort geändert und wieder zurückkopiert wird (siehe Abschnitt 5.2.3.4). Bei höheren Änderungsraten λ_{uc} wird dieser Prozess öfters angestoßen, so dass der Ressourcenbedarf zunächst steigt. Um die Konsistenz der Tabellen zu wahren, dürfen Änderungsvorgänge allerdings nicht nebenläufig durchgeführt werden. Eine neue Anforderung zum Ändern des Regelsatzes muss daher bei ihrer Ankunft ggf. erst auf das Ende des gerade in Bearbeitung befindlichen Änderungsvorgangs warten. Steigt λ_{uc} an, erhöht sich die Wahrscheinlichkeit, dass bis zum Ende dieses Vorgangs gleich mehrere Anforderungen eingetroffen sind, die dann auf einmal bearbeitet werden.

Ein Paketfilter mit Whitelist-Konfiguration vergleicht die Charakteristika jedes eintreffenden IP-Pakets mit dem Regelsatz. Wird eine passende Regel gefunden, darf das Paket passieren, ansonsten wird es verworfen. In der Netfilter-Standardkonfiguration werden die Regeln in einer Liste angeordnet, in der linear gesucht wird [117]. Unter der Annahme, dass sich die zu erlaubenden Pakete gleichmäßig auf die Regeln verteilen, werden somit pro RTP-Paket im Mittel $\frac{U}{2}$ Regelvergleiche benötigt, bis das passende Pinhole gefunden wird. Unerlaubte Pakete müssen hingegen mit allen U Regeln verglichen werden, bis feststeht, dass sie abgewiesen werden müssen. In Bezug auf die Anzahl benötigter Regelvergleiche stellt der Angriffsverkehr somit eine höhere Belastung für den Paketfilter dar als der legitime Verkehr. Andere Ansätze zum Speichern und Durchsuchen der Regelliste (z. B. [126]), die auf Hash-Tabellen o. ä. basieren, können diese Abhängigkeit von der Regel-Anzahl reduzieren.

Geht man davon aus, dass zu jeder Multimedia-Sitzung zwei Medienströme gehören und dass für alle Medienströme der selbe Codec verwendet wird, der eine mittlere Paketrate λ_M erzeugt, und vernachlässigt man den Signalisierverkehr, so ergibt sich am Paketfilter eine mittlere Paketrate von

$$\lambda_i = 2 \cdot m \cdot \lambda_M + \lambda_X \ , \tag{5.6}$$

wobei λ_X die von den Angreifern erzeugte Paketrate ist. Die Rate der zur Filterung dieses Verkehrs erforderlichen Vergleichsoperationen ist bei linearem Suchen

$$\lambda_F = U \left(m \cdot \lambda_M + \lambda_X \right) \ . \tag{5.7}$$

Diese Operationen konkurrieren mit anderen Aufgaben (s. o.) um die selben CPU-Ressourcen.

5.5.3 Messungen an Linux/Netfilter

In der ersten Messreihe wurde die Abhängigkeit der Aufenthaltsdauer eines Paketes im Paketfilter δ_F und die der Paketverlustwahrscheinlichkeit p_F von der Anzahl gleichzeitiger IP-Telefonie-Verbindungen untersucht, wobei auf den Auf- und Abbau von Sitzungen während der Messung und auf die Abwehr von Angriffsverkehr zunächst verzichtet wurde, d. h. der SIMCO-

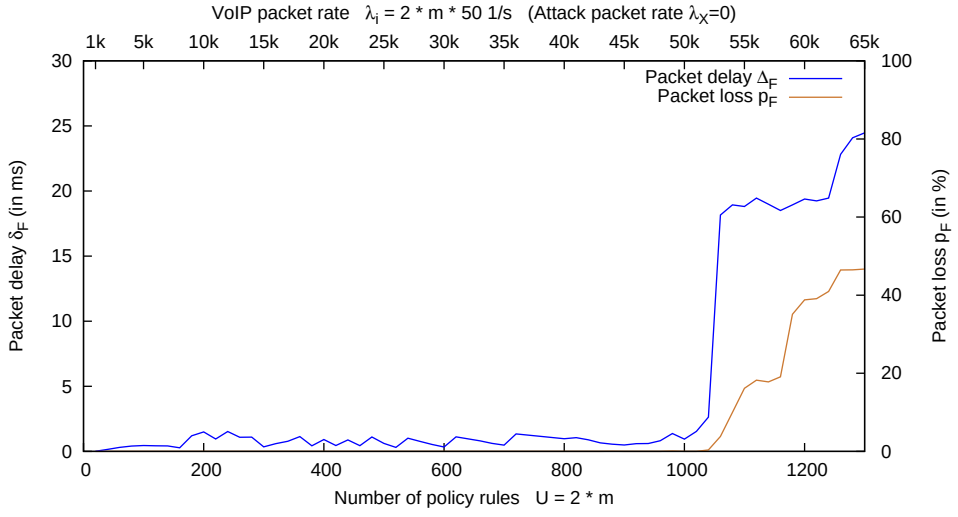


Abbildung 5.6: Paketverzögerung und -verlustwahrscheinlichkeit mit Linux Netfilter

Server kam hier nicht zum Einsatz. Der Paketfilter wurde auf einem Rechner mit 2.8 GHz Intel Pentium 4 Prozessor und zwei Intel E1000 Gigabit-Ethernet-Karten betrieben. Abbildung 5.6 zeigt die Messwerte für δ_F und p_F über der Anzahl der vor Beginn der Messung in den Paketfilter eingetragenen Pinholes U, von welchen $u = 2$ pro Multimedia-Sitzung benötigt werden. Mit Hilfe von Lastgeneratoren wurde durch jedes dieser Pinholes ein Flow von UDP-Paketen, die 160 Byte Nutzlast tragen, mit einer konstanten Paketrate von $\lambda_M = 50\frac{1}{s}$ geleitet. Im dargestellten Parameterbereich lasten diese die Ethernet-Segmente nur im einstelligen Prozentbereich aus. Es zeigt sich, dass bis zu $\hat{m} = 500$ Sitzungen, entsprechend einer Gesamt-Paketrate von $\hat{\lambda}_i = 50000\frac{1}{s}$ und $\hat{U} = 1000$ Regeln, die Verzögerung sich im Bereich der Messgenauigkeit befindet und sich praktisch nicht von den Messwerten bei abgeschaltetem Paketfilter (d. h. Rechner arbeitet als Router) unterscheidet, sowie dass keine Paketverluste auftreten. Oberhalb dieser Stabilitätsgrenze steigen die Paketverluste und die Verzögerung sprunghaft an.

Aufgabe einer Firewall ist die Abwehr unerwünschten Verkehrs, insbesondere Paketfluten von DoS-Angriffen. Dabei sollte es nicht zu Beeinträchtigungen des erwünschten Verkehrs in Form von Paketverlusten oder erhöhten Verzögerungen kommen. Die maximal mögliche Paketrate, die so abgewehrt werden kann, wurde gemessen und ist in Abbildung 5.7 dargestellt.

Ausgangspunkt ist dabei wieder die Anzahl simultaner Sitzungen m. Mit Hilfe von SIMCO-Server und -Lastgenerator wurden während der Messung Pinholes mit einer konstanten Gültigkeitsdauer von $h = 180\,s$ eingetragen. Der Mittelwert der negativ-exponentiell verteilten Zwischenankunftszeiten so wurde eingestellt (vgl. Abschnitt 5.2.5.2), dass im Mittel $U = 2m$ Pinholes (untere X-Achse) geöffnet waren. Mit zuschaltbaren Paketgeneratoren wurde durch jedes dieser Pinholes ein Flow mit einer konstanten Paketrate von $\lambda_M = 50\frac{1}{s}$ geleitet, so dass die mittlere Gesamt-Rate der zu erlaubenden Pakete $m \cdot 100\frac{1}{s}$ betrug (unterer, grüner Anteil der Balken, obere X-Achse). In diesem Zustand wurde die mittlere Verzögerung bei der Aktuali-

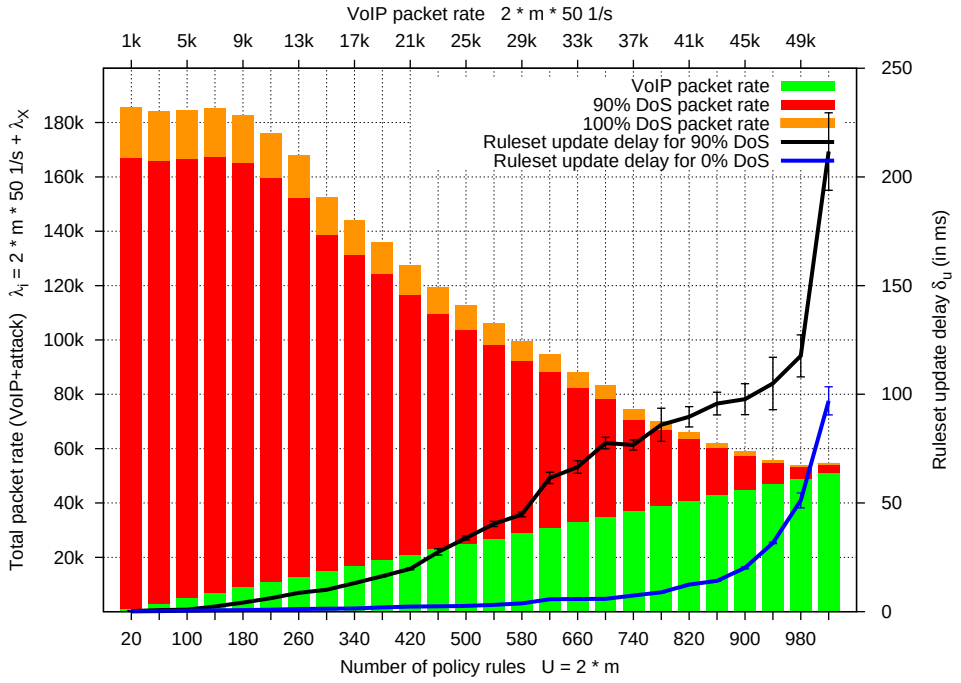


Abbildung 5.7: Maximale Paketrate und Regeleintrag-Verzögerung mit Linux Netfilter

sierung des Regelsatzes δ_u bestimmt (untere, blaue Kurve, rechte Y-Achse). Wiederum stellte sich diese Verzögerung über weite Bereiche als gering heraus, mit sehr deutlichem Anstieg ab ca. 450...500 gleichzeitiger Sitzungen.

In einem zweiten Schritt wurde mit Hilfe eines weiteren Lastgenerators zusätzlich Angriffsverkehr erzeugt, dessen Pakete mit ebenfalls 160 Bytes UDP-Nutzlast zu keinem der Pinholes passen und deshalb verworfen wurden. Die Paketrate dieses Paketstroms wurde bei Null beginnend so weit erhöht, bis die legitimen Medienströme so weit in Mitleidenschaft gezogen wurden, dass dort Paketverlustwahrscheinlichkeiten über 0.1 % auftraten (mittlere, rote und obere, orange Anteile der Balken, linke Y-Achse). Von ca. 180 000 Paketen pro Sekunde bei nur zehn Sitzungen geht dieser maximal abwehrbare DoS-Verkehr bis auf Null bei $m \approx 500$ zurück. Folglich sollte bei einer sinnvollen Netzdimensionierung der Arbeitspunkt eines solchen Paketfilters deutlich unterhalb der prinzipiell möglichen ca. 500 simultanen Multimedia-Sitzungen liegen, um noch genügend Leistungsreserven für die Abwehr von Angriffsverkehr zu haben.

Bei dieser maximalen Paketrate wird die komplette CPU-Leistung für die Paketfilterung benötigt; der mit niedrigerer Priorität ausgeführte Prozess zur Regelsatz-Aktualisierung kommt zum Erliegen. Deshalb wurde in einem dritten Schritt die Paketrate des Angriffsverkehrs auf 90 % des zuvor bestimmten Maximums gesenkt, bevor erneut die Verzögerung beim Regeleintrag gemessen wurde (obere, schwarze Kurve). Diese liegt zwar deutlich höher als bei der Messung ohne DoS-Verkehr, für $m < 500$ aber noch in einer akzeptablen Größenordnung (vgl. Kapitel 7).

5.6 Zusammenfassung und Fazit

Der Aufbau einer Testumgebung hat die Machbarkeit einer Integration von SIP-basierter IP-Telefonie und Pfad-entkoppelter Firewall-Steuerung am Beispiel der IETF MIDCOM-Architektur und dem SIMCO-Protokoll demonstriert. Dabei stellte sich jedoch heraus, dass das Vorhandensein von MIDCOM für SIP nicht so transparent ist, wie in [RFC 3303] suggeriert wird [10]. Dies betrifft insbesondere die Frage, wie im Bereich eines der beiden Signalisierprotokolle auftretende Fehler zu einer sinnvollen Reaktion des jeweils anderen Protokolls führen können.

Während sich die bei SIMCO vorhandenen Protokollmechanismen als ausreichend erwiesen haben, reichen die Mechanismen von SIP nach [RFC 3261] nicht für eine korrekte Erkennung und Behandlung aller identifizierten Fehlermöglichkeiten aus. In der Zwischenzeit wurden zwar für alle benötigten Mechanismen entsprechende Protokollerweiterungen von der IETF standardisiert, jedoch muss damit gerechnet werden, dass diese Erweiterungen nicht in allen Protokollinstanzen implementiert werden. Solche Interoperabilitätsprobleme können dazu führen, dass Teilnehmer nicht beliebige SIP-Endpunkte an einem so geschützten Netz betreiben können.

Ergänzend wurde durch Messungen die Leistungsfähigkeit des Prototypen bestimmt. Auch wenn der Einsatz einer Medienkomponente auf Basis preiswerter PC-Hardware, Linux/Netfilter und dem SIMCO-Prototypen in einer „carrier grade“ IP-Telefonie-Plattform nicht besonders wahrscheinlich ist, konnte mit der Filterung des (emulierten) Verkehrs mehrerer hundert simultaner IP-Telefonie-Verbindungen eine durchaus beachtliche Leistung erreicht werden. Bei der Dimensionierung eines solchen Firewall-Systems muss darauf geachtet werden, dass genügend Leistungsreserven zur Abwehr des Angriffsverkehrs zur Verfügung stehen.

6 Optimierter Transport von Signalisiernachrichten über IP

Bei den betrachteten Architekturen zur Steuerung von Firewalls in IP-Telefonie-Plattformen muss sich die Signalisierkomponente nicht notwendigerweise in netztopologischer Nähe der gesteuerten Medienkomponente(n) befinden. Abbildung 4.7 verdeutlicht dies am Beispiel eines Szenarios, in welchem alle Medienkomponenten an den Netzübergängen einer Domäne von einer zentralen Signalisierkomponente (hier: so genannter *Softswitch*, SSw.) gesteuert werden.

In solchen Szenarien, bei denen Signalisiernachrichten nicht nur über ein lokales Netz, sondern über größere Entfernungen transportiert werden, steigt der Einfluss des Nachrichtentransports auf die Transaktions-Antwortzeiten. Während Latenzen beim Transport von IP-Paketen und ggf. auftretende Paketverluste Eigenschaften des Netzes sind, die oft nur wenig beeinflusst werden können, gibt es Freiheitsgrade bei der Wahl des Transportschichtprotokolls und seiner Parametrisierung.

In diesem Kapitel sollen daher zunächst verschiedene Transportschichtprotokolle und Konfigurationsvarianten für den Transport von Signalisiernachrichten über IP-Netze kurz vorgestellt werden. Dabei wird insbesondere auf den verzögernden Effekt des Head-Of-Line Blocking eingegangen. Anschließend wird am konkreten Beispiel von SIMCO untersucht, welche Eigenschaften bzw. Anpassungen an einem Signalisierprotokoll für das Zusammenspiel mit verschiedenen Transportschichtprotokollen jeweils notwendig sind. Die darauf folgend vorgestellten Messungen zur Verzögerung beim Nachrichtentransport erfolgten in der in Abschnitt 5.2.6 vorgestellten SIMCO-Testumgebung. Die Ergebnisse sind jedoch, genau wie die dazugehörige Modellierung und analytische Berechnung, auch auf andere Signalisierprotokolle übertragbar.

6.1 Transportschichtprotokolle

Über lange Zeit hinweg gab es in der IP-Protokollfamilie nur zwei relevante Transportschichtprotokolle, das *User Datagram Protocol* (UDP) und das *Transmission Control Protocol* (TCP). Beim Transport von Signalisiernachrichten (z. B. zur Firewall-Steuerung) haben beide Protokolle gewisse Nachteile. Deshalb wurde bei der IETF ein weiteres Transportschichtprotokoll, das *Stream Control Transmission Protocol* (SCTP) entworfen und standardisiert. Tabelle 6.1 gibt einen Überblick über diese Protokolle, die im Folgenden kurz vorgestellt werden.

6.1.1 User Datagram Protocol (UDP)

Das *User Datagram Protocol* (UDP) [RFC 768] fügt zu den Eigenschaften und Diensten der IP-Schicht im Wesentlichen nur ein Multiplexing auf Basis so genannter Port-Nummern hinzu. Damit können verschiedene Flows zwischen den selben beiden Endsystemen unterschieden werden und ggf. vom Initiator eines Flows mit Hilfe des Well-Known Port Numbers-Konzeptes ein Anwendungsschichtdienst ausgewählt werden.

UDP bietet somit einen verbindungslosen Transport einzelner Nachrichten („Datagramme“) ohne Fehlersicherung an. Es sind keine Protokollmechanismen zur Erkennung und Korrektur von Übertragungsfehlern (z. B. Paketverluste), zur Reihenfolgesicherung, Flusssteuerung oder Überlastabwehr, etc. vorhanden – diese müssen bei Bedarf in den darüberliegenden Protokollschichten implementiert werden.

Tabelle 6.1: Vergleich der Transportschichtprotokolle UDP, TCP und SCTP

Protokoll	UDP	TCP	SCTP
Kennung im *IP header*	17	6	132
Verbindungsorientiert	nein	ja: *Connection*	ja: *Association*
Verbindungsaufbau	–	*3-way Handshake*	*4-way Handshake*
Verbindungsneustart	–	nein	ja
Multihoming	–	nein	ja
Transport von Nutzerdaten	Nachrichten-Blöcke	kontinuierlicher Datenstrom	Nachrichten-Blöcke
Aufteilen zu großer Nachrichten	nein	ja (*Segments*)	ja (*Chunks*)
Flusskontrolle	nein	ja	ja
Überlastabwehr	nein	ja	ja
Logische Unterkanäle	nein	nein	$1 \ldots 2^{16}$ *Streams*
Prüfsummen z. Fehlererkennung	16 bit (optional)	16 bit	32 bit
Sequenznummern + Bestätigung	nein	ja	ja
Autom. Neuübertragung	nein	ja	ja
Verwerfen von Duplikaten	nein	ja	ja
Reihenfolgesicherung	–	ja	ja, aber **nur im selben *Stream***
Zusätzliche Transportmöglichkeit ohne Reihenfolgesicherung	–	eingeschränkt (*URG Flag*)	ja (*Unordered Flag*)
Schutzmaßnahmen gegen …		optional, teilw.	ja (*Cookies*, *Verification Tags*)
DoS- & blind Spoofing-Angriffe	nein	(*SYN Cookies*)	
Man-in-the-Middle-Angriffe	nein	nein	nein

6.1.2 Transmission Control Protocol (TCP)

Im Unterschied dazu bietet das *Transmission Control Protocol* (TCP) [RFC 793] einen verbindungsorientierten Transport eines kontinuierlichen Byte-Stroms mit Fehlersicherung, Reihenfolgesicherung, Flusssteuerung und Überlastabwehr. Es ist das am meisten verwendete Transportschichtprotokoll im Internet. sowohl für den Massentransport von Daten (z. B. mit dem *File Transfer Protocol*, FTP), als auch für interaktive Anwendungen (z. B. *Secure Shell*, SSH).

Nachteilig ist, dass die Mechanismen von TCP nicht einzeln abschaltbar sind. Dies hat dazu geführt, dass einige Anwendungsprotokolle auf UDP aufsetzen und bestimmte Protokollmechanismen von TCP (z. B. Sequenznummern und Bestätigungen) in der Anwendungsschicht selbst implementieren, da sie Probleme mit anderen TCP-Mechanismen oder deren Auswirkungen (z. B. zusätzliche Verzögerungen durch Reihenfolgesicherung) haben.

6.1.3 Stream Control Transmission Protocol (SCTP)

Das *Stream Control Transmission Protocol* (SCTP) [RFC 2960] wurde ursprünglich im Umfeld der IETF SIGTRAN-Arbeitsgruppe als Teil der SIGTRAN-Architektur [RFC 2719] zum Transport von SS7-Nachrichten über IP entworfen. Das Zeichengabesystem Nr. 7 (engl. *Signaling System No. 7*, SS7) [Q.700] ist eine Familie von Signalisierprotokollen zur Steuerung der leitungsvermittelnden PSTN- und ISDN-Netze; der Nachrichtentransport erfolgt dabei paketorientiert über zentrale Zeichengabekanäle. Der Transport von SS7-Nachrichten über IP kann z. B. bei der Zusammenschaltung von IP-Telefonie-Plattformen mit dem ISDN (siehe Abschnitt 2.1.5.11) sinnvoll sein, oder wenn Mehrwertdienste des ISDN auf IP-basierten Servern erbracht werden sollen. Die benötigten Mechanismen für dieses sehr spezielle Anwendungsszenario werden nicht von SCTP selbst, sondern von einer Reihe von SIGTRAN-Anpassungsschichten (*User Adaptation Layer*) oberhalb von SCTP erbracht, z. B. M2PA, M2UA, M3UA, SUA, IUA, oder V5UA. Diese Anpassungsschichten stellen jeweils einen anderen Dienstzugangspunkt (engl. *Service Access Point*, SAP) des SS7-Protokollstapels zur Verfügung; die Funktionen der darunterliegenden SS7-Protokollschichten werden dabei auf Basis des SCTP/IP-basierten Transports emuliert. SCTP selbst hingegen wurde als generisches Transportprotokoll für IP-Netze entworfen, welches neben UDP und TCP in den IP-Protokollstapel eingeordnet werden kann. Dieser universelle Ansatz ist auch daran erkennbar, dass die Verantwortung für die Weiterentwicklung des Protokolls mittlerweile an die für Transportschichtprotokolle zuständige IETF Transport Area Working Group (TSVWG) übergeben wurde.

SCTP wurde als Transportschichtprotokoll für den Transport von Signalisiernachrichten in Umgebungen mit besonders hohen Zuverlässigkeits- und Sicherheitsanforderungen entworfen. Es unterstützt so genanntes *Multihoming*, d. h. Endpunkte können mehrere physikalische Netzschnittstellen mit je einer IP-Adresse haben. Die Verfügbarkeit der Pfade zwischen diesen Schnittstellen wird ständig überwacht und der Nachrichtentransport wird automatisch auf einen alternativen Pfad umgeschaltet, sobald der primäre Pfad nicht mehr verfügbar sein sollte. SCTP verwendet so genannte *Verification Tags* und einen *State Cookie*-Mechanismus, um sich gegen Angriffe gegen die Verfügbarkeit (*Denial-of-Service*-Attacken, DoS) und das „blinde“ Einschleusen von Nachrichten (d. h. durch Angreifer, die den legitimen Verkehr nicht abhören können) zu schützen [11].

Bezüglich des Nutzdatentransports kombiniert SCTP Eigenschaften von UDP und TCP und fügt neue hinzu. Auf Basis des verbindungslosen, ungesicherten Pakettransports, den die IP-Schicht anbietet, stellt SCTP den darüberliegenden Protokollen (*Upper Layer Protocol*, ULP) einen verbindungsorientierten, fehlergesicherten Transport von Nachrichten zur Verfügung. Anders als UDP kann SCTP Paketverluste, duplizierte IP-Pakete und Bitfehler erkennen und die betroffenen Nachrichten erneut übertragen bzw. verwerfen. SCTP verwendet Mechanismen zur Flusskontrolle und Überlastabwehr, die denen von TCP sehr ähnlich sind. Anders als TCP bewahrt SCTP die Grenzen zwischen Nachrichten: Einzelne Byte-Blöcke werden vom bzw. zum ULP übertragen, anstelle eines kontinuierlichen Byte-Stroms wie bei TCP. Somit werden im ULP keine Längen-Felder oder Markierungen (*Frame Delimiter*) benötigt, um den Datenstrom empfängerseitig wieder in einzelne Nachrichten zu zerlegen (*Message Delineation*).

Eine *SCTP Association* (SCTP-Terminus für Verbindung) kann in bis zu 65536 logische Unterkanäle pro Richtung, so genannte *Streams* aufgeteilt werden. Deren Anzahl wird während des Assoziationsaufbaus zwischen den Endpunkten ausgehandelt. Jede Nachricht wird in einem Stream transportiert, der vom ULP ausgewählt werden kann. Nachrichten, die vom Sender mit dem „U-Flag“ (für *unordered*) versehen werden, werden empfängerseitig sofort an das ULP ausgeliefert; für alle anderen Nachrichten stellt SCTP innerhalb des jeweiligen Streams eine getrennte Reihenfolgesicherung zur Verfügung. Eine Assoziation mit mehreren Streams hat – verglichen mit mehreren Assoziationen mit je einem Stream – den Vorteil, dass weniger Ressourcen für Assoziationsaufbau und -verwaltung benötigt werden und dass die Effizienz des von TCP übernommenen *Fast Retransmit*-Algorithmus (siehe Abschnitt 6.5.3.1) höher ist, da er auf den gesamten Nachrichtenfluss in allen Streams einer Assoziation angewendet werden kann.

6.2 Head-of-Line Blocking in Transportschichtprotokollen

So genanntes *Head-Of-Line Blocking* kann auftreten, wenn ein Transportschichtprotokoll einen fehler- und reihenfolgegesicherten Transport anbietet, wie dies z. B. bei TCP der Fall ist. Wenn eine Nachricht im Netz verlorengeht oder Bitfehler auftreten, so muss sie erneut übertragen werden. Darauf folgende Nachrichten, die während der so verursachten Wartezeit beim Empfänger ankommen, müssen bis zum Ende der Übertragungswiederholung gepuffert werden, damit alle Nachrichten in der richtigen Reihenfolge an die darüberliegenden Protokollschichten ausgeliefert werden können. Dies ist für eine TCP-Verbindung in Abbildung 6.1(a) illustriert.

Unter Fehlersicherung versteht man im Kontext von Transportschichtprotokollen den Schutz gegen Nachrichtenverlust und Bitfehler. Reihenfolgesicherung bezeichnet hingegen, dass Nachrichten in der selben Reihenfolge an die empfängerseitige Anwendungsschicht übergeben werden, in der sie senderseitig übernommen wurden. Dies sind zwei unabhängige Eigenschaften von Protokollen [127]. Viele Signalisieranwendungen haben hohe Zuverlässigkeitsanforderungen und benötigen daher Fehlersicherung, jedoch nicht notwendigerweise auch vollständige Reihenfolgesicherung. Mögliche Anforderungen bezüglich Reihenfolgesicherung können klassifiziert werden [127] in: (1) *vollständige*, (2) *teilweise* oder (3) *keine* Reihenfolgesicherung. Im zweiten Fall muss die Einhaltung der Reihenfolge nur innerhalb von Teilmengen aller transportierten Nachrichten gewährleistet werden, welche vom darüberliegenden Protokoll spezifiziert werden müssen.

SCTP erlaubt einen solchen teilweise reihenfolgegesicherten Transport: wenn eine Nachricht aufgrund eines Fehlers wiederholt übertragen werden muss, ist nur der jeweilige Stream vom Head-Of-Line Blocking betroffen, wohingegen andere Streams unbeeinträchtigt weiter Nachrichten an das ULP ausliefern können (siehe Abbildung 6.1(b)). Dies reduziert den Einfluss der Reihenfolgesicherung auf die mittlere Ende-zu-Ende-Verzögerung.

Viele Signalisierprotokolle müssen hohe Anforderungen bzgl. der Verzögerungen erfüllen. Es kann zwar davon ausgegangen werden, dass in einem angemessen dimensionierten Signalisiernetz Störungen wie Paketverluste oder Bitfehler nur selten vorkommen. Falls sie aber dennoch gelegentlich auftreten, sind bei Assoziationen, die eine hohe Signalisierlast tragen, sofort recht viele Nachrichten von verzögernden Effekten betroffen. Desweiteren werden in dieser Arbeit auch Architekturen untersucht, bei denen Signalisiernachrichten im Zugangsnetz oder über die (evtl. drahtlose) Teilnehmerschnittstelle übertragen werden, wo Paketverluste u. U. häufiger auftreten können. Aus diesem Grund ist es sinnvoll, den Einfluss von Paketverlusten auf die Antwortzeiten von Signalisier-Transaktionen zu untersuchen.

6.3 Konfigurationsvarianten für den Transport von Signalisiernachrichten über IP

In Tabelle 6.2 wird ein Überblick über verschiedene Konfigurationsvarianten für den Transport von Signalisiernachrichten über IP gegeben, die im Folgenden am Beispiel von SIMCO untersucht werden sollen. Dabei wird insbesondere auf die Vermeidung von Head-Of-Line Blocking geachtet. Dazu müssen zunächst die Anforderungen von SIMCO bzgl. Reihenfolgesicherung untersucht werden.

6.3.1 Anforderungen von SIMCO bezüglich Reihenfolgesicherung

Wie viele andere Anwendungsschichtprotokolle auch, ist SIMCO nicht auf eine korrekte Funktion für den Fall ausgelegt, dass die Reihenfolge der SIMCO-Nachrichten beim Transport durch

(a) Eine TCP-Verbindung

(b) Eine SCTP-Assoziation mit 3 Streams

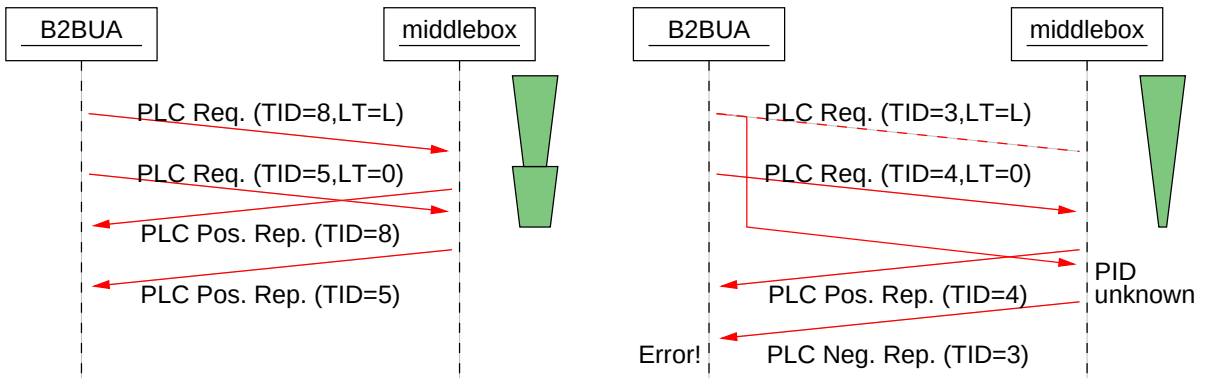
Abbildung 6.1: Head-of-line Blocking bei Transportschichtprotokollen mit vollständiger bzw. teilweiser Reihenfolgesicherung (schematisch)

Tabelle 6.2: Möglichkeiten für den Transport von Signalisiernachrichten über IP

Reihenfolge-sicherung	UDP	TCP	SCTP
vollständig		eine Verbindung	eine Assoziation mit einem Stream, keine *unord.*-Flags
teilweise		mehrere parallele Verbindungen, einzeln verwaltet	eine Assoziation mit mehreren Streams, keine *unord.*-Flags
keine	ein Flow, Fehlersicherung in Anwendungsschicht		eine Assoziation mit einem Stream, *unordered*-Flags gesetzt

die darunterliegenden Protokollschichten beliebig vertauscht wird. Als Beispiel für einen solchen Problemfall soll das Szenario aus Abbildung 4.6 dienen. Es soll angenommen werden, dass der B2BUA zunächst die Gültigkeitsdauer eines Pinholes, welches zu einer seit längerem bestehenden Multimedia-Sitzung gehört, mit Hilfe einer *PLC*-Nachricht (mit einer neuen Gültigkeitsdauer ungleich Null) verlängert. Wird vom B2BUA unmittelbar darauf eine SIP *BYE*-Nachricht empfangen, die diese Sitzung beendet, so sendet der B2BUA eine *PLC*-Nachricht mit Argument Null, um das zugehörige Pinhole zu löschen. Im Normalfall wird der SIMCO-Agent im B2BUA zwei positive Antworten erhalten. Wird hingegen die Reihenfolge der *PLC*-Anforderungen vertauscht, so wird die Nachricht zur Verlängerung der Gültigkeitsdauer von der Middlebox erst bearbeitet, nachdem das Pinhole gelöscht wurde. Die dadurch verursachte negative Antwort kann beim SIMCO-Agent einen Alarm auslösen (siehe Abbildung 6.2).

SIMCO selbst hat keine Mechanismen wie z. B. Sequenznummern, um Reihenfolgefehler zu erkennen. Die Transaktions-Bezeichner (TID) müssen zwar eindeutig, aber nicht in zusammenhängender, aufsteigender Reihenfolge vergeben werden. Das Transportschichtprotokoll muss folglich sicherstellen, dass die Reihenfolge von Transaktionen, die sich auf die selbe Policy Ru-



(a) Normalfall

(b) Reihenfolge der Nachrichten in der Transportschicht vertauscht

Abbildung 6.2: Einfluss vertauschter Nachrichten auf SIMCO (Beispiel)

le beziehen, gewahrt bleibt, um Fehlerfälle wie den oben beschriebenen zu vermeiden. Nachrichten, die sich auf verschiedene Policy Rules beziehen, können hingegen vertauscht werden, ohne dass es zu Problemen kommt. SIMCO ist somit ein Beispiel für ein Anwendungsschichtprotokoll, welches von einem Transport mit teilweiser Reihenfolgesicherung profitieren kann.

6.3.2 TCP-basierter Transport

6.3.2.1 Eine TCP-Verbindung

Die SIMCO-Spezifikation [RFC 4540] schreibt vor, dass alle SIMCO-Implementierungen einen TCP-basierten Transport unterstützen müssen; es werden jedoch auch zusätzlich alternative Transportschichtprotokolle erlaubt. Im Regelfall werden alle Nachrichten zwischen einem SIMCO-Agent und Middlebox über eine persistente TCP-Verbindung gesendet. Da TCP einen fehler- und reihenfolgegesicherten Transport bietet, benötigt SIMCO keinerlei Mechanismen zur Erkennung und Behandlung von Nachrichtenverlusten. Dieser vollständig reihenfolgegesicherte Transport, welcher von SIMCO eigentlich nicht benötigt wird, ist jedoch anfällig für Head-Of-Line Blocking.

Da TCP für den Transport eines kontinuierlichen Byte-Stroms statt einzelner Nachrichten entworfen wurde, muss jede SIMCO-Protokollinstanz mit Hilfe der Längen-Felder in den Nachrichtenköpfen den eingehenden Byte-Strom wieder in einzelne Nachrichten zerlegen. Desweiteren muss damit gerechnet werden, dass ein Systemaufruf zum Lesen von einer TCP-Verbindung nur ein Teilstück einer Nachricht liefert. Dieses muss vor der Verarbeitung gepuffert werden, bis der Rest der Nachricht gelesen werden kann. Dazu werden Puffer und ggf. zusätzliche Zustände in den Protokollautomaten benötigt.

6.3.2.2 Mehrere, parallele TCP-Verbindungen

Ein Ansatz zur Verringerung von Head-Of-Line Blocking ist die Verwendung mehrerer paralleler TCP-Verbindungen zwischen zwei Endsystemen. Wenn eine dieser Verbindungen von Head-Of-Line Blocking betroffen ist, können die anderen unbeeinträchtigt Nachrichten transportieren. Um eine teilweise Reihenfolgesicherung zu erzielen, müssen alle Nachrichten, zwischen denen Reihenfolge-Abhängigkeiten bestehen, über die selbe Verbindung transportiert werden.

Für diese Lösung muss kein neues Transportschichtprotokoll spezifiziert werden. Verglichen mit SCTP (s. u.) gibt es allerdings mehrere Nachteile: Da jede dieser Verbindungen getrennt aufgebaut, aufrechterhalten und wieder geschlossen werden muss, ergibt sich ein zusätzlicher Aufwand. Auch der Aufwand für die Verwaltung der Puffer zum Zwischenspeichern von Nachrichten-Teilstücken (s. o.) erhöht sich. Desweiteren muss die Fehlererkennung und -behandlung von jeder TCP-Verbindung unabhängig von den anderen Verbindungen durchgeführt werden. SCTP hingegen kann Fehlersicherung und Überlastabwehr auf den Gesamtverkehr aller Streams anwenden, was effizienter ist [128, 2].

6.3.3 UDP-basierter Transport

Ein UDP-basierter Nachrichtentransport ist von der SIMCO-Spezifikation nicht vorgesehen. Dieser würde erhebliche Änderungen am Protokoll erfordern. Zur Vervollständigung sollen im Folgenden einige Probleme angerissen werden, die dazu gelöst werden müssten.

Ein *SIMCO-über-UDP*-Ansatz müsste zunächst einen Mechanismus zur Erkennung von Übertragungsfehlern spezifizieren, z. B. Sequenznummern, Timer und eine Strategie zur Anforderung von Übertragungswiederholungen, sowie einen Mechanismus zum Erkennen und Verwerfen von Duplikaten. Desweiteren muss die Reihenfolge aller Nachrichten gesichert werden, die sich auf die selbe PID beziehen. Auch für die Übertragung asynchroner Benachrichtigungen von der Middlebox zu den SIMCO-Agents werden entsprechende Mechanismen benötigt.

Letztendlich würde eine solche Lösung ähnlich aussehen wie der UDP-basierte Transport von SIP-Nachrichten, welcher in [RFC 3261] spezifiziert ist. Dennoch würden wesentliche Eigenschaften eines TCP- oder SCTP-basierten Transports fehlen, z. B. Flusskontrolle und Überlastabwehr. Das Problem der Überlastabwehr könnte evtl. mit dem *Datagram Congestion Control Protocol* (DCCP) [RFC 4340], einem Protokoll zum verbindungsorientierten Transport von Nachrichten mit Überlastabwehr, aber ohne Fehler- und Reihenfolgesicherung, gelöst werden; die anderen Probleme des UDP-basierten Transports würden bei diesem Protokoll aber auch auftreten. Ein Hinzufügen entsprechender Mechanismen zu SIMCO würde bedeuten, dass wesentliche Teile der TCP- bzw. SCTP-Protokollmaschinen in der Anwendungsschicht reimplementiert würden. Dies wäre eine mühselige und fehleranfällige Vorgehensweise, die die Prinzipien der Protokollschichtung und der Softwaremodularisierung verletzt.

Verglichen mit TCP oder SCTP bietet der UDP-basierte Transport den Vorteil, dass vor dem Senden der ersten Signalisiernachricht nicht erst eine Transportschichtverbindung aufgebaut werden muss, was eine gewisse Zeit in Anspruch nimmt. Auch werden keine Ressourcen für das Aufrechterhalten des Verbindungszustands benötigt. Diese Vorteile kommen dann zum Tragen, wenn die Nachrichtenrate auf einer bestimmten Signalisierassoziation sehr niedrig ist, z. B. bei der SIP-Signalisierung auf der Teilnehmerschnittstelle. Wird SIMCO zur Steuerung von Firewalls am Netzübergang verwendet, ist dieser Vorteil jedoch von untergeordneter Bedeutung, da die SIMCO-Assoziationen samt den darunterliegenden Transportschichtverbindungen schon beim Hochfahren des Systems und nicht erst beim Eintreffen einer neuen Verbindungsanforderung aufgebaut werden. Da an solchen Netzübergängen die Signalisierlast i. d. R. recht hoch ist, sind hier TCP und SCTP mit ihren effizienteren Mechanismen zur Fehlererkennung und -behandlung (insbesondere *Fast Retransmit*) im Vorteil.

6.3.4 SCTP-basierter Transport

6.3.4.1 Transport über einen Stream mit Reihenfolgesicherung

Wenn alle Nachrichten ohne *Unordered-Flag* über eine SCTP-Assoziation mit nur einem Stream pro Richtung übertragen werden, erhält man einen vollständig reihenfolgegesicherten Transport, der bezüglich Fehler- und Reihenfolgesicherung dem Transport über eine TCP-Verbindung entspricht. Aufgrund des nachrichtenorientierten Dienstzugangspunkts von SCTP werden

in der Anwendungsschicht jedoch keine Puffer zum Zwischenspeichern von in Teilen empfangenen Nachrichten benötigt, was die Implementierung dieser Protokollmaschine vereinfacht.

6.3.4.2 Transport über mehrere Streams

Eine SCTP-Assoziation mit mehreren Streams pro Richtung stellt einen fehlergesicherten Transport mit teilweiser Reihenfolgesicherung zur Verfügung, wenn die Nachrichten ohne *Unordered-Flag* gesendet werden. Falls das Anwendungsschichtprotokoll keine vollständige Reihenfolgesicherung benötigt, kann so der Einfluss von Head-Of-Line Blocking reduziert werden; allerdings muss in der Anwendungsschicht ein Algorithmus zur Auswahl des Streams für eine zu sendende Nachricht vorhanden sein, welcher deren Reihenfolge-Abhängigkeiten berücksichtigt. Abschnitt 6.4 beschreibt den Entwurf einer solchen Anpassungsschicht für SIMCO.

6.3.4.3 Transport ohne Reihenfolgesicherung

Wenn Nachrichten bei der Übergabe an die SCTP-Instanz entsprechend gekennzeichnet werden, werden sie mit gesetztem *Unordered-Flag* transportiert. Empfängerseitig werden diese ohne Behandlung durch den Mechanismus zur Reihenfolgesicherung direkt an die Anwendungsschicht ausgeliefert. Wenn alle Nachrichten in einer Assoziation auf diese Weise transportiert werden, spielt die Zahl der verwendeten Streams keine Rolle, d. h. alle Nachrichten können über einen Stream pro Richtung übertragen werden. Man erhält einen fehlergesicherten Transport ohne Reihenfolgesicherung.

Head-Of-Line Blocking wird in dieser Konfiguration in der Transportschicht komplett vermieden, allerdings muss das Anwendungsschichtprotokoll trotz evtl. auftretender Reihenfolgefehler korrekt arbeiten bzw. diese selbst korrigieren, was u. U. zu zusätzlichen Verzögerungen dort führen kann. SIMCO hat keine eigenen Protokollmechanismen, um mit vertauschten Nachrichten umgehen zu können. Für SIP wird hingegen diese Variante des SCTP-basierten Transports verwendet [RFC 4168].

6.3.4.4 Verwendung von SCTP für andere Anwendungsschichtprotokolle

Eine Übersicht über verschiedene Anwendungsschichtprotokolle, die einen SCTP-basierten Transport unterstützten, wird in [1] gegeben. Alle diese Protokolle wurden entweder explizit für die Verwendung mit SCTP entworfen, oder sie unterstützen sowohl TCP-, als auch UDP-basierten Transport. Eine UDP-Unterstützung impliziert, dass das Anwendungsschichtprotokoll eigene Mechanismen zur Fehler- und ggf. Reihenfolgesicherung haben muss. Dies vereinfacht eine Anpassung an SCTP erheblich, da z. B. alle Nachrichten über einen Stream und mit gesetztem *Unordered-Flag* transportiert werden können. SIMCO ist in dieser Hinsicht besonders, da es ursprünglich nur für einen vollständig reihenfolgegesicherten Transport über TCP entworfen wurde. Deshalb muss besonders auf Abhängigkeiten zwischen SIMCO-Nachrichten geachtet werden, wenn diese zur Verringerung von Head-Of-Line Blocking auf mehrere SCTP Streams verteilt werden sollen.

6.4 Anpassung von SIMCO an SCTP-basierten Transport

SIMCO ist ein Protokoll, welches eine teilweise Reihenfolgesicherung beim Nachrichtentransport benötigt und somit vom reduzierten Head-Of-Line Blocking durch Transport über mehrere SCTP Streams profitieren kann. Auch andere Eigenschaften von SCTP, z. B. Multihoming sind in Umgebungen mit hohen Verfügbarkeitsanforderungen von Vorteil. Zwei Probleme müssen für einen effizienten Einsatz von *SCTP Multistreaming* gelöst werden. Das Erste ist, wie viele Streams verwendet werden sollen, um einerseits einen Vorteil gegenüber TCP bzgl. der Verzögerungen zu haben, andererseits aber nicht unnötig viele Ressourcen zu verwenden. Diese Frage wird in Abschnitt 6.5 detailliert untersucht.

Das zweite Problem ist, wie die Nachrichten möglichst gleichmäßig auf die Streams verteilt werden können, unter Berücksichtigung der Anforderungen an die teilweise Reihenfolgesicherung. Wie in Abschnitt 6.3.1 dargestellt, darf die Transportschicht die Reihenfolge von Nachrichten, die sich auf die selbe Policy Rule beziehen, nicht vertauschen; betreffen die Nachrichten hingegen verschiedene Policy Rules, muss die Reihenfolge nicht aufrecht erhalten werden. Die Grundidee ist daher, zunächst aus je zwei unidirektionalen Streams bidirektionale Stream-Paare zu bilden. Alle SIMCO-Nachrichten, die die Erstellung einer neuen Policy Rule anfordern, werden reihum (oder mit einem ähnlichen, einfachen Verfahren, wie z. B. „Transaktions-Bezeichner modulo Zahl der Streams“) auf die Streams verteilt. Sobald diese Zuordnung einmal festgelegt ist, müssen alle Folgenachrichten, die sich auf die fragliche Policy Rule beziehen, das selbe Stream-Paar benutzen. Die Anforderungen bezüglich der Reihenfolge-Abhängigkeiten werden so erfüllt. Da für das Etablieren, Aufrechterhalten und Löschen einer Policy Rule nur wenige Transaktionen benötigt werden, die in großen Zeitabständen gesendet werden, und weil eine SIMCO-Session zur Steuerung der Medienkomponente einer Firewall an einem stark frequentierten Netzübergang i. d. R. sehr viele Policy Rules kontrolliert, wird mit dieser Methode auch eine hinreichend gleichmäßige Verteilung erreicht.

Die Umsetzung dieser recht naheliegenden Idee wird durch die Tatsache erschwert, dass die Regel-Bezeichner (PID) von der Middlebox und nicht vom SIMCO Agent vergeben werden (siehe Abschnitt 4.4.3 und Abbildung 4.9). Beim Senden der initialen Anforderungs-Nachricht ist dem Agent daher die PID noch nicht bekannt, weshalb er erst auf die Antwort-Nachricht warten muss, die er über den Transaktions-Bezeichner (TID) zuordnen kann. Die Zuordnung von Regel-Bezeichner (PID) zu SCTP Stream-Bezeichner wird in eine Tabelle im Agent eingetragen und kann dann für alle Folge-Nachrichten verwendet werden.

Beim Entwurf der SCTP-Unterstützung für SIMCO wurde besonders darauf geachtet, dass möglichst wenig Zusatzaufwand für das Halten von Zustandsinformationen oder für Berechnungen benötigt werden. Insbesondere muss der SIMCO-Server in der Middlebox keine zusätzlichen Zustandsinformationen pro Regel halten, da die Regeln für das Versenden von asynchronen Benachrichtigungen entsprechend entworfen wurden. Die vollständige Spezifikation, inklusive aller Spezialfälle, wurde in [5] als *Internet Draft* dokumentiert.

6.5 Modellierung von Verzögerungen in der Transportschicht

Im Folgenden werden die in Abschnitt 6.3 vorgestellten Varianten zum Transport von Signalisiernachrichten bezüglich ihres Beitrags zur Antwortzeit von Signalisiertransaktionen untersucht und verglichen. Dabei wird der Einfluss von Paketverlusten auf das Verhalten der Mechanismen zur Fehler- und Reihenfolgesicherung in der Transportschicht modelliert. Es wird insbesondere auf den Effekt des Head-Of-Line Blocking eingegangen, der je nach verwendetem Transportschichtprotokoll und seiner Parametrisierung unterschiedlich stark auftreten kann. Die Untersuchungen werden am Beispiel von SIMCO durchgeführt; dennoch sind die allermeisten Aspekte des Modells unabhängig von SIMCO. Deshalb können die Ergebnisse sehr einfach auch auf andere transaktionsbasierte Signalisierprotokolle übertragen werden.

Verzögerungen durch Mechanismen zur Reihenfolgesicherung sind ein allgemein bekannter Effekt; die vorhandenen Modelle (siehe z. B. [129]) bilden jedoch die speziellen Mechanismen der IP-basierten Transportschichtprotokolle nicht ab. Analytische oder simulationsbasierte Studien zur Leistungsfähigkeit von TCP bzw. SCTP beschäftigen sich hingegen überwiegend mit dem Durchsatz bei der Übertragung großer Datenmengen, nicht jedoch mit den Antwortzeiten bei der Übertragung von Signalisiertransaktionen. Das erste analytische Modell für Verzögerungen durch Head-Of-Line Blocking, welches die spezifischen Mechanismen von TCP und SCTP sowie eine beliebige Anzahl von SCTP Streams abdeckt, wurde in [4] vorgestellt und in [2] sowie [1] bis zur hier vorliegenden Darstellung weiter verfeinert und erweitert. In [1] wird auch ein Überblick über weitere, verwandte Arbeiten gegeben.

6.5.1 Annahmen bei der Modellierung

Es wird davon ausgegangen, dass auf dem Pfad zwischen den Endpunkten eine konstante, unidirektionale Verzögerung Δ und somit eine minimale *Round-Trip Time* $RTT = 2\Delta$ auftritt. Ferner wird davon ausgegangen, dass dieser Pfad unter symmetrischen, zufälligen Paketverlusten mit einer Paketverlustwahrscheinlichkeit p_L leidet. Diese Paketverluste können z. B. durch Übertragungsfehler oder übergelaufene Warteschlangen in überlasteten Routern verursacht werden. In einem sorgfältig dimensionierten Signalisiernetz sollten beide Effekte kaum auftreten und p_L somit sehr klein sein. Dennoch ist es wichtig, den Einfluss von Paketverlusten auf die Verzögerung von Signalisiervorgängen zu quantifizieren, um Richtlinien für die Netzdimensionierung ableiten zu können. Desweiteren werden in dieser Arbeit auch Architekturen untersucht, bei denen die Signalisierung über die Teilnehmerschnittstelle transportiert werden muss. Hierbei kann es sich u. U. um eine drahtlose Übertragung mit einer deutlich höheren Rate von Übertragungsfehlern handeln. Auch die Wirksamkeit von Mechanismen zur Überlastabwehr ist hier nicht immer sichergestellt. Dies kann dazu führen, dass p_L deutlich höher ist als im Kernnetz. Da Paketverluste in beiden Richtungen auftreten können, ist die Wahrscheinlichkeit, dass ein Datenpaket und seine Quittierung erfolgreich übertragen werden, $p_S = (1 - p_L)^2$.

Desweiteren wird in dieser Analyse zunächst davon ausgegangen, dass das Sendefenster die maximale Zahl unbestätigter Segmente nicht beschränkt. Dies bedeutet, dass ein eventuell vorhandener Einfluss der Mechanismen zur Überlastabwehr im Netz (engl. *Congestion Control*) vernachlässigt wird. Im Verlauf der Untersuchung wird gezeigt werden, dass dies für kleine Werte von p_L eine sinnvolle Annahme ist.

Die wichtigste Kenngröße für die Geschwindigkeit transaktionsbasierter Signalisierprotokolle ist die Transaktions-Antwortzeit R. Diese muss z. B. beim SIMCO-Protokoll möglichst gering sein, um die Rufaufbauverzögerung zu minimieren, welche von den Teilnehmern als störend empfunden wird. Im Transportschichtprotokoll vorhandene Mechanismen zur Fehlerbehandlung sowie ein von diesen ggf. verursachtes Head-Of-Line Blocking vergrößern die unidirektionale Verzögerung beim Pakettransport $RTT/2$ um eine zusätzliche Komponente W, die von der Wahl des Transportschichtprotokolls und dessen Parametrisierung abhängt. Die beiden Transport-Richtungen werden getrennt betrachtet, d. h. es wird davon ausgegangen, dass in Folge von Paketverlusten auftretende Verzögerungen in der einen Richtung keinen Einfluss auf den Transport in der Gegenrichtung haben. Da Signalisiertransaktionen aus einer Anfrage und einer Antwort bestehen und weil Head-Of-Line Blocking beim Nachrichtentransport in beiden Richtungen auftreten kann, folgt die mittlere Transaktions-Antwortzeit als

$$R = RTT + 2W + \delta \,, \tag{6.1}$$

wobei δ die Dauer der Verarbeitung im Server (z. B. Middlebox mit SIMCO) angibt.

6.5.2 Modellierung der Signalisierlast

Es wird davon ausgegangen, dass die betrachtete Signalisierverbindung eine mittlere Signalisierlast von λ_T Transaktionen pro Zeiteinheit transportiert, die einer sehr großen Zahl parallel und unabhängig ablaufender Prozesse entspringt, z. B. der Verbindungssteuerung für eine sehr große Zahl von Teilnehmern. Jeder einzelne dieser Prozesse erzeugt nur eine geringe Rate von Transaktionen, die keine Reihenfolge-Abhängigkeiten zu Transaktionen anderer Prozesse haben. Bei Transaktionen, die aus einer Anfrage und einer Antwort bestehen, entspricht diese Transaktionsrate einer mittleren Zwischenankunftszeit (engl. *Interarrival Time*, IAT) der Nachrichten von $d = \frac{1}{\lambda_\mathrm{T}}$ pro Richtung.

In Gleichung (5.3) wird die Zwischenankunftszeit d der SIMCO-Transaktionen hergeleitet, die benötigt werden, um eine große Zahl von parallelen Telefongesprächen über eine Firewall hinweg zu ermöglichen.

6.5.3 Mechanismen zur Fehlererkennung bei SCTP

Ähnlich wie auch TCP hat auch SCTP zwei getrennte Mechanismen, um Paketverluste zu erkennen und eine entsprechende Übertragungswiederholung auszulösen: den so genannten *Fast Retransmit* (FRTX) und ein auf einer Zeitüberwachung (engl. *Retransmission Timeout*, RTO) basierender Mechanismus. Nach einer kurzen Beschreibung dieser Mechanismen wird ihr Einfluss auf die Ende-zu-Ende-Verzögerung beim Transport von Signalisiernachrichten untersucht.

6.5.3.1 Fast Retransmit

Eine SCTP-Instanz sendet bei Empfang eines Datenblocks (*Data Chunk*) i. d. R. eine Quittierungsnachricht (engl. *Selective Acknowledgment*, SACK), die den Empfang bestätigt und ggf.

eine Liste der Sequenznummern (engl. *Transmission Sequence Number*, TSN) noch ausstehender, älterer Blöcke enthält. Dabei spielt es keine Rolle, welchem Stream die Datenblöcke zugeordnet sind, d. h. die Quittierung einer erfolgreich auf einem Stream übertragenen Nachricht kann den Sender auch über noch ausstehende Nachrichten in anderen Streams informieren. Da Paketverluste in beiden Richtungen auftreten können, ist die Wahrscheinlichkeit, dass die Quittierung für ein Datenpaket beim Sender ankommt, $p_S = (1-p_L)^2$. Eine SCTP-Instanz überträgt einen Datenblock erneut, wenn drei aufeinanderfolgende SACK-Nachrichten die entsprechende Sequenznummer als fehlend melden [RFC 4460].

Dieser so genannte *Fast Retransmit* ist in Abbildung 6.3 illustriert. In dieser Abbildung wird davon ausgegangen, dass die SCTP-Assoziation $N = 3$ Streams in der betrachteten Senderichtung hat und dass die Datenblöcke reihum darauf verteilt werden, d. h. die Blöcke mit den Sequenznummern 0, 3, 6, ... werden auf Stream Nr. 0 übertragen, die Blöcke mit den TSN 1, 4, 7, ... bzw. 2, 5, 8, ... auf den Streams Nr. 1 bzw. Nr. 2. Um die Abbildung übersichtlicher zu halten, sind nicht alle dieser Nachrichten eingezeichnet. Zur Vereinfachung der folgenden Berechnungen wird angenommen, dass die Zwischenankunftszeit d der Nachrichten konstant ist.

In dem in der Abbildung dargestellten Beispiel wird davon ausgegangen, dass das IP-Paket, welches den Datenblock mit der Sequenznummer 0 trägt, einem Paketverlust zum Opfer fällt. Dabei kennzeichnet t_0 den Zeitpunkt, an dem das Paket abgesendet wurde; zum Zeitpunkt t_1 wäre es bei fehlerfreier Übertragung beim Empfänger angekommen. Zum Zeitpunkt $t_2 = t_0 + RTT + 3d$ sind beim Sender drei Quittierungen für Folgepakete angekommen. Hierbei spielt es keine Rolle, zu welchem Stream diese Folgepakete gehören. Eine Übertragungswiederholung wird ausgelöst und zum Zeitpunkt $t_3 = t_2 + RTT/2$ kommt der Datenblock Nr. 0 tatsächlich beim Empfänger an. Erst jetzt kann dessen SCTP-Instanz alle zwischenzeitig empfangenen und in einem Puffer (engl. *Resequencing Queue*) zwischengespeicherten Datenblöcke des Stream Nr. 0 in der richtigen Reihenfolge an die Anwendungsschicht-Instanz übergeben.

Wie in Abbildung 6.3 dargestellt, beträgt die minimale Dauer zur Erkennung des Paketverlustes $D_{\mathrm{FRTX,min}} = t_2 - t_0 = RTT + 3d$. Dies kann jedoch auch länger dauern, da auch Folgepakete und deren Quittierungen verloren gehen können. Die Wahrscheinlichkeit, dass die dritte Quittierung beim Sender ankommt, nachdem genau i Datenblöcke gesendet wurden, ist

$$P(i) = \binom{i-1}{2} p_S^2 (1-p_S)^{i-1-2} \cdot p_S = \frac{(i-1)(i-2)}{2} p_S^3 (1-p_S)^{i-3} \,, \tag{6.2}$$

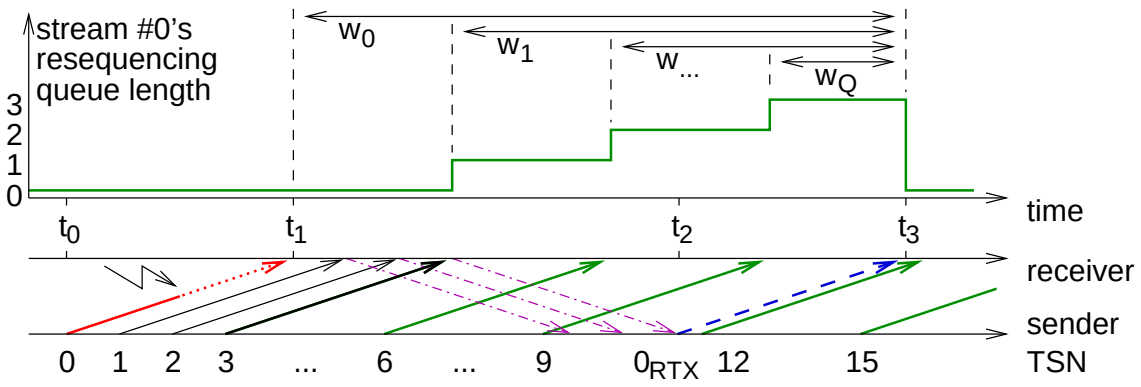


Abbildung 6.3: Wartezeiten bei Fast Retransmit

da auf eine Bernoulli-Kette der Länge $(i-1)$ mit genau 2 Erfolgen (an beliebiger Position) unmittelbar ein dritter Erfolg folgen muss. Daraus folgt

$$D_{\mathrm{FRTX}} \approx RTT + d\sum_{i=3}^{\infty} P(i)\,i = RTT + \frac{3d}{(1-p_{\mathrm{L}})^2}\ . \tag{6.3}$$

6.5.3.2 *Retransmission Timeout*

Zusätzlich zu dem oben beschriebenen Mechanismus verwendet SCTP eine Zeitüberwachung zur Erkennung von Paketverlusten. Dies ist in Abbildung 6.4 illustriert. Ein *Timer* wird immer dann neu gestartet, wenn eine Quittierung eintrifft, die den Wert nach oben verschiebt, bis zu dem alle davor liegenden Sequenznummern lückenlos quittiert wurden. Läuft dieser Timer ab, so werden noch nicht quittierte Datenblöcke erneut übertragen. Die Dauer vom Senden eines Datenblocks bis zur Erkennung, dass dieser verloren gegangen ist, ist bei diesem Mechanismus

$$D_{\mathrm{RTO}} = RTO + \max\left(RTT - d, 0\right)\ . \tag{6.4}$$

Der für den Timer verwendete Startwert RTO wird von optimierten Protokoll-Implementierungen an die geschätzte Paketumlaufzeit RTT angepasst und ist daher nicht notwendigerweise konstant. Bei Netzen mit geringen Latenzen entspricht er jedoch i. d. R. dem in der SCTP-Protokollspezifikation auf 1 s festgelegten Minimalwert.

6.5.3.3 *Gemeinsame Betrachtung beider Mechanismen*

Die wiederholte Übertragung eines Data Chunk wird ausgelöst, sobald einer der beiden Mechanismen zur Erkennung von Paketverlusten anspricht. Die Zeitdauer dafür ist

$$D = \min\left(D_{\mathrm{FRTX}}, D_{\mathrm{RTO}}\right)\ . \tag{6.5}$$

Falls auch die erneut übertragenen Datenpakete verloren gehen, tritt grundsätzlich ein *Retransmission Timeout* auf. Der Verlust mehrerer aufeinanderfolgender Pakete kann überlappende Übertragungswiederholungen auslösen. Diese sind durch ein einfaches Modell schwer zu erfassen. Beide Effekte werden hier vernachlässigt, da sie bei geringer Paketverlustwahrscheinlichkeit p_{L} nur selten auftreten.

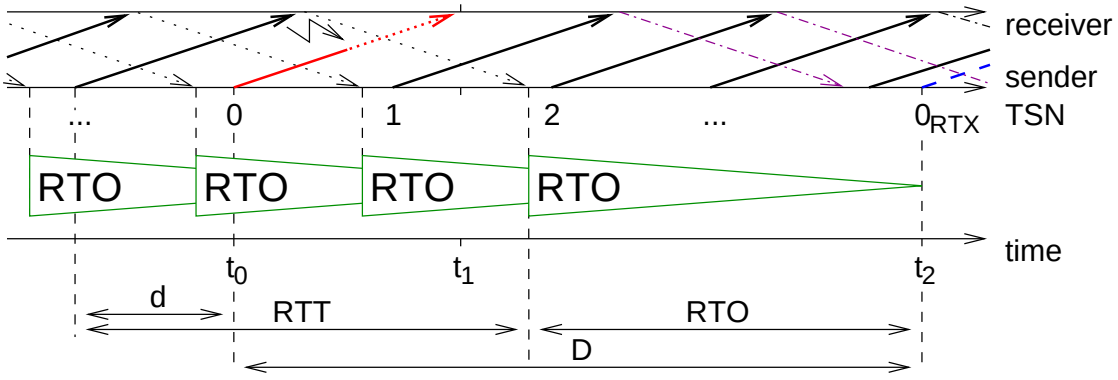


Abbildung 6.4: Wartezeiten bei Timeout-basierter Erkennung von Paketverlusten

6.5.4 Reihenfolgesicherung bei SCTP

Datenblöcke, die im selben Stream transportiert werden und dabei vor einem älteren, noch ausstehenden Block beim Empfänger eintreffen, müssen in einer Resequencing Queue in der SCTP-Instanz zwischengespeichert werden, um die Auslieferung an die Anwendungsschicht in der richtigen Reihenfolge zu ermöglichen. Während der Zeitspanne D, die für die Erkennung des Paketverlustes und das Auslösen der Übertragungswiederholung benötigt wird, werden vom Sender

$$Q = \lfloor \frac{D}{dN} \rfloor \tag{6.6}$$

weitere Blöcke abgesendet, die nach einer als verlustfrei angenommenen Übertragung alle beim Empfänger entsprechend gepuffert werden müssen. Die Wartezeit des ersten Blocks, der nach dem verlorengegangenen Block eintrifft, beträgt $w_1 = D - Nd$, die der darauf folgenden Blöcke entsprechend $w_2 = D - 2Nd, \ldots, w_Q = D - QNd$ (siehe Abbildung 6.3). Die Dauer der Übertragungswiederholung kann als „virtuelle Wartedauer“ $w_0 = t_3 - t_1 = t_2 - t_0 = D$ des verlorengegangenen Blocks modelliert werden.

Die mittlere Wartezeit aller Blöcke ist die Summer aller w_i dividiert durch die Anzahl der Blöcke, die in einer Periode zwischen zwei Verlust-Ereignissen übertragen werden. Diese Anzahl ist $1/p_{\mathrm{L}}$. Somit ergibt sich eine zusätzliche mittlere Verzögerung beim Ende-zu-Ende-Transport von

$$W = p_{\mathrm{L}} \sum_{i=0}^{Q} w_i = p_{\mathrm{L}} \left((Q+1) \cdot D - \tfrac{Q(Q+1)}{2} Nd \right) . \tag{6.7}$$

6.5.5 Optimale Anzahl von SCTP Streams

Eine wichtige Fragestellung, die schon in Abschnitt 6.4 aufgeworfen wurde, betrifft die optimale Anzahl von Streams. Die Verwendung einer sehr großen Zahl von Streams kann ineffizient sein, da dafür Ressourcen wie z. B. Speicher für Sequenznummern-Zähler in den Endpunkten benötigt werden. Ein Head-Of-Line Blocking kann komplett vermieden werden, wenn in einem blockierten Stream keine weiteren Datenblöcke in der Resequencing Queue des Empfängers ankommen, bevor die Übertragungswiederholung abgeschlossen ist, d. h. wenn $Q = 0$. Dies ist erfüllt für $Nd \geq D$, d. h. für $N \geq M$ mit $M = \lceil \frac{D}{d} \rceil$. Laut Gleichung (6.3) hat die Paketverlustwahrscheinlichkeit p_{L} nur einen geringen Einfluss auf D. Für Senderaten, die hoch genug sind, dass Fast Retransmit zum Einsatz kommen kann, folgt die optimale Anzahl von Streams daraus als

$$M \approx \lceil RTT \cdot \lambda_{\mathrm{T}} + 3 \rceil , \tag{6.8}$$

und daraus mit Gleichung (5.3) für SIMCO

$$M_{\mathrm{SIMCO}} \approx \left\lceil RTT \cdot 2\lambda_{\mathrm{C}} \left(2 + (e^{L/h} - 1)^{-1} \right) + 3 \right\rceil . \tag{6.9}$$

6.5.6 Übertragung mit SCTP ohne Reihenfolgesicherung

Bei SCTP können einzelne Data Chunks mit dem „u“-Flag (*unordered*) markiert werden. Diese werden beim Empfänger ohne Reihenfolgesicherung direkt an die oberen Protokollschich-

ten ausgeliefert. Wenn alle Datenblöcke in diesem Modus transportiert werden, gilt $w_0 = D$, $w_1 = w_2 = \cdots = 0$; mit Gleichung (6.7) folgt

$$W = p_{\mathrm{L}} \cdot D \,. \tag{6.10}$$

Die Anzahl der Streams spielt dabei keine Rolle, d. h. es kann $N = 1$ gewählt werden.

Diese Betriebsart setzt allerdings voraus, dass zwischen den Nachrichten keinerlei Abhängigkeiten bzgl. der Reihenfolge bestehen, oder dass die über SCTP liegenden Protokollschichten selbst Mechanismen zur Reihenfolgesicherung haben, welche ihrerseits Verzögerungen verursachen können. Da SIMCO sich auf das Vorhandensein solcher Mechanismen in der Transportschicht verlässt, ist dieser Modus für SIMCO nicht geeignet.

6.5.7 Anpassung des Modells an TCP

Die bei SCTP verwendeten Mechanismen zur Fehlererkennung und zum Auslösen von Übertragungswiederholungen entsprechen denen von TCP; laut SCTP-Spezifikation sind die dabei relevanten Parameter gleich wie bei TCP zu wählen. Von einem theoretischen Standpunkt betrachtet sollte die Verzögerung beim Transport der Signalisiernachrichten über eine TCP-Verbindung demnach identisch sein mit der beim Transport über eine SCTP-Assoziation mit einem Stream und nicht deaktivierter Reihenfolgesicherung. Somit kann ein Modell für die Verzögerung beim Transport von Signalisiernachrichten über TCP einfach erhalten werden, indem $N = 1$ in Gleichung (6.7) eingesetzt wird.

Prinzipiell ist es möglich, Signalisiernachrichten auf mehrere parallele TCP-Verbindungen zu verteilen, analog zur Verteilung auf SCTP Streams. Auch so wäre bei einem evtl. auftretenden Head-Of-Line Blocking nur ein Teil der Nachrichten betroffen. Anders als die SCTP Streams würden diese parallelen TCP-Verbindungen jedoch Fehler unabhängig voneinander detektieren, d. h. eine Quittierung in einer TCP-Verbindung könnte nicht auf noch ausstehende Segmente einer anderen Verbindung hinweisen. Dadurch vergeht mehr Zeit bis zum Auslösen des Fast Retransmit, die Ende-zu-Ende-Verzögerung ist somit höher als bei einer SCTP-Assoziation. Eine genauere Untersuchung dieses Ansatzes kann in [2] gefunden werden.

6.5.8 Modell für UDP

Zur Vervollständigung soll noch ein UDP-basierter Transport untersucht werden. Wie bereits in Abschnitt 6.3.3 erläutert, würde dies erhebliche Änderungen an SIMCO voraussetzen, die auch nicht standardisiert wurden. Deshalb soll für die folgende Analyse von Mechanismen ausgegangen werden, wie sie für den Transport von SIP über UDP in [RFC 3261] spezifiziert sind.

In diesem Szenario wird eine Anforderung von der Anwendungsschicht-Protokollinstanz des Clients erneut gesendet, wenn innerhalb einer bestimmten Zeitdauer keine entsprechende Antwort empfangen wurde. Diese Zeitdauer wird zunächst auf $TO = 500\,\mathrm{ms}$ initialisiert und nach jedem fehlgeschlagenen Übertragungsversuch verdoppelt. Die Signalisiertransaktion ist abgeschlossen, wenn nach i erfolglosen Versuchen ein Übertragungsversuch erfolgreich war. Dies geschieht mit einer Wahrscheinlichkeit $q_i = (1 - p_{\mathrm{S}})^i \cdot p_{\mathrm{S}}$. Die Wartezeit vor dem Senden des

letzten, erfolgreichen Versuchs beträgt $W_i = TO\left(2^i - 1\right)$. Die mittlere Antwortzeit einer Signalisiertransaktion folgt daraus als

$$R_{\mathrm{UDP}} = RTT + \delta + \sum_{i=0}^{\infty} q_i \cdot W_i = RTT + \delta + TO\frac{1 - p_{\mathrm{S}}}{2\,p_{\mathrm{S}} - 1}\ . \tag{6.11}$$

6.6 Vergleich verschiedener Transport-Konfigurationen

In diesem Abschnitt werden die in Abschnitt 6.5 vorgestellten Modelle sowie Messungen an dem in Abschnitt 5.2 beschriebenen SIMCO-Prototypen verwendet, um unterschiedliche Konfigurationen verschiedener Transportschichtprotokolle bezüglich Verzögerung beim Transport von Signalisiertransaktionen zu vergleichen.

Eine Übersicht über die Testumgebung für die Messungen wird in Abschnitt 5.2.6 gegeben. Die SIMCO-Software wurde unter den Betriebssystemen Linux und Solaris installiert. Für die Messungen unter Linux Version 2.6.16 wurde das *lksctp*-Modul oder die in dieser Linux-Version standardmäßig enthaltene TCP-Implementierung genutzt, welche selektive Bestätigungen (engl. *Selective Acknowledgment*, SACK) und den *BIC*-Algorithmus zur Überlastabwehr verwendet. In beiden Fällen wurde die *Nodelay*-Option der Socket-Schnittstelle verwendet. Für Solaris 10 werden hier nur die Messergebnisse für TCP-basierten Transport vorgestellt, da die Messungen zu SCTP von längeren Stillständen der SCTP-Assoziation verfälscht wurden, die gelegentlich ohne ersichtlichen Grund auftraten. Verglichen mit TCP ist SCTP immer noch ein sehr junges Protokoll, dessen Implementierungen noch wenig ausgereift und optimiert sind.

Grundlage für die Parametrisierung der Messungen war ein angenommenes Szenario, bei dem ein zentraler Softswitch eine Medienkomponente an einem stark frequentierten Netzübergang steuert. Sofern nicht anders angegeben, erzeugte der Lastgenerator neue Pinholes mit einer negativ-exponentiell verteilten Zwischenankunftszeit mit Mittelwert $\frac{1}{\lambda_{\mathrm{u}}} = 30\,\mathrm{ms}$ und negativ-exponentiell verteilter Gültigkeitsdauer mit Mittelwert $h = 180\,\mathrm{s}$. Mit einer Auffrischung der Gültigkeitsdauer in konstanten Abständen von $L = 120\,\mathrm{s}$ folgt aus Gleichung (5.3) eine mittlere Zwischenankunftszeit der SIMCO-Transaktionen von $d \approx 10\,\mathrm{ms}$. Mit zwei Pinholes pro Multimedia-Sitzung entspricht dies $m = 3000$ gleichzeitigen Sitzungen (vgl. Abschnitt 5.2.5.2).

Die tatsächliche Filterung einer so großen Zahl simultaner Medienströme übersteigt die Leistungsfähigkeit eines software-basierten Paketfilters (vgl. Abschnitt 5.5). Daher wurden bei den hier vorgestellten Messungen die Pinholes nur im „Hauptprozess“ des SIMCO-Servers verwaltet, aber nicht in den Betriebssystemkern eingebracht; es wurden auch keine Medienströme oder Angriffsverkehr gesendet. Die Leistungsfähigkeit kommerziell verfügbarer, hardware-basierter Medienkomponenten übersteigt hingegen die hier angenommenen Werte; beispielsweise wird in [130] von erfolgreichen Messungen – unter Berücksichtigung der Medienströme – an einem Session Border Controller mit den Parametern $\lambda_{\mathrm{C}} = 440\,\frac{1}{\mathrm{s}}$, $h = 180\,\mathrm{s}$, $m = 79200$ berichtet.

Die vom WAN-Emulator zu erzeugende Paketverzögerung wurde i. d. R. auf $\Delta = 10\,\mathrm{ms}$ je Richtung eingestellt. Alle angegebenen Messergebnisse sind die Mittelwerte über die Antwortzeiten der *PER*-Transaktionen während einer Messdauer von 1000 s, die erst begonnen wurde, nachdem der Messaufbau den eingeschwungenen Zustand erreicht hatte. In [3] sind die Ergebnisse

weiterer Messungen dokumentiert, die einen größeren Parameterraum abdecken, mit sehr ähnlichen Ergebnissen. In diesem Dokument ist auch der Messaufbau inklusive der Versionsnummern aller verwendeten Soft- und Hardwarekomponenten beschrieben.

6.6.1 Einfluss von Paketverlusten auf SCTP

Zunächst wird die Verwendung mehrerer SCTP Streams zum Transport von SIMCO-Nachrichten untersucht (vgl. Abschnitt 6.4). Abbildung 6.5 zeigt die mittlere Antwortzeit von SIMCO-Transaktionen als Funktion der Paketverlustwahrscheinlichkeit p_L. Es zeigt sich, dass die Verwendung von mehr als einem Stream pro Richtung die Antwortzeit R deutlich verbessern kann, selbst bei moderaten Paketverlustwahrscheinlichkeiten wie z. B. $p_L = 2\,\%$. Der Vorteil wird bei höheren p_L noch größer, allerdings sollten solche Situationen in sinnvoll dimensionierten Signalisiernetzen normalerweise nicht vorkommen.

Abbildung 6.6 zeigt die mittlere Antwortzeit als Funktion der Anzahl verwendeter Streams N. Die Messwerte stimmen mit den Vorhersagen des analytischen Modells entsprechend Gleichungen (6.1) und (6.7) sehr gut überein, wenn für die lokale Bearbeitungszeit im SIMCO-Server ein Wert von $\delta = 0.5\,\text{ms}$ angenommen wird. Für Paketverlustwahrscheinlichkeiten über ein Prozent unterschätzt das Modell die Antwortzeiten leicht. Dies liegt vermutlich am Einfluss überlappender Fast Retransmit- bzw. Timeout-Phasen, die bei hohen Paketverlustwahrscheinlichkeiten nicht ignoriert werden können. Abbildung 6.6 bestätigt auch, dass eine Erhöhung der Stream-Anzahl N über den optimalen Wert M hinaus (für die hier gewählte Parametrisierung ergibt sich $M = 6$) zu keiner nennenswerten Verbesserung der Antwortzeit führt.

Die komplementäre, kumulierte Häufigkeitsverteilung (engl. *Complementary Cumulative Distribution Function*, CCDF) der Antwortzeit ist in Abbildung 6.7 für eine Paketverlustwahrscheinlichkeit $p_L = 1\,\%$ dargestellt. Wenn die Anzahl der Streams N vergrößert wird, nähert

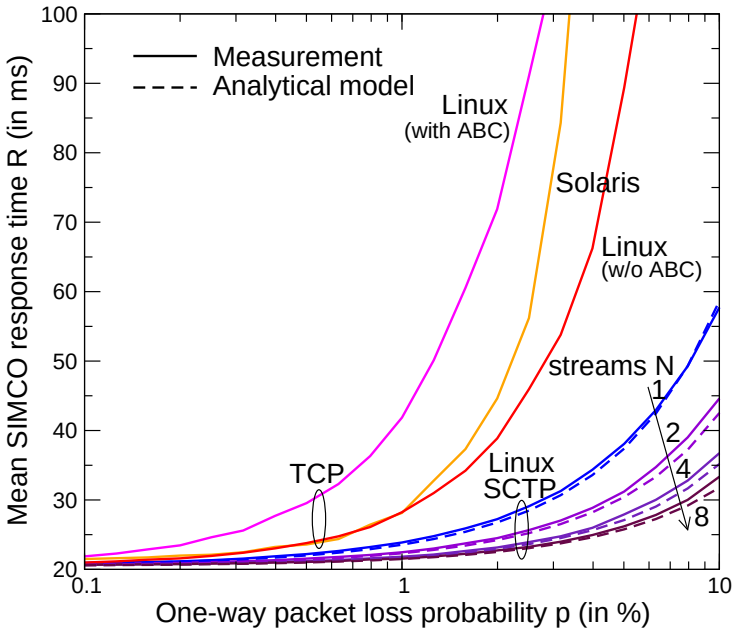


Abbildung 6.5: SIMCO-Antwortzeit bei verschiedenen Transportprotokollen

sich die CCDF asymptotisch einem Grenzwert an, der dem Transport ohne Reihenfolgesicherung entspricht. Für $N \geq M$ haben Multistreaming und Transport ohne Reihenfolgesicherung die selben Antwortzeiten. Die kleinen Stufen in der CCDF haben eine Breite von ca. 4 ms, die auf den Takt des Schedulers im Linux-Betriebssystemkern von 250 Hz zurückgeführt werden können. Bei $R = 30$ ms kann ein etwas größerer Sprung beobachtet werden. Eine genauere Untersuchung während der Messungen aufgezeichneter Paket-Traces ergab, dass die SCTP-Implementierung von Linux gelegentlich zwei Nachrichten zu einem Paket bündelt, obwohl dieses Verhalten mit der *Nodelay*-Option eigentlich deaktiviert sein sollte. Dabei wird die erste Nachricht um eine Transaktions-Zwischenankunftszeit verzögert, d. h. im Mittel um $d \approx 10$ ms.

6.6.2 Einfluss von Paketverlusten auf TCP

Abbildung 6.5 zeigt auch die Messergebnisse für TCP, sowohl mit Linux, als auch mit Solaris. Aufgrund der Tatsache, dass die wesentlichen Mechanismen zur Fehler- und Reihenfolgesicherung von SCTP praktisch identisch zu denen von TCP spezifiziert sind, sollte man erwarten, dass ein TCP-basierter Transport zu ähnlichen Antwortzeiten führt wie SCTP mit einem Stream pro Richtung. Den hier durchgeführten Messungen zufolge ist die mittlere Antwortzeit jedoch erheblich größer, insbesondere für Paketverlustwahrscheinlichkeiten $p_L > 1\,\%$.

Bei Linux kann die Antwortzeit reduziert werden, indem *Appropriate Byte Counting* (ABC) deaktiviert wird. ABC [RFC 3465] ist eine experimentelle Erweiterung der TCP-Mechanismen zur Überlastabwehr, die in der für die Messungen verwendeten Linux-Version 2.6.16 standardmäßig aktiviert ist. Es wurde jedoch festgestellt, dass ABC Anwendungen unfair behandelt, die nur kleine Datenmengen senden wollen. Die hier betrachtete SIMCO-Signalisierung ist ein Beispiel für eine solche Anwendung. Aufgrund dieser Probleme wurde ABC ab Linux-Version 2.6.18 wieder standardmäßig deaktiviert. Ohne ABC erreicht Linux ähnliche Werte wie Solaris, trotz der Tatsache, dass die beiden Betriebssysteme unterschiedliche TCP-Implementierungen

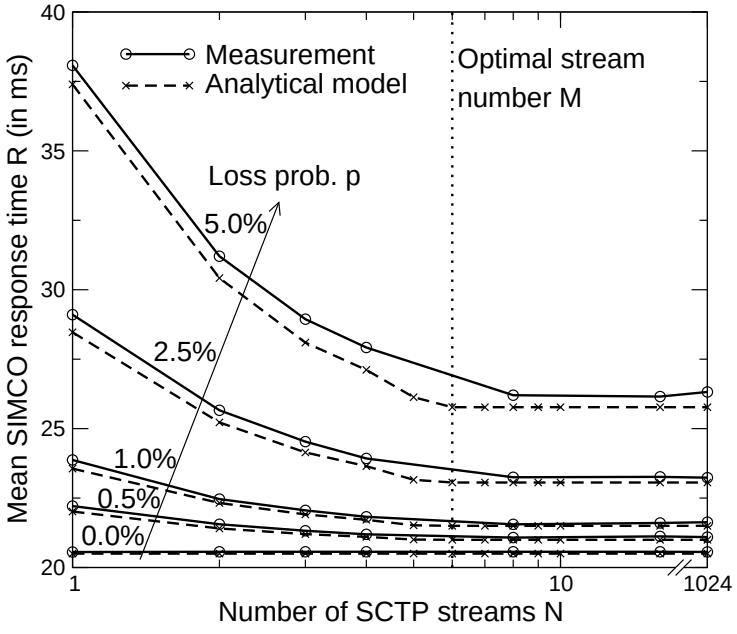


Abbildung 6.6: Einfluss der Anzahl der SCTP-Streams auf die SIMCO-Antwortzeit

verwenden und auf Rechnern betrieben wurden, die sich bzgl. Hardware-Architektur und Leistungsfähigkeit deutlich unterscheiden. Beide TCP-Implementierungen konnten die angebotene Last von 100 Transaktionen/s ab $p_L > 7\,\%$ nicht transportieren, was sich durch Überlauf der Sendepuffer bemerkbar machte. Dies kann mit der Überlastregelung von TCP erklärt werden, die den Durchsatz einer Verbindung limitiert, wenn Paketverluste auftreten.

Die CCDF in Abbildung 6.7 zeigt eine nicht vernachlässigbare Wahrscheinlichkeit großer Verzögerungen, selbst bei einer recht geringen Paketverlustwahrscheinlichkeit ($p_L = 1\,\%$). Das 99 %-Quantil der Antwortzeit liegt in der Größenordnung von 200 ms. Für Solaris TCP gibt es eine signifikante Wahrscheinlichkeit von Verzögerungen von etwa 500 ms. Diese werden scheinbar von Retransmission Timeouts (vgl. Abschnitt 6.5.3.2) mit einer minimalen Dauer von $RTO \approx 500\,\text{ms}$ verursacht. Linux verwendet einen kleineren Wert $RTO \approx RTT + 200\,\text{ms}$ [131], für die Zeitüberwachung (d. h. im hier betrachteten Szenario $RTO \approx 220\,\text{ms}$) und leidet daher nicht so stark unter langen Verzögerungen. Dennoch treten auch hier kleine Verzögerungen mit einer nicht zu vernachlässigenden Wahrscheinlichkeit auf. Eine Analyse der Paket-Traces zeigt, dass die Linux TCP-Implementierung Datensegmente manchmal nicht sofort abschickt, sondern zu größeren Paketen bündelt, obwohl die *Nodelay*-Option gesetzt und der so genannte *Nagle*-Algorithmus damit eigentlich deaktiviert sein sollte.

Um eine Beeinflussung durch die Bearbeitung der SIMCO-Nachrichten ausschließen zu können, wurden die Messungen mit einem sehr einfachen Testprotokoll wiederholt, dessen Nachrichten neben einem Zeitstempel nur aus Füll-Bytes bestehen, die nicht bearbeitet werden. Die Protokollinstanzen verwenden so wenig wie möglich zusätzlichen Code um die Systemaufrufe zum Senden bzw. Empfangen von Nachrichten herum. Die so erzielten, nahezu identischen Ergebnisse [2] bestätigen die Unterschiede zwischen TCP und SCTP und zeigen, dass die beobachteten Effekte nicht spezifisch für den Transport von SIMCO-Nachrichten sind.

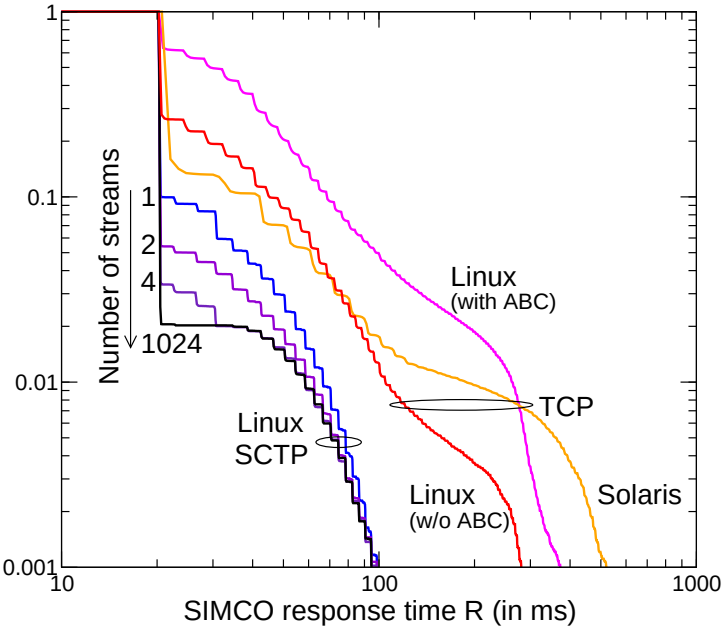


Abbildung 6.7: Komplementäre, kumulierte Häufigkeitsverteilung der SIMCO-Antwortzeit ($p_L = 1\%$, $d = 10\text{ms}$)

6.6.3 Variable Last

Abbildung 6.8 zeigt die mittlere Transaktions-Antwortzeit über der angebotenen Last λ_T, welche von einer recht niedrigen Rate von einer Transaktion pro Sekunde auf bis zu 10.000 Transaktionen/s erhöht wurde. Die Paketverlustwahrscheinlichkeit wurde im WAN-Emulator konstant auf $p_L = 1\,\%$ je Richtung eingestellt. Das Diagramm kann in drei Bereiche eingeteilt werden:

1. Für geringe Transaktionsraten λ_T läuft nach einem Paketverlust i. d. R. der Timer zur Zeitüberwachung ab, bevor ein Fast Retransmit ausgelöst wird. SCTP schneidet in diesem Bereich aufgrund des hohen RTO-Wertes von 1 s am schlechtesten ab. Mit steigender Transaktionsrate steigt in diesem Bereich die mittlere Antwortzeit R, da zusätzlich Head-Of-Line Blocking auftritt, außer wenn hinreichend viele SCTP Streams verwendet werden.

2. Für $\lambda_T > \alpha = \frac{3}{RTO}$ erlaubt der Fast Retransmit-Algorithmus eine deutlich schnellere Fehlererkennung und reduziert somit die Ende-zu-Ende-Verzögerung erheblich. Bei Linux SCTP und Solaris TCP stimmen die Messergebnisse bis zu einer Senderate von ca. $100\,\frac{1}{s}$ gut mit den Vorhersagen des Modells überein, welche im Bereich oberhalb α mit dem Modell für SCTP mit einem Stream zusammenfallen. Bei Linux TCP (mit ABC) kann eine zusätzliche Verzögerung von 10 ms bis 20 ms beobachtet werden.

3. Für hohe Transaktionsraten zeigt sich ein stark abweichendes Verhalten: Bei Linux SCTP liegen die Antwortzeiten nur sehr knapp über der Paketverzögerung RTT, sofern genügend Streams verwendet werden. Die maximale Senderate, die mit dieser immer noch als „experimentell“ gekennzeichneten Version der SCTP-Implementierung übertragen werden konnte, war ca. 2.000 Transaktionen/s. Bei Solaris TCP steigen die Verzögerungen für

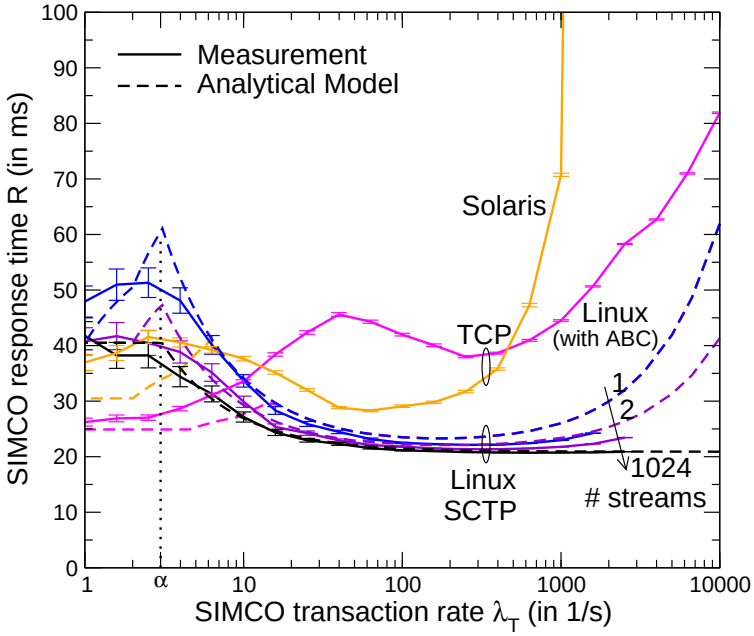


Abbildung 6.8: SIMCO-Antwortzeit als Funktion der Last ($p_L = 1\%$)

$\lambda_T > 500\,\frac{1}{s}$ sehr deutlich an, bevor es zu Überläufen des Sendepuffers und einem Abbruch der Messung kommt. Hier scheint der Algorithmus zur Überlastabwehr die Senderate zu begrenzen. Dieser Effekt wird von dem hier vorgestellten Modell nicht berücksichtigt; daher wird die Antwortzeit diesem Bereich unterschätzt. Interessanterweise kann die TCP-Implementierung von Linux mehr als die vierfache Last tragen, bevor die Antwortzeiten steigen. Die Algorithmen und Parametrisierung von Linux scheinen einen höheren Durchsatz mit höheren Latenzen zu erkaufen.

6.6.4 Hypothetischer UDP-Transport

Wie in Abschnitt 6.3.3 erläutert, würde ein UDP-basierter Transport erhebliche Änderungen an SIMCO erfordern, bei offensichtlichen Nachteilen bzgl. Implementierungsaspekten. Er wurde deshalb bei der Erstellung der prototypischen Implementierung nicht berücksichtigt. Dementsprechend kann für den Vergleich der Leistungsfähigkeit hier nur das analytische Modell für UDP nach Gleichung (6.11) mit dem Modell für TCP und SCTP nach Gleichungen (6.1) und (6.7) verglichen werden. Tabelle 6.3 zeigt beispielhaft die damit vorhergesagten mittleren Transaktionsantwortzeiten bei verschiedenen Transportschichtprotokollen und zwei verschiedenen Transaktionsraten.

Geht man von einer Strategie zum Erkennen von Paketverlusten aus, wie sie für den UDP-basierten Transport von SIP in [RFC 3261] standardisiert wurde, beginnt die erste Übertragungswiederholung jeder verlorenen gegangenen Nachricht nach Ablauf eines Timers von $TO = 500\,\text{ms}$. Dieser Wert liegt in der selben Größenordnung wie der minimale RTO-Wert von TCP und SCTP. Daher ergibt das einfache Erkennungsschema auf Basis von UDP ähnliche mittlere Verzögerungen wie TCP- oder SCTP-basierter Transport, solange die Senderate so niedrig ist, dass auch diese auf RTO-basierte Erkennung zurückgreifen müssen. Bei moderaten bis hohen Senderaten (z. B. 100 Transaktionen/s) können sie hingegen vom Fast Retransmit-Algorithmus profitieren, so dass die mittlere Antwortzeit deutlich unter der des UDP-basierten Transports liegt (siehe Tabelle 6.3).

Prinzipiell wäre es natürlich möglich, die Zeitdauer bis zur Erkennung eines Paketverlustes bei UDP-basiertem Transport zu verringern, z. B. durch dynamisches Anpassen des TO-Wertes an die geschätzte Paketumlaufzeit RTT oder durch Implementieren eines Fast Retransmit-artigen Algorithmus in der Anwendungsschicht. Ein solcher Ansatz würde jedoch recht schnell darin enden, wesentliche Teile von TCP bzw. SCTP in der Anwendungschicht zu reimplementieren, was bzgl. der Modularisierung ein fragwürdiges Vorgehen wäre (vgl. Abschnitt 6.3.3).

6.7 Zusammenfassung und Fazit

In diesem Kapitel wurde am Beispiel von SIMCO der Einfluss des Transportschichtprotokolls auf die Antwortzeit transaktionsbasierter Signalisieranwendungen untersucht. Dabei wurde insbesondere auf den verzögernden Effekt des Head-Of-Line Blocking eingegangen. Dieser tritt auf, wenn infolge eines Paketverlustes Nachrichten erneut übertragen werden und darauffolgende Nachrichten zur Reihenfolgesicherung empfängerseitig gepuffert werden müssen. Messungen an dem im Zuge dieser Arbeit erstellten SIMCO-Prototypen haben gezeigt, dass der in der SIMCO-Spezifikation vorgesehene, vollständig reihenfolgegesicherte Transport der Transaktionen über TCP selbst bei moderaten Paketverlustwahrscheinlichkeiten signifikant mehr Zeit in Anspruch nimmt als die Paketumlaufzeit in der IP-Schicht.

Da SIMCO nur eine teilweise Reihenfolgesicherung benötigt, kann Head-Of-Line Blocking durch den Transport über eine SCTP-Assoziation mit mehreren Streams komplett vermieden werden. Die dazu notwendigen Anpassungen an SIMCO wurden in einem *Internet Draft* spezifiziert. Die optimale Anzahl zu verwendender Streams kann mit einem analytischen Modell bestimmt werden, welches die Mechanismen zur Erkennung von Paketverlusten sowie das Verhalten der empfängerseitigen Reihenfolgesicherungs-Puffer berücksichtigt. Eine prototypische Implementierung von „SIMCO over SCTP“ demonstriert die Realisierbarkeit des Vorschlages; die mit Hilfe eines WAN-Emulators und eines Lastgenerators gewonnenen Messwerte zeigen eine gute Übereinstimmung mit den vom Modell vorhergesagten Werten, sowie eine gegenüber TCP-basiertem Transport deutlich reduzierte mittlere Transaktions-Antwortzeit.

Ein Vergleich verschiedener Konfigurationsvarianten für den Transport transaktionsbasierter Signalisierprotokolle über IP zeigt, dass eine schnelle Erkennung und Behandlung von IP-Paketverlusten am besten dann gelingt, wenn die Nachrichtenrate zwischen zwei Protokollinstanzen recht hoch ist. In diesem Fall greift der Fast Retransmit-Algorithmus von TCP bzw. SCTP, welcher Paketverluste deutlich schneller erkennen kann als ein Timeout-basiertes Schema. Bezüglich niedriger Antwortzeiten können daher Signalisierprotokolle und -architekturen von Vorteil sein, die den Signalisierverkehr möglichst vieler Teilnehmer über eine gemeinsame TCP-Verbindung bzw. SCTP-Assoziation übertragen.

Tabelle 6.3: Mittlere Transaktionsantwortzeit für SCTP, TCP und UDP laut Modell

Transportschicht-protokoll	Mittlere Antwortzeit R bei niedriger Last $\lambda_T = 1\,\frac{\text{Trans.}}{\text{s}}$	Mittlere Antwortzeit R bei hoher Last $\lambda_T = 100\,\frac{\text{Trans.}}{\text{s}}$
SCTP (1 Stream)	40.5 ms	23.5 ms
SCTP (1024 Streams)	40.5 ms	21.5 ms
TCP	40.5 ms	23.5 ms
UDP	30.9 ms	30.9 ms
Parameter	$RTO = 1\,\text{s}$ $TO = 500\,\text{ms}$ $\delta = 0.5\,\text{ms}$	$RTT = 20\,\text{ms}$ $p_L = 1\,\%$

7 Netzweite Sicht – Vergleich der Architekturen zur Firewall-Steuerung

Basierend auf den Untersuchungen einzelner Mechanismen in den vorangegangenen Kapiteln sollen im Folgenden die beiden grundsätzlichen Signalisierverfahren zur Steuerung der Medienkomponenten verteilter Firewalls, die Pfad-entkoppelte und die Pfad-gekoppelte Signalisierung, miteinander verglichen werden. Dabei soll von einem Szenario ausgegangen werden, in dem die IP-Telefonie-Plattformen mehrerer Betreiber zusammengeschaltet wurden. Die funktionalen und die sicherheitsrelevanten Aspekte dieses Vergleichs sollen mit einem Anforderungskatalog abgedeckt werden; desweiteren soll der Einfluss der Verfahren auf die Dienstgüte und den Ressourcenverbrauch quantifiziert werden.

7.1 Funktionale und sicherheitsrelevante Eigenschaften

7.1.1 Einfluss von Netztopologie und Verkehrslenkung

Die folgenden Anforderungen betreffen den Einfluss von Netztopologie, dynamischer Verkehrslenkung und ggf. Adressumsetzungen auf die Anwendbarkeit der betrachteten Signalisierverfahren zur Firewall-Steuerung.

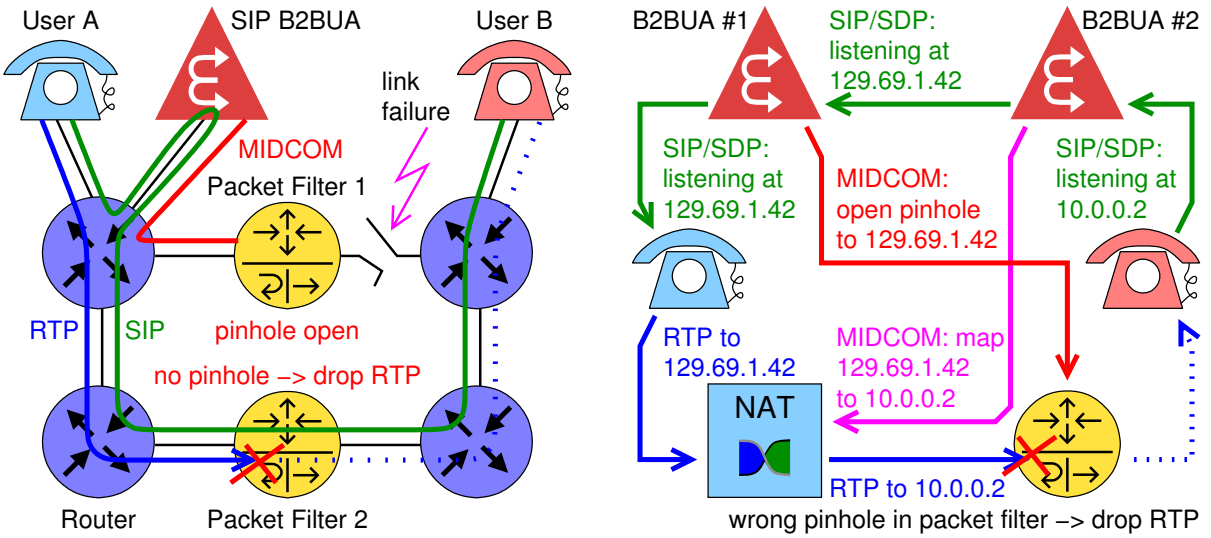
Anforderung 1 *Beim Aufbau einer neuen Multimedia-Sitzung müssen alle Medienkomponenten auf dem Medienpfad gefunden werden, auch bei Einsatz von dynamischem Routing.*

Bei der **Pfad-entkoppelten Signalisierung** senden die – in der Netztopologie u. U. zentral platzierten – Signalisierkomponenten (z. B. SIP B2BUA) Steuerkommandos an die dezentral installierten Medienkomponenten, um das Öffnen von Pinholes für die Medienströme zu veranlassen. Dazu müssen die Signalisierkomponenten die Netztopologie und die Verkehrslenkung kennen, um die Steuerkommandos an die „richtigen“ Medienkomponenten zu schicken, d. h. an die Middlebox, über die der jeweilige Medienstrom tatsächlich fließen wird. Dies ist nur für triviale Netztopologien einfach, z. B. wenn sich an einem Netzübergang nur ein Firewall-Element befindet, durch welches der Medienstrom zwangsläufig hindurchfließen muss (siehe die in der MIDCOM-Architekturbeschreibung und in Abbildung 4.6 dargestellte Netztopologie). Um eine erhöhte Ausfallsicherheit und eine automatische Lastverteilung zu erreichen, ist es jedoch sinnvoll, einen Netzübergang mit mehreren Firewall-Elementen und somit mehreren möglichen

Pfaden auszuführen, und die Verkehrslenkung mit Hilfe eines dynamischen Routingprotokolls (z. B. OSPF [RFC 2328], ggf. in Kombination mit BGP [RFC 4271]) zu steuern.

Abbildung 7.1(a) illustriert anhand eines einfachen Beispiels die Probleme, die auftreten können, wenn dynamisches Routing von einer Pfad-entkoppelten Firewall-Steuerung nicht berücksichtigt wird. Das dargestellte Szenario entspricht grundsätzlich dem von Abbildung 4.6, allerdings ist der Netzübergang mit zwei Paketfiltern realisiert. Unter normalen Umständen würden IP-Pakete zwischen den beiden User Agents und dem B2BUA – wegen der geringeren Anzahl von Transitknoten – über den Paketfilter Nr. 1 transportiert. Dieser wird auch vom B2BUA über ein Pfad-entkoppeltes Signalisierprotokoll (z. B. MIDCOM/SIMCO) konfiguriert, um die Medienströme einer Multimedia-Sitzung passieren zu lassen. Aufgrund eines Leitungsausfalls rechts von Paketfilter Nr. 1 wurde der Verkehr jedoch durch das dynamische Routingprotokoll automatisch über Paketfilter Nr. 2 umgeleitet. Für die SIP-Instanzen ist diese Umleitung normalerweise transparent; für die SIP-Signalisierung stellt sie auch kein Problem dar, da diese i. d. R. über statische Paketfilterregeln erlaubt wird. Die Medienströme hingegen werden von Paketfilter Nr. 2 verworfen, da dort kein Pinhole konfiguriert wurde. Dieser Problematik kann auf verschiedene Weise begegnet werden:

- Es können Zwangspunkte oberhalb der IP-Schicht geschaffen werden, indem der Paketfilter durch eine Medienkomponente ersetzt wird, die den RTP-Strom in der Anwendungsschicht weiterleitet (so genannter *Packet-to-Packet Gateway*) und dazu auf der IP-Schicht explizit adressiert wird. Für diese Lösung, die in der Klassifikation nach Abbildung 4.1 als *Distributed Session Border Controller* bezeichnet wird, müssen die Adressen in den SDP-Nachrichten entsprechend umgeschrieben werden. Nachteilig ist, dass der jeweilige Medienstrom dann fest an diese Medienkomponente gebunden ist; der Ausfall einer Verbindung zu dieser Medienkomponente kann dann nicht durch dynamisches Rerouting



(a) Dynamisches Rerouting nach Leitungsausfall

(b) Wechselwirkung zwischen Adressumsetzern und Paketfiltern

Abbildung 7.1: Probleme bzw. Fehlkonfiguration der Pfad-entkoppelten Firewall-Steuerung bei komplexen Netztopologien

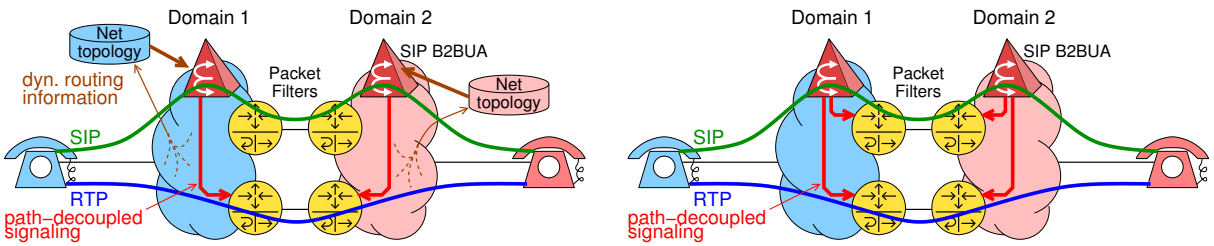
auf der IP-Schicht abgefangen werden. Um dennoch die Ausfallsicherheit zu erhalten, die zur Einführung redundanter Pfade geführt hat, wird ein Verfahren zum Rerouting in der Anwendungsschicht benötigt. Dies würde die Komplexität der Lösung deutlich erhöhen und hätte Auswirkungen auf der Sitzungssignalisierung, für die die Firewall-Steuerung jedoch möglichst transparent sein soll (vgl. Abschnitt 7.1.4).

- In [132, 133] werden Verfahren vorgestellt, mit denen das Verhalten von OSPF, einem der wichtigsten verteilten Routing-Protokolle, dynamisch nachvollzogen werden kann. Prinzipiell wäre es denkbar, die so gewonnenen Informationen über die derzeitige Verkehrslenkung an die Signalisierkomponente zur Auswahl der „richtigen“ Medienkomponenten weiterzuleiten (siehe Abbildung 7.2(a)). Bei dieser Vorgehensweise würden in bzw. bei der Signalisierkomponente Module benötigt, die mit dem jeweils verwendeten Routingprotokoll interagieren. Dies läuft der Bestrebung bei der Konzeption der Next Generation Networks zuwider, Anwendungen und Datentransport so weit als möglich zu entkoppeln.

- Eine simple Lösung des Problems ist, ein Pinhole einfach in alle Medienkomponenten einzutragen, die möglicherweise auf dem Medienpfad liegen (siehe Abbildung 7.2(b)). Dies führt allerdings dazu, dass in eine Medienkomponente auch Pinholes eingetragen werden, die dann gar nicht verwendet werden, weil die Medienströme über einen andern Pfad durch das Netz laufen. Diese unnötigen Pinholes können die Leistungsfähigkeit der Medienkomponente beeinträchtigen (siehe Abschnitt 5.5). Es existieren verschiedene Lösungsansätze, um diesen Nachteil abzumildern, die auch kombiniert werden können:

  – Die Netztopologie und die Strategie bei der Wegesuche (z. B. „Kürzester Pfad“) können im Voraus „off-line“ analysiert werden. Somit kann die Liste aller Firewalls bestimmt werden, durch die ein Medienstrom zwischen zwei gegebenen Endpunkten fließt, solange nicht mehr als eine gegebene Anzahl von Link- und Knotenausfällen in beliebiger Kombination auftreten. Nur diese Firewalls müssen für den fraglichen Medienstrom konfiguriert werden. In [134] wird ein solches Verfahren beschrieben für Netze, in denen OSPF zum Einsatz kommt. Solche Analyseverfahren können auch schon bei der Planung der Netztopologie zum Einsatz kommen, um einerseits die Zahl der möglichen Pfade durch das Netz einzuschränken, andererseits aber noch genügend Ersatz-Pfade für eine ausreichende Ausfallsicherheit zu gewährleisten.

  – Eine Medienkomponente kann eine asynchrone Benachrichtigung an die Signalisierkomponente senden, sobald das erste Paket erkannt und weitergeleitet wurde,



(a) Mit Topologie-Wissen: Nur die benötigten Pinholes werden eingetragen

(b) Ohne Topologie-Wissen: Pinholes in allen Medienkomponenten, die auf dem Pfad liegen könnten

Abbildung 7.2: Pfad-entkoppelte Signalisierung mit und ohne Kenntnis der Netztopologie

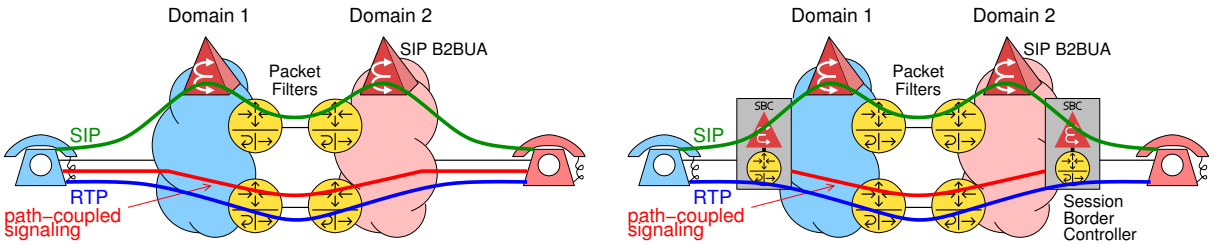
welches zu einem neu eingetragenen Pinhole passt. Somit ist ein Verfahren denkbar, bei dem die Signalisierkomponente die Pinholes für eine neue Multimedia-Sitzung zunächst in alle Medienkomponenten einträgt, um dann auf Rückmeldungen zu warten, wo diese tatsächlich benötigt werden. Die nicht benötigten Pinholes in den anderen Medienkomponenten können dann gelöscht werden, um diese Systeme zu entlasten. Eine solche asynchrone Benachrichtigung gehört derzeit allerdings nicht zum Umfang der SIMCO-Protokollspezifikation.

Mit diesen Verfahren kann – auch bei Einsatz von dynamischem Routing – sichergestellt werden, dass die zur Freigabe der Medienströme benötigten Pinholes an die Medienkomponenten auf dem Pfad der Medienströme signalisiert werden. Diese Verfahren unterscheiden sich bzgl. Komplexität und Effizienz; welches von ihnen für eine bestimmte Netztopologie am geeignetsten ist, hängt u. a. von der Zahl der für einen Medienstrom möglichen Netzübergänge ab.

Bei der **Pfad-gekoppelten Signalisierung** werden die Nachrichten zur Firewall-Steuerung mit Hilfe der selben Routing-Tabellen weitergeleitet, die auch für die Lenkung der Medienströme verwendet werden. Somit erreichen sie „automatisch“ alle Medienkomponenten auf dem Pfad zwischen den Endpunkten der Firewall-Signalisierung, d. h. es wird kein zentrales Wissen über die dortige Netztopologie benötigt. Alle Medienkomponenten müssen in der Lage sein, die Signalisiernachrichten als solche zu erkennen und entsprechend darauf zu reagieren, z. B. mit dem Öffnen von Pinholes. Desweiteren muss sichergestellt werden, dass die Endpunkte der Firewall-Signalisierung auf dem Medienpfad liegen. Dies kann auf verschiedene Weisen erreicht werden:

- Die Pfad-gekoppelte Signalisierung kann **Ende-zu-Ende** eingesetzt werden, indem ihre Endpunkte auf den selben Netzelementen platziert werden wie die Endpunkte der Medienströme, d. h. in den Multimedia-Endgeräten (siehe Abbildung 7.3(a)).

- Falls es Zwangspunkte gibt, an denen der Medienstrom auf jeden Fall vorbeikommen muss, kann Pfad-gekoppelte Signalisierung auch nur **abschnittsweise** zwischen diesen Punkten eingesetzt werden. Solche Zwangspunkte können von der Netztopologie vorgegeben sein, z. B. der „Edge Router“ an der Teilnehmerschnittstelle zu einem Teilnehmer mit nur einer Anschlussleitung. Alternativ können sie auch künstlich geschaffen werden, z. B. durch einen Eingriff in die SIP/SDP-Signalisierung, der bewirkt, dass eine Medienkomponente explizit adressiert wird.



(a) Ende-zu-Ende

(b) Abschnittsweise zwischen Zwangspunkten auf der Anwendungsschicht (Session Border Controller)

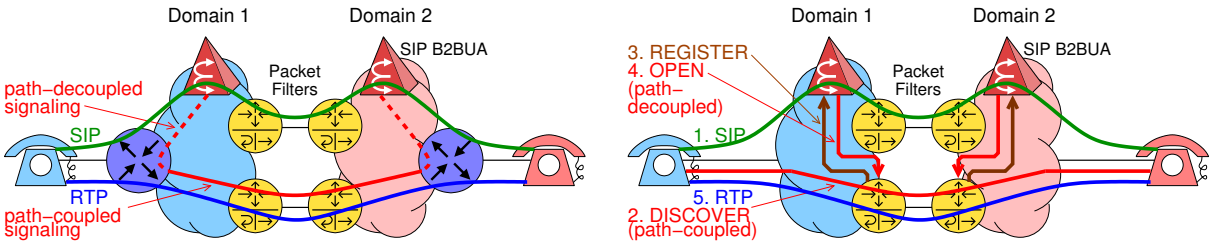
Abbildung 7.3: Pfad-gekoppelte Signalisierung: Ende-zu-Ende oder abschnittsweiser Einsatz

Die Zwangspunkte für die Medienströme können auch an der Sitzungssignalisierung teilnehmen, z. B. Session Border Controller (SBC), siehe Abbildung 7.3(b)). Ist dies nicht der Fall, wird eine Schnittstelle zu den SIP-Servern der Control Plane benötigt, da die Entscheidungen bezüglich des Öffnens von Pinholes dort getroffen werden. Abbildung 7.4(a) zeigt schematisch eine mögliche Konfiguration, bei der die SIP-Server Pfad-entkoppelte Signalisierung verwenden, um bekannte Zwangspunkte auf dem Medienpfad dazu aufzufordern, Pfad-gekoppelte Signalisiernachrichten entlang eines Abschnitts des Medienpfads zu senden.

Ein wichtiger Vorteil der Ende-zu-Ende-Konfiguration ist, dass neben den Multimedia-Endgeräten keine festen (d. h. explizit adressierten) Punkte auf dem Medienpfad geschaffen werden. Diese können aufgrund möglicher Ausfälle die Verfügbarkeit senken oder Mechanismen u. a. im Protokoll zur Sitzungssignalisierung erforderlich machen, mit denen im Fehlerfall auf redundante Komponenten umgeschaltet werden kann. Bei einem Ende-zu-Ende-Einsatz Pfad-gekoppelter Signalisierung müssen die Endpunkte der Firewall-Signalisierung in den Multimedia-Endgeräten platziert werden. Unter Gesichtspunkten der Ausfallsicherheit ist dies unproblematisch, da bei einem Ausfall eines dieser Netzelemente sowieso keine Multimedia-Sitzung aufgebaut oder aufrecht erhalten werden kann. Diese Platzierungs-Variante hat jedoch eine ganze Reihe von weiteren – positiven und negativen – Auswirkungen, so dass bei der Diskussion von vielen der folgenden Anforderungen unterschieden werden muss, ob Pfad-gekoppelte Firewall-Signalisierung Ende-zu-Ende zwischen den Multimedia-Endgeräten oder abschnittsweise zwischen Punkten „im Netz“ eingesetzt wird (siehe insbesondere die Anforderungen 4 und 5).

Anforderung 2 *Werden die Medienströme einer bestehenden Multimedia-Sitzung infolge dynamischen Reroutings durch eine andere Medienkomponente geleitet, darf die Sitzung nicht dauerhaft abgebrochen werden; die Dauer einer evtl. auftretenden Unterbrechung soll möglichst gering sein.*

Bei der **Pfad-entkoppelten Signalisierung** hängt das Verhalten davon ab, welches der bei Anforderung 1 diskutierten Verfahren zum Umgang mit dynamischen Routing eingesetzt wird. Falls die Signalisierkomponente das Routing-Protokoll verfolgt und erst bei Änderungen die Medienkomponente auf dem neuen Pfad ermittelt und konfiguriert, so können die Medienströme erst nach einer Ausfallzeit wieder fließen, die der Dauer zum Ermitteln der „richtigen“



(a) Pfad-gekoppelte Signalisierung zwischen Pfad-entkoppelt gesteuerten Zwangspunkten auf IP-Schicht

(b) Auffinden der Medienkomponenten Pfad-gekoppelt, Steuerung dann Pfad-entkoppelt

Abbildung 7.4: Hybride Firewall-Signalisierschemata

Medienkomponente plus der Dauer R einer Konfigurations-Transaktion entspricht. Werden die Pinholes hingegen von vornherein in alle in Frage kommenden Medienkomponenten eingetragen, können die Medienströme unmittelbar weiterfließen, d. h. das Vorhandensein der Firewall verursacht keine zusätzliche Verzögerung beim Rerouting.

Bei der **Pfad-gekoppelten Signalisierung** werden prinzipbedingt keine Pinholes in Netzelementen abseits des derzeitigen Medienpfades konfiguriert. Nach einem Rerouting muss somit grundsätzlich erst mit Hilfe entsprechender Signalisierung die Medienkomponente auf dem neuen Pfad konfiguriert werden. Während dieses Vorgangs sind die Medienströme unterbrochen.

Im konkreten Fall der NSIS-Protokollfamilie beobachten die NSIS-Protokollinstanzen (NSIS Forwarder) die Routing-Tabelle des Netzelements, auf dem sie platziert sind. Falls Änderungen dieser Tabelle dazu führen, dass ein bestimmter Flow einen anderen Pfad durch das Netz nehmen wird, senden sie eine Nachricht an den Initiator der zugehörigen NSIS-Signalisierung (z. B. im Multimedia-Endgerät), so dass dieser durch erneute Ende-zu-Ende-Signalisierung die NSIS-Instanzen auf dem neuen Pfad konfigurieren kann. Hierfür und für die u. U. erneut notwendige Authentisierung müssen eine ganze Reihe von Nachrichten ausgetauscht werden, was für eine entsprechende Latenz sorgt, während der die Medienströme verworfen werden.

Anforderung 3 *Es sollen auch dann „passende“ Pinholes geöffnet werden, wenn die Adressen in den Medienströmen unterwegs durch Network Address and Port Translation (NAPT) umgesetzt werden.*

NAPT wird verwendet, um in einem Transit-Netzelement auf dem Pfad eines Flows die in den IP-Paketen enthaltenen Quell- oder Zieladressen durch andere zu ersetzen, z. B. um Rückschlüsse auf die Netztopologie zu erschweren oder um durch N : 1-Abbildungen global eindeutige IP-Adressen einzusparen. Damit werden Parameter verändert, die die Grundlage für die Zugriffskontroll-Entscheidung der hier betrachteten Medienkomponenten darstellen. Falls im Umfeld der betrachteten IP-Telefonie-Lösungen die Adressen von RTP-Medienströmen umgesetzt werden sollen, ist i. d. R. eine Interaktion mit der Sitzungssignalisierung und eine dynamische Konfiguration der Adressumsetzer notwendig, aus den selben Gründen, die dies auch bei Paketfiltern und ähnlichen Medienkomponenten erforderlich machen. Da diese Aufgaben eng miteinander verwandt sind, unterstützen sowohl MIDCOM/SIMCO, als auch der NSIS NAT/FW NSLP die Steuerung von Paketfiltern und Adressumsetzern.

Bei der **Pfad-entkoppelten Signalisierung** muss der bzw. den steuernden Signalisierkomponente(n) die Netztopologie bekannt sein; dazu gehört auch die Platzierung und aktuelle Konfiguration evtl. vorhandener Adressumsetzer. Ist dieses Wissen nicht vorhanden, kann es zu Problemen kommen, die in Abbildung 7.1(b) anhand eines Beispiels illustriert werden: Der SIP B2BUA Nr. 1 konfiguriert auf Basis der SIP/SDP-Signalisierung ein Pinhole in einem Paketfilter, ohne zu wissen, dass die Zieladressen der RTP-Pakete von einem davor liegenden NAT umgesetzt werden, welcher von einem anderen B2BUA (Nr. 2) gesteuert wird. Somit „passt“ das Pinhole nicht und die Medienströme werden fälschlicherweise verworfen.

Bei der **Pfad-gekoppelten Signalisierung** können solche Probleme vermieden werden. Im Fall von NSIS passt ein Netzelement mit NAPT-Funktionalität den *Flow Identifier* in den Signa-

lisiernachrichten entsprechend an, während der *Session Identifier* konstant bleibt (vgl. Abschnitt 4.6.3).

In „Internet-Szenarien“ sind NAPT-Geräte sehr häufig anzutreffen, teilweise sogar kaskadiert (z. B. wenn ein WLAN-Accesspoint mit eingebautem NAPT-Router an einen ADSL-Router angeschlossen wird, welcher ebenfalls über NAPT-Funktionalität verfügt). Ob solche schwierigen Topologien in NGN-Szenarien relevant werden, bleibt abzuwarten, da zumindest das Problem der IPv4-Adressknappheit vermieden werden kann, indem solche Plattformen von vornherein mit IPv6 aufgebaut werden.

7.1.2 Authentisierung & Autorisierung, Selbstschutz der Firewall-Steuerung

Die betrachten Signalisier-Architekturen sollen bezüglich ihrer Widerstandsfähigkeit gegen bestimmte Angriffe verglichen werden, die sich aus folgendem Angreifermodell ergeben: Die Betrachtung soll vom Standpunkt des Betreibers einer IP-Telefonie-Plattform erfolgen, in welcher die jeweilige Signalisier-Architektur zur Firewall-Steuerung verwendet wird. Es sollen nur Angreifer außerhalb der eigenen Vertrauensdomäne betrachtet werden, d. h. mögliche Angreifer sind die Teilnehmer und die Betreiber anderer Plattformen, die mit der eigenen Plattform zusammengeschaltet sind. Es werden zwei verschiedene Motivationen für Angriffe untersucht:

- Der Angreifer versucht, Medienströme als Transitverkehr durch die betrachtete Domäne zu leiten, ohne dass vorher der Aufbau einer Multimedia-Sitzung signalisiert, ein entsprechender Verbindungszustand in der Control Plane etabliert, und ggf. die Erfassung von Verbindungsentgelten ausgelöst wurde. Die Abwehr dieses Angriffs wird im Zuge von Anforderung 4 untersucht.
- Ein zweiter Angriff versucht, die Verfügbarkeit oder die Integrität von Komponenten der betrachteten Domäne zu beeinträchtigen, indem von außen Nachrichten an entsprechend gewählte Einfallstore gesendet werden. Dies wird unter Anforderung 5 untersucht.

Bei beiden betrachteten Angriffen wird davon ausgegangen, dass der Angreifer u. U. große Mengen beliebig formatierter Nachrichten von außen an einen oder mehrere Netzübergänge der betrachteten Domäne senden kann. Desweiteren wird davon ausgegangen, dass der Angreifer keinen physischen Zugriff auf die Infrastruktur der betrachteten Domäne hat, dass er keine Nachrichten sehen kann, die zwischen den Netzelementen der betrachteten Domäne ausgetauscht werden, und dass er nicht in der Lage ist, die verwendeten kryptographischen Verfahren zu brechen.

Anforderung 4 *Pinholes zum Erlauben von Medienströmen dürfen nur geöffnet werden, wenn die dazugehörige Sitzung von der (SIP-basierten) Control-Plane autorisiert wurde und dort entsprechende Zustandsinformationen vorliegen. Dies muss von den Medienkomponenten ggf. zuerst geprüft werden.*

Bei der **Pfad-entkoppelten Signalisierung** ist diese Anforderung erfüllt, da bei dieser Architektur die steuernde Signalisierkomponente (z. B. ein SIP B2BUA) Teil der Control Plane ist und eine statische Zuordnung zu den gesteuerten Medienkomponenten (z. B. Paketfilter) besteht, die ihr vertrauen. Da die Signalisier-Assoziationen domänenintern sind, können diese leicht geschützt werden (z. B. durch ein domäneninternes, separates Signalisiernetz oder durch kryptographische Schutzmechanismen). Somit werden i. d. R. keine weiteren Prüfungen in der Medienkomponente benötigt.

Bei der **Pfad-gekoppelten Signalisierung** tauscht eine Protokollinstanz auf dem Pfad eines Flows Signalisiernachrichten mit ihren Nachbarinstanzen aus, sowohl „stromaufwärts“, als auch „stromabwärts“. Da ein Firewall-Element immer auf einer Grenze zwischen Vertrauensdomänen platziert wird, ist mindestens eine dieser Nachbarinstanzen nicht vertrauenswürdig. Auch mindestens ein Endpunkt der Pfad-gekoppelten Firewall-Signalisierung liegt prinzipbedingt außerhalb der eigenen Vertrauensdomäne. Falls die Signalisierung Ende-zu-Ende eingesetzt wird, befinden sich die Endpunkte in den Multimedia-Endgeräten, die unter der physischen und ggf. administrativen Kontrolle der Teilnehmer liegen, bei abschnittsweisem Einsatz hingegen z. B. im Einflussbereich konkurrierender Netzbetreiber, mit denen eine Netzzusammenschaltung vereinbart wurde. In allen Fällen dürfen die Endpunkte der Pfad-gekoppelten Signalisierung keine Pinholes ohne Zustimmung der eigenen Control-Plane anfordern. In Abschnitt 4.6.5 wird ein Verfahren beschrieben, bei der die Firewall-Signalisierung zwar vom Endpunkt z. B. im Endgerät ausgeht, aber mit Hilfe kryptographischer Autorisierungs-Tokens autorisiert wird, welche von der Control Plane ausgestellt wurden. Die Effizienz dieses Verfahrens, mit dem die hier betrachtete Anforderung erfüllt werden kann, wird in Abschnitt 7.2 untersucht.

Anforderung 5 *Die Protokolle und Instanzen zur Firewall-Steuerung müssen gegen Angriffe gegen sie selbst geschützt sein, insbesondere auch gegen DoS-Attacken.*

Für Firewall-Steuerung selbst sind folgende Schutzziele von besonderer Bedeutung: Maßnahmen zum Schutz der Integrität und Authentizität der Signalisiernachrichten sorgen – zusammen mit einer entsprechenden Autorisierung – dafür, dass nur berechtigte Instanzen Pinholes in einer Firewall öffnen können (siehe Anforderung 4). Aus dem selben Grund und um zu verhindern, dass ein kompromittiertes Firewall-Element als Ausgangspunkt für weitere Angriffe dienen kann, muss die funktionale Integrität der Firewall-Elemente geschützt werden. Desweiteren muss die *Verfügbarkeit* der Firewall-Steuerung sichergestellt werden, um zu verhindern, dass ein Angreifer das Öffnen von Pinholes durch berechtigte Instanzen und somit das Zustandekommen legitimer Kommunikationsverbindungen blockieren kann.

Nur von untergeordneter Bedeutung ist i. d. R. der Schutz der Vertraulichkeit: Falls ein Angreifer in der Lage ist, die Signalisiernachrichten der Firewall-Steuerung abzuhören, kann er aus diesen im Wesentlichen die Parameter von Flows entnehmen, die kurze Zeit später durch das Netz fließen werden. Ohne Zugriffsmöglichkeit auf die betroffenen Flows ist diese Information

vergleichsweise uninteressant. Hat der Angreifer hingegen einen entsprechenden Aufschaltpunkt unter seiner Kontrolle, kann er diese Information selbst ermitteln – neben dem Zugriff auf die eigentlichen Inhalte der Medienströme, die i. d. R. sehr viel schutzwürdiger sind.

Ein wichtiges Kriterium für den Vergleich der beiden Architekturen zur Firewall-Steuerung ist, welche Protokolle und Schnittstellen von einem potenziellen Angreifer außerhalb der eigenen Vertrauensdomäne direkt bzw. indirekt erreichbar sind (siehe Tabelle 7.1). Die jeweiligen Angriffsflächen werden auch bei einem Vergleich der Abbildungen 4.6 und 4.14(a) illustriert, in welchen sich der nicht vertrauenswürdige Teilnehmer jeweils auf der rechten Seite befindet. Hilfsprotokolle wie z. B. DNS, die bei beiden Architekturen benötigt werden und die bei einer vollständigen Sicherheitsbewertung der Gesamtplattform als weiterer möglicher Angriffsvektor berücksichtigt werden müssten, sind in diesen Abbildungen nicht dargestellt, um die Übersichtlichkeit zu erhöhen.

Tabelle 7.1: Erreichbarkeit von Schnittstellen für Angreifer in anderen Domänen

Schnittstelle	**Pfad-entkoppelte Sig.**	**Pfad-gekoppelte Sig.**
Signalisierkomponente (SIP Server)	direkt	direkt
Medienkomponente (Paketfilter o.ä.)	direkt	direkt
Steuerschnittstelle der Medienkomponente	**indirekt** (über Signalisierkomponente)	**direkt**
Authentisierungsserver	(nicht benötigt)	indirekt (über Signalisierkomponente)
Hilfsprotokolle (z. B. DNS)	– Prinzipiell gleich konfigurierbar –	

Wie bereits im Zuge von Anforderung 4 diskutiert, existiert bei der **Pfad-entkoppelten Signalisierung** i. d. R. eine feste Zuordnung und ein klares Vertrauensverhältnis zwischen gesteuerten Medienkomponenten und steuernden Signalisierkomponenten. Die statischen Signalisierassoziationen liegen vollständig innerhalb einer Vertrauensdomäne und können daher vergleichsweise einfach geschützt werden, z. B. durch kryptographische Protokolle (z. B. IPsec) und statische Firewall-Regeln, die einen Zugriff auf die Firewall-Steuerung von außen komplett unterbinden. Ein Angriff auf die Firewall-Steuerung kann von einem Angreifer außerhalb der Vertrauensdomäne daher nur indirekt erfolgen, z. B. indem erst die SIP-Protokollinstanz der Signalisierkomponente kompromittiert wird, um dann als Ausgangspunkt für Angriffe über das Firewall-Signalisierprotokoll auf die Medienkomponente zu dienen. Auch DoS-Attacken durch Überfluten der Steuerschnittstelle einer Medienkomponente ist nur indirekt möglich, z. B. indem viele SIP *INVITE*-Nachrichten an die Signalisierkomponente gesendet werden. Dies würde aber hauptsächlich die Signalisierkomponente überlasten. Entsprechende Maßnahmen zum Schutz der SIP-Server in der Control Plane sollen hier jedoch nicht genauer untersucht werden, da auch bei Verwendung von Pfad-gekoppelter Signalisierung die selben Angriffe gegen SIP-Server denkbar sind und abgewehrt werden müssen.

Bei der **Pfad-gekoppelten Signalisierung** tauschen die in den Firewall-Elementen platzierten Protokollinstanzen des Steuerprotokolls Nachrichten mit ihren (logischen) Nachbarknoten

aus, welche sich prinzipbedingt nicht alle in der selben Vertrauensdomäne befinden. Aufgrund dieser direkten Erreichbarkeit von außerhalb muss die Protokollinstanz besonders gut gegen Implementierungsfehler geschützt werden, die die Kompromittierung des Firewall-Elements ermöglichen könnten (die zugrundeliegenden Probleme der in [60] beobachteten Schwachstellen in SIP-Implementierungen sind nicht spezifisch für SIP; somit sind solche Schwachstellen auch bei anderen Protokollen denkbar).

Die direkte Erreichbarkeit der Protokollinstanz für die Pfad-gekoppelte Signalisierung von Knoten außerhalb der eigenen Vertrauensdomäne stellt neben SIP (s. o.) einen zusätzlichen Angriffspunkt für Denial-of-Service-Angriffe dar. Dies gilt insbesondere dann, wenn das zur Erfüllung von Anforderung 4 vorgeschlagene und in Abschnitt 4.6.5 beschriebene, Token-basierte Autorisierungsverfahren zum Einsatz kommt. Die kryptographische Prüfung der Authentizität des Autorisierungs-Tokens benötigt einen gewissen Rechenaufwand. Somit kann entweder das Firewall-Element selbst oder der Autorisierungs-Server, an den das Token zur Prüfung weitergeleitet wird, überlastet werden, indem mit hoher Rate Anforderungen zum Öffnen eines Pinholes gesendet werden, die ein vom Angreifer generiertes, syntaktisch korrektes, aber ungültiges Token enthalten. Dies ist in Abbildung 7.5 für das Szenario aus 4.14(a) illustriert, wobei „User B“ in der Rolle des Angreifers ist. Wirkungsvolle Maßnahmen zur Abwehr solcher DoS-Angriffe sind insbesondere dann schwierig zu implementieren, wenn die konkrete Ausprägung des Protokolls zur Pfad-gekoppelten Signalisierung das Token bereits in der ersten Nachricht vom Angreifer zum Firewall-Element überträgt. In diesem Fall kann der Angreifer gefälschte Absenderadressen in den Nachrichten verwenden, so dass der Rückschluss auf seine wahre Identität oder eine Ratenbegrenzung auf Basis der Absenderadresse erschwert wird.

Falls wegen einer Überlastung keine Nachrichten der Pfad-gekoppelten Firewall-Signalisierung mehr autorisiert werden können, wird nicht nur der Aufbau neuer Multimedia-Sitzungen unmöglich gemacht, auch bestehende Sitzungen brechen nach einiger Zeit ab, wenn die Gültigkeitsdauern der Pinholes nicht mehr verlängert werden können.

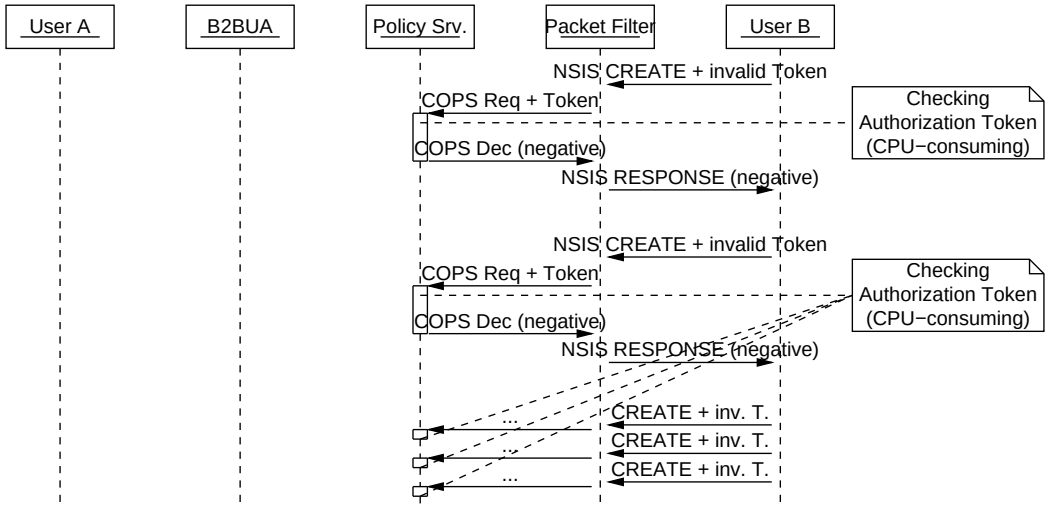


Abbildung 7.5: Denial-of-Service-Angriff von User B gegen Autorisierungs-Server bei Pfad-gekoppelter Signalisierung

7.1.3 Zusammenspiel mit der Sitzungs-Signalisierung

Anforderung 6 *Kann ein Pinhole für einen Medienstrom beim Aufbau der Sitzung nicht geöffnet werden, oder muss es während einer bestehenden Sitzung vorzeitig geschlossen werden, so muss der Abbruch der Multimedia-Sitzung standardkonform signalisiert werden und die Nutzer mit einer aussagekräftigen Fehlermeldung darüber informiert werden.*

Die im Protokoll zur Sitzungssignalisierung (z. B. SIP) benötigten Mechanismen zur Behandlung möglicher Fehler bei der **Pfad-entkoppelten Firewall-Signalisierung** wurden in Abschnitt 5.4 untersucht und dargestellt. Diese werden auch benötigt, falls Pfad-gekoppelte Firewall-Signalisierung nur abschnittsweise zum Einsatz kommt, d. h. deren Endpunkte nicht mit den Endpunkten der Multimedia-Sitzung übereinstimmen.

Wird **Pfad-gekoppelte Firewall-Signalisierung** hingegen Ende-zu-Ende eingesetzt, kommen eventuelle Fehlermeldungen direkt in den Endgeräten der Teilnehmer an und können dort angezeigt werden, ohne erst über mehr oder weniger geeignete Fehlercodes des Protokolls zur Sitzungssignalisierung transportiert werden zu müssen. Auch bei dieser Konfiguration sollte das Protokoll zur Sitzungssignalisierung die für die Firewall-Konfiguration benötigten Parameter rechtzeitig vor der Alarmierung des gerufenen Teilnehmers zur Verfügung stellen, so dass es nicht zu *Ghost Rings* kommen kann (vgl. Abschnitt 5.4).

Anforderung 7 *Wird das Ende einer Multimedia-Sitzung signalisiert, so müssen die Pinholes für die dazugehörigen Medienströme auch ohne Zutun der Teilnehmer möglichst zeitnah geschlossen werden, so dass das tatsächliche Ende der Sprachverbindung nicht wesentlich nach dem von der Control-Plane in den Call Detail Records (CDR) protokollierten Zeitpunkt liegt.*

Bei der **Pfad-entkoppelten Firewall-Signalisierung** wird das Öffnen von Pinholes für Medienströme von Signalisierkomponenten angefordert. Diese SIP-Server (i. d. R. B2BUA), die Teil der Control-Plane sind, halten Zustandsinformationen bzgl. aktiver Multimedia-Sitzungen und zugehöriger Pinholes und können auch über eine entsprechende Signalisier-Transaktion das Schließen dieser Pinholes anfordern, sobald sie die Sitzung für beendet halten. Diese Anforderung ist somit erfüllt.

Die bei der MIDCOM-Architektur und SIMCO vorhandenen Soft States dienen ausschließlich der Fehlerbehandlung. Sie sollen verhindern, dass Policy Rules unbegrenzt lange bestehen bleiben, falls der MIDCOM-Agent in der Signalisierkomponente unkontrolliert terminiert oder die MIDCOM-Session unterbrochen wird. Da solche Fehler im Netz eines professionellen Betreibers nur sehr selten vorkommen sollten, können die Intervalle recht groß gewählt werden (bei dem im Zuge dieser Arbeit erstellten Prototypen und für die Analysen wurde der Wert $L = 120\,\mathrm{s}$ gewählt); die Nachrichten zur Verlängerung der Gültigkeitsdauern (*RLC* bzw. *PLC*) erzeugen somit eine vergleichsweise geringe Signalisierlast.

Wird **Pfad-gekoppelte Signalisierung** Ende-zu-Ende verwendet, gehen die Signalisiernachrichten zum Öffnen und Schließen von Pinholes von den Multimedia-Endpunkten bei den Teilnehmern aus. Diese können von der Control Plane nicht dazu gezwungen werden, Nachrichten zu versenden, die das Schließen eines Pinholes anfordern.

Um dennoch zu erreichen, dass die Teilnehmer die Pinholes einer aus Sicht der Control Plane beendeten Sitzung nicht beliebig lange offen halten können, können die Medienkomponenten nur sehr kurze Gültigkeitsdauern für die Soft States der Pinholes akzeptieren. Somit muss die Gültigkeit einer Regel entsprechend oft verlängert werden, wobei jedesmal die Berechtigung geprüft werden kann (z. B. mit einem Autorisierungs-Token, siehe Abschnitt 4.6.5). Nachdem eine Multimedia-Sitzung aus Sicht der Control Plane beendet ist, kann keine Verlängerung mehr durchgeführt werden. Diese Lösung kann z. B. mit den vorhandenen Protokollmechanismen von NSIS implementiert werden. Je kürzer die Intervalle gewählt werden, desto kürzer ist auch das maximal mögliche „Überziehen“ der Sitzung, allerdings erfordert jede Verlängerung der Gültigkeitsdauer den Austausch mehrerer Nachrichten und die Überprüfung des Autorisierungs-Tokens, so dass die Signalisierlast und die Belastung des Autorisierungs-Servers steigt. Deshalb muss ein Kompromiss zwischen maximal tolerierbarer Ungenauigkeit der CDR und Ressourcenverbrauch gefunden werden.

Ein alternativer Ansatz ist die Verwendung einer hybriden Lösung aus Pfad-gekoppelter und Pfad-entkoppelter Firewall-Signalisierung, die in Abbildung 7.4(b) schematisch dargestellt ist. Mit einem Pfad-gekoppelten Verfahren entsprechend Abschnitt 4.6.5 können zunächst alle Medienkomponenten auf dem Medienpfad gefunden werden. Das Zurücksenden des Autorisierungs-Tokens von der Medienkomponente an den Autorisierungs-Server führt jedoch noch nicht zum Öffnen eines Pinholes. Stattdessen dient dieser Schritt dazu, die betroffenen Medienkomponenten bei dem Autorisierungs-Server zu registrieren, so dass dieser in Zusammenarbeit mit den SIP-Servern der Control Plane und unter Verwendung Pfad-entkoppelter Signalisierung die benötigten Pinholes öffnen und wieder schließen kann. Dabei könnten deutlich längere Intervalle zur Gültigkeitsauffrischung zum Einsatz kommen (s. o.). Um diesen Ansatz mit Protokollen zu implementieren, die bereits standardisiert wurden oder derzeit werden, wird allerdings ein recht komplexes Zusammenspiel mehrerer Protokolle (z. B. NSIS + COPS + SIMCO) und eine entsprechend große Zahl ausgetauschter Signalisiernachrichten benötigt.

Kommt Pfad-gekoppelte Signalisierung hingegen abschnittsweise zum Einsatz, und liegen die Endpunkte unter der Kontrolle der Control Plane, ist die Anforderung erfüllt.

Anforderung 8 *Kommt es zu einem nicht standardkonformen Abbruch der Multimedia-Sitzung, z. B. durch unkontrolliertes Terminieren („Absturz“) der Protokollinstanzen in einem Multimedia-Endpunkt oder durch Abbruch der IP-Konnektivität, so muss dies erkannt und die Pinholes in den Medienkomponenten entfernt werden.*

In Abschnitt 5.4.2 werden Mechanismen benannt, mit denen SIP-Server den Abbruch einer Sitzung auf SIP-Ebene erkennen können, um bei Einsatz von **Pfad-entkoppelter Signalisierung** die Pinholes wieder schließen zu können. Diese Mechanismen werden auch benötigt, wenn Pfad-gekoppelte Signalisierung abschnittsweise verwendet wird.

Wird **Pfad-gekoppelte Signalisierung** Ende-zu-Ende verwendet, sind solche Mechanismen aus Sicht der Firewall-Steuerung nicht notwendig, da in diesen Fehlerfällen davon ausgegangen werden kann, dass auch keine Nachrichten zur Verlängerung der Gültigkeit der Pinholes gesendet werden. Da in einem realen Szenario aber nicht nur die Pinholes für die Medienströme geschlossen, sondern auch die Zustandsinformationen in den SIP-Servern gelöscht werden müssen, werden die genannten (oder ähnliche) Mechanismen dennoch benötigt.

7.1.4 Konfiguration, Verwaltung, Erweiterbarkeit

Anforderung 9 *Es soll möglich sein, eine IP-Telefonie-Plattform, die die betrachtete Architektur zur Firewall-Steuerung verwendet, mit einem ungesicherten Netz (z. B. dem Internet) oder einer anderen Plattform mit einer anderen Firewall-Steuerungsarchitektur zusammenzuschalten. Dabei müssen die Sicherheitsanforderungen der eigenen Domäne erfüllt bleiben; die eigene Firewall-Steuerung soll im anderen Bereich keine besonderen Maßnahmen zur Interaktion erforderlich machen.*

Bei der **Pfad-entkoppelten Signalisierung** handelt es sich um eine rein *Domäneninterne* Signalisierung, d. h. die gesteuerte Medienkomponente und die steuernde Signalisierkomponente befinden sich in der selben Vertrauensdomäne; die Firewall-Signalisierung verlässt diese Domäne nicht. Somit ist eine Netzzusammenschaltung mit Domänen, die gar keine Firewalls oder eine andere Domäneninterne Lösung (z. B. Session Border Controller) verwenden, ohne besondere Absprachen möglich; es ist allerdings vorteilhaft, wenn alle Multimedia-Endgeräte die Mechanismen zur korrekten Behandlung von Fehlerfällen (vgl. Anforderung 11) unterstützen.

Bei der **Pfad-gekoppelten Signalisierung** handelt es sich hingegen prinzipbedingt um eine Domänenübergreifende Signalisierung, da ein Firewall-Element seiner Aufgabe entsprechend immer auf einer Domänengrenze platziert wird und bei Pfad-gekoppelter Signalisierung Nachrichten mit Instanzen „auf beiden Seiten“ austauscht. Deshalb muss in beiden Domänen ein Endpunkt für die Pfad-gekoppelte Signalisierung vorhanden sein – entweder in den Multimedia-Endgeräten der Teilnehmer für Ende-zu-Ende-Signalisierung oder auf einem Knoten „im Netz“ bei abschnittsweisem Einsatz. Die Anforderung ist somit nicht erfüllt.

Ein Spezialfall tritt auf, wenn in einer Gruppe von Domänen Pfad-gekoppelte Signalisierung abschnittsweise verwendet wird und der Endpunkt dieser Signalisierung auf einem Netzelement am Übergang zu einer Domäne liegt, die keine Pfad-gekoppelte Signalisierung verwendet. In diesem Fall werden keine Absprachen benötigt; der betrachtete Netzübergang wird aber auch nicht im eigentlichen Sinne mit der Pfad-gekoppelten Signalisierung gesteuert, und es müssen die im Zuge von Anforderung 1 gemachten Überlegungen zur Ausfallsicherheit beachtet werden.

Anforderung 10 *Zwischen den Betreibern zusammengeschalteter IP-Telefonie-Plattformen sollte möglichst wenig Koordinationsaufwand notwendig sein, um die Steuerung der Firewalls bei domänenübergreifenden Multimedia-Sitzungen zu ermöglichen.*

Die bei Anforderung 9 gemachten Überlegungen zur **Pfad-entkoppelten Signalisierung** sind auch dann gültig, wenn jede Domäne intern, ohne Absprachen mit den anderen Domänen, die selbe Signalisier-Architektur verwendet.

Soll **Pfad-gekoppelte Signalisierung** domänenübergreifend eingesetzt werden, sind Absprachen bzgl. der zu verwendenden Protokolle und ggf. ihrer Parametrisierung notwendig.

Anforderung 11 *Die Firewall-Steuerung sollte den Teilnehmer nicht in der Wahl seiner Multimedia-Endgeräte einschränken.*

In Abschnitt 5.4 wurde untersucht, welche Mechanismen im Protokoll zur Sitzungssignalisierung (z. B. SIP) bzw. seinen Protokollinstanzen benötigt werden, um mit **Pfad-entkoppelter Signalisierung** zusammenarbeiten zu können, insbesondere auch in Fehlerfällen. Diese Mechanismen sind allerdings nicht zwingend erforderlich: Ein User Agent kann abgefragt werden, ob er den optionalen Mechanismus zum Signalisieren von Vorbedingungen unterstützt. Ist dies nicht der Fall, kann es zu den unerwünschten *Ghost Rings* kommen. Ist der bei Fehlern in der Firewall-Steuerung verwendete SIP-Status-Code dem User Agent nicht bekannt, kann u. U. keine aussagekräftige Fehlermeldung angezeigt werden; aufgrund der Einteilung der Status-Codes in Klassen (vgl. Abschnitt 2.1.5.4) kann der User Agent sich aber immerhin bzgl. der Protokollspezifikation korrekt verhalten. Somit ist eine Kompatibilität zu SIP User Agents, die diese Mechanismen nicht unterstützten, gewahrt; diese können mit gewissen Einschränkungen dennoch verwendet werden. Lediglich auf Mechanismen zur Erkennung eines Sitzungsabbruchs kann nur schwer verzichtet werden – dies gilt aber für alle betrachteten Architekturen (vgl. Anforderung 8). Die selben Überlegungen gelten auch, wenn Pfad-gekoppelte Signalisierung abschnittsweise verwendet wird.

Falls **Pfad-gekoppelte Signalisierung** Ende-zu-Ende verwendet werden soll, müssen die dafür benötigten Protokollinstanzen (z. B. für NSIS: GIST und NATFW NSLP) im Multimedia-Endgerät implementiert werden. Falls ein Token-basiertes Autorisierungsverfahren (vgl. Abschnitt 4.6.5) zum Einsatz kommen soll, muss eine Schnittstelle zur SIP-Protokollinstanz geschaffen werden. Diese muss die Protokollmechanismen zum Transport des Tokens und zum Erkennen eines Sitzungsabbruchs (s. o.) unterstützen. Auch hier sollte das Signalisieren von Vorbedingungen möglich sein, um *Ghost Rings* zu vermeiden.

Anforderung 12 *Neue Protokolle zur Sitzungssignalisierung (d. h. ergänzend oder alternativ zu SIP) sollen möglichst leicht in die bestehende Architektur integrierbar sein.*

Die in Abschnitt 5.4 identifizierten Defizite von SIP bezüglich Mechanismen zur Behandlung von Fehlern bei **Pfad-entkoppelter Signalisierung** können bei der Spezifikation neuer Signalisierprotokolle recht einfach berücksichtigt bzw. vermieden werden. Eine wesentliche Anforderung ist, dass die zur Konfiguration der Medienkomponente benötigten Informationen (z. B. Adressen) so rechtzeitig ausgetauscht werden, dass bei einem Fehler in der Firewall-Steuerung der Aufbau der Multimedia-Sitzung möglichst ohne negative Folgen (z. B. Belästigung des gerufenen Teilnehmers) abgebrochen werden kann. Geeignete Protokollinstanzen (z. B. Signalisier-Proxies, u. U. auch Endpunkte) müssen um eine Schnittstelle erweitert werden, über welche die Parameter der benötigten Pinholes an die Client-Instanz der Firewall-Steuerung (z. B. MIDCOM-Agent) übergeben werden können.

Zur Unterstützung **Pfad-gekoppelter Signalisierung** muss das Protokoll zur Sitzungssignalisierung um Nachrichtenformate zum Transport von Autorisierungs-Tokens (vgl. Abschnitt 4.6.5) erweitert werden. Die Schnittstelle zur Übergabe von Pinhole-Parametern und Autorisierungs-Tokens befindet sich i. d. R. im Multimedia-Endgerät.

7.1.5 Zusammenfassung

In Tabelle 7.2 werden die vorangegangenen Überlegungen mit einem dreistufigen Schema bewertet und zusammengefasst. Auf eine feingranulare Bewertung und Bildung einer „Gesamtpunktezahl“ soll hier bewusst verzichtet werden, da die einzelnen Anforderungen gewiss eine unterschiedliche Wichtigkeit haben, eine objektive, allgemeingültige und Szenarien-unabhängige Festlegung von Gewichtungsfaktoren jedoch praktisch nicht möglich ist.

Tabelle 7.2: Zusammenfassung der funktionalen und sicherheitsrelevanten Eigenschaften

Anforderung		Firewall-Signalisierung		
		Pfad-entk.	Pfad-gekoppelt	
			Ende-zu-Ende	Abschnittsw.
Nr.	Kurzbeschreibung			
1	Finden aller Medienkomp. bei dyn. Routing	○	+	+
2	Kein Sitzungsabbruch bei Rerouting	○	+	+
	Keine Unterbrechung bei Rerouting	+[1]	−	−
3	Öffnen „passender“ Pinholes auch bei NAPT	○	+	+
4	Autorisierung von Pinholes durch SIP-Server	+	○	○
5	Selbstschutz, insbes. gegen DoS-Attacken	+	−	○[2]
6	Sauberes Beenden der Sitzung bei Fehler	○	+	○
7	Sofortiges Schließen nach Sitzungsende	+	○	+
8	Schließen nach Sitzungsabbruch	○	+	○
9	Zusammenschaltung mit ungeschütztem Netz	+	−	+
10	Interdomänen-Koordinationsaufwand	+	○	○
11	Anforderungen an Endgeräte	+	○	+
12	Neue Protokolle zur Sitzungssignalisierung	+	○	+

Legende

+ Die Anforderung wird vollständig und besonders effizient erfüllt.

○ Zur Erfüllung der Anforderung ist erheblicher zusätzlicher Aufwand notwendig oder die Anforderung kann nur in bestimmten Szenarien erfüllt werden.

− Die Anforderung wird nicht erfüllt.

Anmerkungen

[1] Unter der Annahme, dass Pinholes in alle in Frage kommenden Medien komponenten eingetragen wurden, was nicht sehr effizient ist (vgl. Anforderung 1).

[2] Schutz gegen Angriffe von Teilnehmern, jedoch nicht gegen solche aus benachbarten IP-Telefonie-Plattformen (i. d. R. Mitbewerber).

7.2 Einfluss auf Dienstgüte und Ressourcenverbrauch

Firewalls auf dem Medienpfad sowie die dazugehörigen Architekturen und Protokolle zur Steuerung können beim Netzbetreiber zu einem erhöhten Ressourcenbedarf führen und einen negativen Einfluss auf die vom Nutzer wahrgenommenen „klassischen“ Dienstgüte-Parameter haben (das Einhalten bestimmter Schutzziele wurde in jüngerer Vergangenheit vermehrt dem Themenkomplex „Dienstgüte“ zugeordnet – in diesem Abschnitt soll der Fokus jedoch ausschließlich auf Effekte wie z. B. Verzögerungen beim Verbindungsaufbau liegen, während Sicherheitsaspekte an anderer Stelle diskutiert werden). Solche negativen Einflüsse können sich verstärken, wenn sich mehrere durch Firewalls abgesicherte Domänengrenzen auf dem Pfad zwischen den beiden Teilnehmern befinden. In Szenarien mit mobilen Teilnehmern, auf die im Folgenden nicht im Detail eingegangen werden soll, können solche Verzögerungen nicht nur beim Verbindungsaufbau auftreten, sondern auch z. B. bei einem so genannten *Handover* in eine benachbarte Funkzelle. In diesem Abschnitt sollen die verschiedenen Verfahren zur Steuerung der Medienkomponenten miteinander verglichen werden.

7.2.1 Betrachtete Szenarien

Zum Vergleich der verschiedenen Verfahren zur Steuerung von Medienkomponenten soll folgendes Grund-Szenario angenommen werden: Zwischen dem rufenden und dem gerufenen Teilnehmer liegen N_D Transitdomänen, die alle ihre Netzübergänge schützen. Jede dieser Domänen besitzt einen SIP B2BUA samt zugehöriger AAA-Infrastruktur (*Authentication, Authorization and Accounting*), der Anforderungen bzgl. neuer Multimedia-Sitzungen autorisieren und ggf. an die richtige Nachbardomäne weiterleiten kann. Zur Lastverteilung und zur Erhöhung der Verfügbarkeit ist jede Domäne mit ihren Nachbardomänen über jeweils N_C Netzübergänge verbunden, die von Signalisierung und Medienströmen genutzt werden können. Es sollen folgende Verfahren zur Steuerung von Medienkomponenten betrachtet werden:

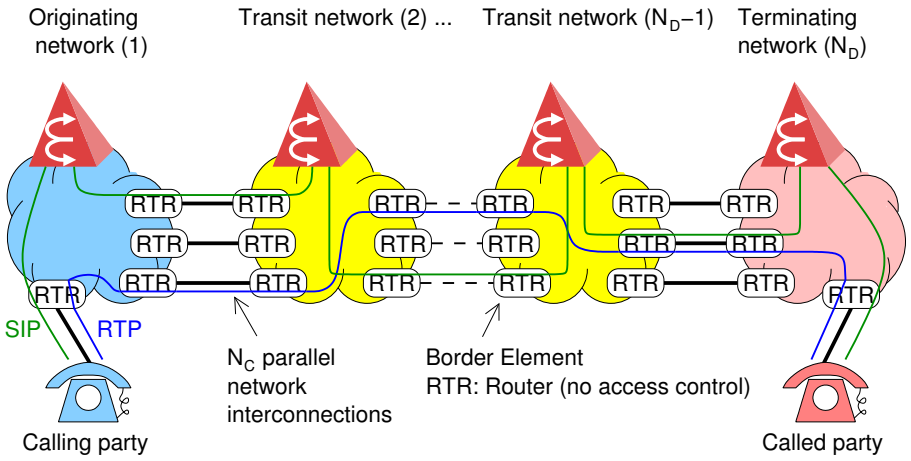


Abbildung 7.6: Szenario 1: keine Schutzmechanismen am Netzübergang

- **Szenario 1:** In diesem Szenario soll zu Vergleichszwecken angenommen werden, dass keine Zugriffskontrollmechanismen am Netzübergang platziert werden, d. h. die Randelemente sind einfache Router und es wird kein Steuer-Protokoll verwendet (siehe Abbildung 7.6).

- **Szenario 2:** Es werden Paketfilter als Randelemente eingesetzt, die vom B2BUA der jeweils eigenen Domäne über ein Protokoll zur Pfad-entkoppelten Signalisierung (z. B. IETF MIDCOM) gesteuert werden (siehe Abbildung 7.7).

  Um sicherzustellen, dass auch bei Einsatz dynamischer Routing-Protokolle alle Verbindungen zustande kommen können, soll das einfache, aber ineffiziente Verfahren angewendet werden, bei dem Pinholes in alle in Frage kommenden Medienkomponenten eingetragen werden (vgl. Abschnitt 7.1, Anforderung 1).

- **Szenario 3:** Es werden Paketfilter als Randelemente eingesetzt, die vom über ein Protokoll zur Pfad-gekoppelten Signalisierung (z. B. IETF NSIS) gesteuert werden, dessen Signalisiernachrichten Ende-zu-Ende, zwischen den Multimedia-Endpunkten der Teilnehmer gesendet werden. Zur Autorisierung der Steuernachrichten soll das in Abschnitt 4.6.5 beschriebene, Token-basierte Verfahren zum Einsatz kommen (siehe Abbildung 7.8).

- **Szenario 4:** Alle Netzübergänge werden durch Session Border Controller (SBC) geschützt. Die zentralen B2BUA sind dennoch vorhanden, zur Verkehrslenkung und Autorisierung in der Anwendungsschicht (siehe Abbildung 7.9)

7.2.2 Beeinflusste Parameter

Aus der Sicht der Teilnehmer können Firewalls folgende Einflüsse auf die Dienstgüte haben:

- Verzögerungen und Paketverluste beim Filtern (legitimer) Medienströme können einen negativen Einfluss auf die Übertragungsqualität haben. Diese Effekte sind unabhängig von den Architekturen und Protokollen zur Firewall-Steuerung, dafür hängen sie stark von

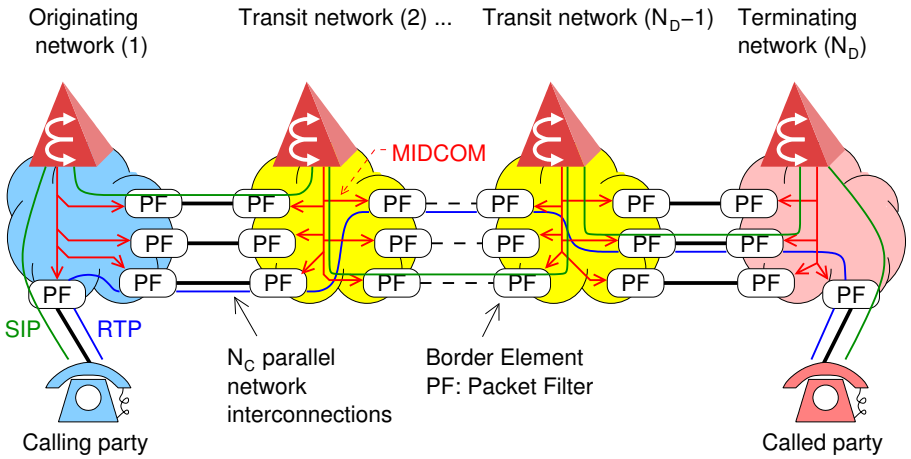


Abbildung 7.7: Szenario 2: Paketfilter mit Pfad-entkoppelter Signalisierung

der konkreten Implementierung des filternden Netzelementes ab. Sie sollen hier deshalb nicht betrachtet werden; in Abschnitt 5.5 wurden entsprechende beispielhafte Messungen an Linux/Netfilter diskutiert.

- Die Maßnahmen, die zur Konfiguration der Firewalls auf dem Medienpfad notwendig sind, können den Aufbau einer neuen Multimedia-Sitzung verzögern.

  In [E.721] werden Grenzwerte für den so genannten Rufverzug (engl. *Post-selection Delay*) und den Meldeverzug (engl. *Answer Signal Delay*) beim Aufbau von Verbindungen im ISDN empfohlen (siehe Tabelle 7.3), welche dort in der Terminologie des ISDN und seiner Protokolle definiert sind. Diese Definitionen können auf SIP übertragen werden:

  – Der *Rufverzug P* sei als die Dauer definiert, die vom Senden der *INVITE*-Nachricht bis zum Empfang der *180 RINGING*-Nachricht – jeweils durch den User Agent des rufenden Teilnehmers – verstreicht.

  – Der *Meldeverzug A* sei als die Dauer definiert, die vom Senden einer *200 OK*-Nachricht, die eine *INVITE*-Transaktion abschließt, durch den User Agent des gerufenen Teilnehmers bis zum Empfang durch den rufenden Teilnehmer verstreicht.

  Zur Illustration sind beide Zeitspannen in das Nachrichtensequenzdiagramm in Abbildung 4.10 eingezeichnet, wie auch die Bearbeitungsdauer δ einer *PER*-Nachricht in der Middlebox und die Antwortzeit R einer *PER*-Transaktion aus Sicht des MIDCOM-Agents.

Während für den Rufverzug relativ hohe Werte toleriert werden können, ist der Meldeverzug recht kritisch, d. h. die Dauer zwischen dem Abnehmen des Telefonhörers durch den gerufenen Teilnehmer und dem bidirektionalen Durchschalten des Sprachkanals, verbunden mit dem Abschalten des Freizeichens. Ist er zu groß, können die ersten Worte einer Konversation verloren gehen, was von den Teilnehmern als störend empfunden wird. Dementsprechend werden in Tabelle 7.3 für den Meldeverzug deutlich engere Grenzwerte als für den Rufverzug gefordert.

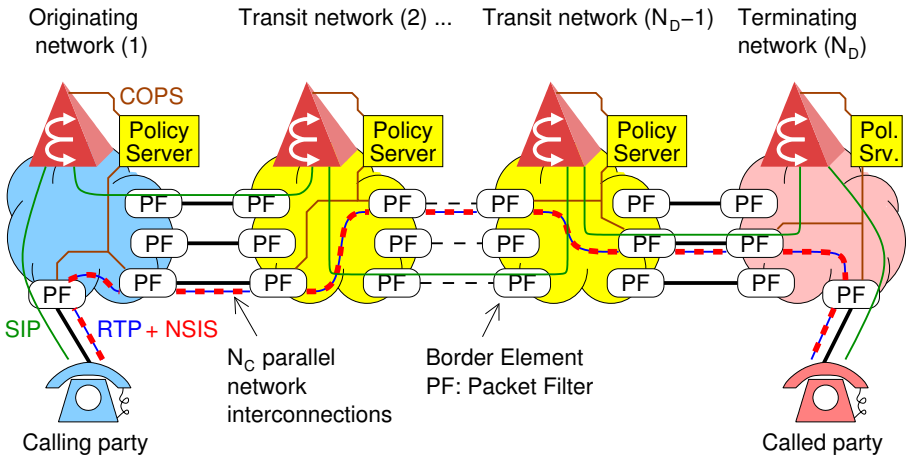


Abbildung 7.8: Szenario 3: Paketfilter mit Pfad-gekoppelter Ende-zu-Ende-Signalisierung

Tabelle 7.3: Empfohlene Grenzwerte für Rufverzug (engl. *Post-selection Delay*) und Meldeverzug (engl. *Answer Signal Delay*) im ISDN nach [E.721]: Mittelwerte und 95%-Quantile für Orts-, Fern- und internationale Verbindungen bei normaler und hoher Netzauslastung

GOS parameter	Normal Load		High Load	
	Mean	95%	Mean	95%
Post-selection delay (*en bloc* sending)				
Local connection	3.0 sec	6.0 sec	4.5 sec	9.0 sec
Toll connection	5.0 sec	8.0 sec	7.5 sec	12.0 sec
International connection	8.0 sec	11.0 sec	12.0 sec	16.5 sec
Answer Signal Delay				
Local connection	0.75 sec	1.5 sec	1.0 sec	2.0 sec
Toll connection	1.5 sec	3.0 sec	2.0 sec	4.0 sec
International connection	2.0 sec	5.0 sec	3.3 sec	6.5 sec

Der Ressourcenbedarf beim Netzbetreiber hängt von den konkreten Implementierungen der Systeme ab. Dennoch können folgende Parameter einen Anhaltspunkt liefern:

- Die Anzahl der zusätzlichen Signalisiernachrichten N_M, die aufgrund der Firewall-Steuerung für den Aufbau einer Multimedia-Sitzung benötigt werden.
- Die Anzahl zusätzlicher kryptographischer Operationen N_H in der Anwendungsschicht, z. B. zum Erstellen oder Prüfen von Autorisierungs-Tokens.
- Die Gesamtzahl von Policy Rules N_U, die für eine neue Multimedia-Sitzung in Medienkomponenten eingetragen werden, unabhängig davon, ob sie letztendlich verwendet werden oder nicht.

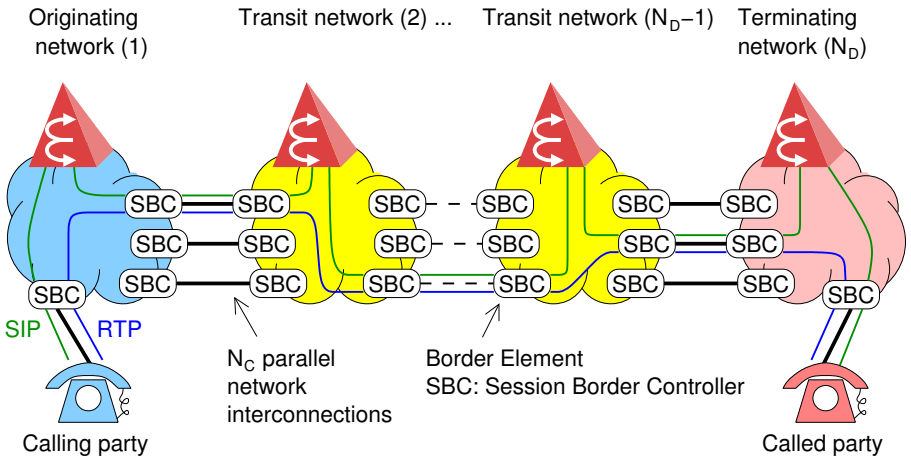


Abbildung 7.9: Szenario 4: Schutz mit “Session Border Controller”

7.2.3 Annahmen bei der Modellierung

Die zu bestimmenden Werte für Ruf- und Meldeverzug werden im Wesentlichen von der Bearbeitung von Signalisiernachrichten in den beteiligten Protokollinstanzen, sowie vom Transport der Nachrichten zwischen ihnen beeinflusst.

Für die lokalen Bearbeitungdauern werden folgende Werte angenommen:

- Bearbeitungsdauer einer SIP-Nachricht, die keine *INVITE*-Nachricht ist, in einem User Agent, B2BUA oder der Signalisierkomponente eines Session Border Controllers: δ_S
- Bearbeitungsdauer einer SIP *INVITE*-Nachricht im B2BUA oder SBC: δ_{SI}.

  Dieser Wert kann deutlich höher sein als δ_S, da Strukturen zum Speichern der Zustandsinformationen über die neue Sitzung erzeugt werden müssen. Ferner müssen evtl. externe Verzeichnisdienste (z. B. DNS) zum Routing der Nachricht oder zur Authentisierung des Teilnehmers abgefragt werden.
- Bearbeitungsdauer für die Nachrichten einer Transaktion zur Firewall-Signalisierung: δ. Diese Dauer wird für alle in Frage kommenden Protokolle (z. B. MIDCOM, NSIS, COPS, etc.) angenommen und tritt in der Medienkomponente und in der steuernden Instanz auf.
- Dauer zum Eintragen eines Pinholes in die Medienkomponente: δ_u.
- Dauer der Berechnung einer kryptographischen Hash-Funktion zum Erstellen oder Prüfen eines Autorisierungs-Tokens (vgl. Abschnitt 4.6.5): δ_H.

Um die Verzögerungen beim Transport der Signalisiernachrichten bestimmen zu können, werden zunächst folgende unidirektionale Verzögerungen auf der IP-Schicht zu Grunde gelegt (siehe Abbildung 7.10):

- Auf dem Zugangslink („access“) vom Teilnehmer bis zum Rand er ersten Domäne: Δ_A
- Zwischen den Netzelementen innerhalb einer Domäne („internal“): Δ_I
- Zwischen den zentralen Komponenten einer Domäne (z. B. SIP B2BUA und Autorisierungs-Server) innerhalb eines lokalen Netzes im Rechenzentrum („LAN“): Δ_L
- Auf den Links zwischen zwei Domänen („transfer“): Δ_T

Entsprechend werden die Paketverlustwahrscheinlichkeiten $p_{L,A}$, $p_{L,I}$, $p_{L,L}$, $p_{L,T}$ angenommen. Desweiteren werden noch Verzögerungen und Verlustwahrscheinlichkeiten für Pfade definiert, die aus dem Verketten mehrerer der oben genannten Strecken entstehen:

- Zwischen dem Endgerät und dem SIP B2BUA der benachbarten Transitdomäne:
  $\Delta_{AI} = \Delta_A + \Delta_I$, $p_{L,AI} = 1 - (1 - p_{L,A})(1 - p_{L,I})$
- Zwischen den B2BUA zweier benachbarter Transitdomänen:
  $\Delta_{ITI} = 2\Delta_I + \Delta_T$, $p_{L,ITI} = 1 - (1 - p_{L,I})^2 (1 - p_{L,T})$

In Kapitel 6 wurde der Einfluss des Transportschichtprotokolls auf die Dauer von Signalisier-Transaktionen untersucht und modelliert. Werden die Links als symmetrisch angenommen, können daraus die Verzögerungen beim unidirektionalen Transport von Nachrichten zwischen zwei Protokollinstanzen der Anwendungsschicht V ermittelt werden, die sich aus der unidirektionalen Verzögerung auf der IP-Schicht plus einer zusätzlichen Verzögerung W durch das Transportschichtprotokoll ergeben:

$$V = \frac{R-\delta}{2} = \Delta + W \tag{7.1}$$

Für UDP-basierten Transport ist diese Berechnung nur gültig, wenn alle betrachteten Nachrichten durch Bestätigungen in der Anwendungsschicht quittiert werden, was bei den hier betrachteten Protokollen – insbesondere SIP – der Fall ist.

In Kapitel 6 wurde gezeigt, dass W u. a. von der Wahl des Transportschichtprotokolls, der Verzögerung auf der IP-Schicht, der Paketverlustwahrscheinlichkeit und der Transaktionsrate abhängt. Ferner wurde gezeigt, dass für verschiedene Transaktionsraten λ_T unterschiedliche Transportschichtprotokolle die jeweils geringsten zusätzlichen Verzögerungen W verursachen. Dementsprechend soll davon ausgegangen werden, dass zwei verschiedene Transportmodi zur Verfügung stehen und entsprechend der Signalisierlast genutzt werden:

Für Signalisier-Assoziationen, an denen das Multimedia-Endgerät eines Teilnehmers beteiligt ist und die folglich nur „niedrige“ Transaktionsraten transportieren müssen, soll ein für diese Transaktionsrate optimales (z. B. UDP-basiertes) Transportverfahren zum Einsatz kommen. Es wird angenommen, dass dieses beim unidirektionalen Nachrichtentransport die mittleren Verzögerungen $V_{L,A}$ bzw. $V_{L,AI}$ verursacht, wenn die Verzögerung auf der IP-Schicht Δ_A bzw. Δ_{AI} und die Paketverlustwahrscheinlichkeiten $p_{L,A}$ bzw. $p_{L,AI}$ sind. Können hingegen zwischen zwei Netzelementen im Kernnetz die Signalisiernachrichten vieler Teilnehmer in einer Assoziation gebündelt werden, kann ggf. ein anderer (z. B. auf SCTP Multistreaming basierender) Transport gewählt werden, der bei dieser „hohen“ Transaktionsrate je nach Einsatzort die Verzögerungen $V_{H,I}$, $V_{H,L}$, $V_{H,T}$ bzw. $V_{H,ITI}$ verursacht.

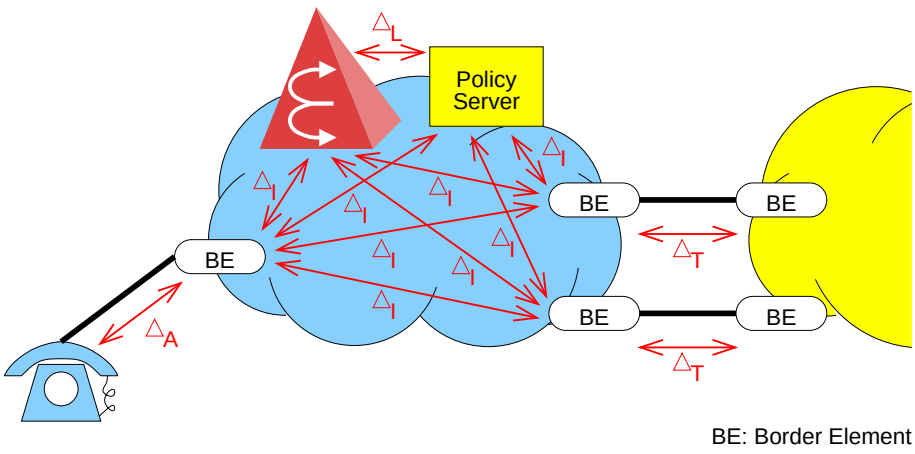


Abbildung 7.10: Angenommene Verzögerungen auf der IP-Schicht

7.2.4 Bestimmung von Ruf- und Meldeverzug sowie Ressourcenverbrauch

7.2.4.1 Szenario 1: Keine Zugriffskontrollmechanismen am Netzübergang

Für Szenario 1 (siehe Abbildung 7.6) erfolgt die Signalisierung beim Verbindungsaufbau prinzipiell wie in Abbildung 2.8 dargestellt.

Die *INVITE*-Nachricht muss zunächst vom User Agent des rufenden Teilnehmers bis zum B2BUA der ersten Transitdomäne übertragen werden. Nach der Verarbeitung dort erfolgt die Weiterleitung zu und Verarbeitung in $(N_D - 1)$ weiteren SIP-Servern, sowie der Transport zum User Agent des rufenden Teilnehmers. Dieser benötigt die lokale Bearbeitungsdauer δ_S zum Erstellen der *180 Ringing*-Nachricht, die über die selbe Kette von SIP-Servern zurücktransportiert wird. Auch die *200 OK*-Nachricht wird auf diese Weise transportiert. Ruf- und Meldeverzug berechnen sich daraus wie folgt:

$$P_1 = \tau_{SI} + \delta_S + \tau_S \tag{7.2}$$

$$A_1 = \tau_S \tag{7.3}$$

$$\text{mit} \quad \tau_{SI} = V_{L,AI} + N_D \cdot \delta_{SI} + (N_D - 1) \cdot V_{H,ITI} + V_{L,AI} \tag{7.4}$$

$$\text{und} \quad \tau_S = V_{L,AI} + N_D \cdot \delta_S + (N_D - 1) \cdot V_{H,ITI} + V_{L,AI} \tag{7.5}$$

Da in diesem Szenario keine Firewalls vorkommen, gilt für den zusätzlichen Ressourcenverbrauch: $N_{M,1} = 0$, $N_{H,1} = 0$ und $N_{U,1} = 0$.

7.2.4.2 Szenario 2: Pfad-entkoppelte Signalisierung

Soll das in Abbildung 7.7 dargestellte Szenario 2 mit Pfad-entkoppelter Signalisierung ohne den optionalen SIP Precondition-Mechanismus implementiert werden, erfolgt die Signalisierung wie in Abbildung 4.10 dargestellt.

Der Transport der SIP-Nachrichten erfolgt prinzipiell auf die selbe Weise wie in Szenario 1, so dass sich hier die selben Verzögerungen ergeben. Zusätzlich veranlasst jeder B2BUA beim Bearbeiten der *INVITE*- bzw. *200 OK*-Nachricht das Öffnen eines Pinholes in jeder Medienkomponente der eigenen Domäne. Das Erzeugen der entsprechenden Anforderungsnachrichten verursacht die Verzögerung δ. Für den Transport zu den Medienkomponenten wird jeweils eine Signalisier-Assoziation benötigt, die die Signalisierung für alle Multimedia-Sitzungen tragen kann. Im Fall der Domänen Nr. 1 und Nr. N_D sind $N_C + 1$ Medienkomponenten beteiligt, für die Domänen Nr. $2 \ldots N_D - 1$ sind es $2 \cdot N_C$.

Das Eintragen kann für alle Medienkomponenten der selben Domäne gleichzeitig erfolgen, so dass nur die längste Antwortzeit $\widehat{R_{N_C+1}}$ bzw. $\widehat{R_{2 \cdot N_C}}$ der $N_C + 1$ bzw. $2 \cdot N_C$ parallelen Transaktionen relevant ist. Geht man vereinfachend davon aus, dass alle Antwortzeiten dem Mittelwert R entsprechen, ergibt sich für dieses Szenario:

$$
\begin{aligned}
P_2 &= \tau_{\mathrm{SI}} + 2\left(\delta + \widehat{R_{N_\mathrm{C}+1}}\right) + (N_\mathrm{D}-2)\cdot\left(\delta + \widehat{R_{2\cdot N_\mathrm{C}}}\right) + \delta_\mathrm{S} + \tau_\mathrm{S} \\
&\approx P_1 + \tau_0 \qquad (7.6)\\
A_2 &= \tau_\mathrm{S} + 2\left(\delta + \widehat{R_{N_\mathrm{C}+1}}\right) + (N_\mathrm{D}-2)\left(\delta + \widehat{R_{2\cdot N_\mathrm{C}}}\right) \\
&\approx A_1 + \tau_0 \qquad (7.7)\\
\text{mit}\quad \tau_0 &= N_\mathrm{D}\cdot(2V_{\mathrm{H,I}} + 2\delta + \delta_u) \qquad (7.8)
\end{aligned}
$$

Bei der angenommenen, einfachen Strategie zum Umgang mit dynamischen Routing, bei der Pinholes einfach in alle in Frage kommenden Medienkomponenten eingetragen werden, werden bei zwei Pinholes pro Medienkomponente (Vorwärts- und Rückwärtsrichtung) insgesamt

$$N_{\mathrm{U},2} = 2\,(1 + (N_\mathrm{D}-1)\cdot 2\cdot N_\mathrm{C} + 1) = 4\,((N_\mathrm{D}-1)N_\mathrm{C} + 1) \qquad (7.9)$$

Pinholes geöffnet. Dazu werden $N_{\mathrm{M},2} = 2N_{\mathrm{U},2}$ Nachrichten (Anfrage + Bestätigung) benötigt. Bei diesem Verfahren werden i. d. R. keine kryptographischen Verfahren in der Anwendungsschicht benötigt (vgl. Abschnitt 7.1, Anforderung 4), d. h. $N_{\mathrm{H},2} = 0$.

Gegenüber dem ungeschützten Szenario 1 erhöhen sich sowohl Ruf- als auch Meldeverzug. Da insbesondere der Meldeverzug kritisch ist, kann dessen Erhöhung zu Lasten einer weiteren Erhöhung des Rufverzuges vermieden werden, indem der SIP Precondition-Mechanismus (siehe Abbildung 5.5) angewendet wird. Dies hat auch den Vorteil, dass eventuell mögliche *Ghost Rings* vermieden werden (vgl. Abschnitt 5.4). Die Bearbeitung der *INVITE*- und *180 Ringing*-Nachrichten erfolgt dabei identisch wie im Szenario ohne Precondition-Mechanismus. Zwischen diesen beiden Nachrichten wird zusätzlich die *183 Session Progress*-Nachricht verarbeitet, die das Öffnen der Pinholes in Vorwärtsrichtung auslöst. Die Bearbeitung dieser Nachricht erfordert so viel Zeit wie die *200 OK*-Nachricht im Szenario ohne Precondition-Mechanismus. Desweiteren muss noch der Transport der *UPDATE*-Nachricht abgewartet werden. Die in Abbildung 5.5 dargestellten Nachrichten F10, F11, F12, F13, F16 und F17 (*PRACK*, bzw. *200 OK*) liegen hingegen nicht auf dem „kritischen Pfad“, da vor dem Versenden der oben genannten Nachrichten nicht auf diese Bestätigungen gewartet werden muss. Wenn der Precondition-Mechanismus verwendet wird, sind alle Pinholes konfiguriert, bevor das Multimedia-Endgerät des gerufenen Teilnehmers mit dem Klingeln beginnt. Die *200 OK*-Nachricht nach dem Annehmen des Rufes kann daher wie im Szenario ohne Firewalls transportiert werden. Somit ergeben sich folgende Verzögerungen:

$$
\begin{aligned}
P_{2\mathrm{P}} &= P_2 + A_2 + \tau_\mathrm{S} \approx P_1 + 2\tau_\mathrm{S} + 2\tau_0 \qquad (7.10)\\
A_{2\mathrm{P}} &= A_1 \qquad (7.11)
\end{aligned}
$$

Durch die Preconditions kommen sieben weitere SIP-Nachrichten (*183*, *PRACK*, *200*,*UPDATE*, *200*, *PRACK*, *200*) hinzu, die zwischen beiden User Agents übertragen werden. Somit erhöht sich die Zahl der zusätzlichen Nachrichten auf $N_{\mathrm{M,2P}} = N_{\mathrm{M},2} + 7(N_\mathrm{D}+1)$. Die Zahl der zu öffnenden Pinholes und die Zahl der kryptographischen Operationen bleiben unverändert, d. h. $N_{\mathrm{U,2P}} = N_{\mathrm{U},2}$ und $N_{\mathrm{H,2P}} = N_{\mathrm{H},2}$.

7.2.4.3 Szenario 3: Pfad-gekoppelte Ende-zu-Ende-Signalisierung

Wird Pfad-gekoppelte Signalisierung Ende-zu-Ende verwendet (siehe Abbildung 7.8), kann die Signalisierung ohne SIP Preconditions wie in Abbildung 4.14 dargestellt erfolgen. Auch hier erfolgt der Transport der SIP-Nachrichten gegenüber Szenario 1 unverändert. Beim Bearbeiten der *INVITE*- bzw. *200 OK*-Nachricht fordert jeder B2BUA ein Autorisierungs-Token vom Autorisierungsserver der eigenen Domäne an, der dieses erst durch Berechnen eines Hash-Wertes erzeugen muss (Gesamtdauer: τ_1).

Desweiteren muss zum Öffnen der Pinholes für die Medienströme für beide Richtungen je eine Nachricht des Pfad-gekoppelten Firewall-Signalisierprotokolls (bei NSIS: *CREATE*) und ihre Bestätigung zwischen den beiden Multimedia-Endgeräten ausgetauscht werden (τ_2). Vor dem Eintragen der Pinholes leitet jede der $2N_\mathrm{D}$ Medienkomponenten auf dem Pfad die Autorisierungs-Tokens zur kryptographischen Prüfung an den Autorisierungsserver weiter (τ_3).

Die Signalisierung zum Öffnen der Pinholes für Medienströme in Vorwärtsrichtung erfolgt erst nach dem Empfang der *200 OK*-Nachricht durch den User Agent des gerufenen Teilnehmers. Obwohl die dafür benötigte Zeitdauer somit formal nicht mehr zum Meldeverzug beiträgt, wird sie hier dennoch berücksichtigt, da erst nach Abschluss dieses Vorgangs der Sprachkanal vollduplex durchgeschaltet ist. Somit ergibt sich für dieses Szenario:

$$P_3 = P_1 + \tau_1 + \tau_2 + \tau_3 \tag{7.12}$$

$$A_3 = A_1 + \tau_1 + \tau_2 + \tau_3 \tag{7.13}$$

$$\text{mit} \quad \tau_1 = N_\mathrm{D}\left(2V_\mathrm{H,L} + 2\delta + \delta_\mathrm{H}\right) , \tag{7.14}$$

$$\tau_2 = 2\left(2V_\mathrm{L,A} + N_\mathrm{D}\cdot V_\mathrm{H,I} + (N_\mathrm{D}-1)\,V_\mathrm{H,T}\right) + 2\left(2N_\mathrm{D}+2\right)\delta \tag{7.15}$$

$$\text{und} \quad \tau_3 = 2N_\mathrm{D}\left(2V_\mathrm{H,I} + 2\delta + \delta_\mathrm{H} + \delta_u\right) = 2\tau_0 + 2N_\mathrm{D}\cdot\delta_\mathrm{H} \tag{7.16}$$

Bei diesem Verfahren werden nur die tatsächlich benötigten Pinholes eingetragen, d. h. je eines in Vorwärts- und Rückwärtsrichtung in beiden Medienkomponenten jeder Transitdomäne: $N_\mathrm{U,3} = 4N_\mathrm{D}$. Für beide Richtungen wird in jeder Domäne ein Autorisierungs-Token erstellt und zweimal geprüft: $N_\mathrm{H,3} = 6N_\mathrm{D}$. Dafür und für die eigentliche, Pfad-gekoppelte Signalisierung in beiden Richtungen, mit Bestätigungen werden $N_\mathrm{M,3} = 12N_\mathrm{D} + 4(N_\mathrm{D}+1) = 16N_\mathrm{D} + 4$ zusätzliche Nachrichten benötigt.

Werden in diesem Szenario SIP Preconditions zur Vermeidung von Ghost Rings und zur Verringerung des Meldeverzugs eingesetzt, erfolgt die Signalisierung wie in Abbildung 7.11 dargestellt. Die Pfad-gekoppelten Signalisiervorgänge zum Öffnen der Pinholes für Medienströme in Vorwärts- bzw. Rückwärtsrichtung erfolgen hier zeitlich überlappend. Der „kritische Pfad“ zur Bestimmung des Rufverzuges beginnt mit der Übertragung der *INVITE*- und *183*-Nachrichten über die Kette der B2BUA, die dabei jeweils die Erzeugung von Autorisierungs-Tokens anfordern. Anschließend erfolgt die Pfad-gekoppelte Signalisierung in Vorwärtsrichtung und das Senden einer *UPDATE*-Nachricht an den User Agent des gerufenen Teilnehmers. Unter der Annahme, dass die Firewall-Signalisierung in Rückwärtsrichtung nicht wesentlich länger dauert als in Vorwärtsrichtung, ist sie zu diesem Zeitpunkt bereits abgeschlossen, so dass sofort die *180*-Nachricht gesendet werden kann. Da die Konfiguration der Medienkomponenten damit abgeschlossen ist, kann die *200 OK*-Nachricht ohne weitere Verzögerungen übertragen werden. Ruf- und Meldeverzug berechnen sich somit wie folgt:

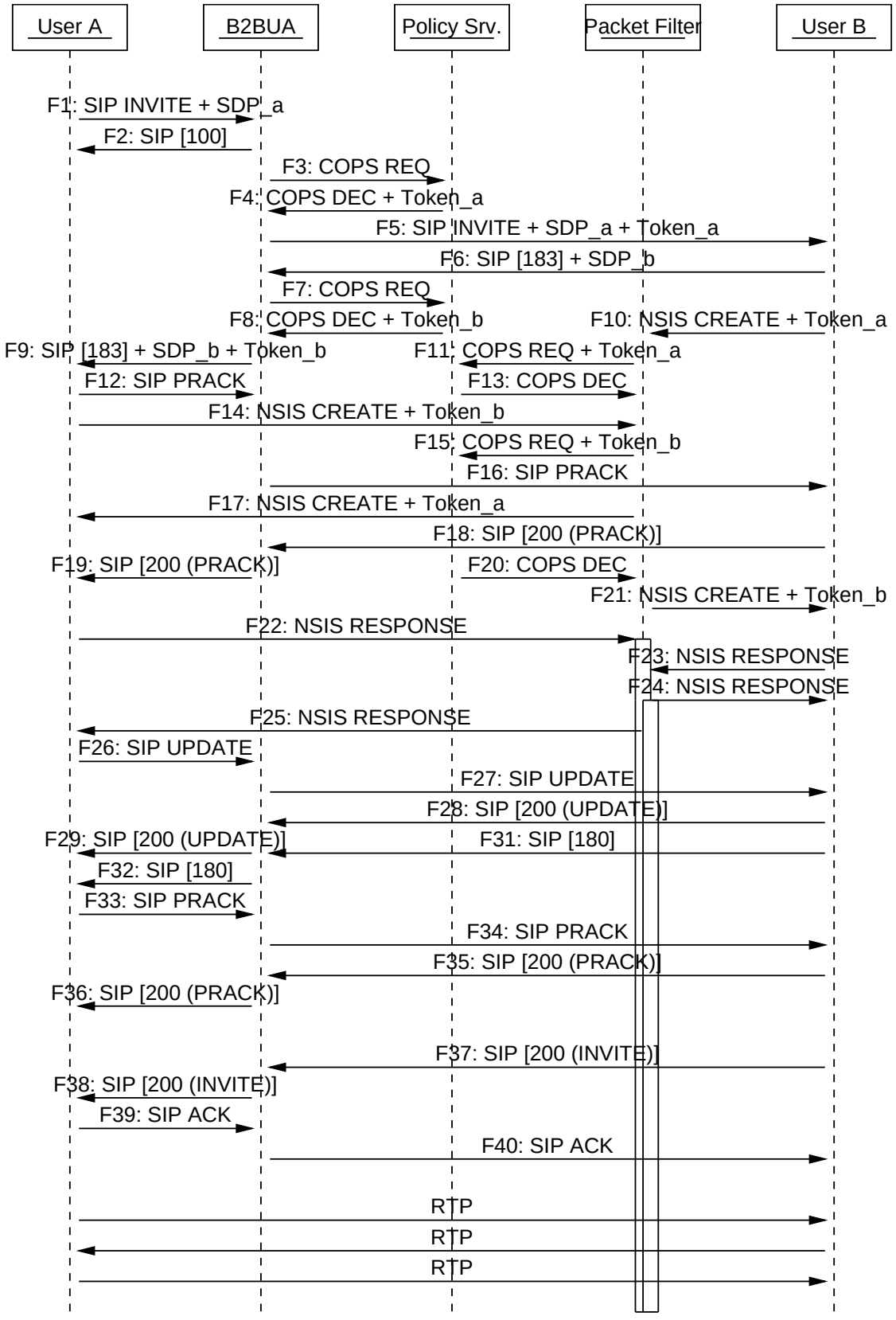


Abbildung 7.11: Kopplung von NSIS- und SIP-Signalisierung mit SIP Preconditions

$$
\begin{aligned}
P_{3P} &= \tau_{SI} + \tau_1 + \delta_S + \tau_S + \tau_1 + \tau_2 + \tau_3 + \tau_S + \tau_S \\
&= P_1 + 2\tau_1 + \tau_2 + \tau_3 + 2\tau_S \qquad (7.17) \\
A_{3P} &= \tau_S = A_1 \qquad (7.18)
\end{aligned}
$$

Durch die Verwendung des Preconditions-Mechanismus kommen in diesem Szenario sechs weitere SIP-Nachrichten hinzu (zwei *PRACK*-, eine *UPDATE*- und jeweils eine zugehörige *200 OK*-Nachricht), weshalb insgesamt $N_{M,3P} = N_{M,3} + 6(N_D + 1)$ zusätzliche Nachrichten benötigt werden. Die Zahl der zu öffnenden Pinholes und die Zahl der kryptographischen Operationen bleiben unverändert, d. h. $N_{U,3P} = N_{U,3}$ und $N_{H,3P} = N_{H,3}$.

7.2.4.4 *Szenario 4: Netzübergänge mit Session Border Controller*

Zum Vergleich soll ein weiteres Szenario ohne verteilte Firewalls betrachtet werden; stattdessen sollen die Netzübergänge hier mit Session Border Controllern geschützt werden, die SIP-Signalisierung und RTP-Medienströme in einem Netzelement bearbeiten (siehe Abbildung 7.9). Es wird in diesem Szenario kein Signalisierprotokoll für die Steuerung abgesetzter Medienkomponenten benötigt, allerdings erhöht sich die Anzahl der beteiligten SIP-Instanzen auf drei pro Transitdomäne (zwei SBC an den beiden Netzübergängen und ein B2BUA für Verkehrslenkung, Abrechnung, etc.). Ein Nachrichtensequenzdiagramm für den Aufbau einer Multimedia-Sitzung über zwei Transitdomänen hinweg ist in Abbildung 7.12 dargestellt.

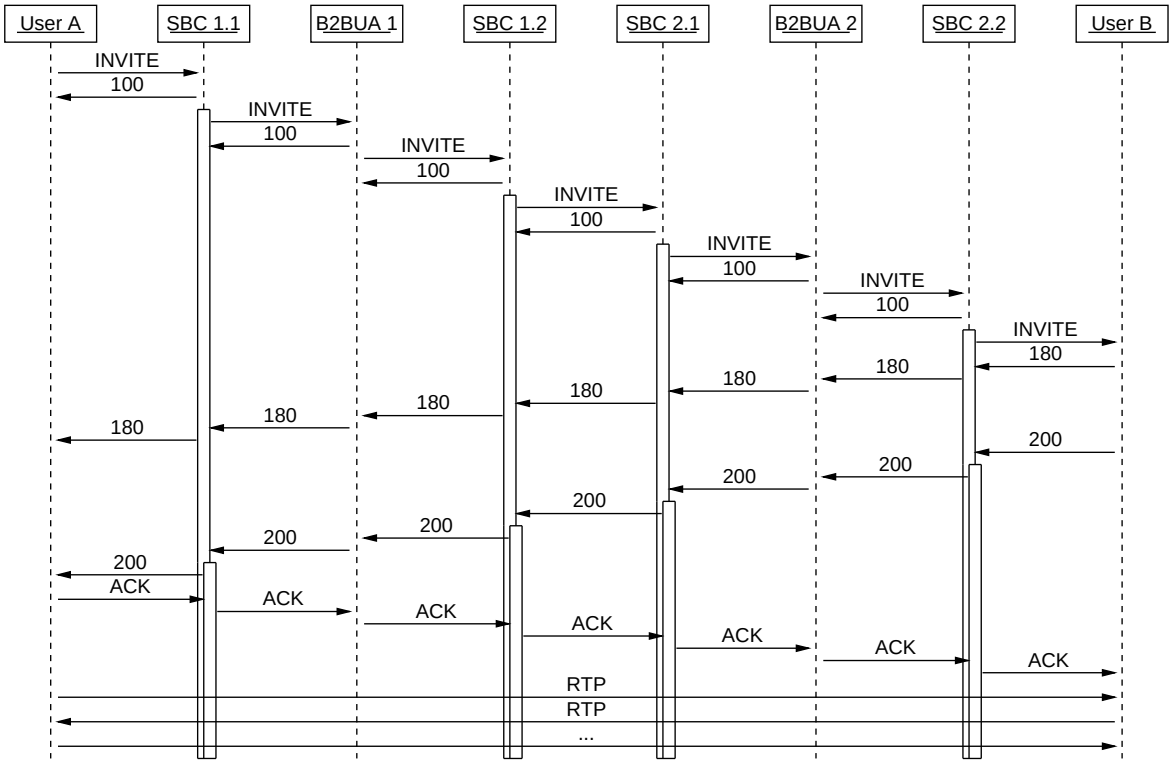


Abbildung 7.12: Nachrichtensequenzdiagramm zu Szenario 4

Wie bereits in Abschnitt 4.1.1 dargestellt, weist der interne Aufbau eines SBC große Ähnlichkeiten zur MIDCOM-Architektur auf. Deshalb sollen die Verzögerungen beim Eintragen von Pinholes analog zu Szenario 2 modelliert werden, wobei der Transport von Nachrichten zwischen Signalisier- und Medienkomponente vernachlässigt wird, da er über eine geräteinterne Schnittstelle abgewickelt wird. Der Rufverzug ergibt sich in diesem Szenario somit im Wesentlichen aus der Dauer für den Transport und die Bearbeitung der *INVITE*- und *180 RINGING*-Nachrichten durch diese, im Vergleich zu den anderen Szenarien dreimal längere Kette von SIP-Instanzen, der Meldeverzug aus Transport und Bearbeitung der *200 OK*-Nachricht:

$$P_4 = \tau_{\mathrm{SI},4} + 2N_\mathrm{D} \cdot \delta_u + \delta_\mathrm{S} + \tau_{\mathrm{S},4} \tag{7.19}$$

$$A_4 = \tau_{\mathrm{S},4} + 2N_\mathrm{D} \cdot \delta_u \tag{7.20}$$

$$\text{mit} \quad \tau_{\mathrm{SI},4} = V_{\mathrm{L,A}} + N_\mathrm{D}\,(3\delta_\mathrm{SI} + 2V_{\mathrm{H,I}}) + (N_\mathrm{D} - 1)\,V_{\mathrm{H,T}} + V_{\mathrm{L,A}} \tag{7.21}$$

$$\text{und} \quad \tau_{\mathrm{S},4} = V_{\mathrm{L,A}} + N_\mathrm{D}\,(3\delta_\mathrm{S} + 2V_{\mathrm{H,I}}) + (N_\mathrm{D} - 1)\,V_{\mathrm{H,T}} + V_{\mathrm{L,A}} \tag{7.22}$$

Durch Verwendung des SIP Precondition-Mechanismus kann der Meldeverzug um die lokalen Bearbeitungszeiten beim Öffnen der Pinholes verkürzt werden, zu Lasten des Rufverzuges. Der Vorteil fällt hier jedoch weit geringer aus als in den Szenarien 2 und 3, da keine Kommunikation über das Netz benötigt wird. Zwischen die *INVITE*- und die *180*-Nachricht schiebt sich bei diesem Verfahren noch eine *183*-Nachricht, bei der die Pinholes für Medienströme in Vorwärtsrichtung geöffnet werden, und eine *UPDATE*-Nachricht. Nicht auf dem „kritischen Pfad“ liegen zwei zusätzliche *PRACK*-Nachrichten und drei *200*-Nachrichten, die *UPDATE* bzw. *PRACK* quittieren. Somit ergeben sich folgende Verzögerungen:

$$P_{4\mathrm{P}} = P_4 + \tau_{\mathrm{S},4} + \delta_\mathrm{S} + 2N_\mathrm{D} \cdot \delta_u + \tau_{\mathrm{S},4} + \delta_\mathrm{S} \tag{7.23}$$

$$= \tau_{\mathrm{SI},4} + 4N_\mathrm{D} \cdot \delta_u + 3\delta_\mathrm{S} + 3\tau_{\mathrm{S},4} \tag{7.24}$$

$$A_{4\mathrm{P}} = \tau_{\mathrm{S},4} \tag{7.25}$$

In diesem Szenario ist auch das Auftreten von Ghost Rings weniger wahrscheinlich, da die geräteinterne Kommunikation weniger fehleranfällig ist. Durch die 1 : 1-Beziehung zwischen Signalisier- und Medienkomponente können drohende Ressourcenengpässe einfacher vorhergesagt und schon bei der Bearbeitung der *INVITE*-Nachricht berücksichtigt werden.

In dem hier betrachteten Szenario werden vier SIP-Nachrichten (*INVITE*, *180*, *200* und *ACK*) zwischen den beiden User Agents durch die Kette der SIP-Instanzen (B2BUA und SBC) gesendet, eine weitere Nachricht (*100*) wird auf allen bis auf der letzten Strecke gesendet. Dies entspricht der Situation in dem ungeschützten Referenz-Szenario 1, allerdings müssen drei statt nur einer SIP-Instanz pro Transitdomäne durchlaufen werden. Somit ergibt sich eine Zahl zusätzlicher Nachrichten von

$$N_{\mathrm{M},4} = (4\,(3N_\mathrm{D} + 1) + 1 \cdot 3N_\mathrm{D}) - (4\,(N_\mathrm{D} + 1) + 1 \cdot N_\mathrm{D}) = 10N_\mathrm{D}\,. \tag{7.26}$$

Werden SIP Preconditions verwendet, kommen sieben weitere Nachrichten (*183*, *UPDATE*, zwei *PRACK*, drei *200*) hinzu, die Zahl zusätzlicher Nachrichten erhöht sich so auf

$$N_{\mathrm{M,4P}} = N_{\mathrm{M,4}} + 7\,(3N_{\mathrm{D}} + 1) = 31N_{\mathrm{D}} + 7\ . \tag{7.27}$$

Bei diesem Netzdesign wird erzwungen, dass die beiden Medienströme über jeweils einen bestimmten SBC an den beiden Grenzen einer Transitdomäne laufen; nur dort müssen entsprechende Regeln eingetragen werden: $N_{\mathrm{U,4}} = 4N_{\mathrm{D}}$. Da die Kommunikation zwischen Signalisier- und Medienkomponente bei einem SBC über eine geräteinterne Schnittstelle stattfindet, werden keine kryptographischen Maßnahmen zum Schutz dieser Informationen benötigt: $N_{\mathrm{H,4}} = 0$.

7.2.5 Bewertung

Betrachtet man die Werte für Ruf- und Meldeverzug, die im vorangegangenen Abschnitt für die Szenarien 1 bis 3 ermittelt wurden, ergeben sich folgende Beziehungen:

$$P_1 < P_2 < P_3 \tag{7.28}$$
$$A_1 < A_2 < A_3 \tag{7.29}$$

bzw.
$$P_{1\mathrm{P}} < P_{2\mathrm{P}} < P_{3\mathrm{P}} \tag{7.30}$$
$$A_{1\mathrm{P}} = A_{2\mathrm{P}} = A_{3\mathrm{P}} \tag{7.31}$$

Unabhängig von der Wahl der zugrundeliegenden Parameter verursacht (aufgrund zusätzlicher, nicht negativer Summanden) die Pfad-gekoppelte Firewall-Signalisierung mit dem Token-basierten Autorisierungsverfahren größere Verzögerungen als die Pfad-entkoppelte Signalisierung, die wiederum zusätzliche Zeit gegenüber dem ungeschützten Szenario benötigt. Durch Verwendung des Precondition-Mechanismus von SIP kann der Meldeverzug bei beiden Verfahren auf den Wert des ungeschützten Szenarios gedrückt werden, zu Lasten des jeweiligen Rufverzuges.

Die größeren Verzögerungen bei der Pfad-gekoppelten Signalisierung können anschaulich mit der Tatsache begründet werden, dass alleine das Zurücksenden des Autorisierungs-Tokens von einem Firewall zur Prüfung im Autorisierungs-Server soviel Zeit in Anspruch nimmt wie eine Pfad-entkoppelte Signalisiertransaktion zum Öffnen eines Pinholes, wobei letztere für alle Medienkomponenten einer Domäne simultan durchgeführt werden können. Selbst wenn auf dieses Token-Verfahren verzichtet wird (was die Erfüllung von Anforderung 4 in Abschnitt 7.1 in Frage stellen kann), gilt $\tau'_1 = 0$, $\tau'_2 = \tau_2$ und $\tau'_3 = 2N_{\mathrm{D}} \cdot \delta_u$, aber $\tau'_2 + \tau'_3 > \tau_0$ und somit $P'_3 > P_2$ und $A'_3 > A_2$, d. h. die Verzögerungen liegen weiterhin über denen der Pfad-entkoppelten Steuerung.

Falls die Paketverlustwahrscheinlichkeit im Netz sehr gering ist, unterscheiden sich die von verschiedenen Transportschichtprotokollen verursachen Verzögerungen nur sehr wenig. Unter den Annahmen

$$V_{\mathrm{H,ITI}} \approx 2V_{\mathrm{H,I}} + V_{\mathrm{H,T}} \tag{7.32}$$
$$V_{\mathrm{L,AI}} \approx V_{\mathrm{L,A}} + V_{\mathrm{H,I}} \tag{7.33}$$

können Ruf- und Meldeverzug in Szenario 4 wie folgt dargestellt werden:

$$\tau_{\mathrm{SI},4} = \tau_{\mathrm{SI}} + 2N_{\mathrm{D}} \cdot \delta_{\mathrm{SI}} \tag{7.34}$$
$$\tau_{\mathrm{S},4} = \tau_{\mathrm{S}} + 2N_{\mathrm{D}} \cdot \delta_{\mathrm{S}} \tag{7.35}$$

$$P_4 = P_1 + 2N_{\mathrm{D}}\,(\delta_{\mathrm{SI}} + \delta_{\mathrm{S}} + \delta_u) \tag{7.36}$$
$$A_4 = A_1 + 2N_{\mathrm{D}}\,(\delta_{\mathrm{S}} + \delta_u) \tag{7.37}$$
$$P_{4\mathrm{P}} = P_1 + 2\,(\tau_{\mathrm{S}} + \delta_{\mathrm{S}}) + 2N_{\mathrm{D}}\,(\delta_{\mathrm{SI}} + 3\delta_{\mathrm{S}} + 2\delta_u) \tag{7.38}$$
$$A_{4\mathrm{P}} = A_1 + 2N_{\mathrm{D}}\delta_{\mathrm{S}} \tag{7.39}$$

Die Verzögerungen beim Verbindungsaufbau in diesem Szenario mit Session Border Controllern entsprechen somit den Verzögerungen beim ungeschützten Szenario zuzüglich weiterer Summanden, die sich aus lokalen Bearbeitungdauern von SIP-Nachrichten ergeben; es kommen jedoch keine weiteren Verzögerungen durch den Transport zusätzlicher Nachrichten hinzu, da solche auch nicht benötigt werden. Verglichen mit den verteilten Firewalls der Szenarien 2 und 3 verursachen die SBC somit geringere Verzögerungen beim Verbindungsaufbau, allerdings nur unter der Annahme, dass die lokale Bearbeitung einer SIP-Nachricht deutlich weniger Zeit in Anspruch nimmt als der Transport einer Nachricht zwischen zwei Netzelementen. Diese Annahme kann sich als nicht erfüllt erweisen, wenn im Zuge der Bearbeitung einer SIP-Nachricht Abfragen an externe Verzeichnisdienste (z. B. DNS, AAA) gestellt werden müssen, die den Transport von Nachrichten erforderlich machen. Der besonders kritische Meldeverzug kann in Szenario 4 auch bei Verwendung von SIP Preconditions nicht auf den Wert der anderen drei Szenarien gesenkt werden.

Bezüglich der Anzahl zusätzlicher Nachrichten ist die Pfad-entkoppelte Signalisierung (Szenario 2) das einzige Verfahren, bei dem diese Zahl von der Anzahl der parallelen Netzübergänge zwischen den Domänen abhängt. Die Ursache dafür ist, dass dies auch das einzige Verfahren ist, bei dem u. U. Pinholes in Medienkomponenten eingetragen werden, die gar nicht auf dem Pfad der Medienströme durch das Netz liegen (d. h. N_{U} unnötig hoch). Ist $N_{\mathrm{C}} = 1$ (weil nur ein Netzübergang vorhanden ist oder weil der Zustand der Routing-Tabellen bekannt ist und so nur die tatsächlich benötigten Pinholes eingetragen werden müssen), verursacht das Verfahren eine geringere zusätzliche Signalisierlast als die beiden konkurrierenden Verfahren der Szenarien 3 und 4. Erst ab $N_{\mathrm{C}} \geq 4$ übersteigt die Nachrichtenzahl bei der Pfad-entkoppelten Signalisierung die der beiden anderen Verfahren auch bei realistischen Szenarien mit relativ wenigen zu durchquerenden Transitdomänen ($N_{\mathrm{D}} < 10$). Die hier vorgestellten Überlegungen berücksichtigen nur die Signalisiernachrichten am Beginn einer neuen Multimedia-Sitzung. Im Verlauf einer solchen Sitzung werden u. U. weitere Nachrichten zur Verlängerung der Gültigkeitsdauern von Policy Rules benötigt. Die Zeitintervalle, in denen solche Nachrichten gesendet werde müssen, können je nach verwendetem Signalisierverfahren stark unterschiedlich sein (vgl. Anforderung 7 in Abschnitt 7.1). Insbesondere bei der Pfad-gekoppelten Signalisierung kann sich die Gesamtzahl der Nachrichten bei langen Multimedia-Sitzungen stark erhöhen. Dies ist auch das einzige Verfahren, bei dem Nachrichten des Firewall-Signalisierprotokolls prinzipbedingt über Domänengrenzen hinweg ausgetauscht und somit kryptographisch gesichert werden müssen (N_{H}).

Zur Illustration wird in Tabelle 7.4 ein Zahlenbeispiel für ein Szenario mit drei Domänen gegeben, z. B. ein Ortsgespräch zwischen zwei Teilnehmern, die Kunden unterschiedlicher Netz-

betreiber sind. Es soll dabei angenommen werden, dass es im Zugangsnetz (z. B. DSL- oder mobilfunkbasiert) zu gewissen Verzögerungen und recht geringen Verlusten beim Transport der IP-Pakete kommt. Verzögerungen, die durch das Verhalten der Transportschichtprotokolle entstehen, werden entsprechend Kapitel 6 berechnet. Dabei werden die schon in Tabelle 6.3 verwendeten Parameter angenommen; insbesondere wurden $\lambda_T = 100\,\frac{\text{Trans.}}{\text{s}}$ als „hohe“ Transaktionsrate zur Berechnung der $V_{H,?}$ und $\lambda_T = 1\,\frac{\text{Trans.}}{\text{s}}$ als „niedrige“ Rate zur Berechnung der $V_{L,?}$ angenommen.

Bei den hier angenommenen Parameterwerten und nur drei Domänenwerden die Anforderungen von [E.721] (siehe Tabelle 7.3) von allen betrachteten Architekturen noch eingehalten, wenn auch mit deutlichen Unterschieden. Für die Bearbeitungsdauer einer SIP *INVITE*-Nachricht wurde in diesem Szenario ein Wert angenommen, der zwar höher liegt als die Bearbeitungsdauern anderer Nachrichten (z. B. aufgrund lokaler Authentizitäts-Prüfungen oder SPIT-Abwehr-

Tabelle 7.4: Zahlenbeispiel für Ruf- und Meldeverzug bei domänenübergreifender Sitzung

Annahmen				
Anz. Domänen	$N_D = 3$		Verbind. zw. Domänen	$N_C = 4$
Nachrichten-	$\delta_S = 1.0\,\text{ms}$	$\delta_{SI} = 10.0\,\text{ms}$		
bearbeitungszeiten	$\delta = 0.5\,\text{ms}$	$\delta_u = 5.0\,\text{ms}$	$\delta_H = 1.0\,\text{ms}$	
Verzögerung IP	$\Delta_A = 20.0\,\text{ms}$	$\Delta_I = 10.0\,\text{ms}$	$\Delta_T = 5.0\,\text{ms}$	$\Delta_L = 1.0\,\text{ms}$
Paketverlustwkt.	$p_{L,A} = 1.0\,\%$	$p_{L,I} = 0.1\,\%$	$p_{L,T} = 0.5\,\%$	$p_{L,L} = 0.0\,\%$
Zwischenergebnisse (vgl. Kapitel 6)				
Verzögerung auf	$V_{L,A} = 25.2\,\text{ms}$	$V_{H,I} = 10.1\,\text{ms}$	$V_{H,T} = 5.2\,\text{ms}$	$V_{H,L} = 1.0\,\text{ms}$
Transportschicht	$V_{L,AI} = 35.7\,\text{ms}$		$V_{H,ITI} = 25.6\,\text{ms}$	

Ergebnisse	**Szenario 1** (ungeschützt)	**Szenario 2** (Pf.-entk. Sig.)	**Szenario 3** (Pf.-gek. Sig.)	**Szenario 4** (SBC)
Ohne SIP Preconditions				
Rufverzug	$P_1 = 279\,\text{ms}$	$P_2 = 358\,\text{ms}$	$P_3 = 642\,\text{ms}$	$P_4 = 373\,\text{ms}$
Meldeverzug	$A_1 = 126\,\text{ms}$	$A_2 = 204\,\text{ms}$	$A_3 = 488\,\text{ms}$	$A_4 = 160\,\text{ms}$
Zusätzl. Nachr.	$N_{M,1} = 0$	$N_{M,2} = 72$	$N_{M,3} = 52$	$N_{M,4} = 30$
Anz. Nachr. insg.	19	91	71	49
Anz. Hashes	$N_{H,1} = 0$	$N_{H,2} = 0$	$N_{H,3} = 18$	$N_{H,4} = 0$
Anz. Pinholes	$N_{U,1} = 0$	$N_{U,2} = 36$	$N_{U,3} = 12$	$N_{U,4} = 12$
Mit SIP Preconditions				
Rufverzug	–	$P_{2P} = 688\,\text{ms}$	$P_{3P} = 905\,\text{ms}$	$P_{4P} = 666\,\text{ms}$
Meldeverzug	–	$A_{2P} = 126\,\text{ms}$	$A_{3P} = 126\,\text{ms}$	$A_{4P} = 130\,\text{ms}$
Zusätzl. Nachr.	–	$N_{M,2P} = 100$	$N_{M,3P} = 76$	$N_{M,4P} = 100$
Anz. Nachr. insg.	–	119	95	119
Anz. Hashes	–	$N_{H,2P} = 0$	$N_{H,3P} = 18$	$N_{H,4P} = 0$
Anz. Pinholes	–	$N_{U,2P} = 36$	$N_{U,3P} = 12$	$N_{U,4P} = 12$

maßnahmen), der jedoch keine zeitaufwändigen Abfragen externer Verzeichnisdienste erlaubt. Werden solche Abfragen durch einen noch höheren Wert für δ_{SI} berücksichtigt, verschlechtert sich die auf Session Border Controllern basierend Lösung relativ zu den anderen, da in diesem Szenario die Nachricht von mehr SIP-Protokollinstanzen bearbeitet werden muss. Geht man von höheren Verzögerungen beim Eintragen der Policy Rules in die Medienkomponenten aus, verschlechtern sich, wie erwartet, die Szenarien 3 und 4 relativ zu Szenario 2.

7.3 Zusammenfassung und Fazit

Ausgehend von einem Szenario mit mehreren zusammengeschalteten IP-Telefonie-Plattformen wurden in diesem Kapitel die beiden grundsätzlichen Signalisierverfahren zur Steuerung der Medienkomponenten verteilter Firewalls, die Pfad-entkoppelte und die Pfad-gekoppelte Signalisierung, miteinander verglichen.

Ein wesentlicher Vorteil der Pfad-gekoppelten Signalisierung ist, dass bei diesem Verfahren keine Informationen über Netztopologie und Verkehrslenkung an zentraler Stelle vorgehalten werden müssen. Da die Signalisiernachrichten auf Basis der selben Routing-Tabellen weitergeleitet werden, die auch für die Lenkung der Medienströme verwendet werden, können somit alle Medienkomponenten auf dem Pfad eines neuen Medienstroms automatisch gefunden und entsprechend konfiguriert werden.

Dieser gewichtige Vorteil wird jedoch dadurch erkauft, dass es sich – anders als bei der Pfad-entkoppelten Signalisierung – nicht um eine rein domäneninterne Lösung handelt, sondern Signalisiernachrichten mit Protokollinstanzen in nicht vertrauenswürdigen Nachbardomänen ausgetauscht werden müssen. Neben dem dadurch notwendigen Abstimmungsbedarf zwischen den Netzbetreibern kann dies u. U. Einfallstore für Angriffe öffnen, z. B. für Denial-of-Service-Attacken oder Angriffe, die Implementierungsfehler zur Kompromittierung der Protokollinstanzen nutzen. Falls Pfad-gekoppelte Signalisierung Ende-zu-Ende zwischen den Endgeräten der Teilnehmer eingesetzt werden soll, müssen dort entsprechende Protokollinstanzen vorhanden sein, was die Teilnehmer in der Wahl ihrer Endgeräte einschränken kann.

Falls Gesprächsentgelte auf Basis der Verbindungsdauer erhoben werden sollen, muss der Netzbetreiber sicherstellen, dass nach dem mit SIP signalisierten Ende einer Sitzung tatsächlich keine Medienströme mehr fließen können. Falls Pfad-gekoppelte Signalisierung Ende-zu-Ende eingesetzt wird, kann dies nur mit sehr kurz gewählten Gültigkeitsdauern der Regeln und einer entsprechend hohen Signalisierlast garantiert werden.

Der Aufbau einer Multimedia-Sitzung erfordert ein Zusammenspiel der Protokolle zur Verbindungs- und Firewall-Signalisierung. Je nach verwendetem Signalisierverfahren kann das Eintragen von Policy Rules in verschiedene Medienkomponenten teilweise nebenläufig erfolgen. Für ein Szenario mit mehreren Transitdomänen zwischen den Teilnehmern wurden die Signalisiervorgänge analysiert und der Einfluss auf Ruf- und Meldeverzug quantifiziert. Es konnte gezeigt werden, dass diese für die Teilnehmer unangenehmen Verzögerungen beim Einsatz Pfad-gekoppelter Signalisierung gegenüber dem ungeschützten Vergleichsszenario stärker anwachsen als bei Pfad-entkoppelter Signalisierung, unabhängig davon, welche konkreten Zahlenwerte für Bearbeitung und Transport der einzelnen Signalisiernachrichten angenommen werden.

8 Zusammenfassung und Ausblick

Firewalls am Übergang zwischen Bereichen mit unterschiedlichen Sicherheitsniveaus und -anforderungen spielen eine wichtige Rolle bei der Absicherung von Kommunikationsnetzen gegen Angriffe. Bei der Verwendung von Multimedia-Anwendungen wie z. B. IP-Telefonie ergeben sich für die Firewall zwei Hauptaufgaben: Zugriffskontrollen auf die Signalisierung und solche auf die Medienströme. Zwischen den beiden Modulen, die die jeweiligen Funktionen übernehmen, wird ein Signalisierprotokoll zur Koordination benötigt. Hierfür existieren zwei verschiedene Grund-Architekturen, die Pfad-entkoppelte und die Pfad-gekoppelte Firewall-Signalisierung, für die derzeit in Arbeitsgruppen der Internet Engineering Task Force jeweils konkrete Signalisierprotokolle spezifiziert werden (MIDCOM bzw. NSIS). Ziel dieser Arbeit ist ein umfassenderer Vergleich der beiden Architekturansätze. Dabei wird von einem Szenario ausgegangen, in dem die IP-Telefonie-Plattformen mehrerer Betreiber zusammengeschaltet wurden.

Nach einer Übersicht über die Grundlagen von VoIP, SIP, der Netzsicherheit und Firewalls in Kapitel 2 werden in Kapitel 3 Bedrohungsszenarien für die IP-Telefonie untersucht und aufgezeigt, wie Firewalls die Sicherheit der IP-Telefonie erhöhen können, aber auch welche Probleme dabei auftreten können. Ferner wird aufgezeigt, dass auf dem Internet Protocol basierende Telefonie nicht notwendigerweise im Internet stattfinden muss; am Beispiel der 3GPP IMS-Architektur werden Ansätze für separate Infrastrukturen illustriert. In Kapitel 4 werden verschiedene Architekturvarianten für VoIP-fähige Firewalls klassifiziert und die beiden Grundverfahren zur Steuerung verteilter Firewalls vorgestellt, inklusive der IETF MIDCOM- und NSIS-Protokolle, die diesen Prinzipien folgen.

Kapitel 5 beschreibt den Entwurf und die prototypische Implementierung von Protokollinstanzen des SIMCO-Protokolls. Der Aufbau einer Testumgebung demonstriert die Machbarkeit einer Integration von SIP-basierter IP-Telefonie und Pfad-entkoppelter Firewall-Steuerung. Dabei stellte sich jedoch heraus, dass das Vorhandensein von MIDCOM bzw. SIMCO für SIP nicht so transparent ist, wie in der MIDCOM-Architekturspezifikation suggeriert wird. Dies betrifft insbesondere die Frage, wie evtl. auftretende Fehler im Bereich eines der beiden Signalisierprotokolle zu einer sinnvollen Reaktion des jeweils anderen Protokolls führen können. Es wurde untersucht und aufgezeigt, wie optionale SIP-Erweiterungen, die teilweise für andere Zwecke spezifiziert wurden, zur Behandlung dieser Fehlerfälle eingesetzt werden können. Ergänzend werden einige Messergebnisse zur Leistungsfähigkeit der Paketfilterung mit dem Betriebssystem Linux präsentiert.

In IP-Telefonie-Plattformen fallen neben dem Medientransport und der SIP-Signalisierung zur Verbindungssteuerung noch diverse andere Signalisieraufgaben an, die zum Ruf- bzw. Meldeverzug beitragen, welche von den Teilnehmern als störend empfunden werden. Um diese Bei-

träge zu verringern, muss neben der Verarbeitung der Signalisiernachrichten in den Netzknoten auch ihr Transport über das IP-Netz optimiert werden. Ein Freiheitsgrad bei der Auswahl und ggf. Parametrisierung des Transportschichtprotokolls ist, ob keine, teilweise oder vollständige Reihenfolgesicherung für die Signalisiernachrichten benötigt wird, da diese so genanntes Head-Of-Line Blocking verursachen kann. Dieser verzögernde Effekt tritt auf, wenn infolge eines Paketverlustes Nachrichten erneut übertragen werden und darauffolgende Nachrichten zur Reihenfolgesicherung empfängerseitig gepuffert werden müssen.

Messungen an dem im Zuge dieser Arbeit erstellten SIMCO-Prototypen zeigen, dass der übliche, TCP-basierte Transport mit vollständiger Reihenfolgesicherung selbst bei moderaten Paketverlustwahrscheinlichkeiten signifikant mehr Zeit in Anspruch nimmt als die Paketumlaufzeit in der IP-Schicht. Das Stream Control Transmission Protocol (SCTP) erlaubt einen Transport mit teilweiser Reihenfolgesicherung, der für SIMCO besonders gut geeignet ist. Die notwendigen Anpassungen an SIMCO wurden in Kapitel 6 untersucht und spezifiziert; eine prototypische „SIMCO over SCTP“-Implementierung demonstriert die Realisierbarkeit.

Verzögerungen durch Mechanismen zur Reihenfolgesicherung sind ein allgemein bekannter Effekt, z. B. im Umfeld der Satelliten-Kommunikation; die vorhandenen Modelle bilden jedoch die speziellen Mechanismen der IP-basierten Transportschichtprotokolle nicht ab. Im Zuge dieser Arbeit wurde ein analytisches Modell für Head-Of-Line Blocking entwickelt, welches die spezifischen Mechanismen von TCP und SCTP – z. B. Erkennung von Paketverlusten durch eine Kombination von Zeitüberwachung, positiven und negativen Quittierungen – sowie eine beliebige Anzahl von SCTP Streams abdeckt. Mit Hilfe eines WAN-Emulators und eines Lastgenerators durchgeführte Messungen am „SIMCO over SCTP“-Prototypen zeigen eine gute Übereinstimmung mit den vom Modell vorhergesagten Werten, sowie eine gegenüber TCP-basiertem Transport deutlich reduzierte mittlere Transaktions-Antwortzeit. Da nur wenige Aspekte der Untersuchung spezifisch für SIMCO sind, können die Resultate auch auf andere Signalisierprotokolle, z. B. SIP oder NSIS, übertragen werden.

Basierend auf diesen Untersuchungen einzelner Mechanismen werden in Kapitel 7 das Pfad-entkoppelte und das Pfad-gekoppelte Signalisierverfahren in einem fiktiven Szenario mit mehreren zusammengeschalteten IP-Telefonie-Plattformen verglichen. Es wird davon ausgegangen, dass eine SIP-basierte Control Plane vorhanden ist, deren SIP-Server neue Multimedia-Sitzungen autorisieren und die dazugehörigen Medienströme kennen, welche von den Medienkomponenten der Firewalls erlaubt werden müssen.

Das Zusammenspiel verschiedener Signalisierprotokolle beim Aufbau einer neuen Multimedia-Sitzung über mehrere Transitdomänen hinweg wurde analysiert und der Einfluss der Firewall-Steuerung auf Ruf- und Meldeverzug quantifiziert. Dabei wurde auch der Nachrichtentransport berücksichtigt. Bei vergleichbaren Latenzen auf der IP-Schicht erfahren Signalisiernachrichten auf der Teilnehmerschnittstelle i. d. R. eine höhere Verzögerung durch das Transportschichtprotokoll als im Kernnetz, da die Paketverlustwahrscheinlichkeiten hier oft höher sind (z. B. in drahtlosen Zugangsnetzen) und aufgrund der niedrigeren Nachrichtenrate der Fast Retransmit-Mechanismus nicht so effizient arbeiten kann. Es konnte gezeigt werden, dass die für die Teilnehmer unangenehmen Verzögerungen beim Einsatz Pfad-gekoppelter Signalisierung gegenüber dem ungeschützten Vergleichsszenario stärker anwachsen als bei Pfad-entkoppelter Signalisierung, unabhängig davon, welche konkreten Zahlenwerte für die Verzögerungen

und Paketverlustwahrscheinlichkeiten auf den Pfaden sowie für die Bearbeitung der Signalisiernachrichten angenommen werden.

Der Vergleich funktionaler und sicherheitsrelevanter Aspekte erfolgte auf Basis eines Katalogs mit zwölf Anforderungen. Ein wesentlicher Nachteil der Pfad-entkoppelten Signalisierung ist, dass den zentralen Instanzen der Control Plane Informationen über Netztopologie und Verkehrslenkung bekannt sein müssen. Diese werden benötigt, um Policy Rules zum Erlauben eines Medienstroms in jene Medienkomponenten eintragen zu können, die tatsächlich auf seinem Pfad durch das Netz liegen. Bei der Pfad-gekoppelten Signalisierung wird hingegen kein zentrales Topologie-Wissen benötigt. Dieser Umstand wird von Befürwortern der Pfad-gekoppelten Firewall-Signalisierung häufig als *das* KO-Kriterium gegen die Pfad-entkoppelte Signalisierung angeführt. Bei einer gemeinsamen Betrachtung aller in dieser Arbeit untersuchten Aspekte zeigt sich, dass dieses Argument durchaus richtig und gewichtig ist; jedoch wird dieser Vorteil der Pfad-gekoppelten Signalisierung mit einer ganzen Reihe von Nachteilen erkauft. Diese ergeben sich überwiegend aus dem Umstand, dass es sich – anders als bei der Pfad-entkoppelten Signalisierung – nicht um eine rein domäneninterne Lösung handelt, sondern Signalisiernachrichten mit Protokollinstanzen in nicht vertrauenswürdigen Nachbardomänen ausgetauscht werden müssen. Neben dem dadurch notwendigen Abstimmungsbedarf zwischen den Netzbetreibern kann dies u. U. Einfallstore für Angriffe öffnen, z. B. für Denial-of-Service-Attacken oder Angriffe, die Implementierungsfehler zur Kompromittierung der Protokollinstanzen nutzen. Falls Pfad-gekoppelte Signalisierung Ende-zu-Ende zwischen den Endgeräten der Teilnehmer eingesetzt werden soll, müssen dort entsprechende Protokollinstanzen vorhanden sein, was die Teilnehmer in der Wahl ihrer Endgeräte einschränken kann.

Somit dürfte die Komplexität und die Dynamik der Netztopologie tatsächlich das *entscheidende* Argument bei der Auswahl des Signalisierverfahrens werden. Ist diese hoch, z. B. in „offenen“, Internet-ähnlichen Szenarien mit dynamischem Routing und Netzelementen unter der administrativen Kontrolle der Teilnehmer („DSL-Router“) sowie kaskadierter Network Address and Port Translation (NAPT) im Zugangsnetz, ist das Sammeln von Topologie-Informationen und somit der Einsatz von Pfad-entkoppelter Signalisierung unverhältnismäßig aufwändig oder unmöglich; die Pfad-gekoppelte Signalisierung ist hier im Vorteil.

IP-Telefonie-Plattformen sind hingegen Netze, in denen zwar die Protokolle der TCP/IP-Protokollfamilie zum Einsatz kommen, die von ihrem Netzdesign jedoch zentralisierter und statischer als das Internet sind, z. B. indem erzwungen wird, dass alle Multimedia-Sitzungen über die zentralen SIP-Server der Control Plane signalisiert werden müssen, oder indem Zwangspunkte im Zugangsnetz geschaffen werden. Verschiedene Ausprägungen dieses Grundgedankens, die derzeit von verschiedenen Organisationen unter verschiedenen Bezeichnungen spezifiziert werden, unterscheiden sich darin, wie restriktiv diese Sicherheitsrichtlinien sein sollen. Auch bei solchen kommerziell betriebenen Netzen ist davon auszugehen, dass zwischen zwei benachbarten Domänen mehrere Netzübergänge vorhanden sind. Die Liste aller für einen bestimmten Medienstrom mit bekannter Quelle und bekanntem Ziel in Frage kommenden Übergänge kann dennoch recht kurz sein; durch entsprechende Maßnahmen bei der Netzplanung (z. B. Festlegen von Kantengewichten im Routingprotokoll) kann sie evtl. weiter reduziert werden. In diesem Fall können beim Aufbau einer neuen Sitzung die zugehörigen Regeln einfach in alle in Frage kommenden Medienkomponenten eingetragen werden; da dies zeitgleich geschehen kann, entsteht so keine wesentliche zusätzliche Verzögerung des Verbindungsaufbaus. Ist die Liste

potenzieller Netzübergänge hingegen sehr lang, erscheint dieser Ansatz ungeeignet; allerdings stellt sich dann u. U. auch die Frage, ob ein derart unscharfe Grenze eine sinnvolle Grenze zwischen Sicherheitsdomänen ist. Somit dürfte in vielen Szenarien das Hauptargument gegen den Einsatz der Pfad-entkoppelten Signalisierung entfallen, die als domäneninterne Lösung keine direkten Schnittstellen zu nicht vertrauenswürdigen Instanzen benötigt und so besser mit den Designprinzipien der „geschlossenen“ IP-Telefonie-Plattformen vereinbar ist.

Die Frage, welche Architektur zur Steuerung verteilter Firewalls besser geeignet ist, ist somit eng an die viel grundsätzlichere Frage geknüpft, wie zukünftige Kommunikationsnetze aussehen werden. Dass dafür IP-basierte Technik zum Einsatz kommen soll, steht derzeit außer Frage; unklar ist hingegen noch, wie „offen“ die Strukturen sein werden. Auf der einen Seite steht hier das Internet, in dem die Erreichbarkeit von Endsystemen durch die Netzbetreiber nicht eingeschränkt wird und in dem Pakettransport und Dienste praktisch vollständig entkoppelt sind. Dies ermöglicht einerseits die schnelle und unkomplizierte Einführung neuer Dienste auch durch Drittanbieter und Privatpersonen, andererseits müssen so auch die meisten Schutzmaßnahmen in oder unmittelbar vor den Endgeräten implementiert werden. Da viele Nutzer mit der Konfiguration dieser Mechanismen überfordert sind, ist der Missbrauch des Internets, z. B. zum milliardenfachen Versand von unerwünschter Werbung oder zum Betrug bei Online-Banking, ein sehr weit verbreitetes Problem. Auf der anderen Seite stehen die geschlossenen Plattformen, deren Strukturierungsprinzipien von den klassischen Telefonnetzen geprägt sind. Die Netzbetreiber haben hier mehr Kontrolle über die Diensterbringung (und -abrechnung), aber auch über netzseitige Sicherheitsmechanismen. Bei der Frage, wie restriktiv die Sicherheitsrichtlinien in solchen Netzen tatsächlich sein sollen, ergibt sich ein breites Spektrum zwischen beiden Positionen – letztendlich wird es eine Entscheidung der Nutzer sein, ob das offene, flexible, teilweise aber auch unsichere Internet oder die von Netzbetreibern mit Hilfe von Firewalls abgeschirmten „Walled Gardens“ eine höhere Akzeptanz finden werden.

Literaturverzeichnis

[1] S. Kiesel and M. Scharf. Modeling and performance evaluation of transport protocols for firewall control. *Computer Networks*, 51(11):3232–3251, August 2007. doi:10.1016/j.comnet.2006.11.031.

[2] M. Scharf and S. Kiesel. Head-of-line blocking in TCP and SCTP: Analysis and measurements. In *Proc. IEEE Globecom*, San Francisco, CA, USA, Nov. 2006.

[3] S. Kiesel, M. Scharf, S. Beutel, and T. Ruschival. Performance measurement results of SIMCO over TCP and SCTP. Interner Bericht 53, Universität Stuttgart, IKR, 2006.

[4] S. Kiesel and M. Scharf. Modeling and performance evaluation of SCTP as transport protocol for firewall control. In *Proc. IFIP-TC6 Networking Conference, Springer LNCS 3976*, pages 451–462, Coimbra, Portugal, May 2006.

[5] S. Kiesel. SIMCO over SCTP. IETF draft - work in progress, IETF, draft-kiesel-midcom-simco-sctp-02, September 2006.

[6] S. Kiesel. SIMCO over SCTP. IETF draft - work in progress, IETF, draft-kiesel-midcom-simco-sctp-01, April 2006.

[7] S. Kiesel. SIMCO over SCTP. IETF draft - work in progress, IETF, draft-kiesel-midcom-simco-sctp-00, October 2005.

[8] S. Kiesel and M. Scharf. Evaluation of SCTP as transport layer protocol for firewall control. In *Proc. Workshop 'Neue Herausforderungen in der Netzsicherheit'*. Universität Duisburg-Essen, Essen, 2005.

[9] M. Stiemerling, C. Cadar, S. Kiesel, and A. Müller. SIMCO protocol implementation interoperability report. IETF draft - work in progress, IETF, draft-stiemerling-midcom-simco-interop-00, August 2004.

[10] A. Müller and S. Kiesel. Issues with the interworking of application layer protocols and the MIDCOM architecture. In *Proc. Eunice Summer School*, pages 188–195, Tampere, Finland, 2004.

[11] S. Kiesel. On the use of cryptographic cookies for transport layer connection establishment. In *Proc. EUNICE Summer School*, pages 177–184, Trondheim, Norway, 2002.

[12] T. Steinert. *Optimierte Steuerung in VoIP-Netzen für eine effiziente Ressourcennutzung*. Dissertation, Universität Stuttgart, IKR, 2005.

[13] Frauenhofer IIS. Die MP3-Geschichte [online]. URI: http://www.iis.fraunhofer.de [Retrieved on 12. Nov. 2007].

[14] IKR. *Communication Networks I*. Manuskript zur Vorlesung. Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart, 2006.

[15] J.-M. Valin and C. Montgomery. Improved Noise Weighting in CELP Coding of Speech - Applying the Vorbis Psychoacoustic Model To Speex. In *Proc. 120th Audio Engineering Society Convention*, May 2006.

[16] B. Goode. Voice over Internet protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495–1517, 2002.

[17] L. Cai, Y. Xiao, X. Shen, L. Cai, and J. Mark. VoIP over WLAN: voice capacity, admission control, QoS, and MAC. *International Journal of Communication Systems*, 19(4):491–508, 2006. doi:10.1002/dac.801.

[18] G. Bandow, H. Gottschalk, D. Gehrmann, W. Hlavac, H. Koch, W. Müller, and D. Schwetje. *Zeichengabesysteme - Eine neue Generation für ISDN und intelligente Netze*. L.T.U.-Vertriebsgesellschaft mbH, 1992, 1995.

[19] D. Flory. The great blue box phone frauds. *Spectrum, IEEE*, 27(11):117–119, 1990.

[20] H. Schulzrinne. Personal Mobility for Multimedia Services in the Internet. In *IDMS '96: Proc. European Workshop on Interactive Distributed Multimedia Systems and Services*, pages 143–161, London, 1996.

[21] H. Schulzrinne. Simple Conference Invitation Protocol. IETF draft - work in progress, IETF, draft-ietf-mmusic-scip-00, February 1996.

[22] M. Handley and E. Schooler. Session Invitation Protocol. IETF draft - work in progress, IETF, draft-ietf-mmusic-sip-00, February 1996.

[23] M. Handley, H. Schulzrinne, and E. Schooler. Session Initiation Protocol. IETF draft - work in progress, IETF, draft-ietf-mmusic-sip-01, February 1996.

[24] J. Rosenberg. A Hitchhikers Guide to the Session Initiation Protocol (SIP). IETF draft - work in progress, IETF, draft-ietf-sip-hitchhikers-guide-01, October 2006.

[25] Third Generation Partnership Project (3GPP). 3GPP Homepage [online]. URI: http://www.3gpp.org [Retrieved on 12. Nov. 2007].

[26] European Telecommunications Standards Institute. Telecoms and Internet converged Services and Protocols for Advanced Networks (TISPAN) Homepage [online]. URI: http://www.etsi.org/tispan/ [Retrieved on 12. Nov. 2007].

[27] F. Andreasen. SDP Capability Negotiation: Requirements and Review of Existing Work. IETF draft - work in progress, IETF, draft-ietf-mmusic-sdp-capability-negotiation-reqts-00, December 2006.

[28] D. Kutscher, J. Ott, and C. Bormann. Session Description and Capability Negotiation. IETF draft - work in progress, IETF, draft-ietf-mmusic-sdpng-08, February 2005.

[29] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible Markup Language (XML) 1.0 (4th. Ed.). Rec xml-20060816, W3C, Aug. 2006.

[30] S. Wanke, M. Scharf, S. Kiesel, and S. Wahl. Measurement of the SIP Parsing Performance in the SIP Express Router. In *Proc. Eunice 2007, Springer LNCS 4606*, pages 103–110, July 2007.

[31] F. Audet. Guidelines for the use of the SIPS URI Scheme in the Session Initiation Protocol (SIP). IETF draft - work in progress, IETF, draft-ietf-sip-sips-02, March 2007.

[32] M. Haberler and R. Stastny. Combined User and Infrastructure ENUM in the e164.arpa tree. IETF draft - work in progress, IETF, draft-ietf-enum-combined-04, January 2007.

[33] R. Pandya. Emerging mobile and personal communication systems. *Communications Magazine, IEEE*, 33(6):44–52, 1995.

[34] J. Calme and R. Ejzak. A common SIP profile for next-generation networks. *Bell Labs Technical Journal*, 11(1):107–122, 2006. doi:10.1002/bltj.20147.

[35] D. R. Evans. *Digital Telephony Over Cable: The PacketCable™Network*. Addison-Wesley, 2001. ISBN: 0-201-72827-3.

[36] AstriCon – The Open Source Telephony Conference & Exhibition [online]. URI: http://www.astricon.net [Retrieved on 12. Nov. 2007].

[37] B. Capouch. IAX: Inter-Asterisk eXchange Version 2. IETF draft - work in progress, IETF, draft-guy-iax-02, October 2006.

[38] Ekiga (formely known as GnomeMeeting) [online]. URI: http://www.ekiga.org [Retrieved on 12. Nov. 2007].

[39] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. ACM SIGCOMM*, pages 149–160, San Deigo, CA, USA, Aug. 2001.

[40] Skype, Ltd. Skype Homepage [online]. 2007. URI: http://www.skype.com [Retrieved on 6. Oct. 2007].

[41] P. Biondi and F. Desclaux. Silver Needle in the Skype. In *Proc. BlackHat Europe Conference*, March 2006.

[42] S. Baset and H. Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[43] R. Sailer. *Sicherheitsarchitektur für mehrseitig sichere Kommunikationsdienste am Beispiel ISDN*. Dissertation, Universität Stuttgart, IND, 1999.

[44] G. Müller und A. Pfitzmann (Hrsg.). *Mehrseitige Sicherheit in der Kommunikationstechnik; Verfahren, Komponenten, Integration*. Addison-Wesley, Bonn, 1997. ISBN 3-8273-1116-0.

[45] J. Rushby and B. Randell. A Distributed Secure System. *IEEE Computer*, 16(7):55–67, July 1983. URI: http://www.csl.sri.com/users/rushby/abstracts/computer83.

[46] R. Sailer und P. J. Kühn. *Integration von Authentikationsverfahren in Kommunikationsnetze unter Verwendung separat sicherbarer Bereiche.* In Müller und Pfitzmann [44], 1997. S. 133 - 165.

[47] R. Sailer und P. J. Kühn. Ein Domain-Konzept zur systematischen und wirtschaftlichen Integration von Sicherheit in Kommunikationsnetzen. *it+ti Informationstechnik und Technische Informatik*, 4:30–33, 1996.

[48] M. Kabatnik. Sichtenbasiertes Kommunikationsmodell zur Beschreibung von Schutzzielen der Mehrseitigen Sicherheit. Interner Bericht 42, Universität Stuttgart, IKR, 2002.

[49] N. Pohlmann. *Firewall-Systeme*. mitp-Verlag, 2003. 5. Auflage, ISBN: 3-8266-0988-3.

[50] U. Roedig. *Firewall-Architekturen für Multimedia-Applikationen*. Dissertation, Universität Darmstadt, KOM, 2002.

[51] CERT/CC. Denial-of-Service Vulnerabilities in TCP/IP Stacks. Advisory CA-2000-21, CERT/CC, November 2000. URI: http://www.cert.org/advisories/CA-2000-21.html.

[52] J. Rosenberg. Rejecting Anonymous Requests in the Session Initiation Protocol (SIP). IETF draft - work in progress, IETF, draft-ietf-sip-acr-code-04, March 2007.

[53] R. Schlegel, S. Niccolini, S. Tartarelli, and M. Brunner. SPam over Internet Telephony (SPIT) Prevention Framework. In *Proc. IEEE GLOBECOM 2006*, San Francisco, CA, USA, November 2006.

[54] S. M. Bellovin and W. R. Cheswick. Network firewalls. *Communications Magazine, IEEE*, 32(9):50–57, 1994.

[55] E. Al-Shaer and H. Hamed. Firewall Policy Advisor for Anomaly Detection and Rule Editing. In *Proc. IEEE/IFIP IM'2003*, pages 17–30, May 2003.

[56] IKR. *Communication Networks II*. Manuskript zur Vorlesung. Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart, 2006.

[57] Internet Assigned Numbers Authority (IANA). TCP and UDP Port Numbers [online]. URI: http://www.iana.org/assignments/port-numbers [Retrieved on 12. Nov. 2007].

[58] A. Adelsbach, A. Alkassar, K. Garbe, M. Luzaic, M. Manulis, E. Scherer, J. Schwenk, and E. Siemens. VoIPSEC - Studie zur Sicherheit von Voice over Internet Protocol. Studie, BSI, Okt. 2005.

[59] R. Kuhn, T. Walsh, and S. Fries. Security Considerations for Voice Over IP Systems. Special Publication 800-58, NIST, January 2005.

[60] C. Wieser. Über die Verwundbarkeit von IP-Telefonie-Systemen. In *Tagungsband zum 13. DFN-CERT Workshop „Sicherheit in vernetzten Systemen“*, Hamburg, 2006. URI: http://www.ee.oulu.fi/research/ouspg/protos/testing/c07/sip/index.html.

[61] BSI. Die Lage der IT-Sicherheit in Deutschland 2007. Lagebericht, Bundesamt für Sicherheit in der Informationstechnik, http://www.bsi.de/literat/lagebericht/index.htm, April 2007.

[62] SPAM-O-METER. Online Spam Statistics [online]. URI: http://www.spam-o-meter.com/stats/ [Retrieved on 11. Mai. 2007].

[63] A. Schwartz. *SpamAssassin – The Open Source Solution to SPAM*. O'Reilly, 2004. ISBN: 0-596-00707-8.

[64] T. A. Radermacher. Spam Prevention in Voice over IP Networks. Diploma thesis, FH Salzburg, June 2004.

[65] C. Jennings and J. Rosenberg. The Session Initiation Protocol (SIP) and Spam. IETF draft - work in progress, IETF, draft-ietf-sipping-spam-04, February 2007.

[66] Y. Rebahi and D. Sisalem. SIP Service Providers and the Spam Problem. In *Proc. Voice over IP Secuity Workshop*, Washington, USA, June 2005.

[67] S. Niccolini. SIP Extensions for SPIT identification. IETF draft - work in progress, IETF, draft-niccolini-sipping-feedback-spit-03, February 2007.

[68] E. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 1998/2001. URI: http://citeseer.ist.psu.edu/friedman98social.html.

[69] J.-E. Horn. Konzept eines IP-basierten Telefonnetzes unter der Verwendung von ENUM. Diplomarbeit, Universität Stuttgart, IKR, Juni 2005.

[70] R. Sailer, H. Federrath, A. Jerichow, D. Kesdogan, and A. Pfitzmann. *Allokation von Sicherheitsfunktionen in Telekommunikationsnetzen*. In Müller und Pfitzmann [44], 1997. S. 325 - 357.

[71] IEEE 802.11 working group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Std 802.11a, IEEE, July 1999.

[72] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-To-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4):277–288, Nov. 1984. URI: http://citeseer.ist.psu.edu/saltzer84endtoend.html.

[73] D. Kaminsky. Black Ops of DNS. In *Proc. 21st Chaos Communication Congress (21C3)*, Berlin, Germany, December 2004. Chaos Computer Club e.V.

[74] A. D. Keromytis, V. Misra, and D. Rubenstein. SOS: an architecture for mitigating DDoS attacks. *Selected Areas in Communications, IEEE Journal on*, 22(1):176–188, 2004.

[75] S. Guha and P. Francis. Towards a Secure Internet Architecture Through Signaling. Technical Report cul.cis/TR2006-2037, Cornell University, Ithaca, NY, 2006.

[76] M. Scharf, M.C. Necker, C.M. Gauger, B. Gloss, C. Hauser, J. Jähnert, S. Kiesel, M. Köhn, P.J. Kühn, A. Reifert, and D. Sass. To Session or not to Session – Session Concepts in Currently Emerging Future Networks. In *Proc. 6th Würzburg Workshop on IP: „Visions of Future Generation Networks“*, Würzburg, 2006.

[77] O. Lendl. Background and Assumptions of the Speermint WG. IETF draft - work in progress, IETF, draft-lendl-speermint-background-00, April 2007.

[78] DeNIC eG. ENUM Statistik [online]. URI: http://www.denic.de/de/enum/statistik/index.html [Retrieved on 20. Jun. 2007].

[79] G. Camarillo and M.-A. García-Martín. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. Wiley, 2006. 2nd. Ed., ISBN: 0-470-01818-6.

[80] R. Penno. SPEERMINT Peering Architecture. IETF draft - work in progress, IETF, draft-ietf-speermint-architecture-01, September 2006.

[81] GSMA. Inter-Service Provider IP Backbone Guidelines. Official Document IR.34 v4.1, GSM Association, Jan. 2007.

[82] GSMA. IMS Roaming & Interworking Guidelines. Official Document IR.65 v3.6, GSM Association, Nov. 2006.

[83] S. Marcus. Interconnection in an NGN Environment. In *Proc. ITU Workshop on "What rules for IP-enabled Next Generation Networks?"*, Genova, March 2006.

[84] J. Hautakorpi. Requirements from SIP (Session Initiation Protocol) Session Border Control Deployments. IETF draft - work in progress, IETF, draft-ietf-sipping-sbc-funcs-03, April 2007.

[85] M. Wedel. SIP connection tracking helper for Linux Netfilter. Studienarbeit, Universität Stuttgart, IKR, 2003.

[86] C. Hentschel. SIP connection tracking helper (for Linux Netfilter). Software module included in Linux kernel version 2.6.18 and later, September 2006.

[87] C. Aoun. *Plan de signalisation Internet pour l'interfonctionnement entre NAT et Firewall*. Dissertation, École Nationale Supérieure des Télécommunications, Paris, 2005.

[88] FhG FOKUS, iptelorg GmbH. SIP Express Router [online]. URI: http://www.iptel.org/ser [Retrieved on 12. Nov. 2007].

[89] J. Kuthan. Internet telephony traversal across decomposed firewalls and NATs. In *Proceedings of the 2nd IP Telephony Workshop*, New York, April 2001.

[90] J. Kuthan, U. Abend, N. Ohlmeier, and J. Janak. Firewall Control Protocol (FCP) – protocol grammar. Protocol specification at berliOS CVS repository, June 2001. URI: http://cvs.berlios.de/cgi-bin/viewcvs.cgi/fcpd/fcp/fcp-spec-19-abnf.txt#rev1.1.1.1.

[91] A. Molitor. Deploying a Dynamic Voice over IP Firewall with IP Telephony Applications. Technical report, ARAVOX Technologies (aquired by Alcatel in 2003), March 2001.

[92] J. Kuthan. Firewall Control Protocol Requirements and Framework. In *47th IETF meeting, FOGLAMPS BOF*, Adelaide, Australia, March 2000.

[93] J. Kuthan and J. Rosenberg. Middlebox Communication – Framework and Requirements. In *49th IETF meeting, MIDCOM BOF*, San Diego, CA, USA, December 2000.

[94] Universal Plug and Play (UPnP) Forum. UPnP Internet Gateway Device (IGD) Standard V1.0. Technical specification, Universal Plug and Play (UPnP) Forum, http://www.upnp.org/, 2001.

[95] U. Roedig, M. Goertz, M. Karsten, and R. Steinmetz. RSVP as Firewall Signalling Protocol. In *Proc. 6th IEEE Symposium on Computers and Communications*, pages 57–62, Hammamet, Tunisia, July 2001.

[96] C. Macian, W. Payer, C. Hauser, K. Dolzer, L. Burgstahler, S. Junghans, and J. Jähnert. Beyond Technology: The Missing Pieces for QoS Success. In *Proc. ACM SIGCOMM 2003 Workshops*, Karlsruhe, 2003.

[97] R. Braden. A Two-Level Architecture for Internet Signaling. IETF draft - work in progress, IETF, draft-braden-2level-signal-arch-01, November 2002.

[98] J. Quittek. Definitions of Managed Objects for Middlebox Communication. IETF draft - work in progress, IETF, draft-ietf-midcom-mib-09, October 2006.

[99] M. Stiemerling. Middlebox Communications (MIDCOM) Protocol Semantics. IETF draft - work in progress, IETF, draft-ietf-midcom-rfc3989-bis-00, May 2007.

[100] J. Manner. NSLP for Quality-of-Service Signaling. IETF draft - work in progress, IETF, draft-ietf-nsis-qos-nslp-13, March 2007.

[101] M. Stiemerling. NAT/Firewall NSIS Signaling Layer Protocol (NSLP). IETF draft - work in progress, IETF, draft-ietf-nsis-nslp-natfw-14, March 2007.

[102] A. Fessi. Framework for Metering NSLP. IETF draft - work in progress, IETF, draft-fessi-nsis-m-nslp-framework-04, March 2007.

[103] H. Schulzrinne and R. Hancock. GIST: General Internet Signalling Transport. IETF draft - work in progress, IETF, draft-ietf-nsis-ntlp-13, April 2007.

[104] X. Fu and H. Tschofenig. Extensible IP Signaling – Architecture, Protocols and Practice. In *Tutorial at IFIP-TC6 Networking Conference*, Coimbra, Portugal, May 2006.

[105] R. Hancock. A Problem Statement for Partly-Decoupled Signalling in NSIS. IETF draft - work in progress, IETF, draft-hancock-nsis-pds-problem-03, March 2006.

[106] C. Werner. NAT/FW NSLP State Machine. IETF draft - work in progress, IETF, draft-werner-nsis-natfw-nslp-statemachine-04, March 2007.

[107] J. Manner. Authorization for NSIS Signaling Layer Protocols. IETF draft - work in progress, IETF, draft-manner-nsis-nslp-auth-03, March 2007.

[108] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. The MIT Press, Cambridge, Ma., 1999. ISBN: 0-262-03270-8.

[109] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990. doi:10.1145/77648.77649.

[110] BSI. IT-Grundschutz-Kataloge. Handbuch/Software, Bundesamt für Sicherheit in der Informationstechnik, 2006. URI: http://www.bsi.de/gshb/index.htm.

[111] Linus Torvalds. Linux Online [online]. URI: http://www.linux.org/info/linus.html [Retrieved on 12. Nov. 2007].

[112] The Free Software Foundation. GNU General Public License [online]. URI: http://www.gnu.org/copyleft/gpl.html [Retrieved on 12. Nov. 2007].

[113] The netfilter.org project. Netfilter Homepage [online]. URI: http://www.netfilter.org [Retrieved on 12. Nov. 2007].

[114] Gerald Combs. Wireshark Network Protocol Analyzer Homepage [online]. URI: http://www.wireshark.org [Retrieved on 12. Nov. 2007].

[115] E. Al-Shaer and H. Hamed. Modeling and management of firewall policies. *IEEE Trans. Network and Service Management*, 1(1), April 2004. URI: http://www.mnlab.cs.depaul.edu/projects/FPA/files/tnsm04.pdf.

[116] C. Blankenhorn. Untersuchung von SCTP als Transportschichtprotokoll für transaktionsbasierte Anwendungen am Beispiel eines Protokolls zur Firewallsteuerung. Studienarbeit, Universität Stuttgart, IKR, 2005.

[117] S. Meier. Design of a Software Interface for Operating System Independent Firewall Configuration. Studienarbeit, Universität Stuttgart, IKR, 2007.

[118] vovida.org Homepage [online]. URI: http://www.vovida.org [Retrieved on 12. Nov. 2007].

[119] A. Müller. Erweiterung eines SIP-Proxies um Sicherheitsfunktionen. Studienarbeit, Universität Stuttgart, IKR, 2004.

[120] IKR. *Teletraffic Theory and Engineering*. Manuskript zur Vorlesung. Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart, 2006.

[121] National Institute of Standards and Technology. NIST Net Homepage [online]. URI: http://www-x.antd.nist.gov/nistnet/ [Retrieved on 12. Nov. 2007].

[122] M. Stiemerling and J. Quittek. Middlebox configuration protocol design. In *Proc. IEEE IPOM 2002 Workshop*, pages 222 – 226, 2002.

[123] Ranch Networks, Inc. „Digium and Ranch Networks Team to Make Asterisk the Most Secure and Scalable VoIP Solution“ [online]. URI: http://www.ranchnetworks.com/news/pressrelease1_20_2006.htm [Retrieved on 22. May 2006].

[124] OpenWRT Homepage [online]. URI: http://openwrt.org [Retrieved on 12. Nov. 2007].

[125] Cisco Systems, Inc. Cisco 12000 Series Routers [online]. URI: http://www.cisco.com/go/12000 [Retrieved on 12. Nov. 2007].

[126] S. Kiesel, J. Kögel, S. Meier, and C. Blankenhorn. A userspace API for netfilter control. In *Proc. 5th Netfilter Workshop*, Karlsruhe, 2007.

[127] P. Amer, C. Chassot, T. Connolly, M. Diaz, and P. Conrad. Partial-order Transport Service for Multimedia and Other Applications. *IEEE/ACM Trans. Netw.*, 2(5):440–456, 1994.

[128] P. Natarajan, J. Iyengar, P. Amer, and R. Stewart. SCTP: an innovative transport layer protocol for the web. In *Proc. WWW '06: 15th intl. conf. on World Wide Web*, pages 615–624, Edinburgh, Scotland, 2006.

[129] Y. Xia and D. Tse. Analysis on Packet Resequencing for Reliable Network Protocols. *Performance Evaluation*, 61:299–328, 2005.

[130] Newport Networks Ltd. 1460 Performance Tests. White paper, Newport Networks Ltd., 2006. Accessed 24.09.2007. URI: http://www.newport-networks.com/cust-docs/90-Performance.pdf.

[131] P. Sarolahti and A. Kuznetsov. Congestion Control in Linux TCP. In *Proc. USENIX Annual Technical Conference*, pages 49–62, Monterey, CA, USA, June 2002.

[132] E. Baccelli and R. Rajan. Monitoring OSPF routing. In *Proc. IEEE/IFIP International Symposium on Integrated Network Management*, pages 825–838, 2001.

[133] A. Shaikh and A. Greenberg. OSPF Monitoring: Architecture, Design and Deployment Experience. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2004. URI: http://www.research.att.com/~ashaikh/publications.html.

[134] B. Tödtmann and E. Rathgeb. Anticipatory distributed packet filter configurations for carrier-grade IP networks. *Computer Networks*, 51(10):2565–2579, July 2007. doi:10.1016/j.comnet.2006.11.013.

[RFC 768] J. Postel. User Datagram Protocol. RFC 768, IETF, August 1980.

[RFC 792] J. Postel. Internet Control Message Protocol. RFC 792, IETF, September 1981.

[RFC 793] J. Postel. Transmission Control Protocol. RFC 793, IETF, September 1981.

[RFC 821] J. Postel. Simple Mail Transfer Protocol. RFC 821, IETF, August 1982.

[RFC 871] M. A. Padlipsky. Perspective on the ARPANET reference model. RFC 871, IETF, September 1982.

[RFC 1034] P. V. Mockapetris. Domain names – concepts and facilities. RFC 1034, IETF, November 1987.

[RFC 1035] P. V. Mockapetris. Domain names – implementation and specification. RFC 1035, IETF, November 1987.

[RFC 1244] J. P. Holbrook and J. K. Reynolds. Site Security Handbook. RFC 1244, IETF, July 1991.

[RFC 1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, IETF, February 1996.

[RFC 2045] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, IETF, November 1996.

[RFC 2131] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, IETF, March 1997.

[RFC 2205] R. Braden (Editor), L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205, IETF, September 1997.

[RFC 2279] F. Yergeau. UTF-8, a transformation format of ISO 10646. RFC 2279, IETF, January 1998.

[RFC 2326] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, IETF, April 1998.

[RFC 2328] J. Moy. OSPF Version 2. RFC 2328, IETF, April 1998.

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

[RFC 2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, IETF, June 1999.

[RFC 2617] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617, IETF, June 1999.

[RFC 2635] S. Hambridge and A. Lunde. DON'T SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*). RFC 2635, IETF, June 1999.

[RFC 2663] P. Srisuresh and M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663, IETF, August 1999.

[RFC 2719] L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, and C. Sharp. Framework Architecture for Signaling Transport. RFC 2719, IETF, October 1999.

[RFC 2748] D. Durham (Editor), J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol. RFC 2748, IETF, January 2000.

[RFC 2806] A. Vaha-Sipila. URLs for Telephone Calls. RFC 2806, IETF, April 2000.

[RFC 2871] J. Rosenberg and H. Schulzrinne. A Framework for Telephony Routing over IP. RFC 2871, IETF, June 2000.

[RFC 2960] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960, IETF, October 2000.

[RFC 2976] S. Donovan. The SIP INFO Method. RFC 2976, IETF, October 2000.

[RFC 3015] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers. Megaco Protocol Version 1.0. RFC 3015, IETF, November 2000.

[RFC 3022] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, IETF, January 2001.

[RFC 3084] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith. COPS Usage for Policy Provisioning (COPS-PR). RFC 3084, IETF, March 2001.

[RFC 3102] M. Borella, J. Lo, D. Grabelsky, and G. Montenegro. Realm Specific IP: Framework. RFC 3102, IETF, October 2001.

[RFC 3113] K. Rosenbrock, R. Sanmugam, S. Bradner, and J. Klensin. 3GPP-IETF Standardization Collaboration. RFC 3113, IETF, June 2001.

[RFC 3170] B. Quinn and K. Almeroth. IP Multicast Applications: Challenges and Solutions. RFC 3170, IETF, September 2001.

[RFC 3198] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser. Terminology for Policy-Based Management. RFC 3198, IETF, November 2001.

[RFC 3234] B. Carpenter and S. Brim. Middleboxes: Taxonomy and Issues. RFC 3234, IETF, February 2002.

[RFC 3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[RFC 3262] J. Rosenberg and H. Schulzrinne. Reliability of Provisional Responses in Session Initiation Protocol (SIP). RFC 3262, IETF, June 2002.

[RFC 3263] J. Rosenberg and H. Schulzrinne. Session Initiation Protocol (SIP): Locating SIP Servers. RFC 3263, IETF, June 2002.

[RFC 3265] A. B. Roach. Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265, IETF, June 2002.

[RFC 3303] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan. Middlebox communication architecture and framework. RFC 3303, IETF, August 2002.

[RFC 3304] R. P. Swale, P. A. Mart, P. Sijben, S. Brim, and M. Shore. Middlebox Communications (midcom) Protocol Requirements. RFC 3304, IETF, August 2002.

[RFC 3311] J. Rosenberg. The Session Initiation Protocol (SIP) UPDATE Method. RFC 3311, IETF, October 2002.

[RFC 3312] G. Camarillo (Editor), W. Marshall (Editor), and J. Rosenberg. Integration of Resource Management and Session Initiation Protocol (SIP). RFC 3312, IETF, October 2002.

[RFC 3323] J. Peterson. A Privacy Mechanism for the Session Initiation Protocol (SIP). RFC 3323, IETF, November 2002.

[RFC 3325] C. Jennings, J. Peterson, and M. Watson. Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks. RFC 3325, IETF, November 2002.

[RFC 3326] H. Schulzrinne, D. Oran, and G. Camarillo. The Reason Header Field for the Session Initiation Protocol (SIP). RFC 3326, IETF, December 2002.

[RFC 3344] C. Perkins (Editor). IP Mobility Support for IPv4. RFC 3344, IETF, August 2002.

[RFC 3398] G. Camarillo, A. B. Roach, J. Peterson, and L. Ong. Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping. RFC 3398, IETF, December 2002.

[RFC 3411] D. Harrington, R. Presuhn, and B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411, IETF, December 2002.

[RFC 3427] A. Mankin, S. Bradner, R. Mahy, D. Willis, J. Ott, and B. Rosen. Change Process for the Session Initiation Protocol (SIP). RFC 3427, IETF, December 2002.

[RFC 3465] M. Allman. TCP Congestion Control with Appropriate Byte Counting (ABC). RFC 3465, IETF, February 2003.

[RFC 3521] L-N. Hamer, B. Gage, and H. Shieh. Framework for Session Set-up with Media Authorization. RFC 3521, IETF, April 2003.

[RFC 3525] C. Groves, M. Pantaleo, T. Anderson, and T. Taylor (Editors). Gateway Control Protocol Version 1. RFC 3525, IETF, June 2003.

[RFC 3550] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF, July 2003.

[RFC 3551] H. Schulzrinne and S. Casner. RTP Profile for Audio and Video Conferences with Minimal Control. RFC 3551, IETF, July 2003.

[RFC 3588] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter Base Protocol. RFC 3588, IETF, September 2003.

[RFC 3711] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). RFC 3711, IETF, March 2004.

[RFC 3761] P. Faltstrom and M. Mealling. The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM). RFC 3761, IETF, April 2004.

[RFC 3850] B. Ramsdell (Editor). Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling. RFC 3850, IETF, July 2004.

[RFC 3851] B. Ramsdell (Editor). Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification. RFC 3851, IETF, July 2004.

[RFC 3856] J. Rosenberg. A Presence Event Package for the Session Initiation Protocol (SIP). RFC 3856, IETF, August 2004.

[RFC 3986] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, IETF, January 2005.

[RFC 3989] M. Stiemerling, J. Quittek, and T. Taylor. Middlebox Communications (MIDCOM) Protocol Semantics. RFC 3989, IETF, February 2005.

[RFC 4028] S. Donovan and J. Rosenberg. Session Timers in the Session Initiation Protocol (SIP). RFC 4028, IETF, April 2005.

[RFC 4080] R. Hancock, G. Karagiannis, J. Loughney, and S. Van den Bosch. Next Steps in Signaling (NSIS): Framework. RFC 4080, IETF, June 2005.

[RFC 4097] M. Barnes (Editor). Middlebox Communications (MIDCOM) Protocol Evaluation. RFC 4097, IETF, June 2005.

[RFC 4123] H. Schulzrinne and C. Agboh. Session Initiation Protocol (SIP)-H.323 Interworking Requirements. RFC 4123, IETF, July 2005.

[RFC 4168] J. Rosenberg, H. Schulzrinne, and G. Camarillo. The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP). RFC 4168, IETF, October 2005.

[RFC 4271] Y. Rekhter (Editor), T. Li (Editor), and S. Hares (Editor). A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, January 2006.

[RFC 4301] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, IETF, December 2005.

[RFC 4340] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). RFC 4340, IETF, March 2006.

[RFC 4346] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, IETF, April 2006.

[RFC 4460] R. Stewart, I. Arias-Rodriguez, K. Poon, A. Caro, and M. Tuexen. Stream Control Transmission Protocol (SCTP) Specification Errata and Issues. RFC 4460, IETF, April 2006.

[RFC 4540] M. Stiemerling, J. Quittek, and C. Cadar. NEC's Simple Middlebox Configuration (SIMCO) Protocol Version 3.0. RFC 4540, IETF, May 2006.

[RFC 4566] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566, IETF, July 2006.

[RFC 4864] G. Van de Velde, T. Hain, R. Droms, B. Carpenter, and E. Klein. Local Network Protection for IPv6. RFC 4864, IETF, May 2007.

[E.164] ITU-T. The international public telecommunication numbering plan. Rec. E.164, ITU-T, February 2005.

[E.721] ITU-T. Network grade of service parameters and target values for circuit-switched services in the evolving ISDN. Rec. E.721, ITU-T, May 1999.

[E.731] ITU-T. Methods for dimensioning resources operating in circuit-switched mode. Rec. E.731, ITU-T, October 1992.

[G.114] ITU-T. One-way transmission time. Rec. G.114, ITU-T, May 2003.

[G.711] ITU-T. Pulse code modulation (PCM) of voice frequencies. Rec. G.711, ITU-T, November 1988.

[G.729] ITU-T. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). Rec. G.729, ITU-T, March 1996.

[H.248.1 v3] ITU-T. Gateway control protocol: Version 3. Rec. H.248.1 v3, ITU-T, September 2005.

[H.248.37] ITU-T. Gateway control protocol: IP NAPT traversal package. Rec. H.248.37, ITU-T, September 2005.

[H.323] ITU-T. Packet-based multimedia communications systems. Rec. H.323, ITU-T, July 2003.

[Q.23] ITU-T. Technical features of push-button telephone sets. Rec. Q.23, ITU-T, November 1988.

[Q.700] ITU-T. Introduction to CCITT Signalling System No. 7. Rec. Q.700, ITU-T, March 1993.

[Q.761] ITU-T. Signalling System No. 7 - ISDN User Part functional description. Rec. Q.761, ITU-T, December 1999.

[Q.931] ITU-T. ISDN user-network interface layer 3 specification for basic call control. Rec. Q.931, ITU-T, May 1998.

[X.680] ITU-T. Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation. Rec. X.680, ITU-T, July 2002.

[Y.2011] ITU-T. General principles and general reference model for Next Generation Networks. Rec. Y.2011, ITU-T, October 2004.

[Z.100] ITU-T. Specification and Description Language (SDL). Rec. Z.100, ITU-T, August 2002.

[Z.120] ITU-T. Message sequence chart (MSC). Rec. Z.120, ITU-T, April 2004.

[TS101671] ETSI Technical Committee Lawful Interception. Handover interface for the lawful interception of telecommunications traffic. Technical Specification TS 101 671 V2.15.1, ETSI, 2006.

[TS 23.002] 3GPP WG S2. Network architecture. TS 23.002 ver. 7.1.0, 3GPP, March 2006.

[TS 23.228] 3GPP WG S2. IP Multimedia Subsystem (IMS); Stage 2. TS 23.228 ver. 7.7.0, 3GPP, March 2007.

[ES282001] TISPAN. NGN Functional Architecture Release 1. ETSI Standard ES 282 001 V1.1.1, ETSI, 2005.