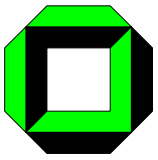
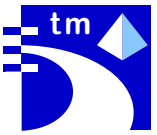


Universität Karlsruhe (TH)
Fakultät für Informatik
Institut für Telematik
76128 Karlsruhe



Netzwerkmanagement und Hochleistungskommunikation

Teil XXIV

Seminar SS 2001

Herausgeber:
Roland Bless
Till Harbaum
Daniel Müller
Anke Speer

Universität Karlsruhe (TH)
Institut für Telematik
Prof. Dr. Martina Zitterbart
http://www.tm.uka.de/

Fakultät für Informatik
Interner Bericht 2001-12
ISSN 1432–7864

Zusammenfassung

Der vorliegende interne Bericht enthält die Beiträge zum Seminar „Netzwerkmanagement und Hochleistungskommunikation“, das im Sommersemester 2001 zum 24. Mal stattgefunden hat.

Ein Block ist der Hochgeschwindigkeits-Technologie gewidmet. Im ersten Beitrag werden *Architekturen für High-Speed-Switches und Router* vorgestellt, die vor dem Hintergrund entwickelt wurden, dass eine gewisse Dienstgüte zu garantieren ist. Der zweite Beitrag beschreibt *Algorithmen für Routing Table Lookup*, d. h. effiziente Verfahren zur Ermittlung des nächsten Knotens während einer Paketweiterleitung. Des Weiteren sind effiziente Verfahren und Algorithmen zur *Paketklassifikation* von IP-Paketen von Interesse, die üblicherweise bei filternden Netzelementen eingesetzt werden und welche bei ständig steigender Geschwindigkeit der Router zunehmend an Bedeutung gewinnen. Darüberhinaus wird das Konzept des *Multiprotocol Label Switchings (MPLS)* vorgestellt, welches gegenüber herkömmlichem Routing eine größere Flexibilität bezüglich der Verkehrslenkung und damit eine bessere Kontrolle der Netzauslastung mit sich bringt.

Ein zweiter Block beschäftigt sich mit verschiedenen Themen aus unterschiedlichen Bereichen. Hier werden zum einen das *Policy-Based Networking* als neueres Paradigma zur Verwaltung von Netzwerken vorgestellt sowie *Sicherheitserweiterungen des DNS* und damit auf im Zeitalter der ständig wachsenden Netze zunehmend wichtige Netzwerkmanagement-Themen eingegangen. Zum anderen werden im Beitrag zu *Mobilitätsprofilen in mobilen Ad-Hoc-Netzen* Mobilitätsaspekte bei Ad-Hoc-Netzen behandelt. Schließlich werden im Thema *Watermarking* Verfahren zum Schutze von multimedialen Daten vorgestellt, welche in jüngster Zeit stark diskutiert wurden, um beispielsweise den Tausch von illegal angefertigen Kopien über das Internet zu unterbinden. Probleme und Herausforderungen, die mit der Zielsetzung verbunden sind, über das Internet telefonieren zu können, waren Gegenstand des Beitrags *Internet Telephony*. Eine Lösung, um den Transport von „herkömmlichen“ Signalisierungsnachrichten für Telefonnetze auch im Internet effizient zu unterstützen, wird im Beitrag *Stream Control Transmission Protocol* vorgestellt. Ein weiterer Beitrag umfasste den Themenbereich der Gruppenkommunikation. Hier werden einerseits *neuere Ansätze zum Multicast-Routing* beschrieben und andererseits eine Auswahl der funktional darüber angeordneten, zahlreichen Multicast-Transportprotokolle vorgestellt.

Abstract

This Technical Report includes student papers produced within a seminar of ‘Network Management and High Performance Communications’. For the 24nd time this seminar has attracted a large number of diligent students, proving the broad interest in topics of network management and high performance communications.

The topics of this report may be coarsely divided into two blocks:

1. One block is devoted to high speed and high performance technology. At first, the concept of modern *High Speed Switches and Routers* with quality-of-service support is described. Subsequently, *Efficient Methods and Algorithms for Routing Table Lookups* as well as *Classification of IP Packets* and *Multiprotocol Label Switching (MPLS)* are presented.

2. A second block deals with various topics such as wireless communications, network management and security. The first article shows advantages of the *Policy-based Networks* to manage todays networks. Furthermore, *Security Extensions of DNS* for secure use

of the domain name service are examined and presented. The next article describes how to use *mobility profiles in mobile ad-hoc networks*. Methods for *watermarking* of multimedia data are discussed in a subsequent article. Moreover, *Technical Challenges and Solutions for IP-telephony* are also presented, whereby the *Stream Control Transmission Protocol* is described separately as an approach to achieve a better transport of signaling messages over the Internet. The last article deals with group communication and shows *New Approaches for Multicast Routing* as well as an overview of some Multicast transport protocols.

Inhaltsverzeichnis

Zusammenfassung i

Vorwort v

Markus Krämer:
Architekturen für High-Speed-Switches/Router 1

Maria Vassiliadou:
Algorithmen für Routing Table Lookup 15

Johann Costin Mihutoni:
Paketklassifikation 27

Michael Wiese:
Multi Protocol Label Switching 45

Georg Kassner:
Policy-based Networking 57

Richard Mager:
Sicherheitserweiterungen des DNS 71

Thomas Richter:
Mobilitätsprofile in mobilen Ad-hoc-Netzen 87

Ivonne Heinemann:
Watermarking: Copy Control for Multimedia 105

Jens Deidersen:
Internet Telephony: Technical Challenges and Solutions 123

Georgios Papadopoulos:
Das Stream Control Transmission Protocol SCTP 137

Anselm Kreuzer:
Multicast - Empowering the Next-Generation Internet 147

Vorwort

Das Seminar „Netzwerkmanagement und Hochleistungskommunikation“ erfreut sich wie auch in den letzten Jahren weiterhin großer Beliebtheit. Gerade heutzutage sind Stichworte wie „Switching“, „Routing“, „Quality of Service“, „Multicast“ oder „Internet“ in aller Munde. Daher sind die Forschungsgebiete in diesen Bereichen auch von allgemeinem Interesse, so dass sie eine derartige Vielzahl von innovativen Arbeiten aufweisen können, deren Behandlung in anderen Lehrveranstaltungen so detailliert nicht möglich ist.

Jetzt liegt auch der nunmehr 24. Seminarband als interner Bericht vor. Durch die engagierte Mitarbeit der beteiligten Studenten konnte so zumindest ein Ausschnitt aus dem komplexen und umfassenden Themengebiet klar und übersichtlich präsentiert werden. Für den Fleiß und das Engagement der Seminaristen sei daher an dieser Stelle recht herzlich gedankt.

Die ausgesprochen gute Resonanz bei den Studenten bestätigt uns darin, auch im Sommersemester 2002 ein derartiges Seminar – natürlich mit geändertem aktuellem Inhalt – durchzuführen, so dass bald ein weiterer interner Bericht mit neuen Forschungsergebnissen aus innovativen Seminarbeiträgen erscheinen wird.

Architekturen für High-Speed-Switches/Router

Markus Krämer

Kurzfassung

Diese Arbeit bietet einen Überblick über die gängigen Konzepte im Bereich der Switch-und Routertechnologie und zeigt neuere Ansätze im Bereich Dienstgüte (Quality-of-Service – QoS) und Skalierbarkeit in den Netzknoten auf. Es wird wird auf die verschiedenen Speicherkonzepte, welche zur Pufferung der zu vermittelnden Pakete in den Netzknoten nötig sind, eingegangen, sowie deren Vor- und Nachteile beschrieben. Nach einer ausführlichen Betrachtung neuerer Ansätze zur Bereitstellung von Dienstgüte für Switches mit Eingangsspeicher, erfolgt die konkrete Vorstellung der beiden Switch-Technologien PRIZMA und Saturn.
Da die Forschung und Entwicklung zu immer leistungsfähigeren Switches/Routern von der enormen Steigerung der Kapazität und Geschwindigkeit heutiger Netze getrieben wird, erfolgt auch eine Vorstellung neuartiger Netztechnologien. Hierbei wird speziell auf die optischen Netze und den Ansätzen, die es in der Paketvermittlung mittels optischer Switches gibt, eingegangen.

1 Einleitung

In den letzten Jahren ist das Datenverkehrsaufkommen im Internet bedingt durch eine stark wachsende Anzahl von Anwendern und damit einhergehend der Bereitstellung vieler neuer Dienste in exponentieller Weise gestiegen. Als weltweites Kommunikationsmedium bietet das Internet die Möglichkeit der Integration von Daten-, Sprach- und Videodiensten. Anwendungsformen wie z.B. das Abhalten einer Videokonferenz, können dabei Echtzeitbetrieb und reservierte Bandbreiten erforderlich machen. Um das Wachstum in Zahlen auszudrücken, bedeutet dies, dass sich schätzungsweise die Zahl der Host-Rechner seit 1989 jedes Jahr sowie die Zahl der Web-Server in den letzten drei Jahren fast halbjährlich verdoppelt hat. Ermöglicht wurde dies durch den weltweit physischen Ausbau der Netze und die immer schnelleren Zugangs- und Übertragungsgeschwindigkeiten. Diese Entwicklung und die durch neuartige Anwendungsformen ausgelöste Nachfrage nach Dienstgüte-Garantien stellen neue Erfordernisse an Hochgeschwindigkeitsnetze und ihre Vermittlungsknoten dar. Um den im Folgenden häufig auftretenden Begriff der Dienstgüte (Quality-of-Service – QoS) verständlich zu machen, sei nun eine kurze Erklärung in Anlehnung an [Kauf00] gegeben: „Mit Quality-of-Service ist im Allgemeinen gemeint, dass ein Netz zwischen zwei Anwendungen bestimmte Grundmerkmale für die Verbindung garantieren kann, wie z.B. Priorisierung, maximale Verzögerung, Garantie einer minimalen Bandbreite, usw.“
Switches und Routern, die in Hochgeschwindigkeitsnetzen eingesetzt werden und einerseits Dienstgüte-Garantien ermöglichen andererseits Skalierbarkeit aufweisen, gilt in der letzten Zeit ein verstärktes Entwicklungsinteresse sowohl in der Industrie als auch in der Akademischen Forschung. Unter Skalierbarkeit ist hierbei zu verstehen, dass die Systeme an Kapazitäts- und Geschwindigkeitänderungen im eingesetzten Umfeld anpassbar und erweiterungsfähig sind. Ein besonderes Augenmerk bei der Entwicklung neuerer Switches/Router

wird auf das Design der Verschaltungseinheiten (Switch-Fabrics) gelegt. Diese Schaltelemente, die in Switches und Routern intern die Eingangsports mit den Ausgangsports verbinden, gilt es für zukünftige Switch- und Routergenerationen so zu konzipieren, dass ein Durchsatz in der Grössenordnung von mehreren Terabit/s möglich ist. Eine besondere Rolle bei dieser Entwicklung nehmen dabei die sog. Schedulingverfahren ein, welche für das Auflösen eventuell auftretender Konflikte zwischen Eingangs- und Augangsports benötigt werden. Diese Konflikte treten in den hier behandelten blockierungsfreien Verschalteinheiten meist dann auf, wenn mehrere Pakete an den Eingängen innerhalb eines Zeitschlitzes zur Übertragung an die selben Ausgänge bereitstehen. Besonders der Entwurf von effizienten Schedulingverfahren spielt im Hinblick auf QoS, Geschwindigkeit und Skalierbarkeit eine grosse Rolle. Nach einer Wiederholung der grundlegenden Funktionsweise eines Switches/Routers und den verschiedenen Queuing-Konzepten, die ebenfalls einen gewichtigen Faktor in Bezug auf Skalierbarkeit und Performance darstellen, wird sich deshalb ein Teil dieser Arbeit mit der Vorstellung neuer Ansätze im Bereich der Schedulingverfahren für Switches/Router befassen.
Durch die Einführung der optischen Übertragungsmedien und der enormen Kapazitäts- und Geschwindigkeitsteigerung mittels der Wellenlängenmultiplex-Technologie (Wavelength-Division-Multiplexing – WDM), drängt sich in letzter Zeit immer mehr die Frage auf, ob die benötigten Portgeschwindigkeiten der Switches mit dieser Entwicklung Schritt halten können oder neue Konzepte nötig sind. Der Schlussteil dieser Arbeit befasst sich darum mit den Ansätzen, die es im Bereich des optischen Paket-Switching gibt, und beschreibt mögliche Architekturen für zukünftige optische Netze.

2 Architekturen von Switches/Routern

Der Aufbau eines modernen Routers oder Switches ist in mehrere funktionale Einheiten gegliedert. In Abbildung 1 sind die wichtigsten Komponenten dargestellt. Zu diesen Komponenten zählen die *Schnittstelle* zur Anbindung verschiedener Übertragungssysteme an den Vermittlungsrechner, der *Netzwerk-Prozessor*, die *Switch-Fabric* (Verschalteinheit: z.B. Crossbar) und der *System-Prozessor*.

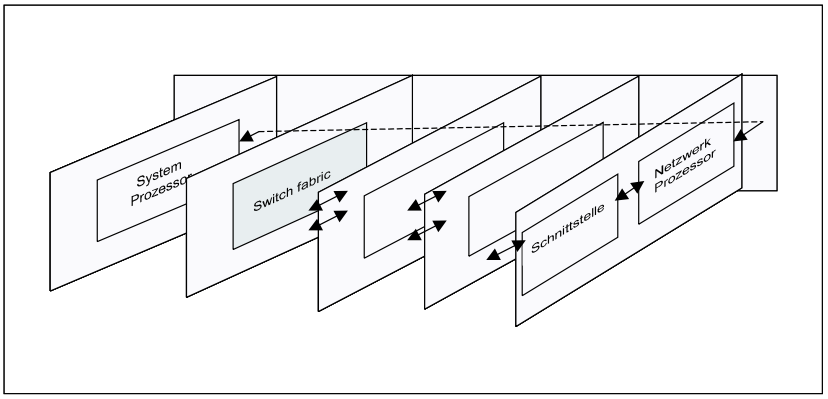


Abbildung 1: Aufbau eines modernen Switchs/Routers

Der *Netzwerk-Prozessor* ist dabei für die Untersuchung von Paketköpfen, die Auswertung der Routingtabelle und die Klassifizierung der Pakete nach ihrer Herkunfts- und Zieladresse zuständig. Des weiteren verarbeitet er Kontrollinformationen und überwacht die Pufferung der Pakete.
Die *Switch-Fabric*, welche im folgenden Abschnitt bei der Beschreibung der verschiedenen

Queuing-Konzepte noch intensiver betrachtet wird, leistet die Vermittlung der Pakete in hoher Geschwindigkeit zwischen den Einheiten des Systems, speziell zwischen Netzwerk- und System-Prozessoren. Die Hauptaufgabe besteht dabei im räumlichen Transfer der Pakete von den Eingangs- zu den Ausgangsports und deren Pufferung, um Konflikte aufzullösen. Steuerungsfunktionen in der Kontrollebene, wie die Berechnung von Routen zur Erstellung der Routingtabelle oder das Netzwerk- und Systemmanagement, werden durch den *System-Prozessor* erbracht.
Bei der Betrachtung heutiger Vermittlungssysteme fällt auf, dass sie mit anwendungsspezifischen Schaltkreisen (Application-Specific-Integrated-Circuits – ASICs) ausgestattet sind, um die Paketvermittlung effizient durchführen zu können. Zwar macht die ständige Weiterentwicklung in der Halbleiterindustrie diese Schaltkreise immer schneller und kleiner, jedoch entstehen durch Auswechseln der Hardware zur Anpassung der Systeme an neue Erfordernisse auch immense Kosten. Eine Möglichkeit diese Unflexibilität und die ständig nötige Entwicklung neuer ASICs zu überwinden, ist der Einsatz der schon erwähnten programmierbaren *Netzwerk-Prozessoren (NP)*. Diese Prozessoren ermöglichen den Herstellern auf den Schichten 3-7 der Paketverarbeitung Funktionen mittels Anpassung der NP-Software hinzuzufügen oder zu verändern, anstatt ganze Hardwarekomponenten austauschen zu müssen. Dieser Vorteil macht deutlich, weshalb gegenwärtig riesige Investitionen seitens der Unternehmen in die NP-Technologie getätigt werden und der Markt für programmierbare Kommunikations-Prozessoren im Jahr 2003 schätzungsweise die Ein-Milliarde US-Dollar Grenze erreichen wird. Um einen tieferen Einblick in das Design von Netzwerk-Prozessoren zu erlangen, sei auf [BDEH+01] verwiesen.

2.1 Warteschlangenkonzepte in Switches/Routern - Queuing Strategien

Queuing-Konzepte nehmen innerhalb der Switch- und Routertechnologie eine zentrale Stellung ein. Bei ihrer Vorstellung wird im Folgenden immer von einer nichtblockierenden Schalteinheit (Nonblocking Switching Fabric) ausgegangen, was soviel bedeutet, dass innerhalb dieser keine Konflikte auftreten, sondern nur extern an Eingangs- bzw. Ausgangsports möglich sind. Des weiteren wird angenommen, dass ein N $\times$ N-Schaltwerk mit N Ein- und N Ausgangsports (vgl. Abbildung 2) immer nur ein Paket pro Zeitschlitz übertragen kann. Arbeitet die Switch-Fabric allerdings mit einer 20-fach höheren Geschwindigkeit, so können auch 20 Pakete pro Zeiteinheit übertragen werden. Man kann sich das so vorstellen, dass der Zeitschlitz in 20 kleinere Schlitze unterteilt wird und somit auch wieder nur ein Paket pro Mini-Zeitschlitz übertragen wird. Bei paketvermittelnden Systemen, wie den hier diskutierten Switches und Routern, besteht die Möglichkeit, dass zur gleichen Zeit mehrere Pakete an den Eingangsports eintreffen, die als Ziel denselben Ausgangsport haben. Falls also z.B. das Schaltwerk mit der Geschwindigkeit S=1 betrieben wird, dann bedeutet dies, dass nur ein Paket sofort übertragen werden kann und die anderen gespeichert werden müssen, um nicht verloren zu gehen. Diese Speicherung und Verzögerung in einer effizienten Art und Weise durchzuführen, stellt ein komplexes Problem dar, und verschiedene Konzepte wurden hierfür schon entwickelt:

- Ausgangsspeicherung – Output Queuing (OQ)
- Zentralspeicherung – Centralized Shared Queuing (CSQ)
- Eingangsspeicherung – Input Queuing (IQ)
- Virtuelle Ausgangsspeicherung – Virtual Output Queuing (VOQ)
- Verteilte Speicherung – Combined Input-Output Queuing (CIOQ)

Hierbei gilt es zu beachten, dass kein Verfahren alle Schwierigkeiten löst, sondern jedes Vor- und Nachteile aufweist.

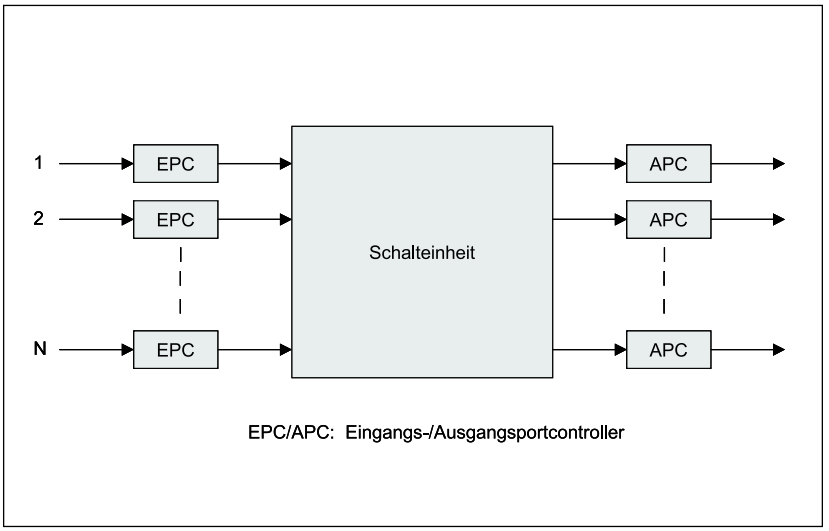


Abbildung 2: Modell einer N × N Switch-Fabric mit Ein- und Ausgangsports

2.1.1 Output Queuing (OQ)

Ankommende Pakete werden sofort nach ihrem Eintreffen an einem Eingangsport zum entsprechenden Ausgangsport vermittelt und dort in einen Pufferspeicher (FIFO-Speicher) geschrieben. Der Nachteil dieser Speichermethode ist, dass die interne Geschwindigkeit der Vermittlungsmatrix grösser als die Geschwindigkeit der ankommenden Pakete sein muß, da es vorkommen kann, dass Pakete, die für den gleichen Ausgang bestimmt sind, zur selben Zeit an verschiedenen Eingängen ankommen. Existieren beispielsweise N Eingangsports, so muss im schlechtesten Fall der Ausgangsspeicher mit einer N-fach höheren Rate arbeiten. Diese Eigenschaft limitiert die Grösse von OQ-Switches. Die Entwicklung dieser Switches ist sehr komplex und damit auch teuer. Ein weiterer Nachteil ist die feste Grösse der Speicher. Ist ein Speicher voll, so kann es zu Paketverlusten kommen.

2.1.2 Centralized Shared Queuing (CSQ)

Alle an den Eingangsports ankommenden Pakete werden in einen gemeinsamen Speicher geschrieben. Dieser kann kleiner als die Summe getrennter Speicher sein, die Steuerung des Zugriffs ist jedoch kompliziert, und es ist eine sehr hohe Geschwindigkeit des Speicherzugriffs erforderlich, da wie bei der Ausgangsspeicherung Pakete simultan ankommen können, die für den gleichen Ausgang bestimmt sind. Auch diese Methode limitiert die Switchgrösse. Ein Vorteil ist jedoch, dass Paketverluste seltener als bei reinen IQ/OQ-Switches vorkommen.

2.1.3 Input Queuing (IQ)

In jedem Eingangsport werden die ankommenden Pakete in einen Speicher nach dem FIFO-Prinzip (First-In-First-Out) gespeichert, d.h. es kann immer nur das erste Paket in der Warteschlange zur Vermittlung ausgewählt werden. Zwar haben Switches, die nach diesem Konzept entworfen sind, keine Grössenbeschränkung und ihr Entwurf ist im Gegensatz zu OQ-Switches recht einfach, jedoch wird auch ein grosser Nachteil deutlich. Dieser besteht darin, dass im FIFO-Speicher wartende Pakete an der Vermittlung gehindert werden, obwohl der entsprechende Ausgang frei ist. Diese Beschränkung der eingangsgepufferten Switches ist auch unter

dem Begriff „Head-of-Line“-Blockierung bekannt. Durch Simulation und mathematische Untersuchungen wurde gezeigt, dass der Durchsatz von IQ-Switches sowohl bei gleichmäßigem als auch burstartigem Verkehr bei 58 Prozent beschränkt ist (siehe hierzu auch [NoHa00]). Ein weiterer Nachteil ist auch hier die feste Grösse der Speicher, die zu Paketverlusten führen kann.

2.1.4 Virtual Output Queuing (VOQ)

Dieses Konzept löst das Problem der Head-of-Line-Blockierung, das bei FIFO-IQ-Switches auftritt, behält aber zugleich deren Vorteil der Skalierbarkeit. Dabei besitzt jeder Eingangsport einen separaten Speicher für jeden Ausgangsport. Das Problem an diesem Verfahren ist, dass ein erhöhter Speicherbedarf nötig ist und die Schwierigkeit besteht, eine hohe Durchsatzleistung zu erreichen. Aus diesem Grund werden Schedulingverfahren benötigt, die dafür sorgen, dass in jeder Zeiteinheit Pakete von den Eingängen zu den Ausgängen übertragen werden. Ansätze für effiziente Schedulingalgorithmen werden daher im Folgenden noch betrachtet.

2.1.5 Combined Input-Output Queuing (CIOQ)

Dieses Verfahren stellt eine Kombination aus Eingangsspeicherung und Ausgangsspeicherung dar. Es wird dabei ein guter Kompromiss zwischen hoher Durchsatzleistung und Skalierbarkeit erzielt. Trotzdem tritt auch hier der Nachteil des erhöhten Speicherbedarfs auf, und ein effizientes Scheduling ist nötig.

Abschliessend ist zu bemerken, dass die neueren Konzepte wie VOQ und CIOQ, genauso wie die traditionellen Konzepte der Eingangsspeicherung und Ausgangsspeicherung, Beschränkungen aufweisen. Sie haben jedoch mittels effizienter Schedulingverfahren das Potential, eine hohe Leistung zu erzielen und dabei skalierbar zu bleiben.

2.2 Scheduling und QoS in VOQ- und CIOQ-Switches

Scheduling wirkt als eine Art „Schiedsrichter“ zwischen den Eingangsports und den Ausgangsports der Switch Fabric, um Konflikte zwischen zur Übertragung anstehender Pakete zu lösen. Es stellt jedoch eine schwierige Aufgabe dar, Schedulingverfahren für Hochgeschwindigkeitsswitches zu entwickeln. In letzter Zeit sind jedoch gute Ansätze für Technologien entwickelt worden, die diese Probleme zu lösen scheinen. Dabei werden einige Anforderungen an das Scheduling gestellt. Ein effizientes Schedulingverfahren sollte bei starkem Datenverkehrsaufkommen einen hohen Durchsatz, niedrige Latenz und eine gute Anpassung bei Veränderungen im Verkehrsaufkommen bieten.
Im Folgenden wird zunächst der sog. *Parallele Iterative Matching (PIM)*- Algorithmus vorgestellt, der selbst keine deterministischen QoS-Garantien unterstützt, jedoch eine gute Basis für ähnlich konzipierte Algorithmen mit QoS-Fähigkeit bietet. Hierbei handelt es sich um die drei Verfahren:

- *Zeitschlitzzuweisung - Timeslot Assignment (TSA)*
- *Maximales Matching (MM)*
- *Stabiles Matching (SM)*

2.2.1 Konzept des Parallelen Iterativen Matchings (PIM)

Der Anspruch des hohen Durchsatzes und der niedrigen Latenz beim Datenverkehr in Hochgeschwindigkeitsnetzen verlangt, dass mittels des Scheduling-Verfahrens möglichst viele konfliktfreie Paare, bestehend aus Eingängen mit wartenden Paketen und den entsprechenden Ausgängen in möglichst kurzer Zeit gefunden werden. Das Auffinden von solchen Paaren kann traditionell als ein Bipartites-Problem angesehen werden für das schon längere Zeit Lösungsalgorithmen existieren. Allerdings weisen diese häufig eine zu grosse Zeitkomplexität ($\geq O(N^2)$) auf, was gerade in einem Hochgeschwindigkeitsnetz sehr ungünstig ist. Das PIM-Verfahren bietet durch Einsatz von Parallelisierung, zufälliger Auswahl und Iterationen eine Verbesserung an. Dabei werden nur solche Paare gesucht, bei denen auch tatsächlich Pakete an den Eingängen zur Übermittlung anstehen und die entsprechenden Ausgänge noch frei sind. Das Verfahren besitzt eine Zeitkomplexität von O(logN) und durchläuft mehrmals hintereinander die drei Schritte Anfrage (Request), Bewilligung (Grant) und Annahme (Accept).

- *Request*: Jeder Eingang, welcher noch kein Matching erfahren hat, sendet Anfragen an alle Ausgänge aus, für die er Pakete zur Übermittlung bereitstehen hat.
- *Grant*: Bekommt ein Ausgang, welcher noch kein Matching erfahren hat, Anfragen, so wählt er zufällig eine darunter aus und meldet dies an den entsprechenden Eingang.
- *Accept*: Erhält ein Eingang Grants, so wählt er einen Ausgang aus und ein neues Paar ist gefunden.

Durch Simulation und mathematische Analyse wurde gezeigt, dass mittels PIM in VOQ-Switches und einem gleichmäßigem Verkehrsaufkommen schon nach dem Durchlauf von vier Iterationen ein Durchsatz von über 99 Prozent möglich ist. Die Switchgrösse spielt dabei keine Rolle. Algorithmen, die einen Durchsatz von 100 Prozent unter gleichmäßigem oder burstartigen Verkehrsaufkommen ermöglichen, sind zwar entwickelt (siehe hierzu auch [McAW96]), weisen aber den Nachteil der zu grossen Zeitkomplexität ($O(N^{2.5})$) auf.
Die Schwierigkeiten in VOQ-/CIOQ-Switches QoS-Garantien zur ermöglichen, liegt daran, dass ihr Switch-Fabric meist in der Grösse begrenzt ist und anstehende Pakete nicht sofort mit der gewünschten Dienstgarantie übertragen werden können. Algorithmen, die Ansätze zur Lösung dieses Problems aufzeigen, werden in den folgenden Abschnitten betrachtet.

2.2.2 Timeslot Assignment (TSA)

Algorithmen, die nach dem Prinzip der festen Zeitschlitzzuweisung arbeiten, legen exakt den Zeitpunkt fest, wann Pakete den VOQ-Switch verlassen. Diese Eigenschaft macht sie besonders für Echtzeitanwendungen geeignet, die eine konstante Bitrate (Constant Bitrate – CBR)und eine feste Verzögerung benötigen.
Ein Ansatz für ein solches Verfahren stellt der sog. gewichtete PIM-Algorithmus (Weighted Probalistic Iterative Matching – WPIM) dar. Dieser verwendet Konzepte des zuvor beschriebenen PIM-Algorithmus und berechnet darüber hinaus noch Gewichte, um die Konflikte zwischen Ein- und Ausgängen zu lösen. Der Vorteil an diesem Verfahren ist, dass auf eine einfache Weise unterschiedlichen Verbindungen flexibel eine feste Bandbreite zugeteilt werden kann. Ein Nachteil des WPIM-Verfahrens ist aber, dass es in jedem Zeitschlitz neu ablaufen muss und eine Zeitkomplexität von $O(N^2)$ besitzt.
Ein weiterer interessanter Ansatz für TSA-Scheduling bildet der sog. gewichtete Reihum-Algorithmus (Weighted Round-Robin – WRR). Dieses verfahren ermöglicht eine feste Bandbreite in VOQ-Switches zu garantieren, hat aber den Nachteil, dass es nur für Verkehr mit

konstanter Bitrate entwickelt wurde. Das letzte der vorgestellten TSA-Verfahren bildet der sog. BATCH_TSA-Algorithmus, welcher eine geringe QoS-Garantie für Verkehr mit variabler Bitrate (VBR) ermöglicht. Das Prinzip beruht darauf, dass die Switch-Fabric mit seinen Ein- und Ausgängen änlich eines TDMA-Netzwerks (Time Division Multiple Access – TDMA) behandelt wird, und somit Lösungen des klassischen TSA-Problems anwendbar sind. Hierbei kann ein Durchsatz von bis zu 100 Prozent in VOQ-Switches erreicht werden. Nachteile weist dieses Verfahren allerdings bei häufig wechselnden Verkehrslasten auf.

2.2.3 Maximales Matching (MM)

Der hohe Durchsatz, welcher mit TSA-basierten Verfahren erreicht werden kann, geht zu Lasten einer niedrigen Zeitkomplexität. Der nach dem Prinzip des Maximal Matching arbeitende LOOFA-Agorithmus (Lowest Output Occupancy First – LOOFA) besitzt dagegen entgegengesetzte Charakteristika: bessere Zeitkomplexität aber niedrigerer Durchsatz. Das Verfahren nutzt die Vorteile von PIM und erzeugt zusätzlich eine umfassende Präferenzliste zur Lösung von Ein-/Ausgangskonflikten. Wird dieses Verfahren in einem VOQ-Switch eingesetzt, so muss die interne Vermittlungsmatrix mit einer mindestens zweifach höheren Geschwindigkeit als die Eingänge betrieben werden, um einen Durchsatz von 50 Prozent zu erreichen. Eine interne Geschwindigkeit mit Faktor vier oder sechs ist erforderlich, um Paketverzögerungen ausreichend zu begrenzen. Algorithmen, die ebenfalls die Verzögerung von Paketen begrenzen und gleichzeitig einen höheren Durchsatz ermöglichen, werden häufig nach dem Konzept des Stabilen Matchings entworfen.

2.2.4 Stabiles Matching (SM)

Das Verfahren des Stabilen Matchings setzt voraus, dass Prioritäten zwischen den zur Übermittlung anstehenden Paketen festgelegt werden. Die Schwierigkeit dabei ist, Informationen der gepufferten Pakete, speziell über die gewünschte QoS-Garantie zu erlangen und somit die Prioritäten festzulegen. Der sog. MUCFA-Algorithmus (Most Urgent Cell First – MUCFA) bietet dafür einen guten Ansatz. Die Arbeitsweise des Verfahrens ähnelt dem LOOFA-ALgorithmus, jedoch werden im Gegensatz dazu Präferenzlisten sowohl für die Eingänge als auch die Ausgänge angelegt. Es wurde gezeigt, dass durch eine interne Geschwindigkeit mit Faktor vier in einem CIOQ-Switch, welcher MUCFA einsetzt, die Nachahmung eines OQ-Switches möglich ist. Da für OQ-Switches schon einige Verfahren zur Garantie von Quality-of-Service entwickelt wurden, ist dies ein grosser Schritt in Richtung der Bereitstellung von QoS in CIOQ-Switches. Allerdings weist dieses Verfahren trotzdem noch einige Nachteile auf:

- Die Methoden, die zum Erzielen von QoS-Garantie entwickelt wurden, gelten nur für Unicast-Verkehr, nicht aber für Multicast-Verkehr.
- Durch Nachahmung von OQ-Switches werden QoS-Garantien erreicht, die für realistische Verkehrsarten von minderer Bedeutung sind.
- Für Echtzeitanwendungen in Hochgeschwindigkeitsnetzen sind die Verfahren meist zu komplex.
- Das Scheduling wurde bisher nur für Pakete mit fester Länge entwickelt, nicht aber für variable Paketlängen.

Algorithmen	**Komplexität**	**Maximaler Durchsatz**	**Verschiedene QoS-Arten**	**Verkehrsart**
Zeitschlitzzuweisung	$O(N^{2.5})$	100%	nicht unterstützt	CBR
Maximal Matching	$O(N^2)$	50%	nicht unterstützt	CBR
Stabiles Matching	$\Omega(N^2)$ oder $O(N^2)$	50%	unterstützt	CBR, VBR

Tabelle 1: Vergleich der Scheduling-Algorithmen

In diesem Abschnitt wurden verschiedene Algorithmen zum Erreichen von Quality-of-Service in Switches vorgestellt, die nach den neuen Konzepten CIOQ und VOQ entworfen sind. In Tabelle 1 werden ihre Performance-Unterschiede nochmals herausgestellt.

3 Switch-Technologien: PRIZMA und Saturn

In den bisherigen Abschnitten wurden die grundlegenden Konzepte der Switch/Router-Technologie beschrieben und neue Ansätze für zukünftige Entwicklungen aufgezeigt. Es folgt nun die Vorstellung zweier konkreter Technologien, welche einige dieser Konzepte verwenden.

3.1 PRIZMA

PRIZMA ist eine Technologie, die von IBM entwickelt wurde und aus einer ganzen Familie von Chips für Switches besteht.
Die PRIZMA-Architektur kombiniert die Konzepte *Input-Queuing* und *Output-Queuing* miteinander. Ein N × N-Chip dieser Technologie arbeitet blockierungsfrei und leitet eintreffende Pakete selbstroutend von den N Eingangsports zu einem oder mehreren der N Ausgangports weiter, basierend auf Switch-internen Portadressen. Pakete werden dabei physikalisch in einem Zentralspeicher abgelegt und mit einem Zeiger auf einen logischen Ausgangsspeicher (VOQ) versehen. Dieses Prinzip ermöglicht ausser Unicast- auch den Multicast-Betrieb, da mehrere Zeiger des jeweils gespeicherten Paketes auf die Speicher verschiedener Ausgänge gerichtet werden können. Die Architektur ist skalierbar und besitzt die Fähigkeit Portgeschwindigkeiten zu verändern, was auch unter dem Begriff „speed expansion“ bekannt ist. Dies wird durch parallele Verschaltung mehrerer Chips erreicht. Darüber hinaus können auch mehrere Chips stufenweise hintereinander geschaltet werden, was eine Veränderung der Portanzahl ermöglicht und unter dem Begriff *„port expansion“* bekannt ist. Der Zusammenhang beider Expansionsformen wird in Abbildung 3 dargestellt. Die mit PRIZMA verarbeitbaren Pakete können ein Länge zwischen 64 und 80 Bytes haben. Diese Pakete können sowohl die 53 Byte grossen ATM-Zellen enthalten, als auch minimale Ethernetrahmen der Länge 64 Byte. Um Quality-of-Service-Garantien zu ermöglichen, besitzt jeder Ausgangsport logische Prioritätsspeicher und beinhaltet einen Mechanismus, der eine feste Bandbreite reservierbar macht.

Die zweite Generation der PRIZMA-Technologie ist durch Chips mit 32 Eingangsports und 32 Ausgangsports charakterisiert. Jeder Port wird dabei mit einer Geschwindigkeit von 2 Gigabit/s betrieben, was zusammengenommen einen Durchsatz von 64 Gigabit/s ermöglicht. Durch Geschwindigkeitsexpansion ist es sogar möglich, 128 Gigabit/s zu erreichen. Bei der Entwicklung der dritten PRIZMA-Generation ist das Ziel in den Terabit/s - Bereich vorzustoßen. Um einen detaillierten Überblick der Leistungsfähigkeit der PRIZMA-Technologie zu erlangen, wird auf [MiEn00] verwiesen. Es werden darin das Verhalten von PRIZMA bei variablen Paketlängen und die erzielbare Performance dargestellt. Ausserdem wird ein Vergleich zu einem klassischen OQ-Switch-Chip namens ATLAS gezogen.

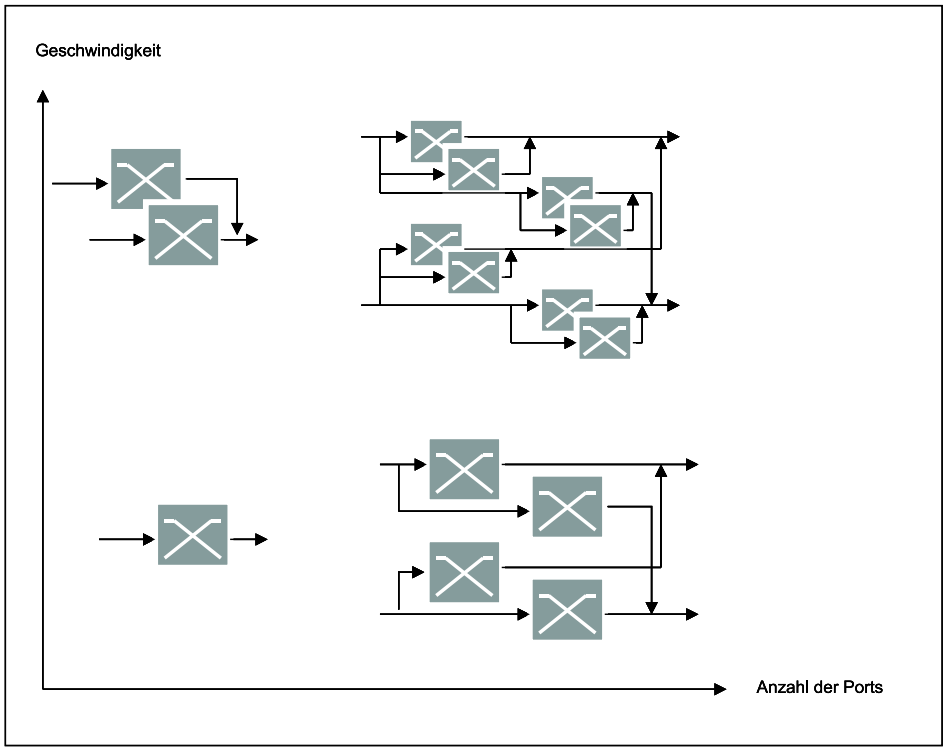


Abbildung 3: PRIZMA Expansionsformen

3.2 Saturn

Saturn (Switch at terabit using dual round-robin) stellt einen Ansatz für eine Switch-Technologie zur Paketvermittlung dar, welche eine Kapazität von 1 Terabit/s erreicht. Das Vermittlungsprinzip dieser Switch-Architektur beruht auf dem sog. Dualen Round-Robin-Verfahren und erreicht auf diese Weise einen hohen Durchsatz mit vergleichsweise statistisch niedriger Paketverzögerung. Die Schaltmatrix eines Saturn-Chips ist nach dem Crossbar-Prinzip entworfen und sowohl Eingangsports als auch Ausgangsports arbeiten jeweils mit einer Geschwindigkeit von 10 Gigabit/s. Das Interessante an dieser Technik ist das Vermittlungsverfahren, welches Konflikte zwischen Eingängen und Ausgängen löst. Ein anderer interessanter Aspekt dieser Technologie ist die Anwendung einer neuartigen Token-Tunnelling-Technik, die den Vorteil einer niedrigen Zeitkomplexität erbringt. Besitzt ein Saturn-Switch beispielsweise N Ports, so liegt die Zeitkomplexität bei $O(\sqrt{N})$. Für eine ausführliche Beschreibung des Token-Tunnelling sei auf [Chao00] verwiesen.
Im Folgenden wird nun der Ablauf des zweifachen Reihum-Verfahren (Dual Round-Robin – DRR) beschrieben.
Jeder Eingang besitzt einen Vermittler (Input Arbiter), der nach dem Round-Robin-Prinzip einen seiner N VOQ-Seicher (siehe Abschnitt 2.1.4) auswählt, der nicht leer ist. Nach dieser Auswahl setzt jeder Eingang eine Anfrage (Request) an einen Ausgangsvermittler (Output Arbiter) ab. Jeder dieser Ausgangsvermittler kann bis zu N Anfragen erhalten und wählt ebenfalls nach Round-Robin-Prinzip einen darunter aus. Der Sender dieses Requests erhält darauf ein Bewilligungssignal (Grant) und das anstehende Paket kann übertragen werden. In Abbildung 4 wird der Ablauf des DRR-Algorithmus an Hand eines Beispiel erläutert. Angenommen Eingang (Input) 1 hat Pakete für die Ausgänge (Outputs) 1 und 2 gespeichert und sein Round-Robin-Zeiger (r1) richtet sich momentan auf 1. Im nächsten Schritt setzt er einen Request an Ausgang 1 ab und verschiebt seinen Zeiger auf 2. Auf Ausgangsseite läuft das Verfahren in ähnlicher Weise ab. Betrachtet man z.B. Ausgang 3, so ist dessen Zeiger (g3)

auf 3 gerichtet. Dies hat zur Folge, dass im nächsten Schritt der Ausgangsvermittler einen Grant an Eingang 3 absetzt und danach seinen Zeiger auf 4 richtet.
Aufgrund der beiden voneinander unabhängigen Round-Robin-Verfahren hat dieses Verfahren den Namen Dual-Round-Robin (DRR) erhalten. Abschliessend gilt zu bemerken, dass Saturn eine interessante elektronische Switch-Technologie für den Einsatz in Hochgeschwindigkeitsnetzen darstellt, was durch die Leistungsvergleiche in [Chao00] unterlegt wird.

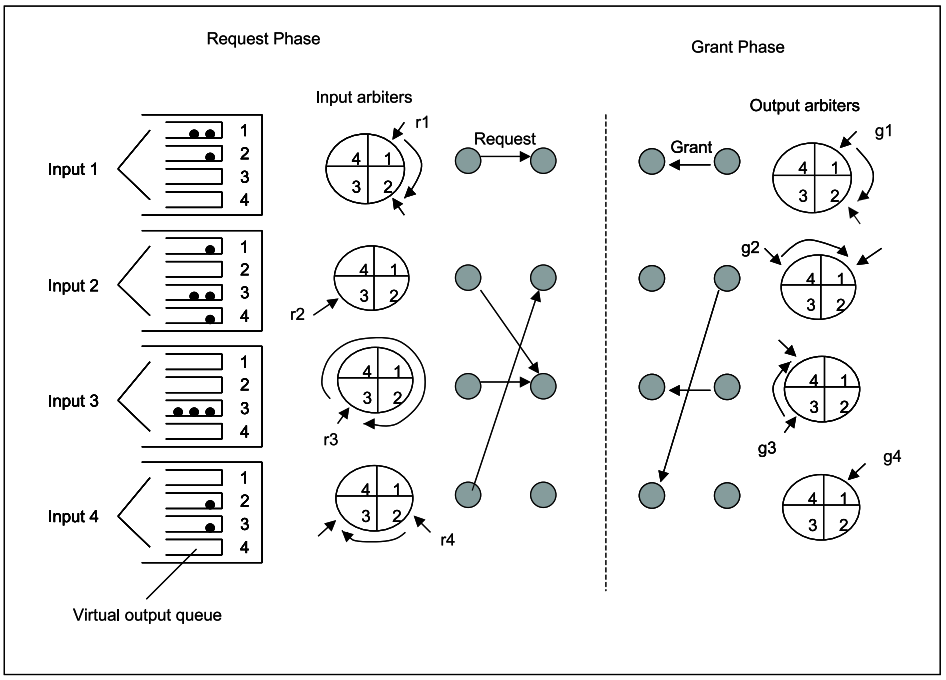


Abbildung 4: Ablauf des Dual-Round-Robin

4 Architekturen für zukünftige optische Netze und Ansätze für optisches Paket-Switching

Wie schon eingangs erwähnt, vollzieht sich die Entwicklung zur Bereitstellung von mehr Übertragungskapazität zwischen den Übertragungsmedien und den vermittelnden Rechnern mit unterschiedlicher Geschwindigkeit. Der rasante Anstieg zu grösserer Bandbreite auf Seite der Übertragungsleitungen wurde durch den Einsatz optischer Technologien ausgelöst. Hierbei nehmen die Verwendung von Glasfaserleitungen und der Einsatz der Wellenlängenmultiplex-Technik (WDM), welche es ermöglicht, flexibel und kostengünstig Anpassungen zur Bereitstellung höherer Bandbreite vorzunehmen, eine entscheidende Rolle ein. Switches und Router, die derzeit noch rein auf elektronische Weise Daten vermitteln, können den häufigen „Upgrades“ zu mehr Übertragungskapazität nicht mehr Schritt halten und stellen daher häufig einen „Flaschenhals“ für die Datenübertragung in den Netzen dar. Im Folgenden soll nun ein Ansatz vorgestellt werden, der mittels optischer Switching-Verfahren eine Lösung dieses Problems aufzeigt, jedoch auch noch einige Nachteile besitzt.

4.1 Optisches Paket-Switching

Das Ziel dieses Verfahrens ist es, einen grossen Teil der zu vermittelnden Daten auf der optischen Schicht zu belassen und somit die hohe Leistungsfähigkeit der WDM-Technik auszunutzen. Dabei kommt es zu einer Trennung der Routing- und Übertragungsmechanismen,

wobei nur die sehr komplexen Headerinformationen elektronisch ausgewertet werden. Die Nutzdaten verbleiben auf der optischen Schicht. Mittlerweile gibt es sogar schon gelungene Versuche dafür, dass ein Teil der Headerauswertung rein auf der optischen Ebene vollzogen werden kann (siehe hierzu auch [Care98]). Diese Entkopplung der Header- von den Nutzdaten erbringt den Vorteil, dass verschiedene Netzprotokolle unterstützt werden können und gleichzeitig die Leistungsfähigkeit der WDM-Technik zum Tragen kommt. Eine entscheidende Rolle für eine erfolgreiche Entwicklung eines optischen Switches für den Einsatz im Internet spielt die Schnittstelle zum IP-Protokoll, welches für darüberliegende Protokolle den Netztyp verbirgt, hier speziell die optische Schicht. Bei der Entwicklung optischer Switching-Verfahren existieren derzeit zwei unterschiedliche Ansätze für die Charakteristik der optischen Pakete:

- Optische Pakete mit fester Länge. Dies bedeutet, dass mehrere Pakete ein IP-Datagramm unter sich aufteilen und somit Fragmentierung und Reassemblierung nötig wird.
- Optische Pakete mit variabler Länge, passend zur Länge der IP-Datagramme.

Das erste Konzept findet sehr viel Forschungsinteresse und bildet daher auch die Grundlage für die weiteren Ausführungen.
Die Dimension der Wellenlänge, in der die Paketdaten codiert sind, ist eine wichtige Grösse im Hinblick auf Konfliktlösung und Übertragungskapazität. Es wird im Folgenden davon ausgegangen, dass sowohl Header- als auch Nutzdaten mit derselben Wellenlänge codiert sind. Der Aufbau eines einfachen optischen Switches (siehe Abbildung 5) gliedert sich in drei Teilbereiche:

- Eine Schnittstelle am Eingang des Switches (Input Interface) synchronisiert die eintreffenden optischen Pakete
- Eine Schaltmatrix (Switching Matrix) leitet die Pakete zu den passenden Ausgängen und löst die dabei auftretenden Konflikte
- Eine Schnittstelle am Ausgang (Output interface), die den Header hinzufügt und Daten regeneriert

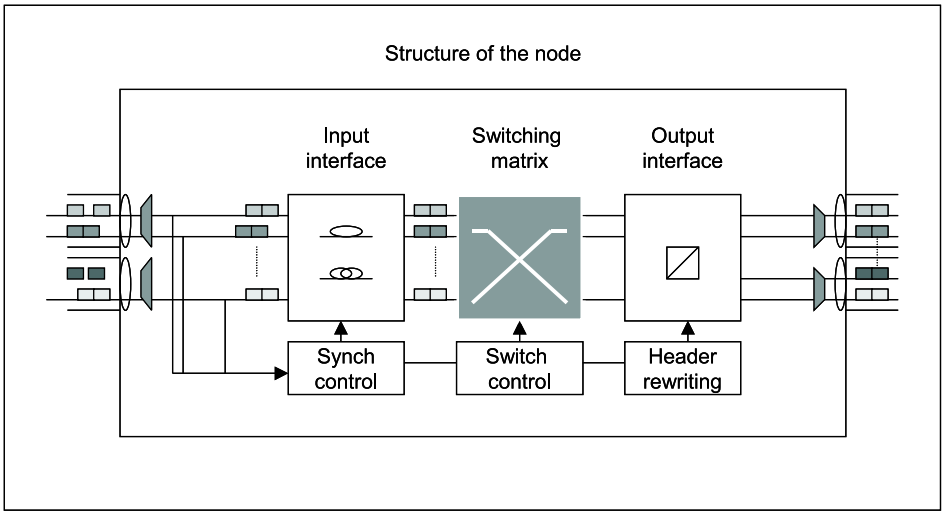


Abbildung 5: Struktur eines optischen Paket-Switches

Die Wahl des Formates für die zu vermittelnden optischen Datenpakete hat grosse Auswirkung auf Verzögerungszeiten und erreichbaren Durchsatz. Im KEOPS-Projekt wurde ein für optisches Switching effizientes Format entwickelt. Details hierzu sind in [HuAn00] zu finden. Obwohl die Pakete bei diesem Konzept eine feste Länge besitzen, ist eine Synchronisierung (Synch control) an den Eingängen nötig, da mehrere Paketströme zur selben Zeit ankommen können. Desweiteren ist auch eine effiziente Pufferung nötig, um auftetende Konflikte zu vermeiden. Vorschläge verschiedener Puffergrössen finden sich ebenfalls in [HuAn00].

4.2 Optisches Netze

Optische Netze haben mit der Anwendung der WDM-Technik begonnen, die auf herkömmlichen Glasfasern zusätzliche Kapazität bereitstellen kann. Heutige WDM-Systeme erhöhen z.B. die Kapazität einer OC3-Faser (Optical Channel – OC), die bisher mit 155 Megabit/s auf ATM benutzt wurde, bis zu 4 Gigabit/s, ohne dabei Zwischenverstärker auf Distanzen bis zu 200 km zu benötigen. Die Komponenten optischer Netze werden dadurch definiert, wie die Wellenlängen im Netz behandelt, übertragen oder implementiert werden. Wichtige Komponenten sind dabei die optischen Verstärker, Multiplexer und Demultiplexer. Ein detaillierter Überblick über verschiedene Multiplex-/Demultiplex-Systeme findet sich in [LiES00].
Die momentan modernste Entwicklung der WDM-Technologie bildet das sog. Dense Wavelength Division Multiplexing (DWDM). Es kombiniert verschiedene Signale auf der gleichen Faser und kann heute zwischen 40 und 80 unterschiedliche Kanäle schaffen. Mittlerweile sind sogar schon 400-500 Kanäle im Gepräch. Durch das Einsetzen der DWDM-Technologie und optischer Verstärker, können in den Netzen eine Vielzahl von Bitraten (z.B. OC-48 oder OC-192) und eine Vielzahl von Kanälen auf einer einzigen Faser bereitgestellt werden. Trotz dieses enormen Fortschritts in der Entwicklung der Netze, gibt es immer noch grosse Schwierigkeiten diese an die Charakteristik des Internetverkehrs basierend auf der Verwendung des IP-Protokolls anzupassen. Hierfür gibt es verschiedene Konzepte, die aber auch einige Nachteile aufweisen:

- IP über ATM über WDM:
  Mittels der ATM-Technik ist es zwar möglich, private virtuelle leicht Netze einzurichten und zu managen, jedoch ist der durch die ATM-Struktur zusätzlich erzeugte Overhead eher ungünstig für den Internetdatenverkehr.

- IP über SDH/Sonet über WDM:
  Die SDH/Sonet-Techinik weist Vorteile bei Ausfall oder Fehler von Übertragungleitungen auf, da in sehr kurzer Zeit auf eine andere Leitung bzw. einen anderen Pfad umgeschaltet werden kann. Diese Eigenschaft kommt jedoch wegen der verteilten Struktur des Internets und der somit inhärenten Ausfallsicherheit weniger zum tragen. Der Nachteil, dass SDH/Sonet nicht für asymetrischen Datenverkehr konzipiert wurde, gestaltet eine Optimierung für den Internetverkehr, der meist diese Asymetrie aufweist, jedoch schwierig.

- IP direkt über WDM:
  Für diese Technik gibt es mehrere Ansätze wie die IP-Pakete in eine Framestruktur gebracht werden können. Allerdings wird dadurch auch ein recht grosser Overhead erzeugt, welcher zu Effizienzverlusten führt.

Eine ausführliche Diskussion über diese drei Techniken lässt sich noch in [LiES00] finden.

5 Schlussbewertung

Diese Arbeit befasste sich mit den neuen Ansätzen, die es im Bereich der Hochgeschwindigkeitsnetze gibt. Es wurde gezeigt, dass die traditionelle Switch-/Router-Architektur den Erfordernissen, welche die rasante Entwicklung der Übertragungskapazitäten mittels Glasfasertechnik und die neuen Dienste des Internets stellen, nicht mehr ganz gewachsen ist. Darauf folgend wurden die neuen Ansätze im Bereich der VOQ-/CIOQ-Switches vorgestellt, aber auch die Schwierigkeiten bei ihrer Entwicklung betrachtet. Die Vorstellung von PRIZMA und Saturn sollte nach Betrachtung der theoretischen Konzepte einen Einblick in die konkrete Chip-Entwicklung für Switches geben. Zum Schluss erfolgte ein Ausblick auf zukünftige optische Netze und Ansätze für optische Switches.
Da die Entwicklung zu immer mehr benötigter Bandbreite und neuen Diensten im Internet auch weiterhin voranschreiten wird, bleiben die Architekturen für Hochgeschwindigkeitsnetze sicherlich auch in Zukunft noch ein wichtiges Thema. Dieses wird noch viel Gelegenheit für Forschung und Entwicklung an den Hochschulen und in der Industrie bieten.

Literatur

[BDEH+01] Werner Bux, Wolfgang E. Denzel, Ton Engbersen, Andreas Herkersdorf und Ronald P. Luijten. *Technologies and Building Blocks for Fast Packet Forwarding*. IEEE Communications Magazine, Januar. 2001.

[Care98] A. Carena. *OPERA: An Optical Packet Experimtental Routing Architecture with Label Swapping Capability*. IEEE/OSA J.Lightwave Tech. vol16. 1998.

[Chao00] Jonathan Chao. *Saturn: A Terabit Packet Switch Using Dual Round-Robin*. IEEE Communications Magazine, Dezember. 2000.

[HuAn00] David K. Hunter und Ivan Andonovic. *Approaches to Optical Internet Packet Switching*. IEEE Communications Magazine, September. 2000.

[Kauf00] Franz-Joachim Kauffels. *Lokale Netze 12.Auflage*. MITP-Verlag. 2000.

[LiES00] Marco Lisanti, Vincenzo Eramo und Roberto Sabella. *Architectural and Technological Issues for Future Optical Internet Networks*. IEEE Communications Magazine, September. 2000.

[McAW96] N. McKeown, V. Ananthram und J. Walrand. *Achieving 100% Throughput in an Input Queued Switch*. IEEE INFOCOM. 1996.

[MiEn00] Cyriel Minkendorf und Ton Engbersen. *A Combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technologie*. IEEE Communications Magazine, Dezember. 2000.

[NoHa00] Ge Nong und Mounir Hamdi. *On the Provision of Quality-of-Service Guarantees for Input Queued Switches*. IEEE Communications Magazine, Dezember. 2000.

Algorithmen für Routing Table Lookup

Maria Vassiliadou

1 Einleitung

In den letzten Jahren die Anzahl der Benutzer, Hosts, Domänen und Netzwerke, die an das Internet angeschlossen sind, explodiert. Der Netzwerkverkehr verdoppelt sich innerhalb weniger Monate. Die Anforderungen nach Multimedia-Anwendungen ergänzen das Problem der hohen Netzlast. Der zunehmende, unkontrollierte Verkehr erfordert drei Faktoren, so dass das Internet weiter gute Dienste anbietet: Link-Geschwindigkeit, Router-data-throughput und Packet-forwarding-Raten.

In diesem Teil des Seminars werden wir uns mit dem dritten Faktor beschäftigen. Die Hauptaufgabe eines Routers ist Pakete weiterzuleiten. Er muss für jedes eintreffende Paket sowohl die Adresse des nächsten Hops als auch den Port, wodurch das Paket geschickt werden soll, kennen. Diese Information ist in einer Tabelle gespeichert, auf die der Router zugreift, um die Ziel-Adresse zu finden. Dies nennt man „address lookup“ und die Tabelle heißt „Routing Tabelle“. Sobald diese Information bekannt ist, kann das Paket weitergeleitet werden.

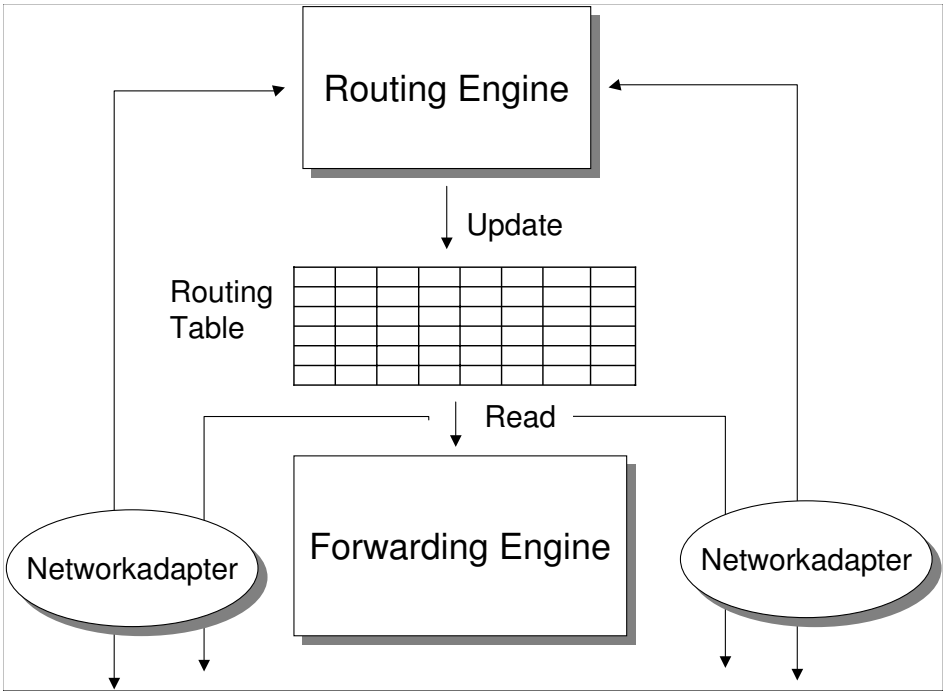


Abbildung 1: Aufgabe eines Routers

1.1 IP-Adressen

Die IP-Adressen in IPv4 sind 32 Bits lang. Traditionell gibt es im Internet 5 Adressklassen: Klasse A, B, C, D und E. Die ersten Bits einer IP-Adresse bestimmen die Klasse. In Abhängigkeit von der Klasse bestimmen die folgenden Bits die Netzwerknummer, eine mögliche Subnetznummer und die Hostnummer. Die Netzwerknummer unterscheidet die einzelnen Netzwerke einer Klasse voneinander, die Subnetznummer unterscheidet die Teilnetzwerke und die Hostnummer die Rechner innerhalb dieses Teilnetzwerkes. Klasse D Adressen sind für Multicast-Pakete reserviert, das sind Pakete, die nur einen Sender aber mehrere Empfänger haben. Klasse E Adressen werden noch nicht verwendet und sind für zukünftige Verwendungszwecke reserviert (vgl. Abb. 2).

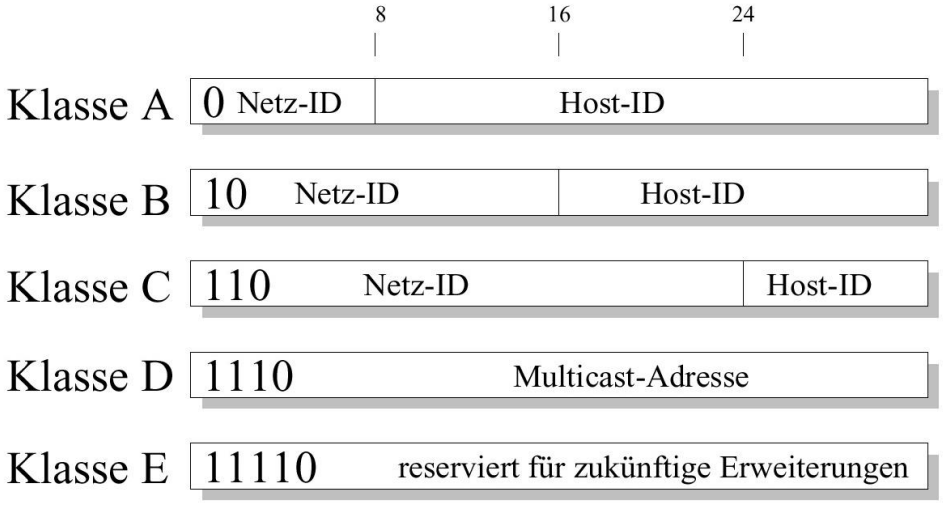


Abbildung 2: IP-Adressen und ihre Klassen

Um in der Routingtabelle nicht einen Eintrag pro IP-Adresse zu benötigen, werden mehrere hintereinander liegende IP-Adressen, die eine identische Next-Hop-Information aufweisen, in einem Eintrag zusammengefasst. Den Bereich der zusammengefassten Adressen nennt man Adresspräfix. Besitzen z. B. die Adressen 1110 und 1111 die gleichen Next-Hop-Informationen, dann werden sie in einem Eintrag mit dem Präfix P1 = 111 zusammengefasst. Dieses beschreibt aber auch die Adressen 1101 und 1100. Die Zuordnung des Präfixes ist also nicht eindeutig, trotzdem werden viele Bereiche von Adressen in nur einem Eintrag zusammengefasst werden. Das Adresspräfix entspricht dann der Netzadresse.

Eine andere Zusammenfassung von Einträgen ergibt sich aus folgender Überlegung. Angenommen alle IP-Adressen, die P = 111 aufweisen, besitzen identische Next-Hop-Informationen und werden in einem einzigen Eintrag in der Routingtabelle zusammengefasst. Dieser Eintrag deckt dann einen bestimmten Bereich A aus dem IP-Adressraum ab (vgl. Abb. 3).

In A befindet sich nun ein kleinerer Teilbereich B von IP-Adressen, die eine von den übrigen Adressen in A abweichende Next-Hop-Information haben. Der Bereich B sei durch das Präfix P2 = 11101 repräsentiert. Auf diese Weise würde der Bereich A in drei Teilbereiche aufgespalten, die durch drei Einträge in der Routingtabelle mit P1a = 11100, P2 = 11101 und P1b = 1111 beschrieben werden müssten, wobei die Einträge mit P1a und P1b identische Informationen enthalten würden. Statt dessen speichert man nur zwei Einträge mit P1 und P2. Wird nun für eine Adresse nach einem Eintrag mit passendem Präfix gesucht, so wählt man den Eintrag mit dem längsten passenden Präfix (LMP). Für Adressen aus dem Bereich B ist P2 der LMP und für Adressen aus den übrigen Teilbereichen von A ist es P1. Auf diese Weise ist eine eindeutige Zuordnung von Adressen zu einem bestimmten Tabelleneintrag sichergestellt.

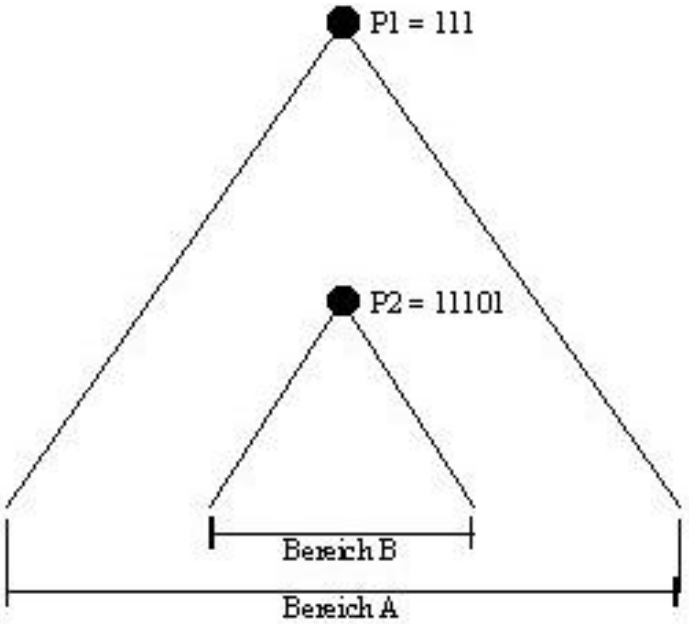


Abbildung 3: P1 ist im Bereich von P2 verdeckt

2 Die Routingtabelle

Das Problem, das im letzten Abschnitt angedeutet wurde, nämlich dass bestimmte Adressen und Adressbereiche mit identischer Next-Hop-Information nicht durch ein gemeinsames Präfix eindeutig beschrieben werden können, führt dazu, dass Tabellen immer größer werden. Heuzutage besitzt eine Routing-Tabelle ungefähr 33000 bis 38000 Einträge. Diese Tatsache mündet in der Forderung nach einer kleinen Lookup-Zeit. Dies könnte erreicht werden, wenn die Größe der Datenstruktur und die Anzahl der Speicherzugriffe minimiert werden. Die Einträge sollten soweit es möglich ist, in enger Beziehung stehen, so dass teure Instruktionen und langsame, ineffiziente bit-Extraktionen vermieden werden. Zu beachten ist, dass eine große Tabelle viel Speicherplatz braucht, folglich auch hohe Kosten aufweist, wenn sie nicht im Secondary-Level-Cache des Prozessors passt. Außerdem gab es lange Diskussionen über das Thema, ob die Realisierung in Software oder Hardware erfolgen soll. Einerseits fragt man sich, wie schnell IP-Routing-Lookups laufen können, um Gigabit-Geschwindigheiten zu erreichen. Andererseits wären die Hardware-Kosten zu hoch.

3 Datenstrukturen für die Routingtabelle

Im nächsten Abschnitt wird erklärt, wie eine Routintabelle realisiert werden kann. Lineare Tabellen über alle 2^{32} möglichen IP-Adressen sind ungeeignet, da sie viel Platz in Anspruch nehmen. Es ist klar, dass eine andere Datenstruktur verwendet werden muss. Besonders gut für diese Aufgabe eignen sich die Binärbäume, da man hier im schlimmsten Fall mit immer noch relativ schnellen Lookup-Zeiten rechnen kann. Hier werden vier verschiedene Datenstrukturen für Routingtabellen vorgestellt.

3.1 Binärbaum mit Unterbäumen

Die Datenstruktur, die im Weiteren vorgestellt wird, passt ganz im Secondary Cache eines Pentium Pro und fast in dem Cache eines Alpha 21164. Sie ist nur 150–160 kbytes groß.

Die Datenstruktur ist ein vollständiger Binärbaum mit gesamter Höhe 32. Tiefe 1 enthält Präfixe der Länge 1, Tiefe 2 der Länge 2 usw. In der untersten Tiefe, Tiefe 32, sind 2^{32} Blätter angehängt; jedes Blatt entspricht einer IP-Adresse. Es wurden aus den 32 Tiefen 3 Ebenen gemacht, nämlich Ebene 1 umfasst Tiefe 0 bis 16, Ebene 2 von 17 bis 24 und die letzte von 25 bis 32 (vgl. Abb. 4).

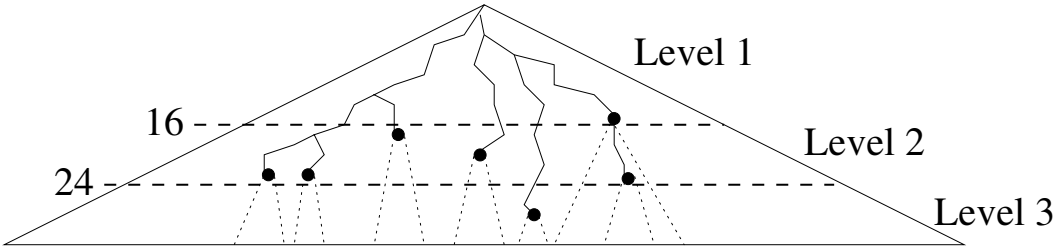


Abbildung 4: Die drei Ebenen

3.1.1 Ebene 1

Die erste Ebene ist ein Knoten mit 1–64 bits Kindern. Jeder Knoten ist folglich ein Paar aus (Präfix, Next-Hop). Ein Bit-Vektor wird benutzt, um den Schnitt in Tiefe 16 zu represäntieren (Abb. 5). Dieser ist 8 kbytes groß, da 1 bit jedem Knoten entspricht (2^{16} bits = 64 kbits = 8 kbytes). Die ersten 16 Bits werden als Index benutzt. Im Vektor wird ein Bit gesetzt, wenn:

- ein Knoten angibt, dass der Baum unter den Schnitt weiterläuft. Dieser heißt „root head“ (Wurzel).
- ein Blatt in Höhe 16 oder weniger liegt. Er heißt dann „genuine head“ (echter Kopf).
- die Bits, die 0 sind heißen „members“ (Mitglieder) und sind von einem Blatt überdeckt, welches in kleinere Höhe als 16 liegt.

Der Bit-Vektor ist in Bit-Masken der Länge 16 unterteilt. Es gibt $2^{12} = 4096$ solche. Die Informationen der „genuine heads“ werden als Index in der Next-Hop-Tabelle gespeichert. „Members“ haben denselben Next-Hop wie der größte „head“ kleiner als der „member“. Die „root heads“ brauchen auch einen Index, der in einem Teilbaum gespeichert wird, welcher zum Unterbaum zeigt. Die „head“ Information ist in 16 bit-Zeiger kodiert und in einem Vektor gespeichert. 2 Bits davon geben die Art des Zeigers an und die restlichen 14 bilden entweder einen Index in die Next-Hop-Tabelle oder in einen Vektor, der den Teilbaum enthält. Es gibt soviele Zeiger, wie die gesetzten Bits in einer Bit-Maske.

Wie findet man nun diese Zeiger? Man implementiert eine neue Datenstruktur, die aus zwei Vektoren besteht: einem Vektor aus Code-Wörtern, soviele wie die Bit-Masken und einem aus Basis-Indices, einer pro vier Code-Wörtern. Ein Wort ist 16bit lang: 10bits für den Wert und 6 für den Offset. Der Offset gibt an, wieviele Zeiger zurückgestzt werden müssen, um den ersten Zeiger zu finden, der dieser Maske entspricht (vgl. Abb. 6).

Der Basis-Index gruppiert eigentlich vier Wörter, da die Anzahl der gesetzten Bits möglicherweise so groß sind, dass sie nicht mit den 6 Bits des Offsets repräsentiert werden kann. Also sind die ersten 12 Bits der IP-Adresse ein Index auf das zugehörige Wort und die ersten 10 ein Index auf den Vektor der Basis-Indices. Dieser Wert ist ein Index auf die Tabelle, der

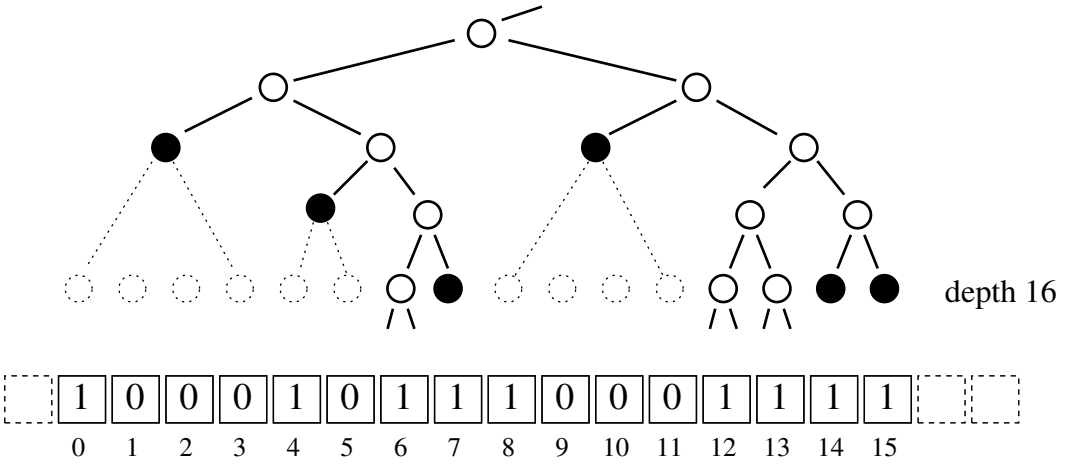


Abbildung 5: Teilbaum mit zugehörigem Bitvektor

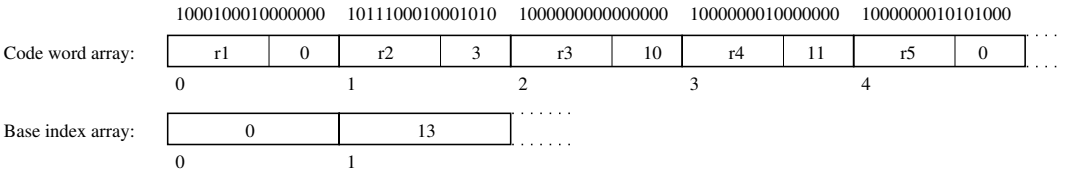


Abbildung 6: Bitmasken vs CodeWörter und BasisIndices

die Bit-Nummer in der IP-Adresse auf die Zeiger-Offsets wiederspiegelt. Die Maptabelle sieht ähnlich aus; die Bitnummern innerhalb einer Bit-Maske werden auf 4 Bit Offsets gespiegelt.

Die erste Ebene benötigt 7 Bytes für die Suche, nämlich 16 Bits für das Code-Wort, 16 Bits für den Basisindex, 4 Bits für die Maptabelle und für die Zeiger 16 Bits.

3.1.2 Ebene 2 und 3

Die zweite und dritte Ebene bestehen dementsprechend aus Unterbäumen der Tiefe 8 und kann $2^8 = 256$ „heads“ enthalten. Es gibt 3 Kathegorien von Unterbäumen. Der Unterbaum heißt:

- „sparse“(dick), wenn er 1 bis 8 „heads“ hat.
- „dense“(dicht besetzt), wenn er 9 bis 64 „heads“ hat.
- „very dense“(sehr dicht besetz), wenn er 65 bis 256 „heads“ hat.

„dense“ und „very dense“ Unterbäume werden wie in der ersten Ebene gesucht. Im „sparse“ Unterbaum werden allerdings die Werte in absteigender Reihenfolge platziert. Man untersucht das vierte Element und entscheidet sich dann, ob das gewünschte Element in der oberen oder unteren Hälfte liegt. Dann folgt linear der Index des Elementes und der Pointer mit diesem Index wird extrahiert. Man braucht maximal 7 Bytes für diese Suche.

Da die Suche nach dem „Longest Matching Prefix“ in den einzelnen Ebenen in konstanter Zeit stattfindet, werden insgesamt maximal 3 Schritte benötigt. Der Aufwand beträgt also

$O(W/8)$, wobei W = Länge der IP-Adresse. Bei einem Umfang der Tabelle von 30000–40000 Einträgen beträgt die Suchzeit auf einem Pentium Pro mit 200 MHz 505 ns. Das entspricht einem Durchsatz von mindenstens 2 Mio IP-Paketen pro Sekunde.

Da die beschriebene Datenstruktur als reine Suchtabelle für die Anwendung in Hochleistungsroutern entwickelt wurde, sind Einfüge- und Löschoperationen nicht vorgesehen. Für diese Aufgaben ist eher eine zweite Tabelle im Router gedacht, die bei jeder Änderung aktualisiert und gegen die alte ausgetauscht wird.

3.2 Binärsuche in Hashtabellen

Dieses Schema basiert auf Hashing, Binärsuche und Vorberechnung. Es wird für jede Präfixlänge eine Hashtabelle angelegt (vgl. Abb. 7).

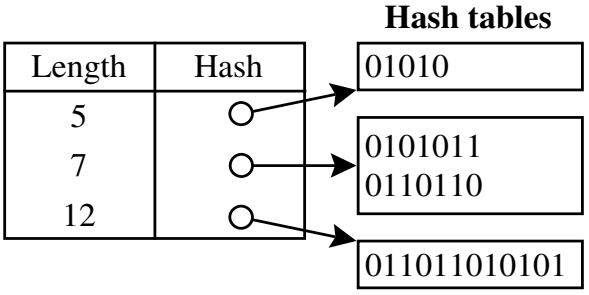


Abbildung 7: Hashtabelle für jede Präfixlänge

Diese liegen dann sortiert in einer Tabelle, die die Struktur eines Tries wiederspiegelt. Um das „Longest Matching Prefix“ zu finden, macht man eine Binärsuche in der Tabelle. Da die Suche relativ komplex werden kann, werden „Markers“ hinzugefügt. Diese vereinfachen die Suche mach dem „Longest Matching Prefix“, da sie den Weg zum zunächst längsten Präfix zeigen, nämlich zur unteren Hälfte der Tabelle. Wenn die Suche misslingt, sucht man in der oberen Hälfte weiter. In den „Markers“ wird das aktuelle Präfix gespeichert. Die Datenstruktur kann bessere Ergebnisse liefern, wenn man sie mutiert. Die Idee, die dahinter steckt, ist folgende: Immer wenn man ein Präfix findet, braucht man nur den Unterbaum weiter binär zu durchsuchen. Also werden diese Teilbäume vorberechnet und in die Datenstruktur hinzugefügt. Das Schema wird in der Abb. 8 verdeutlicht.

Sucht man z. B. das Präfix G einer IP-Adresse, dann fängt man beim ersten Baum an. Die Ebene 16 ist erreicht, man findet E. E enthält eine Beschreibung eines neuen Baumes, Baum 2. Man wird dann F, welches zum nächsten Baum führt, der nur eine Länge hat, G. Der Baum ist von 32 auf nur 5 Längen mutiert. Man braucht dafür maximal $\log_2 H$ Schritte, wobei H = Anzahl der Hashtabellen. Für die Suche werden maximal 5 Speicherzugriffe benötigt, also ist der gesamte Aufwand $O(\log W)$. Die Einfüge- und Lösche-Operationen sind aber sehr komplex, da der Baum nicht ausbalanciert ist. Daher wäre es besser, wenn man die Änderungen gruppiert und die Datenstruktur neu bildet. Das würde eine Komplexität von $O(N \log W)$ haben.

Bei 33000 Einträgen wird eine maximale Suchzeit von 450 ns erfordert, dennoch braucht die Tabelle einen Speicher von 1,2 bis 1,4 Mbytes.

3.3 Dynamische Präfix Tries

Die Datenstruktur, die jetzt vorgestellt werden soll, ist eine besondere Struktur von Bäumen. Diese heißt „Trie“. Ein Knoten besteht aus 6 Komponenten (Abb. 9):

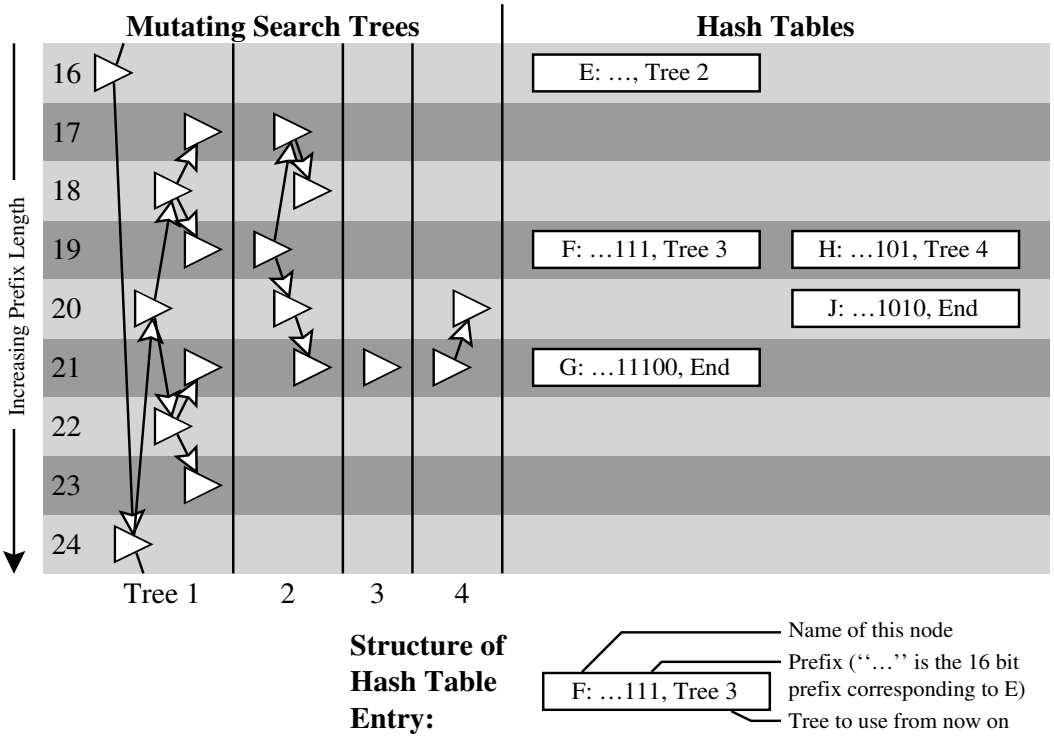


Abbildung 8: Mutierte Binärsuche

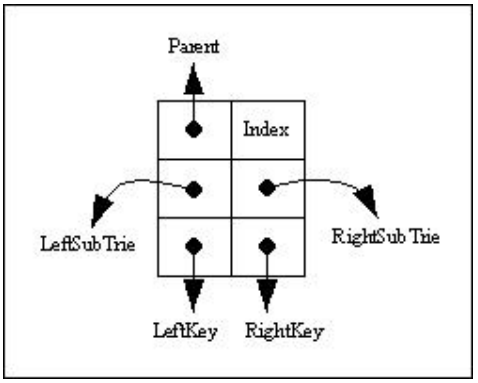


Abbildung 9: Knotenstruktur

- Index(n) gibt die Bitposition an ,ab der die beiden Präfixe des Knotens einen Unterschied aufweisen.
- LeftKey(n) bildet das Präfix, das an dieser Stelle eine Null enthält
- RightKey(n) bildet das Präfix, das an dieser Stelle eine Eins hat
- Parent(n) ist der Zeiger, der auf den übergeordneten Knoten verweist
- LeftSubTrie(n) ist der Zeiger, der auf den linken Unterbaum verweist
- RightSubTrie(n)ist der Zeiger, der auf den rechten Unterbaum verweist

Jeder Knoten des Tries, der kein Blatt ist, ist die Wurzel des Unterbaumes. Somit ist der Präfix in einer solchen Wurzel ein Präfix der unterhalb liegenden Präfixe. Außerdem besitzt ein Trie folgende Eigenschaften:

- Für jeden Knoten n ,Keys(n) beschreibt die Menge der Schlüssel, die sich in n und seinen Teilbäumen befinden. Die Breite dieser Schlüssel ist mindenstens Index(n) und ihre Länge Index(n) + 1.

- Alle Schlüssel in Keys(n) haben dasselbe Präfix mit Länge Index(n) Bits.

- Die Schlüssel im Teilbaum von n teilen ein Präfix mit mindenstens einer Breite von Index(n) + 1 bits.

- Index(n) heißt maximal, wenn ein Blatt nur einen Schlüssel k hat, mit Index(n) = |k|.

- Jeder Knoten hat mindenstens einen Schlüssel oder zwei nichtleere Teilbäume.

- Wenn sowohl der linke(rechte) Schlüssel als auch der linke(rechte) Teilbaum vom Knoten n nicht leer sind:

  |LeftKey(n)| = Index(n) (|RightKey(n)| = Index(n))

- Wenn der linke(rechte) Schlüssel von n leer ist und der linke(rechte) Teilbaum nicht, dann hat der Teilbaum mindenstens zwei Schlüssel.

- Für einen gegebenen Knoten n , der linke(rechte) Schlüssel und alle Schlüssel im linken(rechten) Unterbaum haben eine Null(Eins) Bit in der Position Index(n).

- Die Größe von Präfix(n) nimmt zu, wenn n tiefer im Trie liegt.

- Jedes Pfad ist eindeutig und terminiert beim Blatt, welches mindenstens einen Schlüssel enthält.

Im Gegensatz zum Binärbaum steht die Anzahl der Einträge fest, d. h. sie hängt nicht von der Reihenfolge der Einfüge- und Löschoperationen ab. Also ist eine maximale Tiefe des Tries garantiert.

3.3.1 Einfügen und Löschen

Um einen neuen Schlüssel einzufügen, muss man zuerst in den Blättern die Länge des „Longest Matching Prefixes“ finden, die gleich mit der des Schlüssels ist. Dann durchläuft man den Baum bis man zu einem Knoten mit kleinerer Breite als die des Schlüssels gelangt und fügt ihn da ein (Abb. 10). Wenn ein Schlüssel nicht mehr gebraucht wird, wird er einfach entfernt. Knoten ohne Schlüssel haben nur einen Teilbaum. Bei diesem verbindet man den Teilbaum mit dem Eltern-Knoten bevor man diesen entfernt.

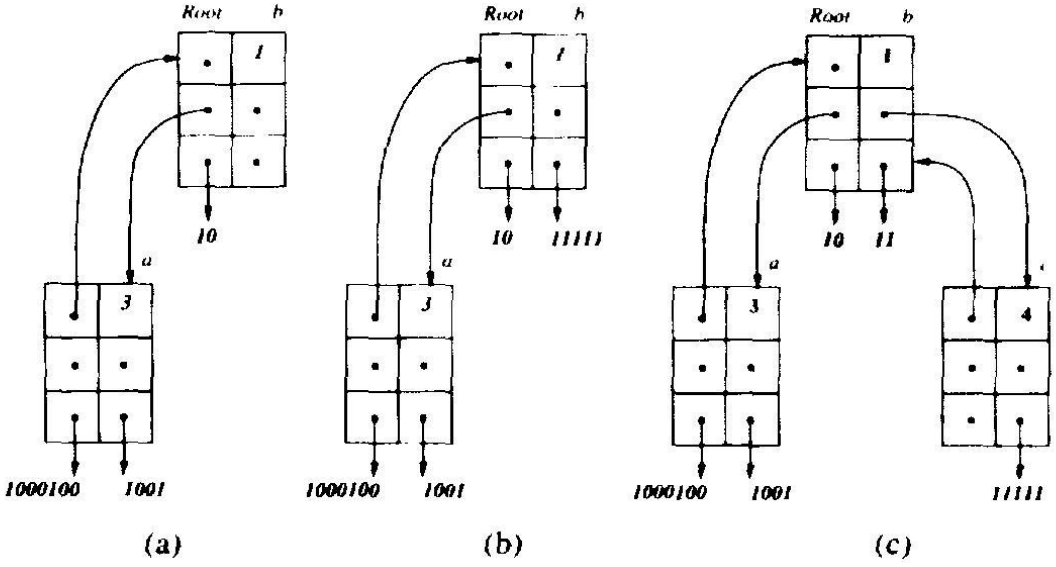


Abbildung 10: Einfügen(a)Einfügen von 10;(b)Einfügen von 11111;(c)Einfügen von 11

3.3.2 Komplexität

Das Einfügen und Löschen hängen nicht von der Größe der Datenstruktur ab. Beide Operationen laufen in linearer Zeit. Die Suchzeit nach dem „Longest Matching Prefix“ beträgt $O(2W)$. Es wird zuerst im Baum an einem Ast bis zu den Blättern hinabgewandert. Der Ast wird aufgrund der Bits in der IP-Adresse bestimmt, d. h. trifft man auf eine Null, wird die Suche im linken Unterbaum fortgesetzt, trifft man auf eine Eins, im rechten. Kommt man bei einem leeren Unterbaum an, wandert man denselben Ast nach oben und vergleicht dabei die Präfixe mit der IP-Adresse. Das erste passende Präfix ist dann das LMP und die Suche ist beendet. Vorteilhaft ist daher, dass beim Hinabwandern im Baum keine Vergleiche stattfinden. Beim Rückweg werden dann deutlich wenige Vergleiche benötigt.

3.4 Multibit Tries

Multibit Tries sind Binärbäume, die zuerst optimiert wurden. Die Idee, die dahinter steckt ist, viele Bits gleichzeitig zu untersuchen, um die Speicherzugriffe zu reduzieren. Bei IPv4 wären dann nur 8 Speicherzugriffe nötig, wenn man gleichzeitig 4 Bits sucht und nicht 32. Diese Datenstruktur weist auf zwei weitere Optimierungsmöglichkeiten, nämlich auf die Wege-Komprimierung („path compressed tries“) und die Ebenen-Komprimierung („Level compressed tries“). Bei der Ersten werden die Wege optimiert, d. h. alle Knoten, die eine einzige Verzweigung besitzen wurden durch einen „Skip-Wert“ ersetzt. Dieser besagt, wieviele Ebenen weggelassen werden. Die Anzahl der weggelassenen Ebenen ist dann die Anzahl der Bits, die man beim Vergleich mit der IP-Adresse auslassen muss. Ein „path compressed trie“ hat genau $2k$ Knoten in der k-ten Ebene. Bei der Ebenen-Komprimierung werden die x höchsten, vollständigen Ebenen zu Verzweigungen eines einzigen Knotens gemacht. Dadurch besitzt die Datenstruktur wesentlich weniger Ebenen und die Suche kann erheblich beschleunigt werden (Abb. 11).

Dieser Vorgang wird rekursiv für jede Ebene wiederholt. Um Speicherplatz zu sparen, sind alle Knoten in einem Vektor gespeichert; zuerst die Wurzel, dann alle Knoten der Ebene2, dann die der Ebene3 usw. Interne Knoten dürfen allerdings keine Präfixe haben. Dagegen besitzt jedes Blatt eine lineare Liste mit Präfixe. Die Suche läuft wie im DP-Trie bis auf die Tatsache,

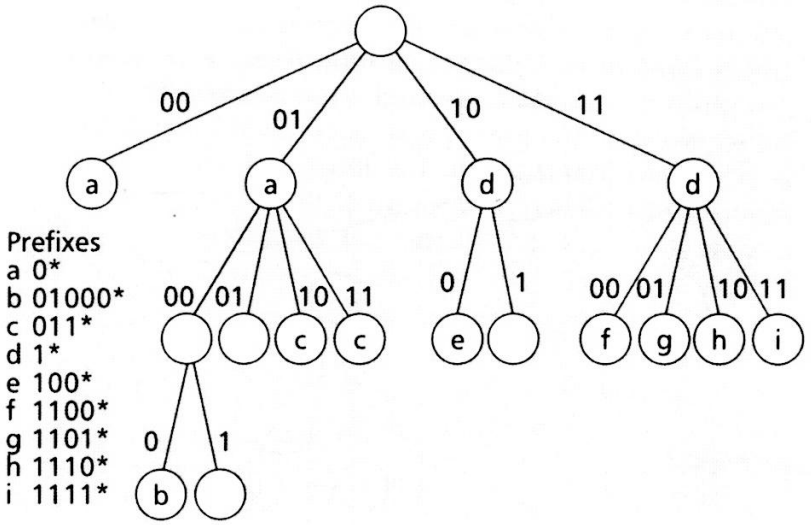


Abbildung 11: Multibit Trie mit variabler Anzahl Bits

dass ein expliziter Vergleich gemacht werden muss, wenn man das Blatt erreicht hat. Wenn das Präfix nicht gefunden wird, muss man in der Liste der Präfixe suchen (Abb. 12).

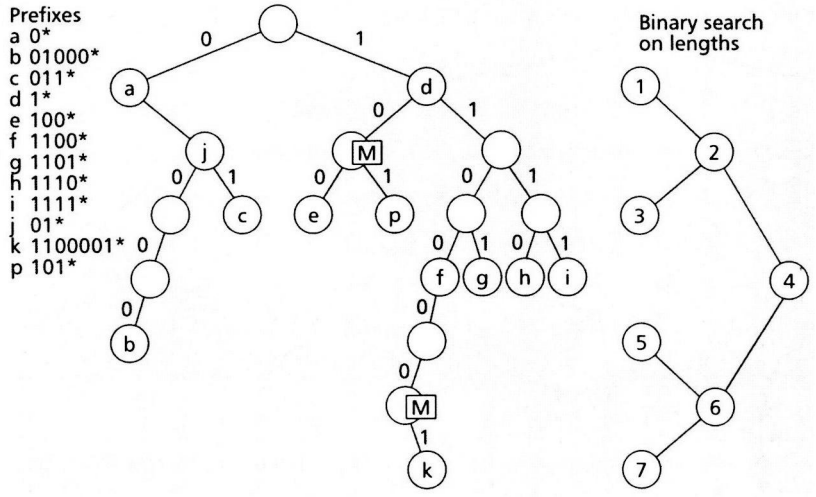


Abbildung 12: Binärsuche

Daher ist eine Erweiterung der Datenstruktur sehr schwierig. Allerdings beträgt der Aufwand für die Suche $O(W/k)$ und für die Speicherzugriffe $O(2kNW/k)$.

4 Vergleich

In der folgenden Tabelle sind kurz die wichtigsten Eigenschaften der vorgestellten Datenstrukturen zusammengefasst. Ihre tatsätzliche Effizienz kann jedoch nicht vorgestellt werden, da sie von der jeweiligen Implementierung abhängt.

	Binärbaum mit Teilbäumen	DP-Trie	Binärsuche in Hashtabellen	Multibit-Tries
Datenstruktur	Trie	Trie	Hashtabelle	Trie
Komplexität d. Suche	$O(W/8)$	$O(2W)$	$O(log_2 W)$	$O(W/k)$
Aufwand f. Einfügen/Löschen	hoch	niedrig	sehr hoch	hoch
Speicherverbrauch	sehr gering	mittel	hoch	mittel
Realisierung	Software	Software	Software	Software
Unterstützung v. IPv6	ja	ja	ja	ja

Der Binärbaum mit Unterbäumen weist eine einfache Struktur auf, die wenig Speicherplatz braucht und somit ganz in den Secondary Cache eines Prozessors passt. Die Suchfunktion findet sehr schnell statt aber die Einfüge- und Lösche-Operationen sind sehr aufwendig. Allerdings könnte diese Datenstruktur in der Hardware implementiert werden. Mit Pipelining wäre dann die Lookup-Zeit gleich mit dem Suchaufwand in der ersten Ebene. Weitere Modifikationen erlauben auch die Unterstützung von größeren Adress wie IPv6.

Die Hashtabelle bietet auch eine schnelle Suchfunktion, wie der Binärbaum aber auch komplexe Grundoperationen. Diese Komplexität könnte verringert werden, wenn die Datenstruktur nach mehreren Änderungen mit Hilfe einer Referenztabelle neu aufgebaut wird. Der Nachteil ist, dass die Datenstruktur ungefähr 1,2 MByte groß ist, was dem zehnfachen Verbrauch der vorigen Lösung entspricht. Trotzdem würden die meist benutzten Hashtabellen für die Längen 8, 16 und 24 fast im Cache passen. Die Implementierung kann sowohl in der Software als auch in der Hardware realisiert werden.

Der DP-Trie bietet eine Suchfunktion mit garantierter maximaler Laufzeit und gleichzeitig effizienten Operationen. Da die Suche bis zu 64 Schritte braucht, eignet er sich eher für Router mit niedrigen Übertragungsraten. Eine Beschleunigung der Suche durch eine Realisierung in Hardware ist auch möglich, da der Suchalgorithmus sehr einfach ist.

In der letzten Datenstruktur spielt die Anzahl der gewählten Bits die Hauptrolle. Man muss einen optimalen Bereich wählen, um die Anzahl der Speicherzugriffe zu reduzieren. Ein Vorschlag dafür ist, die Präfixe, die am meisten vorkommen, zu optimieren.

5 Zusammenfassung

In diesem Beitrag wurden bereits vier verschiedene Datenstrukturen für die Implementierung der Routingtabelle beschrieben. Je nach Einsatzgebiet hat jede einzelne Vorteile und Nachteile. Allerdings wurde ein Ansatz für die Realisierung in Hardware nicht diskutiert.

Literatur

[DBCP97] M. Degermark, A. Brodnik, S. Carlsson und S. Pink. Small Forwarding Tables for Fast Routing Lookups. In *Proceedings of ACM SIGCOMM '97*, September 1997, S. 3–14.

[DoKN96] W. Doeringer, G. Karjoth und M. Nassehi. Routing on Longest-Matching Prefixes. *IEEE/ACM Transactions On Networking* 4(1), Februar 1996.

[Horn98] B. Hornburg. IPv6-Routing. Seminarvortrag, Februar 1998.

[RSBD01] M. A. Ruiz-Sanchez, E. W. Biersack und W. Dabbous. Survey and Taxonomy of IP Adress Lookup Algorithms. *IEEE Network*, March/April 2001.

[WVTP97] M. Waldvogel, G. Varghese, J. Turner und B. Plattner. Scalable High Speed IP Routing Lookups. In *Proceedings of ACM SIGCOMM '97*, September 1997, S. 25–36.

Paketklassifikation

Johann Costin Mihutoni

Kurzfassung

Immer mehr Pakete werden per Internet von einen Computer zu einem anderen geschickt. Bis die Pakete ihr Ziel erreichen, gehen sie durch verschiedene Routers. In jedem Router werden die Pakete mit Hilfe einer Filterdatenbank (Datenbank mit Regeln) gematcht (verglichen). Das nennt man Paketklassifikation. Man braucht Paketklassifikation, um die Quality of Service (Dienstgüte) zu garantieren. Paketklassifikation enthält auch Firewallfunktionen, die für das Filtrieren von Attacken auf Ports (wie: FTP, Telnet, etc) und von anderen böswilligen Programmen zuständig sind. Damit die Pakete ihr Ziel nicht zu spät erreichen, wurden schnelle und effiziente Algorithmen entworfen. Die Algorithmen können entweder per Software oder per Hardware implementiert werden. Je nach Applikationsart kann der passende Algorithmus gewählt werden.

1 Einleitung

Das Thema behandelt, wie schon oben erwähnt, Paketklassifikation. Im nächsten Kapitel wird mit der Theorie begonnen. Es wird kurz beschrieben was für Informationen ein Paket enthält. Es werden ein paar Maßanalysen angegeben mit dem man die Performanz eines Klassifikators untersuchen kann. Zum Beispiel: wieviel Platz die Filterdatenbank besetzt, wie lange das Filtern dauert, wie leicht die Datenbank erweitert werden kann und andere Angaben. Danach folgt eine Liste von Befehlen, die oft in Klassifikationsalgorithmen verwendet werden.

Die dritte Sektion enthält Hardwarearchitekturen: Ternary CAM, Bitmap-Intersektion, ClassiPI Architektur. Sie sind schneller als die Softwarealgorithmen, aber dafür veralten sie auch schneller, weil sich die Hardwaretechnik schneller als die Software weiterentwickelt. Die Software kann auf einen neuen Computer installiert werden, während die Karte mit der alten, langsameren Technologie bleibt. Die vorgestellte Hardware ist die ClassiPI Architektur. Die Komponente, mit der diese Hardware gebaut ist, werden auch beschrieben.

Im darauf folgende Kapitel werden Algorithmen angegeben, die für die Softwareimplementierung verwendet werden. Man kann sie in drei verschiedene Kategorien unterteilen:

- Grunddatenstruktur: lineare Suche, hierarchische Tries, set-pruning Tries.
- Geometrisch basiert: Grid-of-tries, AQT, FIS.
- Heuristisch: RFC, hierarchical cuttings, tuple-space-search.

Das 5. Kapitel schließt das Thema mit einer Zusammenfassung ab. Es werden auch Bücher empfohlen, die Einsteigern und Fortgeschrittenen, die einen Internet Firewall einrichten möchten, von Nutzen sein werden.

2 Theorie

In Abbildung 1 kann man sehen, welche Informationen ein Paket enthält. Die Daten, die oft für die Paketklassifikation verwendet werden, sind: die beiden IP Adressen (Quell- und Zieladresse), der TCP Protokolltyp, der Quell- und Destinationsport von TCP.

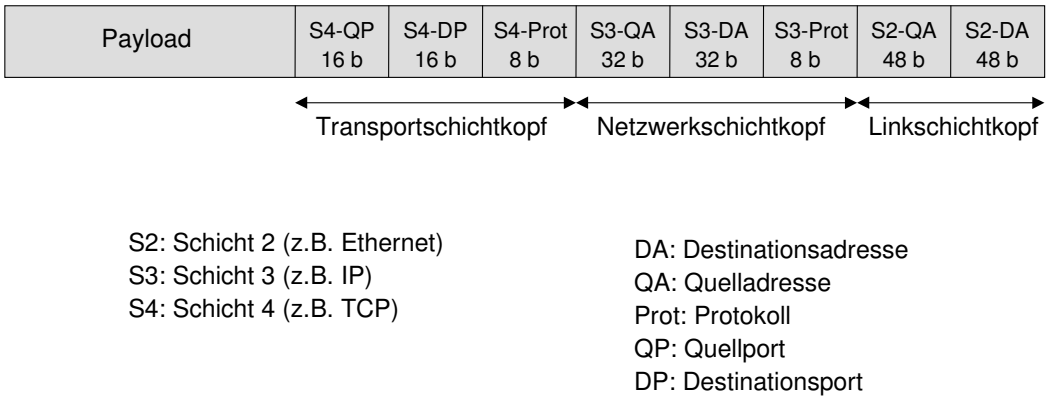


Abbildung 1: Kopffeldinformationen, die für die Paketklassifikation benutzt werden.

2.1 Komplexitätsangaben

Mit Hilfe dieser Angaben kann man die Güte eines Klassifikators beschreiben. In diesen Abschnitt gibt n die Anzahl der Regeln an.

Platzkomplexität (Space Complexity S): $\Theta(f(n))$.

S beschreibt den Speicherplatzbedarf, der gebraucht wird, um die Regeldatenbank zu speichern.

Zeitkomplexität (Time Complexity T): $\Theta(f(n))$.

Diese Größe definiert die maximale Anzahl der Schritte oder Zyklen die benötigt werden, um eine Klassifikation zu vollenden. Um die Zeit zu verkleinern, kann man parallele Vergleichsalgorithmen implementieren. Zum Beispiel: lineare Suche $T = \Theta(n)$, binäre Baumsuche $T = \Theta(\log n)$, Parallele Suche $T = \Theta(1)$.

Leistungskomplexität (Power Complexity P): $\Theta(f(n))$.

Sie definiert die maximale Anzahl der Schritte oder Zyklen, die benötigt werden, um eine Klassifikation durchzuführen. Hier konzentriert man sich auf die Menge der Operationen, die für die Klassifikation benötigt werden. Zum Beispiel: lineare Suche $P = \Theta(n)$, Hash-Tabelle $P = \Theta(b)$ mit $b =$ Behältergroße.

Aktualisierungskomplexität (Update Complexity U): $\Theta(f(n))$.

U gibt an, wie schwierig ist es die Filterdatenbank zu aktualisieren: beim Einfügen, Löschen oder Verändern von Regeln.

Regelnkomplexität (Rule Complexity R): $\Theta(f(n))$

Sie beschreibt die Anzahl der Operationen, die pro Feldregel unterstützt werden. Beispiel: falls man nur den Gleichheitsmatchoperator verwendet, dann gilt: $R = 1 * F$ mit F = Feldgröße. Wenn man neben den Gleichheitsmatchoperator auch ein Maskingoperator benutzt, dann ist $R = 2 * F$, $R = 4 * F$ falls man noch die Operatoren $>$, $<$ hinzufügt.

Optimale Komplexitätsangaben: S = niedrig, T = niedrig, P = niedrig, U = niedrig, R = hoch. In Allgemeinen hat jeder Algorithmus mindestens eine schwache Komplexitätsgröße. Bei jedem Algorithmus kann man die Koplexitätsangaben bestimmen. Nach diesen Angaben kann der geeignete Algorithmus für die eigenen Implementation ausgewählt werden. Nähere Informationen findet man in [IyKS01].

2.2 Operatoren

Angenommen es gibt K Kopffelder, die für das Filtern relevant sind. Jeder Filter F ist ein K Tupel $(F[1], F[2], ..., F[K])$ wobei $F[i]$ ein Bitstring von variabler Präfixlänge ist. Diese Bits enthalten folgende Informationen: IP Destinationsadresse (32 Bits), IP Quelladresse (32 Bits), Protokolltyp (8 Bits), Portnummern für Destination und Quelle (je 16 Bits), und andere Protokollflaggen; siehe Abbildung 1. Eine Datenbank besteht aus N Filtern (Regeln) $F_1, F_2, ..., F_N$. Jeder dieser F_j hat obige Eigenschaften. $F_j[i]$ ist eine Spezifikation vom i-te Feld, man nennt sie auch i-te Dimension. Jeder Filter F_j ist assoziiert mit einer Direktive in der Datenbank, z. B.: ein Firewall kann entweder einen Paket akzeptieren oder ablehnen. Ein Paket P matcht (passt) mit Filter F, wenn es für alle Paketfelder i gilt: $P[i]$ matcht mit $F[i]$. Um ein Paket zu vergleichen, bei N Filtern und bei Dimension K, braucht man eine Speichergröße von $O(N^K)$ und die Zeitkomplexität beträgt $O(N)$. Betrachte 3. Sektion in Artikel [SrSV99].

Sequenzbeschreiber (Π): $\Pi = \Pi_1 \bullet \Pi_2 | (Bool)?\Pi_1 : \Pi_2 | switch\{(Integer) : (\Pi_1, \Pi_2, ..., \Pi_n)\}$

Ist eine Serie von Klassifikationsoperationen.

Sequenzoperator ($\bullet$):

Man kann mehrere Sequenzbeschreiber hintereinander ausführen. $\Pi_1 \bullet \Pi_2$, Π_1 wird vor Π_2 ausgeführt. Die Komplexitätsangaben:

$$\begin{array}{ll} S = S(\Pi_1) + S(\Pi_2) & S = \text{Space (Speicherplatz)} \\ T = T(\Pi_1) + T(\Pi_2) & T = \text{Time (Zeit)} \\ P = P(\Pi_1) + P(\Pi_2) & P = \text{Power (Leistung)} \end{array} \quad (1)$$

Bedienungssequenzoperator: $(Bool)?\Pi_1 : \Pi_2$

Wenn der *Bool*-Operator wahr ist, dann wird Π_1 ausgeführt, ansonsten wird Π_2 ausgeführt. Die Komplexitätsangaben:

$$\begin{array}{l} S = S(Bool) + S(\Pi_1) + S(\Pi_2) \\ T = T(Bool) + max\{T(\Pi_1), T(\Pi_2)\} \\ P = P(Bool) + max\{P(\Pi_1), P(\Pi_2)\} \end{array} \quad (2)$$

Switch-Case Operator: $switch\{(Integer) : (\Pi_1, \Pi_2, ..., \Pi_n)\}$

Dieses Operator wird oft für Fallunterscheidungen verwendet. Integer ist eine Funktion oder eine Variable, die eine positive natürliche Zahl liefert, mit dessen Hilfe der entsprechende Operator ausgeführt wird. Für $Integer = 1$ wird Π_1, für $Integer = 2$ wird Π_2 ausgeführt, usw. Die zugehörigen Komplexitätsangaben werden wie gefolgt ausgerechnet:

$$\begin{aligned} S &= S(Integer) + \sum_{k=1}^{n} S(\Pi_k) \\ T &= T(Integer) + max\{T(\Pi_1), T(\Pi_2), ..., T(\Pi_n)\} \\ P &= P(Integer) + max\{P(\Pi_1), P(\Pi_2), ..., P(\Pi_n)\} \end{aligned} \tag{3}$$

Unter Artikel [IyKS01] erfahren Sie mehr über die verwendete Funktionen.

In den nächsten zwei Kapiteln wird als Beispiel immer die gleiche Tabelle (Abbildung 2) mit 6 Regeln (Filterzahl $N = 6$) und die dazugehörige zweidimensionale Filterdatenbank ($K = 2$) präsentiert.

Mögliche Zuordnung der Dimensionen:

- 1. Dimension enthält einen Präfix der IP Destinationsadresse (32 Bits)
- 2. Dimension enthält einen Präfix der IP Quelladresse (32 Bits)

Rule	F[1]	F[2]
R1	00*	00*
R2	0*	01*
R3	1*	0*
R4	00*	0*
R5	0*	1*
R6	*	1*

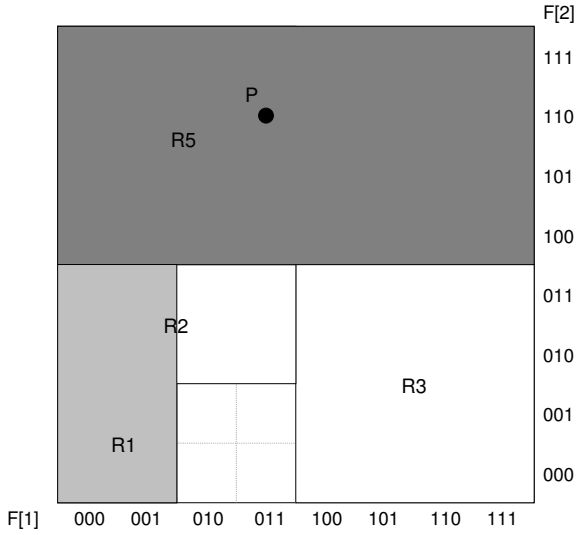


Abbildung 2: Klassifikator mit Regeln und 2 Dimensionen. Rechts von der Tabelle steht einen Bild, der anzeigt, welche Bereiche die Regeln überdecken. Punkt P (011,110) stellt ein Paket dar.

3 Hardware

3.1 Ternary CAM: [GuMc01]

TCAM speichert jedes W-Bit Feld als ein Paar (Wert, Maske), z. B.: $W = 5$, ein Präfix von $10*$ wird als folgendes Paar gespeichert $(10000, 11000)$. Die ersten beiden Einsen der Maske zeigen, dass die ersten beiden Bits der Schlüssel gematcht werden sollen. In Abbildung 3 sieht man den Aufbau eines Ternary CAMs. Der Schlüssel wird als ein Vektor Parallel gematcht

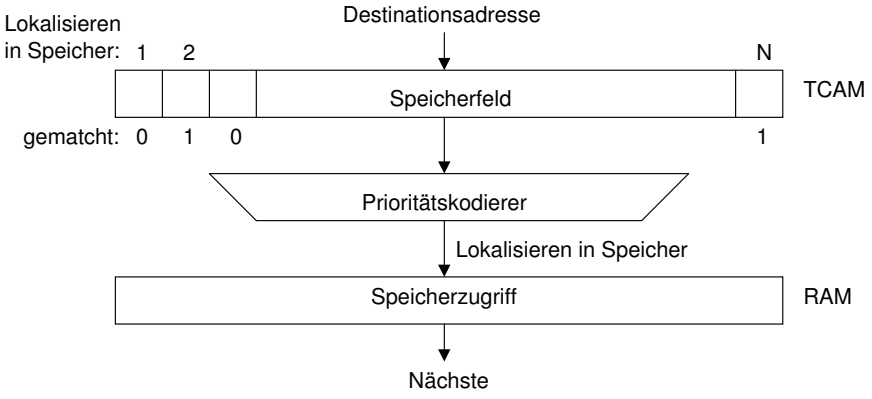


Abbildung 3: Funktionsweise der TCAM

(verglichen). Angenommen N sei die Anzahl der Regeln, dann wird ein N-Bitvektor ermittelt, der anzeigt, welche Regeln zu diesen Schlüssel passen. Mittels dieses Vektors indiziert der Prioritätskodierer die Adresse der höchst priorisierten Match im RAM. Die Struktur der TCAM ist einfach und leicht produzierbar.

3.2 Bitmap-Intersektion: [GuMc01]

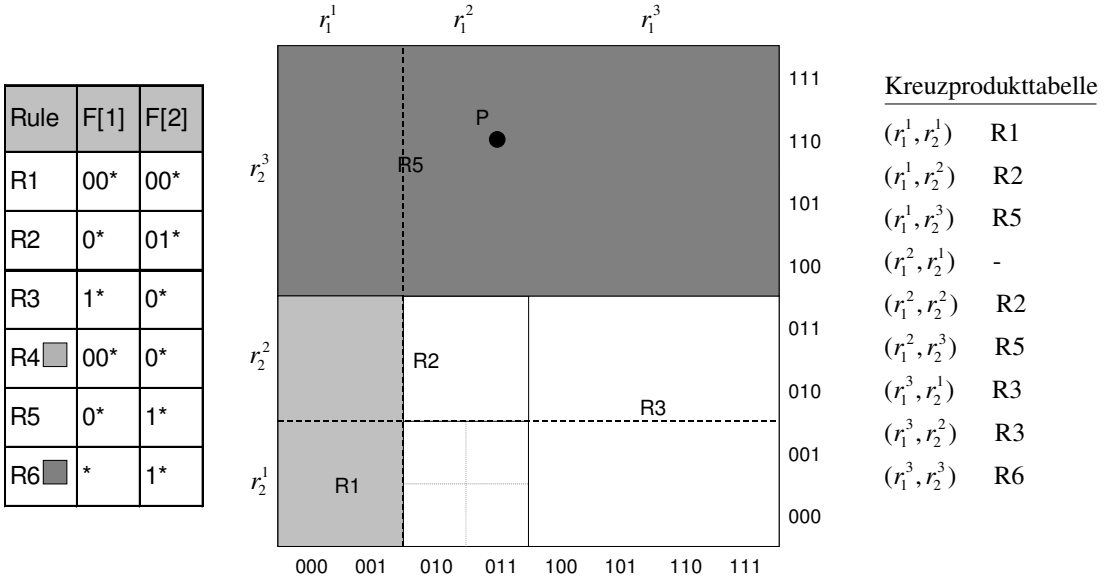


Abbildung 4: Geometrische Darstellung der Kreuzproduktalgorithmus.

Zuerst wird das Kreuzprodukt mit Hilfe eines Algorithmus ausgerechnet, wie man es im Abbildung 4 sehen kann. Für jede Dimension k (hier $max\{k\} = d = 2$) wird die Satzgröße

Dimension 1:		
r_1^1	{R1, R2, R4, R5, R6}	110111
r_1^2	{R2, R5, R6}	010011
r_1^3	{R3, R6}	001001
Dimension 2:		
r_2^1	{R1, R3, R4}	101100
r_2^2	{R2, R3, R4}	011100
r_2^3	{R5, R6}	000011

Tabelle 1: Wurde mit Hilfe der Abbildung 4 gebildet.

	010011	(Dimension 1)
UND	000011	(Dimension 2)
=	000011	=> R5 oder R6

Tabelle 2: UND-Verknüpfung von zwei Bit-Strings.

$s_k = |G_k|$ bestimmt ($s_1 = s_2 = 3$). Im Allgemein es gilt: $1 \leq k \leq d$ und sei $r_k^j, 1 \leq j \leq s_k$ die j-te Komponente in G_k, dann wird die Kreuzprodukttabelle wie gefolgt konstruiert:

$$\left(r_1^{i_1}, r_2^{i_2}, ..., r_d^{i_d}\right), \quad 1 \leq i_k \leq s_k, \quad 1 \leq k \leq d \quad \text{mit der Größe } \prod_{k=1}^{d} s_k \tag{4}$$

Man kann aus der Abbildung 4 erkennen, welche Regeln auf der selben Linie wie die Komponente r_k^l liegen. Sie werden bitweise kodiert. Liegt Ri auf der selben Linie, wird an der i-ten Stelle des Bitvektors eine 1 ansonsten eine 0 eingetragen. Betrachte Tabelle 1.

Wenn man das Paket $P(011, 110)$ sucht, sieht man, dass 011 zu r_1^2 und 110 zu r_2^3 am besten passen. Die Bitvektoren werden dann mit einer UND-Funktion verknüpft (Tabelle 2).

R5 passt am besten, da er ein größere Priorität als R6 hat.

3.3 ClassiPI Architektur: [IyKS01]

ClassiPI ist eine effiziente Pipelinearchitektur, mit der man einen kontinuierlichen Paketfluss klassifizieren kann. Die Komponenten der Geräte sind in Abbildung 5 dargestellt.

System Interface (Systemschnittstelle):

Der ClassiPi Co-Prozessor wird an einem synchronen RAM (SRAM) zusammen mit einem Prozessor, Paketquelle oder DMA-Gerät angeschlossen. Durch diese Schnittstelle werden Pakete und die Ergebnisse der Klassifikation gesandt. Mehrere Pakete werden in den Buffer geladen und unabhängig voneinander klassifiziert.

*F*ield *E*xtaction *E*ngine (Feldextrahierungsmaschine):

Die Aufgabe dieser Komponente ist es den Key- (Schlüssel-) oder die Feldlisteninformationen F zu bilden. Es extrahiert die Kopffelder der IP, TCP und UDP zu jedem Offset. Es ist programmierbar, um bestimmte Informationen aus dem Datenfluss zu extrahieren. Das geschieht durch Kontrollregister oder Ergebnisse der früheren Klassifikationsoperationen.

*C*lassification *E*ngine (Klassifikationsmaschine):

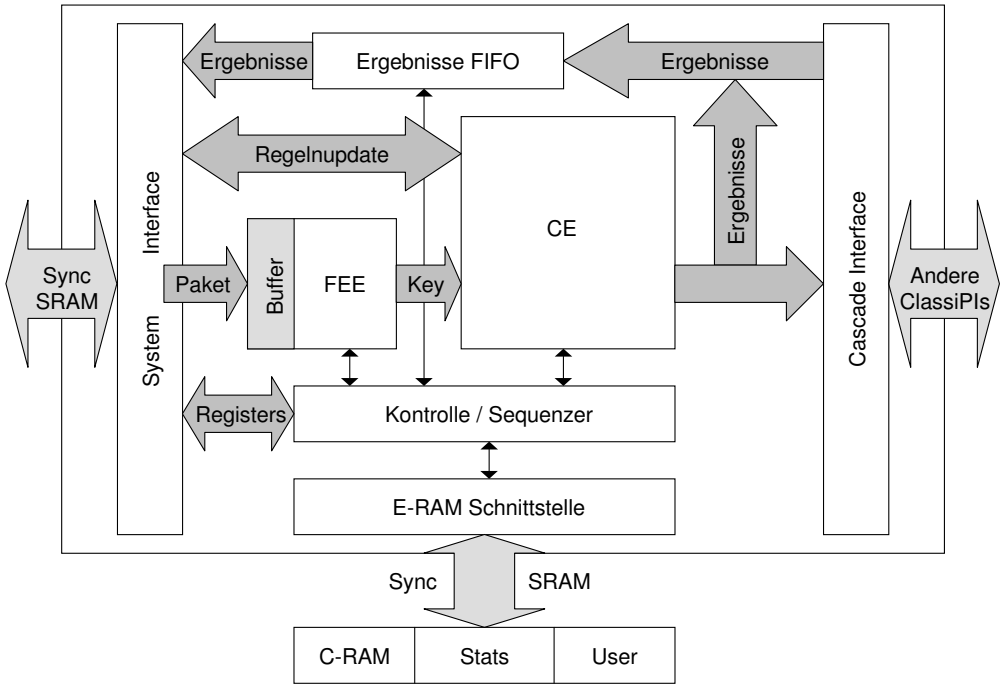


Abbildung 5: Ein ClassiPI-Blockdiagramm.

CE implementiert einen Satz von Klassifikationsfunktionen. Die Anzahl der Regeln beträgt 16K (16000) und kann durch die Kaskadierungsschnittstelle bis zu 128K erweitert werden. Die Architektur unterstützt ein breites Regelnformat. Verwendete Operatoren sind: MASK, AND, OR, NOT, >, <, ==, *. Damit lassen sich verschieden, beliebig komplexe Funktionen mit beliebig breiten Schlüsseln programmieren. Die fundamentale Operation ist eine Suchfunktion. Sie unterstützt sowohl multiple Match als auch höchste Priorität Match. Bei multiple Match werden die passenden Regeln in das „Ergebnisse FIFO Register“ gelegt.

*E*xternal RAM Interface (Externe *R*andom *A*ccess *M*emory Schnittstelle):

Hier werden folgende Daten gespeichert:

- das Klassifikationsprogramm
- benutzerprogrammierbare Daten, die mit jeder Regel assoziiert werden
- statistische Informationen: Paketanzahl, Byteanzahl und Zeitstempel

Kontrolle / Sequenzer:

Dieses Bauteil kontrolliert die beteiligten Komponenten bei der Paketklassifikation.

Cascade Interface (Kaskadierungsschnittstelle):

Man kann maximal acht ClassiPI-Geräte kaskadieren. Damit steigert man die Anzahl der Regeln auf 128K. Jeder Chip empfängt die selben Schlüssel und führt in Parallel unterschiedliche Klassifikationsfunktionen mit entsprechende Regeln aus.

Auf diese Hardwarearchitektur können lineare Sucher und baumbasierte IPv4 Router Algorithmen implementiert werden.

4 Software

Es wurden eine Menge von Softwarealgorithmen entwickelt. Man kann sie in drei Kategorien unterteilen: Grunddatenstrukturen, geometrisch basierte Datenstrukturen und heuristische Datenstrukturen. Zu jeder dieser Kategorien werden ein paar Algorithmen beschrieben.

4.1 Grunddatenstruktur

4.1.1 Lineare Suche: [GuMc01]

Alle Regeln (Filter) sind in einer Liste gespeichert. Je tiefer man in der Liste geht, desto kleiner ist die Priorität der Regel. Die Pakete werden sequentiell mit der Regeln verglichen, bis die Passende gefunden wird. Bei einer großen Datenbank ist dieser Algorithmus ineffizient. Man kann es verwenden, wenn die Suche parallel erfolgt.

4.1.2 Hierarchische Tries: [GuMc01]

Rule	F[1]	F[2]
R1	00*	00*
R2	0*	01*
R3	1*	0*
R4	00*	0*
R5	0*	1*
R6	*	1*

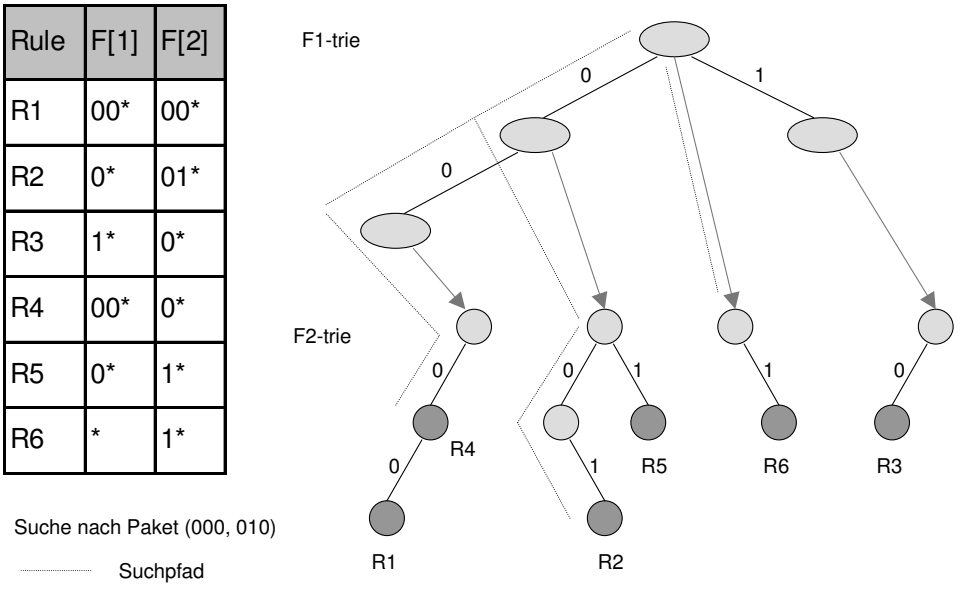


Abbildung 6: Eine hierarchische Trie Datenstruktur. Die grauen Pfeile zeigen das Dimensionswechsel. Die Suche nach Paket (000, 010) wird durch die gestrichelte Linien angezeigt.

Zuerst wird der eindimensionale Trie gebaut. Er wird auch F1-trie genannt. Er hat eine baumförmliche Struktur. 0 bedeutet eine Kante links, 1 bedeutet eine Kante rechts und * ist eine Dimensionswechsel oder eine Regel. Siehe Abbildung 6. Pfeile zeigen einen Dimensionswechsel. Bsp.: von F1-trie nach F2-trie. Die nächsten Dimensionen werden entsprechend konstruiert, bis man zu den Blättern der letzten Trie kommt. Die Speicherkomplexität beträgt: $O(NdW)$ mit N Regeln, d Dimensionen und W ist die maximale Präfixlänge (Anzahl der Ziffern vor dem *).

Suche nach Paket $(v_1, v_2, ..., v_d)$: zuerst wird F1-trie durchsucht basierend auf den Bits von v_1, danach werden die Wege der auftreffende Pfeilen genommen. Nimm den Weg des ersten Pfeils (Dimensionswechsel), fahre fort wie bei F1-Trie für F2-Trie. So werden alle passende Regeln gefunden. Als Beispiel siehe Abbildung 6. Zeitkomplexität: $O(W^d)$ mit W maximale Präfixlänge und d Dimensionen.

4.1.3 Set-pruning Trie: [GuMc01, SrSV99]

Rule	F[1]	F[2]
R1	00*	00*
R2	0*	01*
R3	1*	0*
R4	00*	0*
R5	0*	1*
R6	*	1*

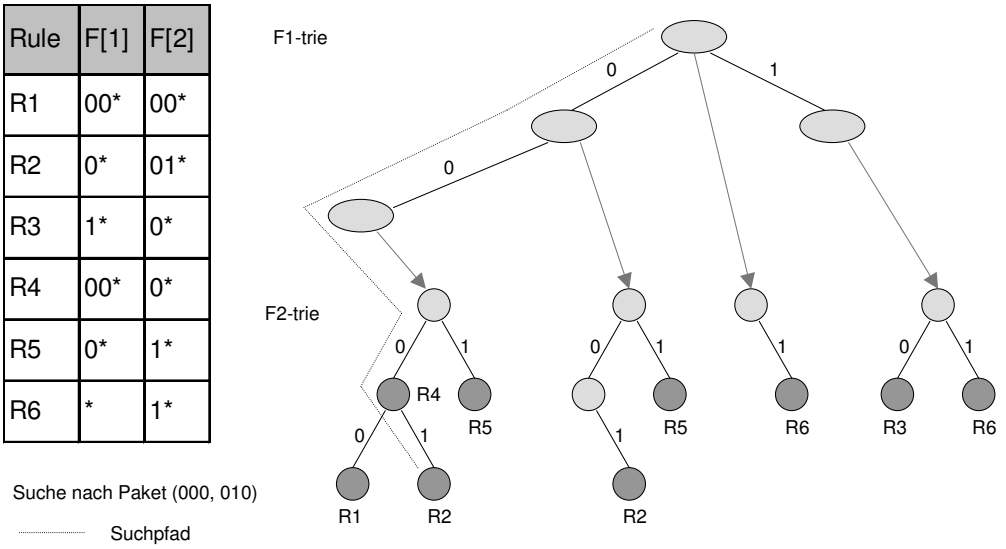


Abbildung 7: Eine set-pruning Trie Datenstruktur. Die grauen Pfeile zeigen das Dimensionswechsel. Die Suche nach Paket (000, 010) wird durch die gestrichelte Linien angezeigt.

Die Suche nach Paket $(v_1, v_2, ..., v_d)$ ist deterministisch: zuerst wird der längste passende Präfix der v_1 in F1-Trie durchgesucht, dann wird der letzte angetroffene Pfeil genommen, falls es einen gibt. Damit wurde die Dimension von F1-trie zu F2-trie gewechselt. F2-trie wird entsprechend durchquert mit Hilfe von v_2. So fährt man fort, bis keine Dimensionswechsel möglich ist. Letzte begegneten Knoten enthält den passendsten Filter. Abbildung 7 zeigt einen Beispiel. Komplexitätsangaben: Zeit $T = O(dW)$, Update $U = O(N^d)$ und Speicherplatz $S = O(N^d dW)$ mit N Regeln, Dimension d und Präfixlänge W.

4.2 Geometrisch basierte Datenstruktur

4.2.1 Grid-of-tries: [GuMc01]

Sie erlaubt schnelle Suche bei wenig Speicherplatz für zwei Dimensionen. Die Regeln werden nicht doppelt kopiert, sondern es werden Zeiger auf die Regeln gesetzt. Das kann man in Abbildung 8 beobachten.

T_x und T_w sind zwei unterschiedliche Tries, die aus dem Präfix der F2-Trie gebildet werden. Die Knoten r und s liegen auf F1-Trie und zeigen auf die beiden Tries. Da der Bitstring von der Wurzel bis w und mit Bit b konkateniert, identisch mit dem Bitstring von der Wurzel bis zu x ist, wird ein Zeiger von w nach x gelegt. w hat kein Kind, der mit Bit

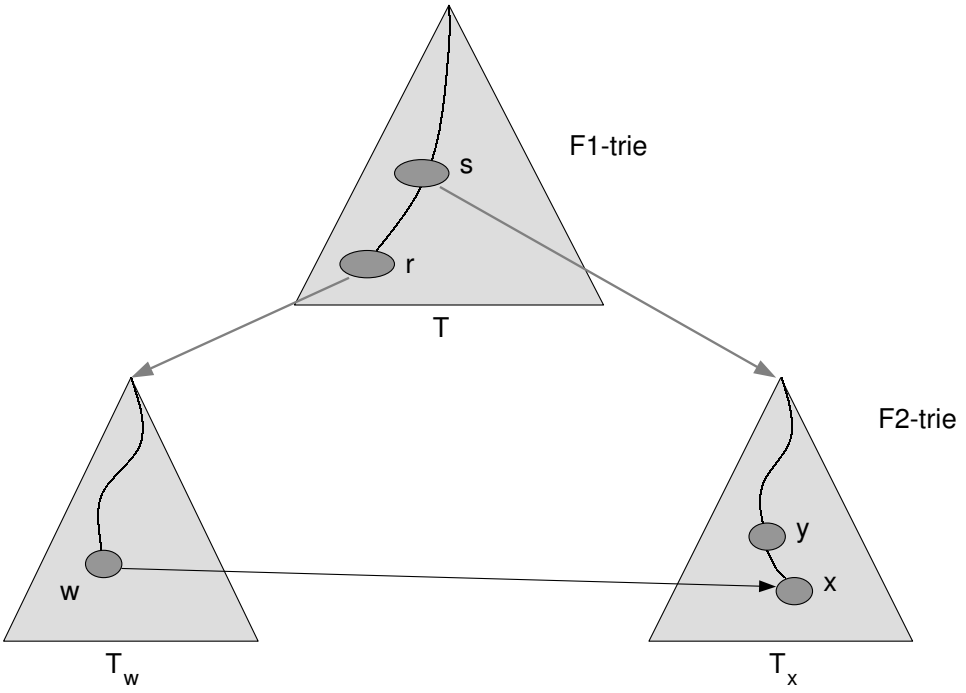


Abbildung 8: Ein Zeiger verweist von w nach x.

b verweist wird. Damit sind folgende Wege gleich: $W1(Wurzel(T), s, r, Wurzel(T_w), w, x)$, $W2(Wurzel(T), s, Wurzel(T_x), y, x)$.

Abbildung 9 zeigt einen Beispiel. Die schwarzen Pfeile sind solche Verweise die auf die Regeln zeigen, um Speicherplatz zu sparen und die Suche zu beschleunigen. Die Suche erfolgt ähnlich wie beim Set-pruning Trie 4.1.3.

Man kann Grid-of-tries auch für mehrere Dimensionen verwenden, indem man die letzte beide Dimensionen einer hierarchische Trie mit dieser Methode behandelt. Die Suchzeit verringert sich bis auf $O(W^{d-1})$, Speicherplatzbedarf: $O(NdW)$.

4.2.2 AQT (Area-Based Quadtree): [GuMc01]

Die Fläche wird in vier Quadranten unterteilt. Jeder der einzelnen Quadranten wird weiter in vier Quadranten unterteilt, bis jeder der kleineren Quadranten nur eine Regel enthält. Diese Methode funktioniert für zwei Dimensionen. Man benutzt eine Baumstruktur. Jeder Knoten hat vier Kinder. Siehe Abbildung 10.

Als Beispiel betrachte Abbildung 11. Eine Regel wird dem Knoten zugewiesen, wenn sie in dem kleinste Quadrat, des zugehörigen Knoten passt. R6 passt nur in die Wurzel, da sie sich von NW bis NE erstreckt.

Es ist leicht zu erkennen, dass R5 in NW und R3 in SE passt. So werden die Regeln dem entsprechende Knoten zugeordnet. Die Komplexitäten einer zweidimensionalen Datenbank mit N Regeln lauten: Zeit $T = O(\alpha W)$ mit einen Integer Parameter α, Speicherplatz $S = O(NW)$.

Rule	F[1]	F[2]
R1	00*	00*
R2	0*	01*
R3	1*	0*
R4	00*	0*
R5	0*	1*
R6	*	1*

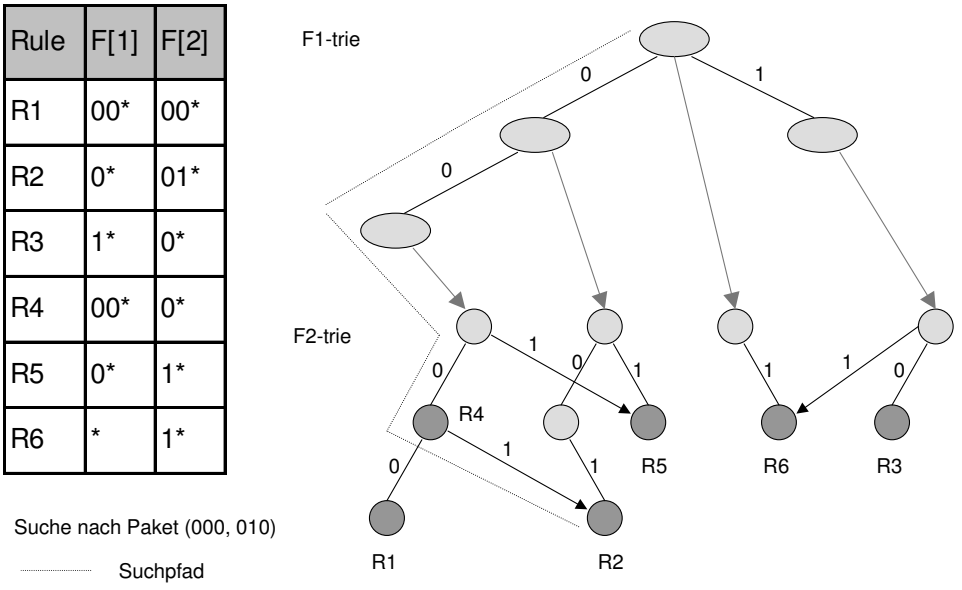


Abbildung 9: Ein Grid-of-tries als Beispiel. Die gestrichelte Linie zeigt der Suchpfad nach der Paket (000, 010).

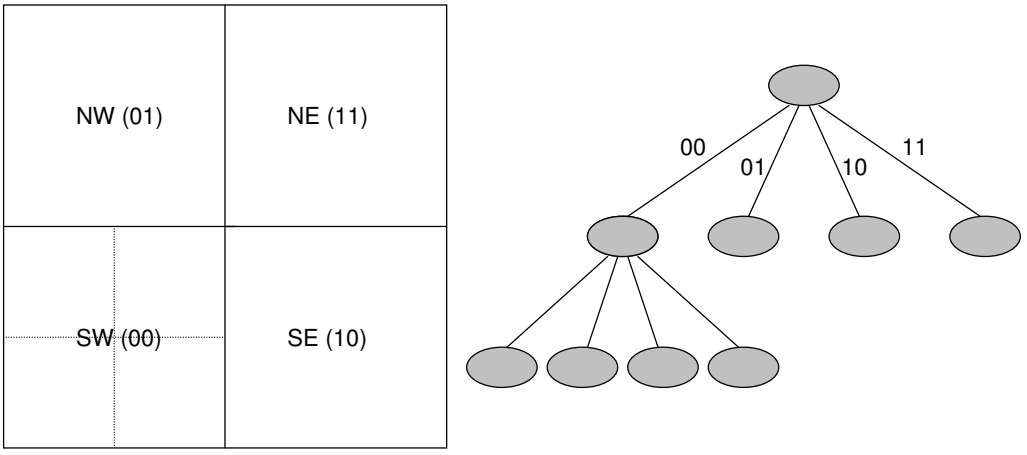


Abbildung 10: Ein Quadtree Konstruktion für ein zweidimensionaler Raum.

4.2.3 FIS (Fat Inverted Segment Tree): [GuMc01]

Ein Segment Baum speichert die Endpunkte eines Satzes von überlappenden Liniensegmenten. Es ist geeignet für eine zweidimensionale Klassifikation. $\{x, y\}$ mit x, y Binärzahlen von eine bestimmte Stellenanzahl, ist einen Bereich von Binärzahl x bis Binärzahl y. Bsp.:

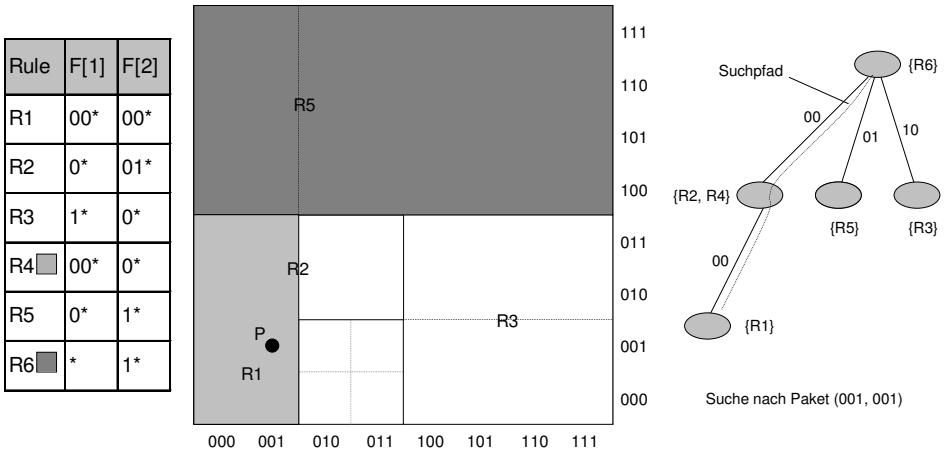


Abbildung 11: Ein AQT Datenstruktur. Die gestrichelte Linie zeigt den Suchpfad des Pakets (001, 001).

$\{000, 011\} \widehat{=} 000, 001, 010, 011$. Es entsteht ein Binärbaum. Die Blätter enthalten die kleinsten Bereiche. Der Elternknoten enthält die Vereinigung der Bereiche ihrer beiden Kinderknoten. In Abbildung 12 kann man an einen Beispiel betrachten, wie solch ein Baum gebildet wird. Das Bild, das aus Rechtecken besteht, zeigt die Bereiche, die die Regeln überdecken. Eine Anfrage traversiert von der Wurzel des Segment Baums und berechnet die höchste Priorität aller auftreffende Segmente.

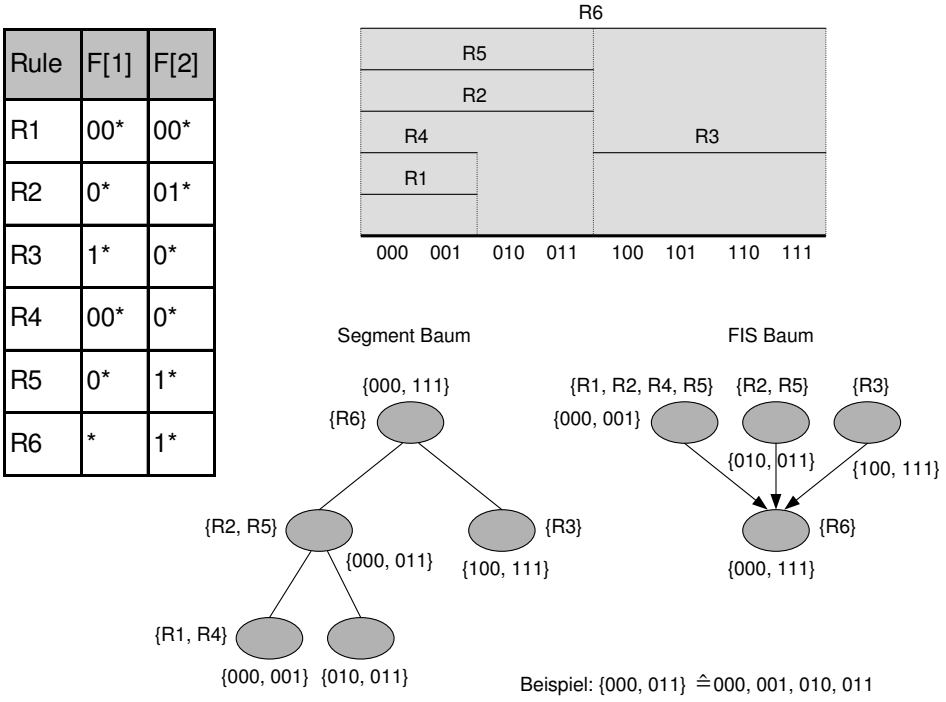


Abbildung 12: Ein Segment Baum und ein 2-Level FIS.

Ein FIS Baum ist ein Segment Baum mit zwei Modifikationen:

- Baum wird komprimiert: Grad des Baumes wird erhöht, dadurch sinkt ihre Höhe und besetzt eine gegebene Ebenenanzahl.
- Aufwärtszeigern von Kind- zu Elternknoten werden eingefügt.

Eine Anfrage nach Paket $P(v_1, v_2)$ beginnt mit F1. Aus v_1 wird ein Blatt des FIS Baums ermittelt. Aus diesem Blattknoten werden die Aufwärtszeiger in Richtung Wurzel verfolgt, dabei wird die höchste Priorität der Regeln, die zu diesem Paket passt, ausgerechnet.

Komplexitäten: Zeit $T = O((l+1)t_{RL})$, Speicherplatz $S = O(ln^{l+1})$ mit einen l ebenen Baum und t_{RL} die Zeit für einen eindimensionalen Bereich. Es ist leichter Regeln einzufügen als zu löschen. FIS ist zu mehrere Dimensionen erweiterbar.

4.3 Heuristische Datenstruktur

4.3.1 Rekursive Fluss Klassifikation (*R*ecursive *F*low *C*lassification): [GuMc01]

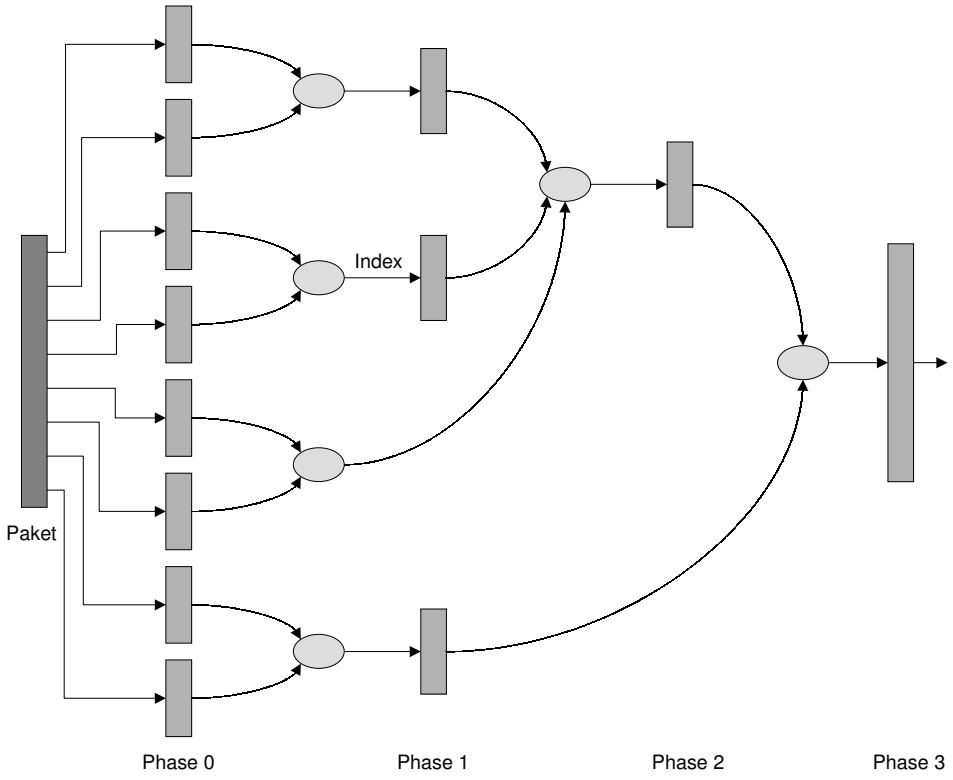


Abbildung 13: Paketfluss Mittels RFC.

Der Sinn des Algorithmus ist: in mehrere Phasen wird die Anzahl der infragekommende Regeln rekursiv reduziert. Sei T ein Bitstring aus dem Paketkopf, $T = \log N$ mit N Regeln. T wird als Index verwendet, um auf dem Speicher zu zugreifen. Dann wird der nächste Bitstring aus dem Ergebnis als Index benutzt, um innerhalb dieses Speichers auf einen kleineren Speicherblock zu zugreifen, usw. Abbildung 13 zeigt eine Verwirklichung dieser Vorgehensweise:

- d Felder (Dimensionen) des Paketkopfs werden in Stücke zerlegt, um sie in multiplen Speichern parallel zu indizieren. Es entstehen mehrere Mengen mit passenden Regeln (Phase 0).

- In den nächsten Phasen werden die Speichern mit dem Ergebnisse früheren Phasen indiziert. Es wird eine Art Schnittmenge aus je zwei Mengen gebildet. Die Schnitttmenge enthält Regeln die sowohl im eine Menge als auch in die andere liegen.
- In der letzte Phase wird die Menge mit der verbliebenen Regeln ermittelt.

Das Problem dieses Verfahren ist das schnelle Wachstum des Speicherplatzbedarfs und der Berechnungszeit mit der Zunahme der Regelanzahl.

4.3.2 hierarchische intelligente Schnitte (*H*ierarchical *I*nteligent *Cut*ting*s*): [GuMc01]

Das Verfahren ähnelt den B-tree Modell. Man hat Behälter in dem maximal B Regeln Platz haben. Diese Behälter sind die Blätter des Baumes. Der Baum kann höchstens so viele Ebenen haben, wie die Anzahl der Dimensionen, da jede Ebene einer Dimension entspricht. Die Kinderknotenanzahl zeigt in mindestens wie viele Teile die entsprechende Dimension unterteilt wird. Als Beispiel siehe Abbildung 14.

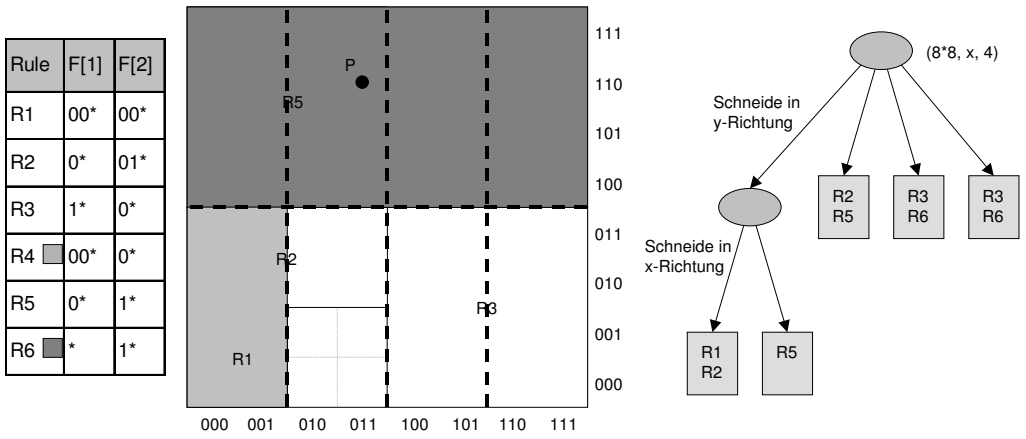


Abbildung 14: HiCuts mit Behältergröße = 2, die Kinderknotenanzahl zeigt in mindestens wie viele Teile die Dimension geteilt wurde.

Die Suche eines Paketes zeigt ein Blatt, beginnend von der Wurzel. Das Blatt verweist auf einen Behälter. In dem Behälter erfolgt die Suche nach dem besten Match sequentiell. Die Charakteristika des Entscheidungsbaums sind: die Höhe, der Grad jedes Knotens und die Suchentscheidung in jedem Knoten.

4.3.3 Tuple-space Suche: [GuMc01]

Eine Hashtabelle wird benutzt, um die passenden Regeln schnell zu finden. Feld Tupel wird als Schlüssel für die Hashtabelle verwendet. Tupel ist ein Vektor der aus d Komponente besteht, d ist die Dimension. Jede Komponente von Tupel gibt die Anzahl der Präfixlänge der zugehörigen Dimension an. Beispiel aus Abbildung 15: R4 mit (00*, 0*) hat den Tupel (2, 1), weil die Präfixlänge der 1. Dimension 2 und der 2. Dimension 1 ist. So werden die Tupeln gebildet.

Für M Tupel braucht man höchstens M Speicherzugriffe. Speicherkomplexität $S = O(N)$ mit N Regelanzahl. Je mehr Dimensionen von einem Paket verglichen werden, um so schneller

Regel	Spezifikation	Tupel
R1	(00*, 00*)	(2, 2)
R2	(0**, 01*)	(1, 2)
R3	(1**, 0**)	(1, 1)
R4	(00*, 0**)	(2, 1)
R5	(0**, 1**)	(1, 1)
R6	(***, 1**)	(0, 1)

Tupel	Hashtabelleneingänge
(0, 1)	{R6}
(1, 1)	{R3, R5}
(1, 2)	{R2}
(2, 1)	{R4}
(2, 2)	{R1}

Abbildung 15: links sind die Tupel, die aus jeder Regel entstehen und rechts die Tupel als Schlüssel für die Hashtabelle.

ist der passende Filter (Regel) auffindbar, weil der Schlüssel länger ist und die Hashtabelle mehrere Eingänge hat.

5 Schluss

Die Anzahl der Personen, die im Internet surfen wächst ständig. Das bedenkt, dass auch die Anzahl der Daten, die per Internet gesendet werden, mitwächst. Die Daten werden als Pakete von Quelle zu Destination gesendet. Es werden Routern und Firewalls benutzt, um die Pakete zu ihre Destination zu befördern und die böswilligen Pakete auszufiltern. Damit die Routern und Firewalls ihre Aufgabe erfüllen können, müssen die Pakete klassifiziert werden. Die Paketklassifikation muss schnell und zuverlässig erfolgen, damit es zu keinem Paketstau kommen kann. Es wurden ein Menge Algorithmen entwickelt (siehe Tabelle 3). Man kann sie sowohl per Software als auch per Hardware implementieren. Sie werden verglichen, indem man die Komplexität zu jedem Algorithmus ausrechnet. Wichtige Komplexitätsangaben sind: Zeit, Speicherplatzbedarf, Update und Filteranzahl (Regelanzahl). Jeder Klassifikator ist in mindestens einer Komplexität schwach. Je nach Regelanzahl, Updatehäufigkeit und Zeitbedarf kann der entsprechende Algorithmus für die Implementation gewählt werden.

Für Personen, die mit Netzwerkmanagement und Hochleistungskommunikation nicht vertraut sind, aber es erlernen möchten, wird ein gutes Buch empfohlen: [PeDa00]. Dieses Buch ist für Einsteiger leicht zu verstehen.

Das Buch [ZwCC01] ist im Jahr 2001 erschienen und enthält die bis jetzt neuesten Information über Internet Firewalls. Es wird ausführlich gezeigt, wie man Firewalls für Internet einrichten kann.

Algorithmen			$\mathbf{T}^-$	$\mathbf{S}^-$
Hardware		Ternary CAM	1	N
		Bitmap-Intersektion	$dW + N/SB$	dN^2
		ClassiPI Architektur	IA	IA
Software	Grunddatenstruktur	Lineare Suche	N	N
		Hierarchische Tries	W^d	NdW
		Set-pruning Tries	dW	N^d
	Geometrisch basiert	Grid-of-Tries	W^{d-1}	NdW
		AQT	αW	NW
		FIS-Baum	$(l+1)W$	$l \times N^{1+1/l}$
	Heuristisch	RFC	d	N^d
		Hierarchical cuttings	d	N^d
		Tuple space search	N	N

Tabelle 3: zeigt eine Zusammenfassung, der behandelten Algorithmen mit dem zugehörigen wichtigsten Komplexitäten. Abkürzungen: $\mathbf{T}^- \hat{=}$ Zeitkomplexität im schlechtesten Fall, $\mathbf{S}^- \hat{=}$ Speicherplatzkomplexität im schlechtesten Fall, $SB \hat{=}$ Speicherbreite und IA $\hat{=}$ Implementationsabhängig. Parameterbeschreibung: N Regeln (Filter), d Dimensionen, W maximale Präfixlänge, α ein Integer Parameter und l ist der Ebenenzahl des Baumes.

Literatur

[GuMc01] Pankaj Gupta und Nick McKeown. IEEE Network. *Algorithms for Packet Classifikation*, March/April 2001, S. 9. Stanford University.

[IyKS01] Sundar Iyer, Ramana Rao Kompella und Ajit Shelat. IEEE Network. *ClassiPI: An Architecture for Fast and Flexible Packet Classification*, March/April 2001, S. 9.

[PeDa00] Larry L. Peterson und Bruce S. Davie. *Computernetze, ein modernes Lehrbuch.* dpunkt.verlag, Ringstraße 19b, 69115 Heidelberg. ISBN 3-932588-69-X, 1. Auflage, 2000.

[SrSV99] V. Srinivasan, Subhash Suri und George Varghese. Packet Classification Using Tuple Space Search. In *SIGCOMM*, 1999, S. 135–146.

[ZwCC01] Elizabeth D. Zwicky, Simon Cooper und D. Brent Chapman. *Einrichten von Internet Firewalls.* O'Reilly, Balthasarstr. 81, 50670 Köln. ISBN 3-89721-169-6, 2. Auflage, 2001.

Multi Protocol Label Switching

Michael Wiese

Kurzfassung

Um dem wachsenden Datenaufkommen im Internet erfolgreich zu begegnen, genügt eine alleinige Kapazitätsausweitung nicht. Das Optimieren des Verkehrsfluss ist eine sinnvolle Ergänzung. Das Multi Protocol Label Switching (MPLS) ist eine neue Technologie, die im Bereich des Traffic Engineering erhebliche Vorteile bietet: Durch das Anbringen eines Labels an ein Datagramm und das Aufbauen eines Label Switched Path (LSP) können Pakete auf Schicht 2 witergeleitet werden und durch einfache Labeloperationen auf andere Routen umgeleitet werden. MPLS bietet den Geschwindigkeitsvorteil des Switching auf Schicht 2, sowie die Flexibilität des Routing auf Schicht 3.

1 Einführung

Einer US-Studie zu folge übertrifft die Geschwindigkeit, mit der das Internet eingeführt wurde alle anderen Technologien, einschließlich Radio, Fernsehen und PC. Das Internet wird sich zu einem Medium entwickeln, in dem Sprach-, Video- und Datenkommunikation verschmelzen werden, was den Verkehr im Internet rapide anwachsen lässt. Man schätzt, dass dich das Verkehrsaufkommen pro Jahr verdoppelt bis verzehnfacht.

Ansatzpunkte, um mit diesem rasanten Wachstum zurecht zu kommen, sind Verbesserungen in der Netzwerkarchitektur, die Erweiterung der Kapazitäten und schließlich die Optimierung des Verkehrsflusses, das Traffic Engineering.

Zeitgleich wächst der Anspruch an die Dienstgüte im Internet, um z.B. Multimediaanwendungen qualitativ hochwertig im Internet anbieten zu können. Vergleicht man das Internet mit dem Logistiknetz der Post erkennt man eine gleichartige Entwicklung: auch beim Brief-, bzw. Paketversand wurden zur ursprünglichen einfachen Zustellung Dienstleistungen, wie Expresszustellung, Kurierdienst und Einschreiben gegen ein entsprechendes Entgelt angeboten.

Besonders die Bedeutung des Traffic Engineering nimmt zu, da eine alleinige Kapazitätsausweitung ohne sinnvolle Verkehrssteuerung nutzlos wäre (vgl. Kapazitätsausweitung in Backbone-Netzen: 1996: 47 Mb/s, 1999/2000: 2,5 Gb/s, vgl. [Awdu99], S. 42).

Multi Protocol Label Switching (MPLS) ist eine relativ einfache Technologie, die im Bereich des Traffic Engineering erhebliche Vorteile bietet. Zur Bereitstellung von Dienstgüte trägt MPLS zwar nicht direkt bei, ist aber ein hilfreiches Werkzeug, um Verzögerungen zu minimieren und das Jittern zu verbessern, zwei Merkmale des QoS.

MPLS wurde in unterschiedlichen Ansätzen durch mehrere Unternehmen (Cisco, Ipsilon,..) voran getrieben und schließlich durch die Internet Engineering Task Force (IETF) standardisiert.

2 Die Welt ohne MPLS

2.1 Klassisches IP-Routing

Das Weiterleiten von IP-Paketen erfolgt in den Routern durch die Analyse der Zieladresse des IP- Paketkopfs und einer internen Routing-Tabelle. Die Routing-Tabelle enthält für jedes Netz oder Subnetz einen Eintrag, der festlegt, über welchen Ausgangslink ein eingetroffenes Paket weitergeleitet wird. Hierzu wird in der Tabelle der passende Präfix gesucht und das Paket an den zugehörigen Ausgang geleitet (vgl. Tabelle 1, [KrRe00], S. 169).

Zieladresse	Ausgang
129.13.*.*	1
128.*.*.*	2
129.1.2.*	1
129.13.42.*	1
129.13.41.*	4

Tabelle 1: Routingtabelle

Lautet die Zieladresse z. B. 129.13.42.7, so wäre 129.13.42. der beste Präfix. Das Paket wird den Router somit an Ausgang 1 verlassen.

Die Einträge der Routing-Tabelle können sowohl manuell (statisches oder nicht adaptives Routing), als auch durch periodischen Informationsaustausch mit benachbarten Routern (dynamisches oder adaptives Routing) initialisiert werden.

Grundsätzlich lassen sich zwei Routing-Verfahren unterscheiden (vgl. [Brau99], S. 30ff):

1. Distanz-Vektor-Routing

Jeder Router kennt die Entfernung zu jedem anderen Router und verteilt diese Informationen periodisch an die benachbarten Router. Ein weit verbreitetes Beispiel hierzu ist das *Routing Information Protocol (RIP).*

2. Link-State-Routing

Jeder Router besitzt Informationen über jeden Link der Domäne (z.B. ein Uninetz) und kann sich hiermit ein Bild der kompletten Netztopologie der Domäne machen. Ein wichtiges Link-State-Routing-Protokoll ist das *Open Shortest Path First (OSPF).* Durch das periodische Verschicken von sog. HELLO-Paketen und das Empfangen der zugehörigen ECHO-Pakete können die Kosten der Teilstrecken ermittelt werden. Diese Informationen werden wiederum mit den Nachbarn ausgetauscht und gespeichert.

Da die jeweiligen Domänen unterschiedliche Routing-Protokolle, sog. *Interior Gateway Protokolle (IGP),* verwenden können (z.B. RIP, OSPF, IS-IS), müssen die Domänen über einheitliche Protokolle, sog. *Exterior Gateway Protokolle (EGP)*, auf höheren Ebenen Informationen austauschen, z. B. über das *Border Gateway Protokoll BGP-4.*

2.2 IP-Switching: das Overlay-Modell, klassisches IP über ATM

Um den Verkehrsfluss in IP Systemen zu verbessern, wurde der Asynchrone Transfermodus ATM als zusätzliche Technologie in die IP-Infrastruktur eingeführt. Um IP-Pakete über ATM Netzen transportieren zu können, sind die ATM-Endsysteme jeweils einem bestimmten logischen IP-Subnetz (LIS), mit eindeutiger IP-Netzadresse, zugeordnet. Die LIS sind durch IP-Router verbunden.

Um ein Datenpaket von einem LIS in ein anderes zu schicken, wird es über den die LIS verbindenden IP-Router geleitet (Schicht 3, Routing). Innerhalb der LIS können die Datenpakete schneller auf Schicht 2 (Switching) weitergeleitet werden. Hierzu werden virtuelle Verbindungen (VCs) aufgebaut und die Pakete anhand kurzer Kennungen im Paketkopf weitergeleitet, was zu einer deutlichen Geschwindigkeitserhöhung führt.

2.3 Probleme

Das ATM-Netz mit seinen Punkt-zu-Punkt Verbindungen, umgeben von IP-Routern weist mehrere Nachteile auf. Hierzu zählt sicher die parallele Existenz zweier Netzwerke mit unterschiedlichen Technologien. Die Hintereinanderschaltung mehrerer Netzwerkelemente auf einem Pfad steigert die Komplexität und senkt die Verlässlichkeit. Betrachtet man die IP über ATM Netzwerktopologie, erkennt man die enorme Anzahl permanenter virtueller Verbindungen (PVCs) (vgl. [Awdu99], S. 44).

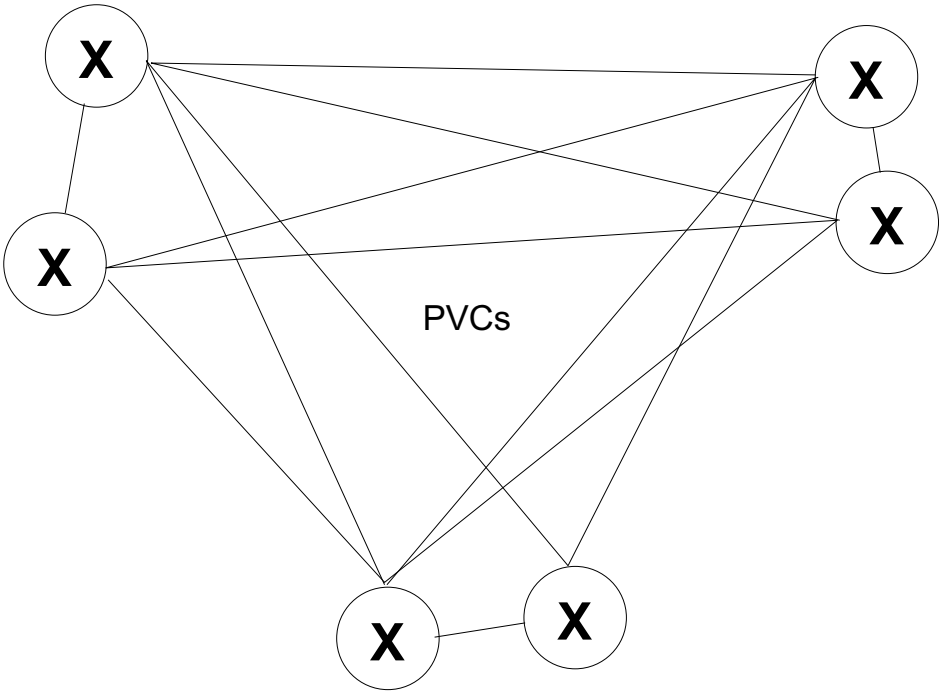


Abbildung 1: Physikalische und logische Netzwerktopologie im Fall IP über ATM.

Im Falle des IP über ATM führt ein einziger Verbindungsabbruch zum Ausfall duzender virtueller Verbindungen (VCs). Diese müssen einzeln wieder aufgebaut werden, was zu einer enormen Verkehrslast durch den Austausch von Routing-Informationen führt: Kommt ein Link-State-Routing-Protokoll beim Ausfall eines Routers in einem Netz von n Routern zum Einsatz, wird jeder der n-1 Nachbarn diesen Router-Ausfall bemerken und eine neue Routing-Tabelle berechnen. Diese wird jeder dieser n-1 Router an die n-2 Nachbar-Router fluten, welche die Informationen an ihre n-3 Nachbarn zurückfluten (vgl. [Swal99], S.54).

In der Praxis kommt es zu einem weiteren Problem (vgl. [Gree00], S. 124): Bei der Weiterleitung der Daten von einem LIS in ein anderes LIS muss das Datenpaket einen IP-Router passieren, auch wenn eine direkte ATM-Verbindung zwischen den Rechnern existieren würde. Um dies zu vermeiden, versuchen die Netzbetreiber logische IP-Subnetze zusammen zu führen. Die Anzahl der virtuellen Verbindungen erhöht sich somit nochmals.

Die Nachteile des klassischen IP-Routing auf Schicht 3 ist die Geschwindigkeit. Jedes IP-Paket wird einzeln bearbeitet, indem die Felder im Paketkopf ausgewertet werden. Die Weiterleitung erfolgt anhand umfangreicher Routing-Tabellen.

3 MPLS Architektur

3.1 Was ist Label Switching ?

Um sich das Label Switching anschaulich zu machen, bietet sich wiederum die Anaolgie zum Postnetz an. Bei der Zustellung eines Briefs schaut sich der Briefträger die Adresse auf dem Umschlag an, um zu entscheiden in welchen Briefkasten er diesen steckt. Beim traditionellen IP-Routing wird anhand der Zieladresse entschieden an welchen Router-Ausgang das IP-Paket geschickt wird. Kommt nun das Label Switching zum Einsatz, wird dem IP-Paket zusätzlich zur Zieladresse ein Label (eine Nummer) aufgeklebt, anhand dessen die Routing-Entscheidung gefällt wird. Im Falle der Post, ist das Label die Postleitzahl, die im Hauptlauf der Post zur Wegewahl benutzt wird. Es wäre sehr viel umständlicher, und somit auch langsamer, bei jeder Weiterleitungsentscheidung im Postleitzahlenbuch nachschauen zu müssen (vgl. Bsp, [Blac01], S.7).

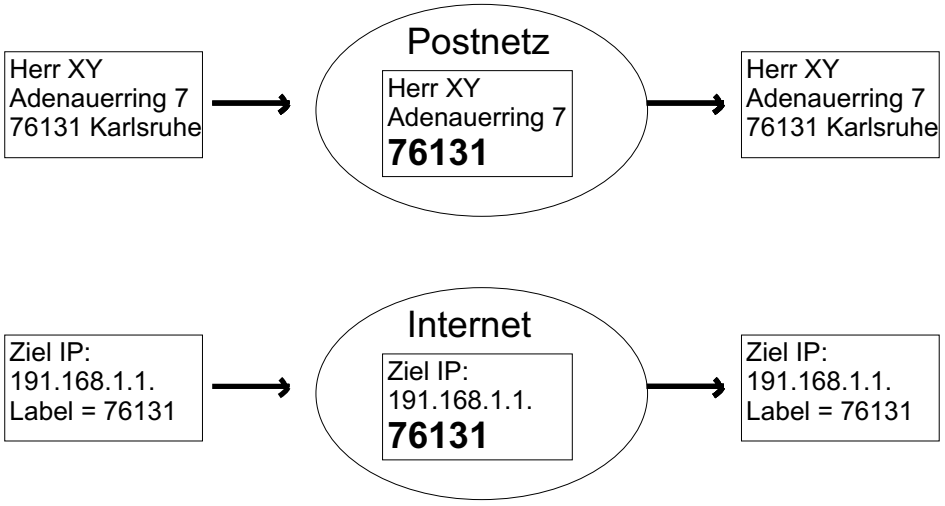


Abbildung 2: Prinzip des Label Switching

Das IP-Datagramm wird zum Label Switching Router (LSR) geschickt, der ein Label an das IP-Paket heftet. Im Folgenden wird das Label bis zum Ende des Label-Pfads zur Weiterleitung benutzt, danach die ursprüngliche IP-Adresse. Es wird somit viel Zeit gespart, da nicht jedes Mal entschieden werden muss, wie das IP-Paket weitergeleitet wird. Ziel des Label Switching ist es Routing-Instanzen aus dem Weiterleitungspfad zu eliminieren und wenn möglich das Datenpaket nur per Switching weiterzuleiten.

Der große Vorteil des MPLS ist die Kombination der Skalierbarkeit und Flexibilität des Routing mit den Ouality of Service- (QoS) und Verkehrsmanagementfähigkeiten des Schicht-2-Switching. Ein Hauptmerkmal des MPLS ist die strikte Trennung von Weiterleitung und Kontrolle.

3.2 Label basierte Weiterleitung

3.2.1 Functional Equivalence Class (FEC)

Zur Paketweiterleitung benutzt das MPLS Functional Equivalence Classes (FECs), die mit den virtuellen Pfaden und Kanälen des ATM vergleichbar sind. Das Konzept der FECs ist jedoch allgemeiner gehalten. Die Datenpakete werden in Gruppen eingeteilt, die nach unterschiedlichen Kriterien gebildet werden können. Eine Gruppe kann durch Pakete gleicher Zieladressen, gleicher Portnummern, aber auch durch Klassen gleicher Dienstgüte gebildet werden. Die Granularität der Ströme wird durch die Kriterien, die zur Klassenbildung herangezogen werden, bestimmt.

3.2.2 Das Label

Es gibt kein verbindliches Labelformat, da MPLS auf jedem

Schicht-2-Protokoll einsetzbar sein sollte (Multi Protocol). Labels können daher ATM-VCI/VPI-Werte, VLAN IDs, Shim Header, usw. sein. Der Shim Header wird zwischen dem Schicht-2- und IP-Kopf eingefügt. Der Shim Header besteht aus einem 20-Bit umfassenden Label-Feld, einem 3-Bit-Feld für experimentelle Zwecke (z.B. für QoS Informationen), einem S-Bit für Stack-Informationen und einem 8-Bit-TTL-Feld, das die Schleifenbildung verhindern soll.

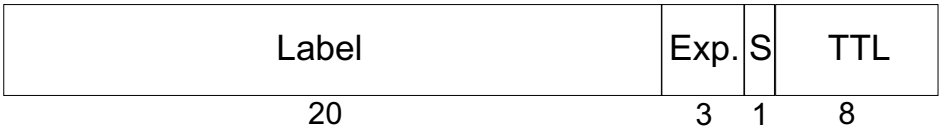


Abbildung 3: Shim Header (vgl. [Brau99], S. 209)

Ein Label kann sowohl einen feinkörnigen Mikrofluss, als auch einen grobkörnigen Makrofluss repräsentieren. Weiterhin kann es für Unicast- oder Multicastverkehr stehen.

3.2.3 Der Label-Stack

MPLS erlaubt es eine willkürliche Anzahl Labels anzubringen, die in einem Stack abgelegt werden, auf dem die Operationen push (hinzufügen eines neuen Labels), swap (austauschen eines Labels) und pop (entfernen eines Labels) zugelassen sind. Der Nutzen eines Label-Stacks ist die Einflussnahme mehrerer Kontrollkomponenten auf die Weiterleitung des Paketes.

3.2.4 Weiterleitung

Die Weiterleitung der Datenpakete erfolgt in einem Label Switching Router (LSR), der Daten sowohl per Switching, als auch per Routing befördern kann. Im Idealfall werden alle Datenpakete durch betrachten des Labels auf Schicht 2 geswitcht. Üblicherweise werden jedoch nicht alle, sondern nur Datenströme hoher Bandbreite durch die Switching-Komponente weitergeleitet. Bei geringer Bandbreite macht es keinen Sinn den Switch so zu konfigurieren, damit man das langsamere Routing umgehen kann.

Passiert ein Datenpaket mehrere Label Switching Router spricht man von einem Label Switching Path (LSP). Mehrere zusammenhängende LSRs können eine MPLS Domäne bilden, innerhalb dieser Pakete nur anhand des Labels weitergeleitet werden.

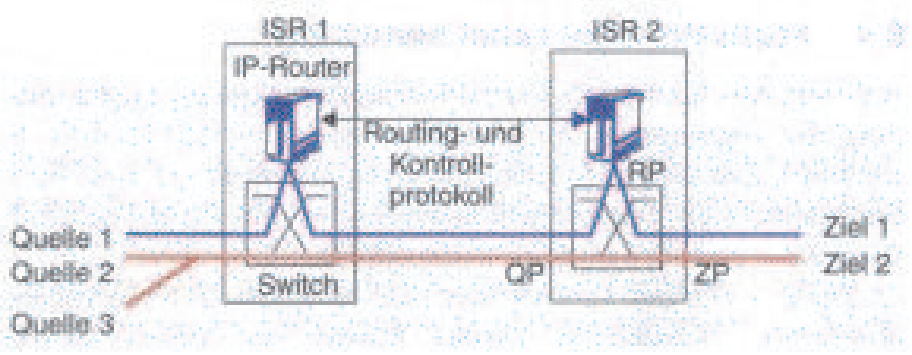


Abbildung 4: Label Switching Router (vgl.[Brau99], S194)

Kommt ein Datenpaket ohne Label in eine MPLS-Domäne, wird es am Ingress-LSR einmalig einer FEC zugeteilt und innerhalb der Domäne nur anhand des nun angehefteten Labels weitergeleitet. Beim Verlassen der MPLS-Domäne wird das Label am Engress-LSR entfernt. Jeder LSR speichert das Label und den dazugehörigen Ausgang als Next-Hop Label Forwarding Entry (NHLFE) in der Weiterleitungstabelle. Kommt ein neues Datenpaket mit Label an, wird dies einem NHLFE der Weiterleitungstabelle zugeteilt Der LSR pflegt eine FEC-zu-NHLFE Abbildung, da mehrere NHLFEs zu einer FEC existieren können. Durch die Abbildung wird ein bestimmter NHLFE ausgewählt, da die Abbildung sich laufend verändern kann, um z.B. die Netzlast auszugleichen (vgl. [ChOh99], S. 59).

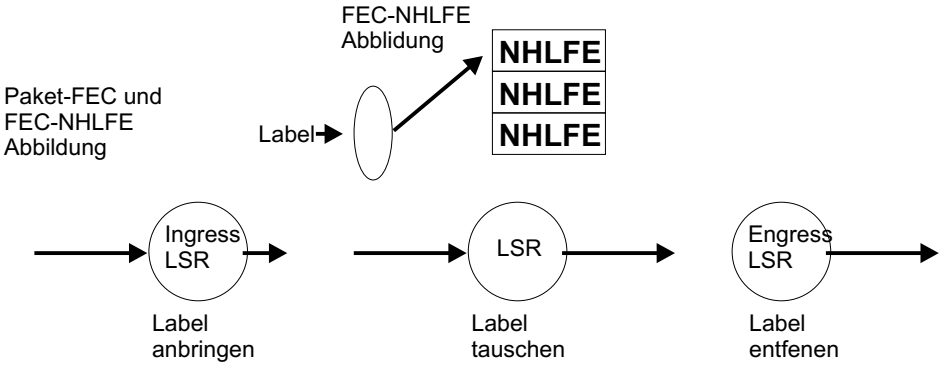


Abbildung 5: Next Hop Label Forwarding Entry

Folgt auf einen LSR ein IP-Router, wird das Label vom Stack entfernt, womit wieder ein normales IP-Paket entsteht, welches auf Schicht-3 weitergeroutet wird.

3.3 MPLS Kontrollkomponenten

Die Aufgaben der Kontrollkomponenten sind

1. Das Verteilen der Routinginformationen an die LSR.

2. Das Erstellen der Weiterleitungtabelle mit Hilfe bestimmter Algorithmen.

Die traditionelle Routing Architektur ist eine Untermenge der Label Switching Kontrollkomponente. Die Kontrolle des MPLS beinhaltet die bekannten traditionellen Routing-Protokolle (OSPF, BGP,....). Zusätzlich hierzu können die LSR

1. Verbindungen zwischen Labels und FECs erstellen.

2. Andere LSR über diese geschaffenen Abbildungen informieren.

1. und 2. zur Erstellung einer Weiterleitungstabelle nutzen und diese zu pflegen (vgl. [DaRe00], S. 36ff)

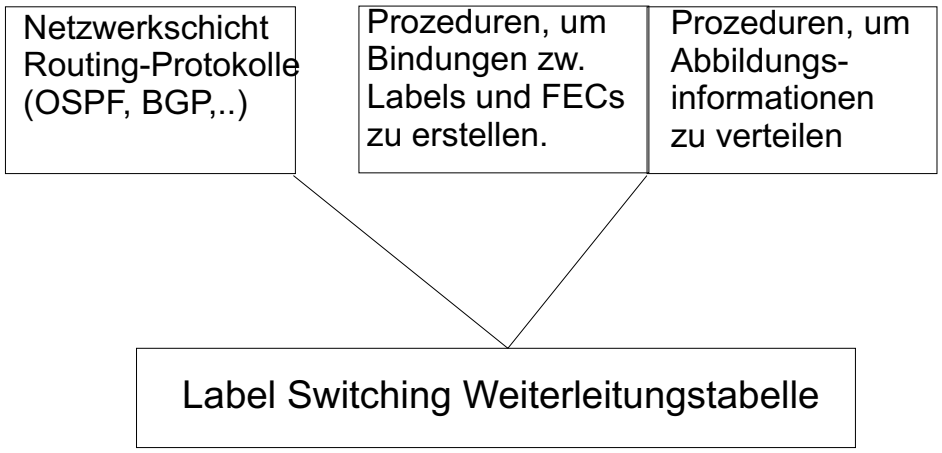


Abbildung 6: Label Switching Kontrollkomponente

Ein LSR bringt das erste Label an das Datenpaket an, falls noch keine Bindung zu einer FEC existiert, d.h. das Paket ohne Label ankommt (Local Binding). Hat ein anderer LSR bereits ein Label angebracht (Remote Binding), kann der LSR, der passiert wird auf bestimmte Ereignisse im Netz reagieren und weitere Labels anheften, entfernen, bzw. das Paket einfach weiterleiten. Dies kann z.B. bei erhöhtem Datenaufkommen geschehen. Es könnte der Fall auftreten, dass die Existenz vieler Datenströme eine grobe Granularität der FECs verlangt, um die Skalierbarkeit des Netzes zu gewährleisten, was zu folgender Situation führen könnte (vgl. [DaRe00], S. 44):

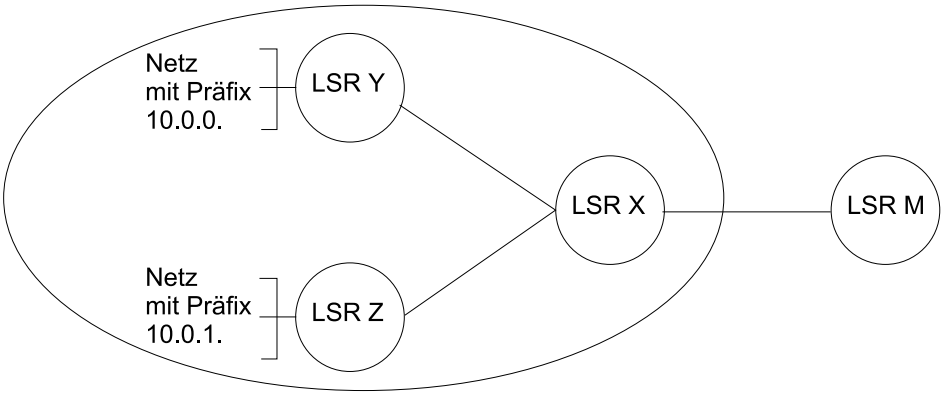


Abbildung 7: Label Switching und IP-Routing

Wird zur Bildung einer FEC der Präfix 10.0.0.*/23 der IP-Adresse herangezogen und dieser FEC ein bestimmtes Label zugeordnet (FEC-Label-Abbildung), kann LSR X ein ankommendes Paket von LSR M nicht auf Schicht 2 weiterswitchen, sondern muss das Datenpaket auf Schicht 3 weiterrouten, da der zur FEC-Bildung genutzte Präfix zu ungenau ist und somit die Weiterleitungsinformation aus dem IP-Kopf benötigt wird. Es muss eine Leistungseinbuße hingenommen werden, um das Netz nicht mit zu vielen feinen Strömen zu belasten.

3.3.1 Label Distribution Protocol

Wurde durch einen LSR eine Verbindung zwischen FEC und Label geschaffen, bzw. zerstört, muss diese Information an andere LSR verteilt werden. Dies geschieht im Falle MPLS nicht durch Piggybacking auf einem existierenden Protokoll (z.B. BGP), sondern durch ein separates Protokoll, dem Label Distribution Protokoll (LDP).

Eine LDP-Nachricht hat folgendes Format:

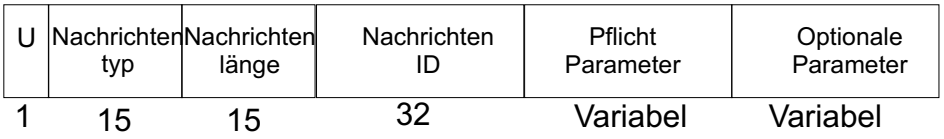


Abbildung 8: Format der LDP Nachricht

Erkennt ein LSR eine Nachricht nicht, signalisiert das U-Bit (Unknown Message), ob der Sender benachrichtigt werden soll. Das 15-Bit-Feld (Message Type) beinhaltet eine der zehn definierten Nachrichtenarten (1. Hello Message, 2. Initialisierung einer LDP Sitzung, 3. Keep Alive Message, falls Nachrichten ausbleiben, 4. Address Message, um Schnittstellenadressen zu melden, 5. Address Withdraw Message, um zuvor gemeldete Schnittstellenadresse zurückzuziehen, 6. Label Mapping, um Label-Bindung zu melden, 7. Label Request, um Label-Verbindung zu erfragen, 8. Label Withdraw, um zuvor erstellte FEC-Label-Verbindung zurückzuziehen, 9. Label Release, um FEC-Label-Verbindung aufzulösen und 10. Notification Message, zum Signalisieren von Meldungen und Fehlern). Es folgt die 16-Bit Nachricht (Message Length) zur Nachrichtenlänge in byte, sowie die 32-Bit Nachrichtenidentifikationsnummer (Message ID), welche z.B. eine Notification Message eindeutig bestimmt. Zur Übertragung von z.B. Class of Service (CoS) oder der Anzahl Hops, werden die Felder Mandatory Parameters und Optional Parameters genutzt.

Zwei LSRs müssen eine bidirektionale LDP Sitzung aufbauen, um Informationen auszutauschen. Hierzu senden die LSR periodische Hello-Nachrichten mit einer LDP Identifikation, die der LSR für diese Schnittstelle nutzen möchte. Das Erkennen eines neuen Nachbarn initiiert eine LDP Sitzung. Die LSRs öffnen eine TCP Verbindung, um die Sitzungsparameter auszuhandeln (z.B. die Protokollversion) und eröffnen die LDP Sitzung schließlich durch eine Keep Alive Nachricht. Hello Nachrichten werden weiterhin ausgetauscht; das Ausbleiben zeigt an, dass ein LSR den ausgehandelten Label-Raum nicht mehr benutzen möchte, oder ein Fehler aufgetreten ist. In beiden Fällen, sowie beim Ausbleiben der Keep Alive Nachricht, wird die LDP-Sitzung beendet (vgl. [ChOh99], S. 60).

Die Label-Bindung kann sowohl upstream, als auch downstream erfolgen, da jeder beliebige LSR in der Weiterleitungstabelle auf die FEC-Label-Abbildung Einfluss nehmen kann. Im Falle des Downstream Label Binding hat ein flussabwärts liegender LSR die FEC-Label-Abbildung erstellt; beim Upstream Label Binding wurde die FEC-Label Zuordnung von einem flussaufwärts liegenden LSR übernommen.

4 Die Welt mit MPLS

Ein offensichtlicher Vorteil den der Einsatz von MPLS gegenüber IP über ATM bietet, ist die stark vereinfachte Netzwerktopologie, die durch das Einrichten der LSPs entsteht. Dies zeigt der Vergleich der logischen IP über ATM Topologie und einer möglichen MPLS Topologie in folgender Abbildung (vgl. [Awdu99], S. 45):

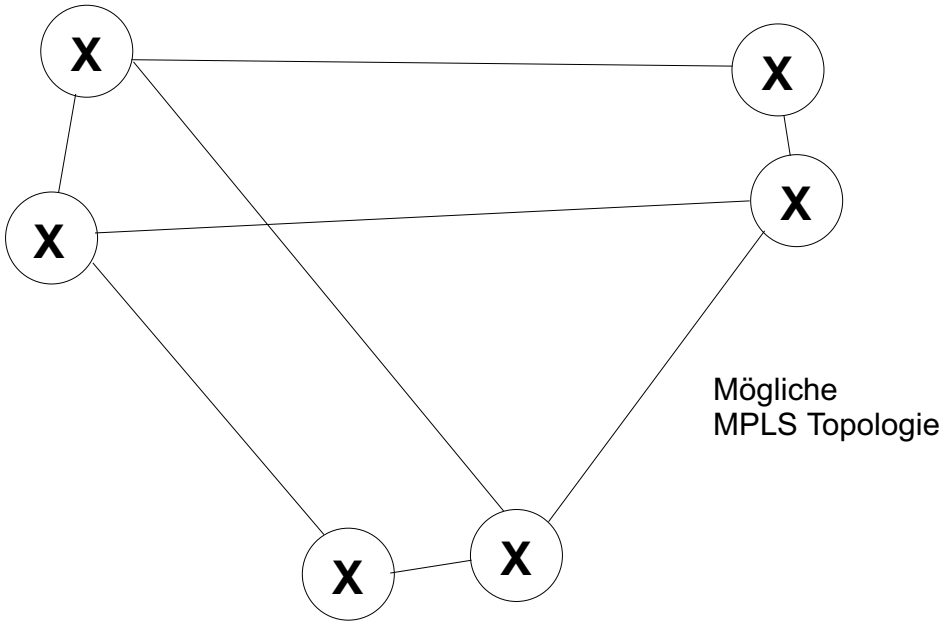


Abbildung 9: Vereinfachte Netzwerktopologie mit MPLS

Der größte Unterschied zu herkömmlichen IP Netzwerken, ist die Art des Traffic Engineering. Ohne MPLS sind stark überlastete Routen neben unausgelasteten Verkehrswegen anzutreffen. Dies kann mit MPLS erkannt und vermieden werden. Für Internet Service Provider stellt die effiziente Auslastung der Bandbreite einen enormen Wettbewerbsvorteil dar.

4.1 Traffic Engineering

Um den Verkehrsfluss im Netz zu optimieren, bietet MPLS große Flexibilität beim Aufbau eines Label Switching Path, der Verkehrszuweisung und Umverteilung, sowie Möglichkeiten des Netzwerk-Monitoring.

Ein Werkzeug des MPLS-Traffic Engineering ist der Einsatz des Constraint Based Routing (CR). CR erlaubt es einen Pfad aufzubauen, der z.B. nach dem Kriterium Anzahl Hops optimal ist und zugleich bestimmten Nebenbedingungengenügt.

4.1.1 Constraint Based Routing

In einem MPLS Netzwerk muss ein Label Switching Path zuerst aufgebaut werden, bevor der Verkehr weitergeleitet werden kann. Dies kann auf zwei Arten geschehen (vgl. [GJFAS+99], S. 51):

1. control-driven: jeder LSR bestimmt den nächsten Wegabschnitt anhand der IP-Weiterleitungstabelle. Die Instandhaltung des LSP erfolgt durch das LDP.

2. constraint based: die optimale Route für den LSP, die gegebene Nebenbedingungen nicht verletzt, ist in einer *setup message* angegeben, die jeden Knoten passiert und, durch das Versenden von label *request messages* zur jeweils nächsten Schnittstelle, den gewünschten Pfad schrittweise aufbaut. Dies geschieht mit Hilfe eines Consraint Based Routing Label Distribution Protokoll (CR LDP), welches das LDP um die expliziten Routeninformationen, die Verkehrsparameter zur Ressourcenreservierung und Optionen zum

Abbauen der CR LSPs erweitert. Das CRLDP unterstütz das Anbieten von Differentiated Services. Hat die Signalisierung dieses CR LSP das Ziel erreicht und entspricht der Pfad den Wünschen der Ressourcenreservierung, wird die Label-Abbildungsinformation im Netz verteilt. Die Signalisierung ist im Erfolgsfall nach einem Umlauf der Signalisierungsdaten abgeschlossen. Falls alternative Routen verfügbar werden, die den Ressourcenrestriktionen genügen, können die CR LSPs umgeleitet werden (adaptive LSPs), oder auf dem festgelegten Pfad verbleiben (nicht adaptive LSPs). CR LSPs werden von einer Netzwerk Management Anwendung kontrolliert und können somit für das Traffic Engineering genutzt werden.

4.1.2 Traffic Monitoring

Das Netzwerk-Monitoring ist aus mehreren Gründen sinnvoll. Es ermöglicht die Überprüfung des QoS, das Erkennen alternativer Routen und das frühzeitige Aufspüren von Fehlerquellen oder Überlastungen. Hierzu hat die IETF das Simple Network Management Protocol (SNMP) um eine Real Time Flow Measurement (RTFM) Komponente zur Verkehrsmessung erweitert (vgl. [ChOh99], S. 61f).

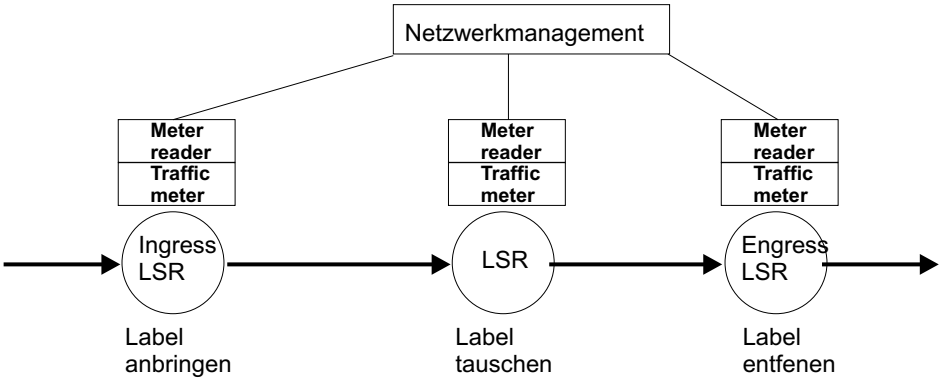


Abbildung 10: Traffic Monitoring

Diese Verkehrmesseinheiten können sowohl das gesamte Netzwerk, wie auch einzelne Flüsse, mit zuvor vom Netzwerkmanager definierten Merkmalen (z.B. Anzahl Pakete, Bytes), beobachten. Die Messergebnisse werden in einer Datenbank angelegt und können über die Meter Reader bei Bedarf abgerufen werden und Applikationen zur Verarbeitung bereit gestellt werden.

Wird z.B. der Fluss einer FEC an mehreren LSR gemessen, kann durch den Vergleich der Daten eine Stauung, bzw. Paketverlust erkannt werden. Für QoS notwendige Messungen, wie z.B. Verzögerung, können aber nicht durchgeführt werden.

4.2 VPN

Herkömmliche IP VPNs basieren auf der Einrichtung von Tunneln, über die IP-Pakete unabhängig von ihrer Zieladresse weitergeleitet werden. Diese Art der Weiterleitung lässt sich in der MPLS Welt durch das Aufbauen eines LSP zwischen zwei Routern eines VPNs realisieren. Der Ingress LSR am Tunnelanfang setzt ein neues Label auf den Label Stack . Die zwischenliegenden LSR leiten das Paket unabhängig von der IP-Zieladresse an das Tunnelende zum Engress LSR (vgl. [Brau99], S. 211).

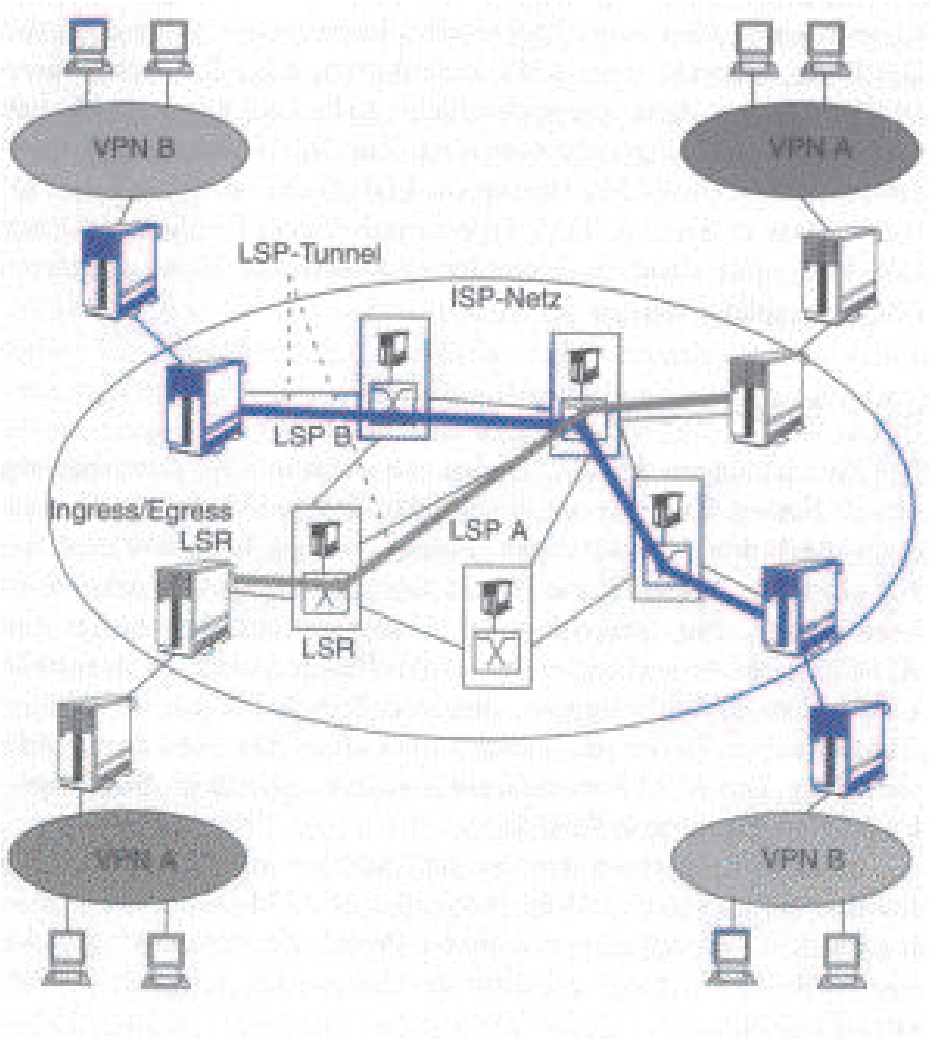


Abbildung 11: VPN und MPLS

5 Zusammenfassung

Die Attraktivität der MPLS Technologie liegt in den Optimierungsmöglichkeiten des Datenflusses. Die Verkehrslast kann durch einfache Label-Operationen umgeleitet werden. Das Monitoring des Netzstatus liefert die hierzu nötigen Impulse und kann in Stau oder Fehlersituationen eingreifen, ohne unzählige virtuelle Verbindungen neu aufbauen zu müssen. MPLS erleichtert das Anbieten von Diensten im Internet, was einerseits durch den Einsatz des Constraint Based Routing zum gezielten Verbindungsaufbau, andererseits durch die Effizienzsteigerung im Netz möglich ist. Zwar bietet MPLS keine direkte Lösung zur Differenzierung von Dienstgüte im Netz, ist aber z.B. durch das Nutzen der experimentellen Bits im Shim Header, oder das Einrichten von separaten FECs für die jeweiligen Güteklassen ein hilfreiches Werkzeug. Im Feld VPNs bietet sich MPLS durch den Aufbau von LSPs an. Einen Geschwindigkeitsvorteil erzielt MPLS durch das mögliche Switching auf Schicht 2. Das langsamere Routing auf Schicht 3 kann, falls sinnvoll, umgangen werden.

MPLS stellt eine leistungsstarke Erweiterung zur existierenden IP-Routing-Architektur dar.

Literatur

[Awdu99] D. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine*, Dezember 1999, S. 42–47.

[Blac01] U. Black. *MPLS and Label Switching Networks*. Prentice Hall, Upper Saddle River, NJ. 2001.

[Brau99] Torsten Braun. *IPnG: Neue Internet-Dienste und virtuelle Netze*. d.punkt Verlag. 1999.

[ChOh99] Th. Chen und T. Oh. Reliable Services in MPLS. *IEEE Communication Magazine*, Dezember 1999, S. 58–62.

[DaRe00] B. Davie und Y. Rekhter. *MPLS:Technology and Applications*. Morgan Kaufman, San Fancisco. 2000.

[GJFAS+99] A. Ghanwani, B. Jamoussi, D. Fedyk, P. Ashwood-Smith, L. Li und N. Feldman. Traffic Engineering Standards in IP Networks Using MPLS. *IEEE Communications Magazine*, Dezember 1999, S. 49–53.

[Gree00] A. Greenville. The Magic behind the Myths. *IEEE Communications Magazine*, Januar 2000, S. 124–131.

[KrRe00] G. Krüger und D. Reschke. *Lehr und Übungsbuch Telematik*. Lehrbuchverlag Leipzig. 2000.

[Swal99] G. Swallow. MPLS Advantages for Traffic Engineering. *IEEE Communications Magazine*, Dezember 1999, S. 54–57.

Policy-based Networking

Georg Kassner

Kurzfassung

Diese Seminararbeit beschäftigt sich mit dem Aufbau, der Funktionsweise und den Einsatzmöglichkeiten von Policy-basierten Netzwerken. Zunächst werden die Vorteile von Policy-based Networking gegenüber herkömmlichen Verfahren erläutert und der Begriff Policy näher erklärt. Daran anschliessend werden mögliche Einsatzräume für Policy aufgezeigt. Es folgen grundsätzliche Anforderungen, die der Einsatz von Policy an ein Netzwerk stellt. Danach wird die Architektur eines Policy-basierten Netzwerks dargestellt, dessen Funktionsweise erklärt und auf die zum Einsatz kommenden Protokolle eingegangen. Die Arbeit schließt mit der Betrachtung von Beispielszenarien für den Einsatz von Policy-based Networking.

1 Einleitung

1.1 Warum Policy?

Policy-based Networking ist in der letzten Zeit zu einem vieldiskutierten Ansatz im Bereich des Netzwerkmanagements geworden. Bedingt durch den Einsatz immer größer werdender IP-Netzwerke müssen zunehmend mehr und unterschiedliche Netzwerkkomponenten verwaltet und konfiguriert werden. Je mehr Komponenten eingeführt werden, desto mehr Aufwand entsteht dadurch den Systemadministratoren bei Konfigurierung und Management. Nach [MBHS00] besteht als Konsequenz dessen eine große Nachfrage nach vereinfachten Möglichkeiten des Netzwerkmanagements. Gewünscht werden unter anderem ein zentralisiertes Management, vereinfachte Managementdaten, die Anwendung gemeinsamer, standardisierter Regeln und Mechanismen, sowie die Möglichkeit deren Wiederbenutzung.

Es wird also eine Möglichkeit gesucht, ein Netzwerk als Ganzes zu kontrollieren und zu verwalten. Würde dies realisiert durch ein traditionelles Management-Werkzeug, welches jedes Netzwerkelement separat konfiguriert und verwaltet, bedeutete das nach [MMSS$^+$99] eine unverhältnismäßig hohe Last für die Management-Applikation. Der Einsatz von Policy stellt einen alternativen Ansatz hierzu dar. Im Gegensatz zu herkömmlichen Netzwerkmanagementansätzen werden hierbei vorgeschriebene Regeln für Management und Konfiguration zentral abgelegt und so das Verhalten der Netzwerkelemente netzwerkweit kontrolliert, anstatt jedes Element oder Protokoll für sich separat zu konfigurieren und zu verwalten.

Die Idee von Policy hat ihren Ursprung im RSVP-Protokoll bei der Reservierung von Ressourcen. Hierbei geht es darum, ob einem Benutzer Zugang zu gewissen Ressourcen gewährt werden soll. Neben der Ressourcenreservierung gibt es aber noch weitere Aspekte von Policy. Zum einen eignet sich Policy dazu, den Zugang von Benutzern zu einem Netzwerk zu regeln und zu verwalten, zum anderen ist Policy ein effizientes Werkzeug zur Konfiguration von Netzwerkelementen.

1.2 Was ist Policy?

Da der Begriff „Policy“ in vielen verschiedenen Zusammenhängen auftritt, ist es unumgänglich, Policy im Zusammenhang mit Netzwerken zu definieren. Im Wesentlichen ist Policy nach [RVKF+99] die vereinheitlichte Regulierung von Zugang zu und Verwaltung von Netzwerkressourcen und -diensten, basierend auf administrativen Kriterien.

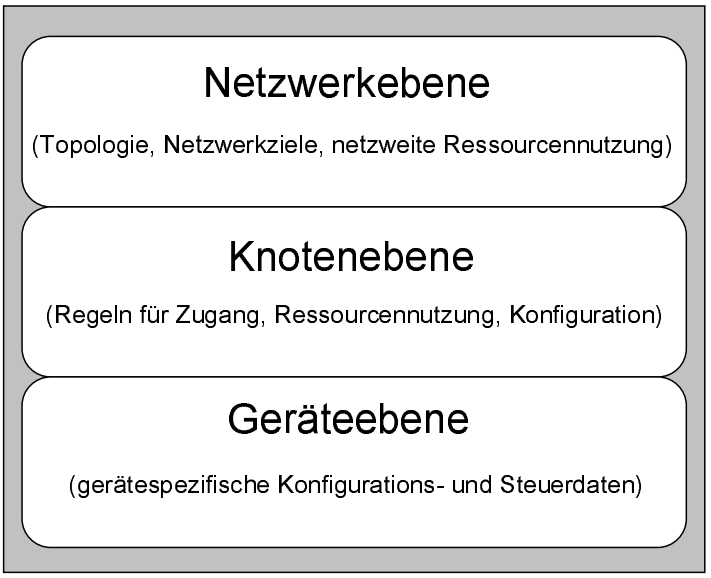


Abbildung 1: Policy Hierarchie nach [RVKF+99]

Abbildung 1 beschreibt verschiedene Schichten in denen Regulierung beschrieben und ausgeführt werden kann. Im Blick auf die *Netzwerkebene* kann man Policy als Gesamtheit von Topologie, Netzwerkzielen und netzweiter Ressourcennutzung betrachten. Dieser Blickwinkel setzt sich aus verschiedenen *Knotenansichten* zusammen, die den Policy-Zielen und -Anforderungen an den verschiedenen Netzwerkknoten entsprechen. Diese wiederum werden durch *Policy Regeln* beschrieben, durch welche die einzelnen Netzwerkknoten kontrolliert werden. Da die einzelnen Netzwerkelemente unterschiedliche, herstellerspezifische Mechanismen für Konfiguration und Ressourcenallokation besitzen, müssen diese Regeln in gerätespezifische Befehle übersetzt werden. Beispielsweise könnte es auf Netzwerkebene die Absicht des Administrators sein, für den gesamten HTTP-Verkehr von Server A zu den Endsystemen B 2Mb/s zu reservieren. Auf *Knotenebene* würde dies bedeuten, dass Server A den IP Header für die von den Endsystemen B aufgerufenen Webseiten entsprechend markiert, damit ein Zugangsrouter diese Pakete einer entsprechend eingerichteten Reservierung zuordnen kann. Diese somit vorgegebenen Regeln müssen nun auf die unterste Ebene, die *Geräteebene*, übersetzt werden, damit die einzelnen unterschiedlichen Netzwerkkomponenten sich entsprechend verhalten und die vom Administrator verlangten 2 Mb/s realisiert werden.

2 Anwendungsgebiete von Policy-based Networking

2.1 Zugangskontrolle

Wie eingangs erwähnt stellt die Zugangskontrolle ein mögliches Anwendungsgebiet von Policy dar. Es geht hierbei darum, die Verwaltung, Kontrolle und Regelung von Zugangsberechtigungen der Benutzer in einem Netzwerk durch Policy möglichst einfach zu gestalten. Im Gegensatz zu traditionellen Verfahren müsste die Management-Applikation nicht mehr jeden

Zugangsknoten separat verwalten, sondern es würden Regeln zur Beschreibung der für den jeweiligen Benutzer gültigen Berechtigungen im Netzwerk deponiert. Im Falle einer Benutzeranfrage könnte der entsprechende Zugangsrouter dann auf dieses Regelwerk zugreifen und den Benutzer entsprechend dieser Vorgaben zulassen oder auch nicht.

2.2 Ressourcenreservierung

Auch für die Reservierung von Ressourcen für bestimmte Benutzer oder Applikationen ist der Einsatz von Policy geeignet. Zwar kann ein Netzwerkknoten lokal entscheiden, ob er für eine Reservierungsanfrage genügend Ressourcen zur Verfügung hat, jedoch kennt er im Regelfall nicht den Gesamtstatus des Netzwerks oder den Anteil der Ressourcen, welche besagter Benutzer bereits insgesamt im Netzwerk belegt. Um hier eine korrekte und schnelle Entscheidung zu treffen, sind Mechanismen gefragt, welche es dem Netzwerkknoten erlauben, notwendige Informationen über den Netzwerkstatus einzuholen und auf deren Grundlage eine Entscheidung über die Reservierungsanfrage zu treffen.

2.3 Konfiguration

Ein dritter Aspekt für den Einsatz von Policy stellt die Konfiguration von Netzwerkelementen dar. Bedingt durch die im Regelfall hohe Zahl verschiedener Netzwerkelemente müsste ein traditionelles Management-Werkzeug detaillierte gerätespezifische Kenntnisse über viele verschiedene Typen von Netzwerkelementen haben, was zwangsläufig zu einer hohen Komplexität des entsprechenden Werkzeugs führen würde. Über den Einsatz von Policy kann die Konfiguration von Netzwerkelementen vereinfacht werden. Die Elemente selbst fragen die für sie passenden Konfigurationsdaten aus dem Netzwerk ab und wenden sie an. Sie müssen also nicht mehr zentral von einer Management-Applikation angesteuert werden, was den Administrationsaufwand deutlich verringert.

3 Grundlagen

In diesem Abschnitt sollen die Grundlagen für Policy-based Networking dargestellt und erläutert werden. In [YaPG00] werden die folgenden Anforderungen an die einem Policy-basierten Netzwerk zugrunde liegenden Methoden und Protokolle aufgezeigt.

- Die Methoden und Protokolle müssen so entwickelt sein, dass sie die Anforderungen der policy-basierten Netzwerkkontrolle für das entsprechende Problem erfüllen. Als Haupteinsatzgebiete von Policy sind hier wiederum Zugangskontrolle, Ressourcenreservierung und Konfiguration der Netzwerkkomponenten zu nennen.
- Die Methoden und Protokolle müssen Verdrängung unterstützen. Verdrängung bedeutet hier, dass die Möglichkeit besteht, zuvor etablierte Einstellungen wieder zu entfernen, um beispielweise neuen Anfragen nach Ressourcen gerecht zu werden, die Konfiguration eines Netzwerkelements zu ändern oder gewährte Zugangsberechtigungen für bestimmte Benutzer wieder aufzuheben.
- Es muss die Möglichkeit für Monitoring und Accounting vorhanden sein. Monitoring bedeutet, dass Policy Status und Ressourcenbenutzung abgefragt werden können, um somit über die vorhandenen Kapazitäten Bescheid zu wissen. Ausserdem ist es wichtig, Benutzungs- und Zugangsinformationen abfragen zu können, um Daten für das Accounting zu haben. Als Beispiel diene hier die Aufzeichnung der Nutzungsinformation für die Abrechnung eines ISP mit seinen Benutzern.

- Es müssen Bestimmungen für Fehlertoleranz und Wiederanlauf nach Fehlern wie Ausfällen von Netzwerkelementen und Kommunikationsunterbrechungen zwischen verschiedenen Einheiten vorhanden sein.

- Auch so genannte Policy Ignorant Nodes, also Netzwerkelemente, die nicht direkt über den Einsatz von Policy verwaltet werden können, müssen unterstützt werden.

- Die Policy-basierte Architektur muss auch in gewissen Belangen Sicherheit garantieren. Die Bedrohung durch zum Beispiel durch „Denial of Service“ muss minimiert werden. Weiterhin muss garantiert sein, dass die in die Policy-Kontrolle eingebundenen Einheiten sich gegenseitig erkennen und verifizieren können, bevor sie miteinander kommunizieren.

4 Architektur, Funktionsweise und Protokolle

Dieser Abschnitt stellt die prinzipielle Architektur eines Policy-basierten Netzwerks vor und erläutert dessen Funktionsweise. Weiterhin wird auf die existierenden Protokolle eingegangen, welche die Bausteine eines Policy-basierten Netzwerks benutzen. Insbesondere wird das Common Open Policy Service (COPS) Protokoll näher beleuchtet.

4.1 Architektur und Funktionsweise des Policy-based Networks

4.1.1 Bausteine

Die Komponenten eines Policy-basierten Netzwerks lassen sich nach [MMSS+99], [RVKF+99] und [YaPG00] in vier funktionale Gruppierungen oder Bausteine einteilen.

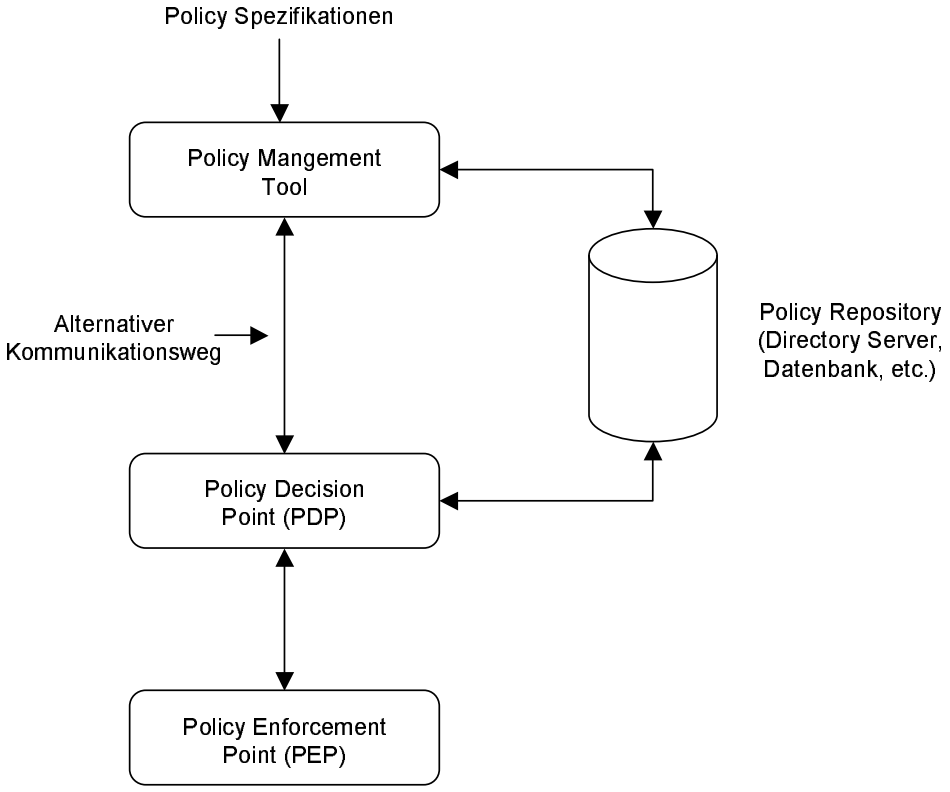


Abbildung 2: Bausteine eines Policy-basierten Netzwerks

Wie in Abbildung 2 zu sehen ist, setzt sich das Netzwerk aus den Bausteinen *Policy-Management-Werkzeug*, *Policy Repository*, *Policy Decision Point (PDP)* und *Policy Enforcement Point (PEP)* zusammmen. Die PDPs sind die Netzwerkelemente, welche auf Grundlage der in einem Policy Repository abgelegten Regeln Entscheidungen treffen, während die PEPs für die Umsetzung dieser Entscheidungen zuständig sind.

Das Policy-Management-Werkzeug stellt die Schnittstelle zwischen System und Benutzer, in diesem Fall der Netzwerkadministrator, dar. Der Administrator kann über das Management-Werkzeug die gewünschten Service Level Objectives (SLOs), Benutzerrechte oder Konfigurationsrichtlinien in das System einspeisen. Da diese zumeist sehr allgemein gehalten sind und noch zu abstrakt sind, dass sie das System ausführen könnte, muss das Management-Werkzeug die Fähigkeit besitzen, diese Daten genauer zu spezifizieren und in für die PDPs verständliche Regeln zu übersetzen. Diese so genannten *Policy Regeln* bestehen aus einer Menge von Aktionen die initiiert werden, wenn eine bestimmte Menge von Bedingungen erfüllt ist. Sie sind also von der Form IF <Menge von Bedingungen> THEN <Menge von Aktionen>. Beispielsweise im Falle des Einsatzes eines solchen Netzwerks für QoS hieße dies, „normale“ oder „premium“ Dienstgüte so genau zu spezifizieren, dass die PDPs mit diesen Informationen umgehen können. Weitere Funktionen des Management-Werkzeugs sind *Regelprüfung* und *Globale Konflikterkennung*. Unter Regelprüfung versteht man die Überprüfung der eingegebenen Regeln auf Korrektheit der Datentypen und auf korrekte Semantik. Die Globale Konflikterkennung überprüft, ob neu eingegebene Regeln Konflikte mit bereits existierenden Regeln aufweisen. Ein Konflikt besteht, wenn die Bedingungen mehrerer Regeln gleichzeitig erfüllt sind, diese Regeln aber sich widersprechende Aktionen verlangen. Hierbei werden so genannte *statische* Konflikte entdeckt, dynamische Konflikte, welche beispielsweise auf zeitbasierten Regeln beruhen, können im Management-Werkzeug noch nicht erkannt werden.

Nach Übersetzung und Validierung einer Regel wird diese im Policy Repository abgelegt. Es ist auch möglich, dass die Regel zuerst direkt an einen PDP weitergegeben wird und erst danach im Policy Repository gespeichert wird (siehe auch Abbildung 2). Auf das Policy Repository wird von den PDPs zugegriffen, um die Regeln abzurufen, welche sie für eine Entscheidung über eine Anfrage benötigen. Das Management-Werkzeug greift im Rahmen der Globalen Konflikterkennung ebenfalls auf das Policy Repository zu, um die notwendigen Informationen über bestehende Regeln für die Konfliktüberprüfung zu erhalten. Das Kommunikationsprotokoll zwischen Policy Repository und Management-Werkzeug, bzw. PDP ist abhängig von der Art des Repositorys. Ist dieses ein Verzeichnis, empfiehlt sich die Benutzung von LDAPv3, bei einer Datenbank beispielsweise SQL.

Die Auswertung der Policy-Bedingungen geschieht im Regelfall durch die PDPs, in bestimmten Fällen kann sie aber auch durch die PEPs oder durch beide Bausteine erfolgen. Der PEP ist der Baustein, der tatsächlich für die Behandlung und Weiterleitung von Daten (Paketen) und für die Anwendung und Ausführung von Policy-Aktionen verantwortlich ist. Er ist eine so genannte operationelle Komponente und führt Aufgaben wie Filtern, Markieren von Paketen, Ressourcenmanagement, etc. aus. Der PDP hingegen entscheidet, welche Aktionen für welche Pakete in Frage kommen. Er wertet Regeln für einen oder mehrere PEPs auf der Basis von Informationen aus Datenpaketen, des Netzwerkzustandes oder auch dynamischen Informationen wie Kontoständen (z.B. Freiminuten bei einem ISP) aus. Beispielsweise entscheidet der PDP ob eine Reservierungsanfrage positiv oder negativ beantwortet wird, in Abhängigkeit vom Ursprung der Anfrage oder der Netzwerkauslastung. Der PDP ist ebenfalls verantwortlich für *lokale Konflikterkennung*. Im Gegensatz zur globalen Konflikterkennung überprüft die lokale Konflikterkennung auf Konflikte zwischen verschiedenen Regeln, welche speziell die vom PDP kontrollierten Geräte betreffen. Ausserdem wird kontrolliert, ob für die Ausführung von Aktionen genügend Ressourcen vorhanden sind.

Neben der in Abbildung 2 gezeigten prinzipiellen Architektur eines Policy-basierten Netzwerks ist es auch denkbar, dass ein PEP noch über einen lokalen PDP, den so genannten LPDP, verfügt (Abbildung 3). In diesem Fall wird der PEP zunächst den LPDP kontaktieren, um zu einer lokalen Entscheidung zu gelangen, und ggf. erst danach Verbindung zu seinem PDP aufnehmen. Die Motivation für den Einsatz eines solchen LPDPs liegt in der damit verbundenen Effizienzsteigerung des Systems. Auch werden dynamische Informationen, die nur den entsprechenden Netzwerkknoten betreffen, besser direkt dort als in einem netzwerkweiten Verzeichnis abgelegt.

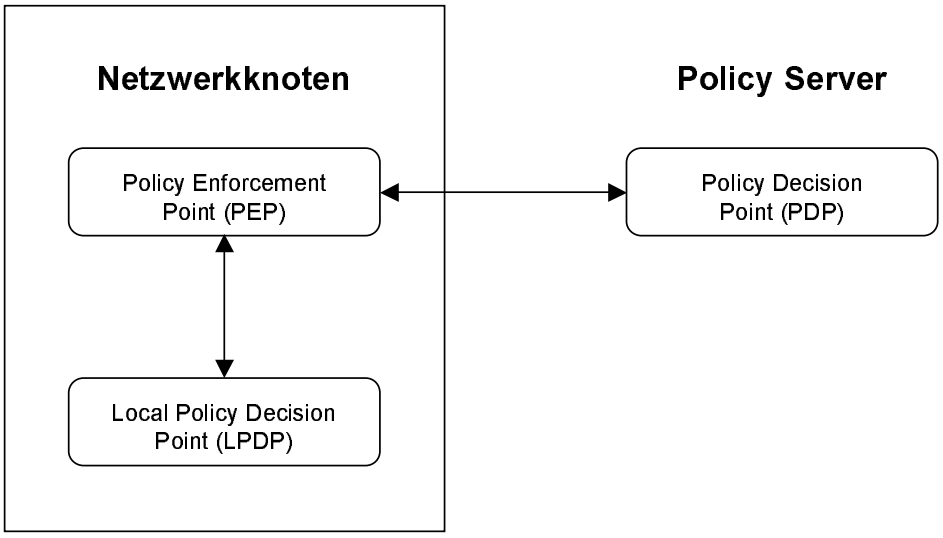


Abbildung 3: PEP, LPDP und PDP

4.1.2 Funktionsweise

Zentrale Elemente der in Abschnitt 4.1.1 vorgestellten Architektur sind PDP und PEP. Das Zusammenspiel der beiden Bausteine beginnt, wie in [YaPG00] und [MMSS+99] beschrieben, im Regelfall mit dem PEP. Der PEP erhält eine Nachricht, die eine Policy-basierte Entscheidung erfordert. Als Beispiel sei die Ankunft eines bestimmten Datenpakets am Netzwerkknoten des PEP genannt. In einem solchen Fall sendet der PEP eine Anfrage nach einer Entscheidung an seinen PDP. Diese Anfrage kann zusätzlich zu den Informationen aus der Nachricht, welche die Anfrage auslöste, ein oder mehrere *Policy-Elemente*, bzw. *Policy-Objekte* enthalten. Ein Policy-Element ist eine Untereinheit eines Policy-Objekts und enthält einzelne, zur Auswertung von Regeln notwendig Informationseinheiten. Es kann kann beispielsweise Benutzer- oder Anwendungsidentität beinhalten. Ein Policy-Objekt besteht aus Policy-Elementen und wird als Anfrage an den PDP oder als dessen Antwort bezüglich einer Entscheidung übergeben. Je nach Anwendungsfall kann es sich um Anfragen nach Ressourcenreservierung, Zugangsberechtigung für Benutzer oder Konfigurierungsdaten handeln. Der PDP greift bei Bedarf auf das Policy Repository zu, um die passenden Regeln abzurufen. Dann wertet der PDP diese Regeln aus, trifft eine Entscheidung bezüglich der Anfrage und gibt diese an den PEP zurück, welcher die Policy durchsetzt, indem er die angeschlossenen Geräte entsprechend konfiguriert. Der PDP kann auch zusätzliche Informationen an den PEP übergeben, die diesem beispielsweise zur Formulierung von Fehlermeldungen dienen. Bei Bedarf kann er zusätzlich auf andere externe Server zugreifen, zum Beispiel zur Benutzerauthentifizierung oder zum Einholen von Kontoinformationen.

Existiert neben dem PDP noch ein LPDP (siehe Abbildung 3), verläuft die Kommunikation wie folgt. Der PEP kontaktiert zunächst seinen LPDP, welcher ihm eine lokale Entscheidung

liefert. Diese Teilentscheidung wird mit der ursprünglichen Anfrage zusammen an den PDP übertragen. Dieser wertet die Anfrage des PEP aus und trifft eine Entscheidung. Diese vergleicht er dann mit der lokalen Entscheidung des LPDP und trifft auf der Grundlage dieser Daten eine endgültige Entscheidung, welche er dann an den PEP zurückgibt. Der PEP muss diese dann ausführen.

Zu beachten ist, dass der PEP den PDP auch dann kontaktieren muss, wenn die Anforderung an den PEP (z.B. eine Reservierungsanfrage) selbst keine Policy-Objekte enthält. Somit ist garantiert, dass eine Anforderung die Policy-Kontrolle nicht umgehen kann. Ist die lokale Entscheidung eines LPDP negativ, muss die Anfrage an den PDP nicht weitergeleitet werden, jedoch muss dieser trotzdem über das negative Ergebnis des lokalen Prozesses informiert werden. Ebenfalls wichtig ist, dass der PDP jederzeit unangefordert Anweisungen an seine PEPs senden kann, um zuvor getroffene und installierte Entscheidungen aufzuheben, Konfigurationen zu ändern oder Fehlermeldungen zu generieren.

Die Regeln werden jedoch nicht für jedes Netzwerkelement separat entwickelt, bedingt durch die Vielzahl verschiedener Systeme wäre dies nicht praktikabel. Vielmehr werden die Regeln für so genannte *Rollen* spezifiziert. Eine Rolle ist eine administrativ spezifizierte Eigenschaft eines Netzwerkelements. Über Rollen ist festgelegt, welche Regeln für ein bestimmtes Netzwerkelement angewandt werden können. Dies bedeutet, dass jedem Netzwerkelement eine oder mehrere Rollen zugeordnet werden. Ein PEP meldet dem PDP bei der Anfrage seine Rollen, woraufhin der PDP entscheiden kann, welche Regeln im speziellen Fall anzuwenden sind.

4.2 Policy Protokolle

Die in Abschnitt 4.1 beschriebene Netzwerkarchitektur und Funktionsweise stellt logischerweise verschiedene Anforderungen an die Kommunikationsprotokolle der einzelnen Bausteine. Im Folgenden wird insbesondere auf die Erfordernisse des Kommunikationsprotokolls zwischen PEP und PDP näher eingegangen. Diese werden in [YaPG00] wie folgt beschrieben:

- *Zuverlässigkeit*: Die Wichtigkeit und Sensitivität von Policy-Kontrolldaten erfordert zuverlässige Verarbeitung. Unbemerkter Verlust von Anfragen oder Antworten kann zu inkonsistenter Netzwerkkonfiguration führen. Insbesondere im Fall von Billing und Accounting ist eine zuverlässige Verarbeitung daher unerlässlich.

- *Kurze Verzögerungen*: Um eine zügige Weiterleitung der am Netzwerkknoten ankommenden Pakete zu gewährleisten, darf die Kommunikation zwischen PEP und PDP keine großen Verzögerungen zu der zur Entscheidungsfindung benötigten Zeit hinzufügen.

- *Unterstützung von PEP-initiierter 2-Weg-Kommunikation*: Das Protokoll muss Verhandlungen zwischen PEP und PDP ermöglichen. Insbesondere muss der PEP Anfragen, Neuverhandlung zuvor erfolgter Entscheidungen, und den Austausch von Policy-Information initiieren können. Auch muss es ihm möglich sein, Monitoring-Information und Policy-Statusänderungen an den PDP zu übermitteln.

- *Unterstützung einseitiger Benachrichtigung*: Dies ist wichtig, um sowohl PEP als auch PDP die Benachrichtigung des jeweils anderen über asynchrone Statusänderungen zu ermöglichen. Zum Beispiel soll der PDP seinen PEP informieren können, wenn der Kredit eines Benutzers aufgebraucht ist, damit der PEP gegebenenfalls die Reservierung für den entsprechenden Benutzer aufheben kann. Gleichfalls muss der PEP befähigt sein, den PDP über eventuelles Versagen bei der Ausführung von Policy-Aktionen zu informieren.

Für die Kommunikation zwischen PEP und PDP bietet sich das Common Open Policy Service (COPS) Protokoll an, welches in Abschnitt 4.3 näher beschrieben wird. Dieses ist zur Zeit das Standardprotokoll für den Austausch von Policy-Information und Entscheidungen. Das Kommunikationsprotokoll zwischen PDP und Policy Repository hängt wie in Abschnitt 4.1.1 erwähnt von der Art des verwendeten Repositorys ab. Handelt es sich um ein Netzwerkverzeichnis, wird zumeist das Lightweight Directory Access Protocol (LDAP) verwendet, beim Einsatz von Datenbanken könnten SQL Anfragen benutzt werden. Zur Authentisierung von Benutzern kann das DIAMETER-Protokoll zum Einsatz kommen.

4.3 Das „Common Open Policy Service“-Protokoll COPS

Dieser Abschnitt gibt einen Überblick über das im Regelfall zur Kommunikation zwischen PEPs und PDPs verwendete Common Open Policy Service Protokoll, welches in [BCDR+00] ausführlich beschrieben ist. Zu den Hauptcharakteristiken dieses Protokolls gehören:

- Das COPS-Protokoll ist Client-Server-basiert. Ein Client, der PEP, sendet Anfragen an einen Server, den PDP, welcher diese verarbeitet und ein Ergebnis an den Client zurückschickt.
- Das Protokoll benutzt TCP als Transportprotokoll, um zuverlässigen Datenaustausch zwischen PEP und PDP zu gewährleisten.
- Das Protokoll ist erweiterbar in der Hinsicht, dass es PEP-spezifische Informationen unterstützt, ohne dass es Veränderungen am COPS-Protokoll selbst bedarf.
- COPS bietet Sicherheit auf Nachrichtenebene für Authentisierung, Replay Protection und Nachrichtenintegrität. Ausserdem kann COPS bestehende Sicherheitsprotokolle wie IPSec oder TLS zur Authentifizierung und Sicherung des Kommunikationskanals zwischen PEP und PDP wiederverwenden.
- COPS basiert auf Zustandshaltung. Das bedeutet, dass Anfragen des PEP vom PDP gespeichert werden, bis der PEP sie explizit löscht. Weiterhin kann der Server Konfigurationsdaten auf dem Client installieren, welche dieser hält, bis sie vom Server wieder entfernt werden.

Wenn der PEP eine Anfrage an den PDP sendet, erwartet er vom PDP, dass dieser Einheiten von Daten über so genannte Entscheidungsnachrichten zurückschickt. Die Art der Daten ist vom jeweiligen Einsatzfall der Policy abhängig, es können Konfigurationsdaten, Entscheidungen über Ressourcenallokation oder Benutzerrechte sein. Nach erfolgreicher Installation einer solchen Einheit auf dem PEP, sendet dieser eine Bestätigungsmitteilung an den PDP. Dieser kann dann im Bedarfsfall diese Konfigurationsdaten über neue Entscheidungsnachrichten aktualisieren oder auch löschen. Wenn der PDP eine solche Entscheidung zur Löschung an den PEP überträgt, löscht dieser die spezifizierte Konfiguration und teilt dies dem PDP mit.

Um zwischen verschiedenen Arten von Clients unterscheiden zu können, muss der Clienttyp in jeder Nachricht identifiziert werden, da unterschiedliche Clients unterschiedliche Arten von Policy-Entscheidungen benötigen können. Dies ist wichtig, da im Regelfall mehrere Clienttypen über eine client-initiierte TCP-Verbindung mit dem Server, also dem PDP, kommunizieren.

Wie bereits in Abschnitt 3 angesprochen, gehört Fehlertoleranz zu den geforderten Eigenschaften des Protokolls. Diese wird dadurch erreicht, dass PEP und PDP ständig ihre Verbindung durch Keep-Alive-Nachrichten überprüfen. Wird hierbei eine Unterbrechung der Verbindung

entdeckt, versucht der PEP diese wiederherzustellen, oder eine Verbindung zu einem alternativen PDP aufzubauen. Solange die Verbindung unterbrochen ist, trifft der PEP lokal seine Entscheidungen. Wird die Verbindung zum PDP wiederhergestellt, informiert der PEP diesen über alle Statusänderungen, die während der Zeit der Unterbrechung eingetreten sind. Der PDP kann seinerseits den PEP anweisen, den Status, der vor Eintritt der Unterbrechung vorlag, wiederherzustellen.

4.3.1 Der Common Header

Jede COPS-Nachricht besteht aus dem COPS-Header, gefolgt von verschiedenen Objekten. Der Header hat das in Abbildung 4 dargestellte Format. Er besteht den folgenden Feldern:

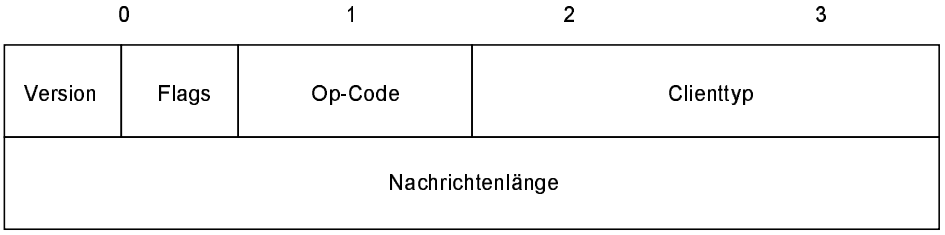


Abbildung 4: Common Header

- Version (4 bit): COPS-Version, zur Zeit 1
- Flags (4 bit): definierte Flag-Werte
- Op-Code (8 bit): bezeichnet die Art der COPS-Nachricht
- Clienttyp (16 bit): Art des Clients
- Nachrichtenlänge (32 bit): Gesamtlänge der COPS-Nachricht inklusive Header und eingebetteten Objekten.

4.3.2 Nachrichtentypen

COPS kennt verschiedene Nachrichtentypen, die im Folgenden kurz dargestellt werden.

- *Request (REQ)*: Anfrage des PEP an den PDP
- *Decision (DEC)*: Antwort des PDP auf eine REQ-Nachricht des PEP
- *Report State (RPT)*: Nachricht des PEP an den PDP über Erfolg oder Misserfolg bei der Ausführung einer Entscheidung
- *Synchronize State Request (SSQ)*: Mitteilung des PDP an den PEP, dass dieser seinen Status zum Zweck der Synchronisierung dem PDP mitteilen soll
- *Client-Open (OPN)*: Nachricht des PEP an den PDP über von ihm unterstützte Clienttypen oder über den zuletzt für einen bestimmten Clienttyp verbundenen PDP
- *Client-Accept (CAT)*: Positive Antwort des PDP auf eine OPN-Nachricht
- *Client-Close (CC)*: Nachricht, dass ein bestimmter Clienttyp nicht mehr unterstützt wird. Eine CC-Nachricht kann sowohl vom PEP als auch vom PDP ausgehen.

- *Keep-Alive (KA)*: Nachricht des PEP an den PDP, die in vorgegebenen Abständen zur Überprüfung der Verbindung gesendet werden muss
- *Synchronize State Complete (SSC)*: Benachrichtigung des PEP an de PDP nach erfolgter Antwort auf einen Synchronize State Request des PDP

4.3.3 Handle-Objekt

Das Handle-Objekt enthält einen Wert zur Identifizierung eines installierten Status. Es wird von den meisten COPS-Nachrichten benutzt. Besondere Beachtung verdient hierbei der Client-Handle. Client-Handles werden vom PEP bei einer Request-Nachricht erzeugt, um diesen Request-Status mit einem bestimmten Clienttyp zu verknüpfen. Mit diesem Client-Handle kann der Request des PEP für einen Clienttyp in allen nachfolgenden Nachrichten eindeutig identifiziert werden.

5 Anwendungsbeispiele

Wie bereits weiter oben mehrfach erwähnt, liegen die Einsatzmöglichkeiten von Policy in der Zugangskontrolle, der Ressourcenreservierung und der Konfigurierung von Netzwerkelementen. Im Rahmen der Zugangskontrolle kann ein Zugangsrouter bei einem PDP anfragen, ob einem bestimmten Benutzer Zugang zu gewähren ist oder nicht. Ressourcenreservierung in Abhängigkeit von Tageszeit, Netzwerkauslastung und Benutzeridentität kann ebenfalls durch den Einsatz von Policy-Regeln vorgenommen werden. Auch die Konfigurierung verschiedener Klassen von Netzwerkelementen auf der Basis von zentral abgespeicherte Regeln stellt eine Einsatzmöglichkeit von Policy dar. Die folgenden zwei Abschnitte zeigen nun an praktischen Beispielen, welche weiteren Optionen Policy bieten kann.

5.1 Prepaid-Account

Eine Möglichkeit des Einsatzes von Policy-based Networking liegt in den Bereichen, in denen ein Benutzer so genannte Prepaid-Karten oder Prepaid-Accounts erwirbt, also einen gewissen Kredit, und diesen durch Nutzung der Leistungen eines Providers (Mobilfunk, Internet, etc.) sukzessive dezimiert. Der User kontaktiert seinen Provider und teilt diesem, beispielsweise über eine RSVP RESV message, die ID Nummer seiner Prepaid-Karte oder seines Prepaid-Accounts mit. Diese ID ist in ein Policy-Objekt eingebettet. Dieses Objekt wird in Policy-fähigen Routern verarbeitet und der Kredit des Benutzers entsprechend verringert.

Angenommen Benutzer U in Abbildung 5 möchte die Dienste seines Providers P in Anspruch nehmen. U generiert daher ein Policy-Objekt, welches die ID Nummer seines Prepaid-Accounts enthält. Dieses Objekt wird zusammen mit anderen Policy-Objekten in einer RESV message an den Provider P übertragen, wo es an einen zuständigen PEP weitergeleitet wird. Der PEP kontaktiert nun seinerseits seinen PDP und stellt an diesen eine Reservierungsanfrage für Benutzer U. Der PDP wird nun eine Datenbank D kontaktieren, um festzustellen, ob der verbleibende Kredit von U ausreichend für die angeforderte Reservierung ist. Entsprechend dieser Information wird der PDP die Reservierung akzeptieren oder nicht, und dem PEP eine entsprechende Entscheidung übermitteln. Ausserdem muss der PDP den Kredit von U in der Datenbank D entsprechend verringern. Um zu wissen, wann der Kredit von U aufgebraucht ist, muss der PDP die Datenbank D regelmäßig konsultieren, den Stand abfragen und auch weiter verringern. Ist der Kredit aufgebraucht, ist die Reservierung vom PDP aufzuheben, d.h. der PEP ist zu informieren, dass die Reservierung von U zu löschen ist.

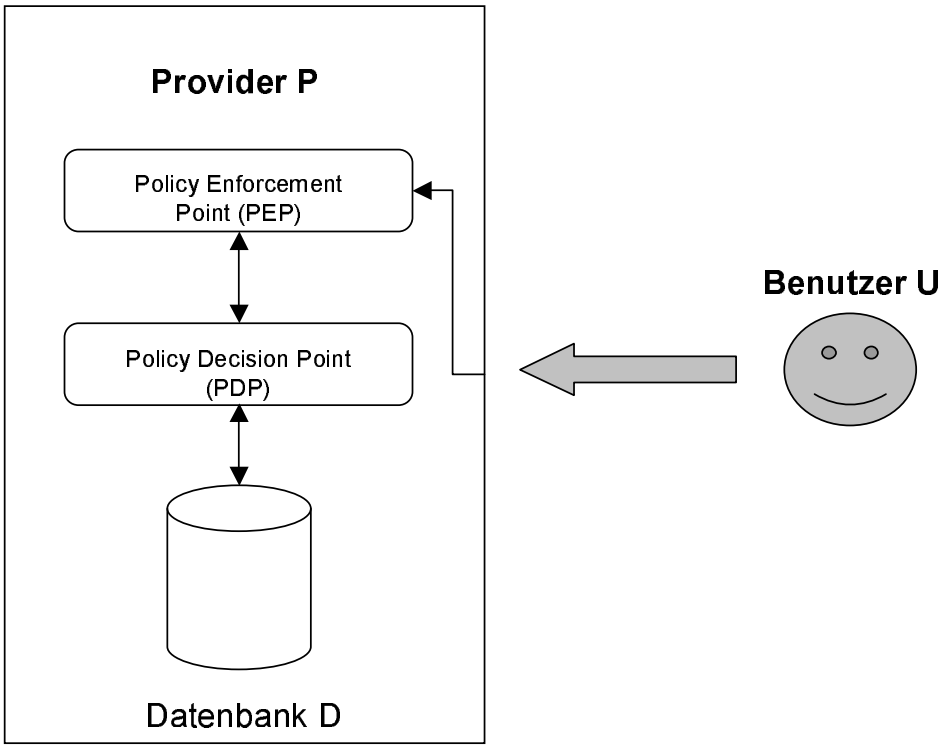


Abbildung 5: Policy-based Networking und Prepaid-Accounts

5.2 Interdomain Policy Architecture

Bisher wurde die Anwendung von Policy nur im Rahmen einer einzelnen administrativen Domäne betrachtet. Es ist aber unrealistisch, in einem Multi-Domänen-Netzwerk davon auszugehen, dass das gesamte Netz über ein einziges Policy-System kontrolliert wird. Dieser Abschnitt zeigt daher, wie Policy-Information zwischen unabhängigen Domänen über bilaterale Mechanismen, die so genannten *Bandwidth Broker (BBs)*, ausgestauscht wird. Hierbei muss man sich die BBs als PDPs vorstellen, welche Anfragen aus anderen Domänen verarbeiten.

Jede Domäne steht unter der Kontrolle eines Bandwidth Brokers. Benachbarte Domänen handeln ein Service Level Agreement (SLA) aus, welches die Art und den Umfangs des Datenverkehrs zwischen den beiden Domänen beschreibt. Angenommen, es bestünde ein SLA zwischen zwei Domänen, in welchem „Premium Service“ vereinbart ist, also Reservierung von Bandbreite für gewisse Verkehrsprofile. Auf dieser Basis treffen die BBs der beiden Domänen ein Traffic Control Agreement (TCA), welches aus einer Menge von Verkehrskontrollparametern für das vereinbarte Verkehrsprofil enthält.

Abbildung 6 zeigt die prinzipielle Architektur eines solchen Netzwerks. Die Router R1 und R2 an den jeweiligen Domänengrenzen spielen eine besondere Rolle bei der Umsetzung des SLA zwischen den Domänen und stellen die zu den BBs gehörenden PEPs dar. Das TCA stellt den Teil der Parameter des SLA dar, welcher für R1 und R2 zur Umsetzung wichtig ist. Angenommen Domäne 1 stelle ein Intranet dar, Domäne 2 einen ISP. Domäne 1 benötige eine 1 Mb/s Verbindung zu seinem ISP. Der Bandwidth Broker von Domäne 1 (BB1) fordert daher ein entsprechendes SLA von BB2 an. BB2 überprüft daraufhin, ob entsprechende Ressourcen vorhanden sind. Ist dies der Fall, sendet er ein von dem SLA abgeleitetes TCA an seinen Router R2 und gibt einen positiven Bescheid an BB1 zurück, worauf BB1 ebenfalls ein ähnliches TCA an seinen Router R1 überträgt.

Sollte Domäne 2 seinerseits mehr Ressourcen, z.B. von einem Backbone-ISP, benötigen, um die Anforderung von Domäne 1 befriedigen zu können, wird BB2 erst den Bandwith Bro-

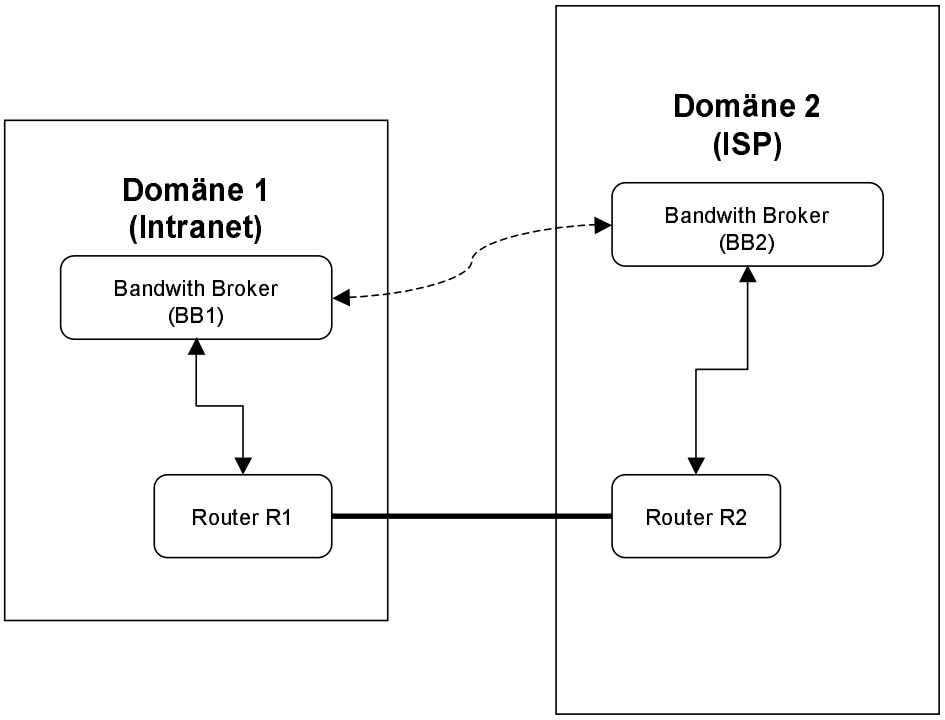


Abbildung 6: Interdomänen Policy-Architektur

ker des Backbone-ISP kontaktieren. Erst wenn dieser einen positiven Bescheid an BB2 zurückgegeben hat und die entsprechenden TCAs installiert sind, kann BB2 seinerseits eine Reservierungsbestätigung an BB1 senden und seinem Router R2 ein entsprechendes TCA übermitteln.

Daraus ist ersichtlich, dass bilaterale Vereinbarungen zwischen verschiedenen Domänen iterativ fortsetzbar sind.

6 Zusammenfassung und Ausblick

Policy-based Networking ist nicht nur ein Schlagwort. Es besteht der Bedarf nach einem System, das es ermöglicht, Netzwerke effizienter als ein Ganzes zu administrieren, und nicht als eine Ansammlung individueller Elemente, von denen jedes zur Administration eigene, spezifische Daten benötigt.

Noch sind nicht alle Punkte im Bereich Policy-based Networking vollständig geklärt, verschiedene IETF-Workgroups arbeiten zur Zeit daran. So sollte zum Beispiel die Architektur eines solchen Netzwerks noch näher definiert werden, um herstellerunabhängige Implementierungen zu ermöglichen. Auch sollten Administratoren informiert werden können, ob die Entwicklung einer Policy erfolgreich war, um nicht erst auf das Feedback der Benutzer warten zu müssen.

Welche Möglichkeiten der Einsatz von Policy tatsächlich bietet, wird sich erst mit der Zeit herausstellen. Noch ist das System in der Praxis wenig erprobt und das wirkliche Potential noch nicht ausgelotet. Sicherlich ermöglicht Policy eine vereinfachte Verwaltung von Netzwerken, allerdings muss sich erst noch herausstellen, inwieweit die praktische Umsetzung erfolgreich sein wird.

Literatur

[BCDR+00] Jim Boyle, Ron Cohen, David Durham, Raju Rajan, Shai Herzog und Arun Sastry. The COPS (Common Open Policy Service) Protocol. RFC2748 (Standards Track), Januar 2000.

[MBHS00] Hugh Mahon, Yoram Bernet, Shai Herzog und John Schnizlein. Requirements for a Policy Management System. Internet Draft draft-ietf-policy-req-02.txt, Mai 2000.

[MESW01] B. Moore, E. Elleson, J. Strassner und A. Westerinen. Policy Core Information Model - Version 1 Specification. RFC3060 (Standards Track), Februar 2001.

[MMSS+99] Hugh Mahon, Robert Moore, Mark Stevens, John Strassner, Glenn Waters, Walter Weiss, Andrea Westerinen und Jeffrey Wheeler. Policy Framework. Internet Draft draft-ietf-policy-framework-00.txt, September 1999.

[RVKF+99] Raju Rajan, Dinesh Verma, Sanjay Kamat, Eyal Felstaine und Shai Herzog. A Policy Framework for Integrated and Differentiaded Services in the Internet. *IEEE Network*, September 1999.

[WSSS+01] Andrea Westerinen, John Schnizlein, John Strassner, Mark Scherling, Bob Quinn, Jay Perry, Shai Herzog, An-Ni Huynh, Mark Carlson und Steve Waldbusser. Terminology. Internet Draft draft-ietf-policy-terminology-02.txt, März 2001.

[YaPG00] Raj Yatvakar, Dimitrios Pendarakis und Roch Guerin. A Framework for Policy-based Admission Control. RFC2753 (Informational), Januar 2000.

Sicherheitserweiterungen des DNS

Richard Mager

Kurzfassung

Das Domain Name System (DNS) ist ein Dienst, von dessen Funktion viele andere Dienste des Internet abhängig sind. Er bietet unter anderem einen Mechanismus zum Abbilden von Rechnernamen in Adressen des Internet-Protokolls (IP) an. Unsichere zugrunde liegende Protokolle und Mangel an Authentifizierung und Integritätsprüfung bieten Angreifern die Möglichkeit, die korrekte Funktion des DNS zu stören. Die Internet Engineering Task Force (IETF) arbeitet deshalb an Erweiterungen, bekannt unter dem Namen DNSSEC, um die Sicherheit innerhalb des DNS zu erhöhen. Die Sicherheitsprobleme und die durch DNSSEC angebotenen Lösungen werden hier dargestellt.

1 Einleitung

Das Domain Name System ist ein verteilt organisierter Verzeichnisdienst, der unter anderem Rechnernamen auf numerische IP-Adressen und IP-Adressen zurück auf Rechnernamen abbildet. Durch die starke Verwendung des DNS z.B. in der Internet-Infrastruktur ist diese Funktionalität für einen großen Teil der angebotenen Dienste unverzichtbar geworden.

Beim Entwurf des DNS im Jahr 1987 wurde allerdings auf Sicherheitsmaßnahmen verzichtet, woraus eine hohe Anfälligkeit gegenüber Manipulationen resultiert. Diese werden durch ein Fehlen von Authentizitäts- und Integritätskontrollen der Daten, die im DNS gespeichert werden, und durch die Verwendung von unsicheren Protokollen ermöglicht. Als Reaktion auf solche Bedrohungen hat die Internet Engineering Task Force (IETF) eine Arbeitsgruppe gebildet, um das bestehende DNS um Sicherheitserweiterungen (DNSSEC) zu ergänzen.

Die vorliegende Arbeit beschäftigt sich mit den Sicherheitserweiterungen, die seit Mitte der neunziger Jahre für das DNS entwickelt worden sind. Die so geschaffenen Möglichkeiten zur Authentifizierung der Datenherkunft, zur Schlüsselverteilung und zur Sicherung der Kommunikationsverbindung sollen hier näher untersucht werden.

2 Das Domain Name System (DNS)

Das IP-Protokoll verwendet 32 bit lange Adressen, die von Benutzern ohne Verwendung eines Verzeichnisdienstes nur schlecht zu merken sind. Darum wurden eindeutige Namen eingeführt, die diese Adressen repräsentieren. In Netzen mit wenigen Endsystemen kann die Auflösung der Namen über eine Datei, in der in tabellarischer Form jeder verwendeten IP-Adresse eine eindeutige ASCII-Zeichenkette zugeordnet wird, erfolgen. Problematisch hierbei ist, dass diese Datei auf jedem Rechner vorhanden sein muss. Die Daten müssen dadurch auf allen Rechnern aktualisiert werden, bzw. die Rechner müssen sich in periodischen Zeitabständen eine Datei von einem Server abholen. Bei einer größeren Zahl an Endsystemen wird diese Datei allerdings schnell sehr groß und eine zentrale Verwaltung nicht mehr möglich. Aus diesen Gründen wurde das verteilte, hierarchisch organisierte Domain Name System eingeführt. DNS ist in [Mock87a] und [Mock87b] definiert.

2.1 Grundlagen

Der Nameserver ist das Programm, das den Server-Teil des Client-Server-Mechanismus des DNS darstellt. Nameserver enthalten Informationen über bestimmte Segmente der verteilten Datenbank, welche sie für Clients, so genannte Resolver zur Verfügung stellen. Die DNS-Daten sind in Verwaltungseinheiten, so genannten Zonen, gegliedert, wobei ein Nameserver für eine oder mehrere Zonen zuständig ist. Der Resolver stellt den Client-Teil des DNS dar und besteht häufig lediglich aus einer Bibliothek, die Anfragen erstellt, diese durch das Netz an den Nameserver schickt und die Antworten der aufrufenden Applikation zur Verfügung stellen.

In den folgenden Abschnitten wird auf grundlegende Funktionen und Eigenschaften eingegangen, weitere Informationen zum DNS finden sich z.B. in [Stev94].

2.1.1 Baumstruktur

Der Namensraum des DNS ist in einer hierarchischen Baumstruktur organisiert, deren oberster Knotenpunkt als Wurzel-Domain („root domain“) bezeichnet wird. Jeder Knoten des Baumes hat einen Namen („label“), der durch eine alphanumerische Zeichenkette dargestellt wird, die einzigartig diesen Knoten von seinen benachbarten Knotenpunkten unterscheidet. Die Labels werden durch eine Punktnotation verbunden und bilden so von den Blättern des Baumes zur Wurzel eindeutige DNS-Namen. Ein Label der Länge Null symbolisiert die Wurzel des Baumes, weswegen alle DNS-Namen mit einem Punkt enden, der meist jedoch nicht geschrieben wird.

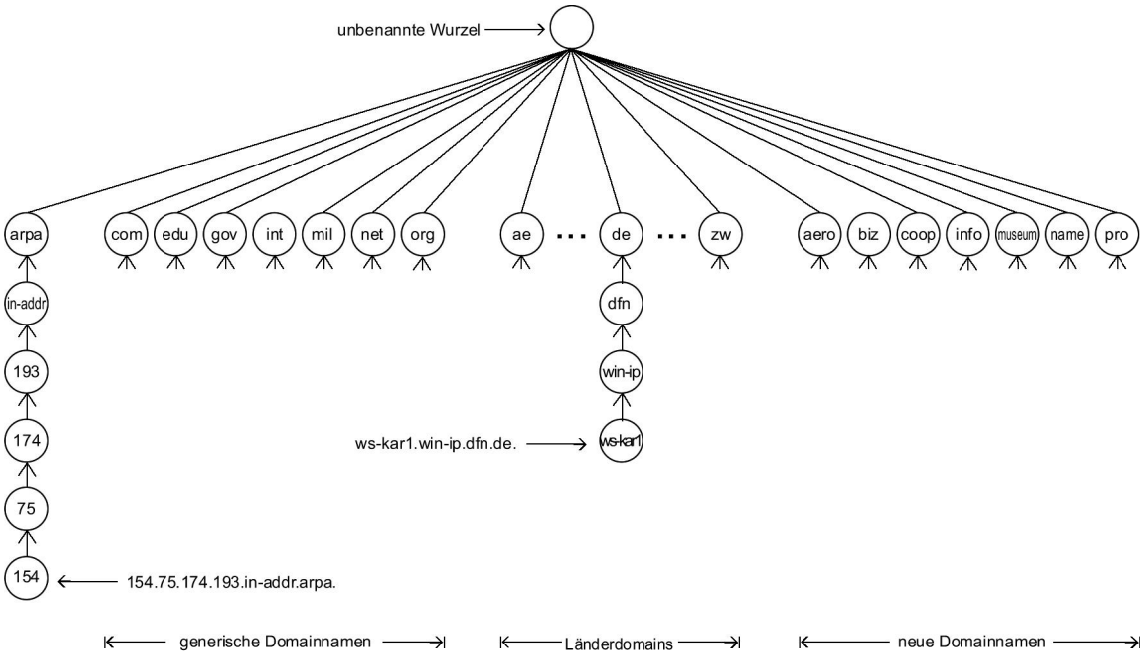


Abbildung 1: Hierarchische Struktur des DNS

Ein Fully Qualified Domain Name (FQDN) (z.B. www.tm.uka.de) ist der vollständige DNS-Name eines Rechners. Das am weitesten links stehende Label bezeichnet den Namen des Endsystems, während das folgende Label die lokale Domain bezeichnet, der der Endsystem angehört. Eine solche lokale Domain kann eine Subdomain einer übergeordneten Domain sein. Der Name dieser übergeordneten Domain wird dann durch das nächste Label bezeichnet. Dies setzt sich so lange fort, bis die Wurzel des DNS-Baumes erreicht ist.

Zur Zuordnung von IP-Adressen zu Domainnamen („Reverse Mapping“ oder „Reverse Lookup“) wurde die Domain „in-addr.arpa“ gebildet, die auch als inverse Domain bezeichnet wird. Mittels der oben beschriebenen Baumstruktur wird in dieser Domain unter der IP-Adresse eines Endsystems der zugehörige Name gespeichert. Hierbei wird die gleiche Notation mit Labeln von rechts nach links verwendet, also von dem am stärksten spezifizierenden Label bis zum letzten. Dies ist ein Gegensatz zu der typischen Darstellung einer IP-Addresse, deren Dezimalnotation von links nach rechts unspezifischer wird. So wird auch bei der Auflösung von IP-Adressen in Namen eine hierarchische Datenstruktur verwendet, die nach dem Suchkriterium sortiert ist.

2.1.2 Anfragen

Stellt nun ein Resolver eine Anfrage an einen Nameserver, für die dieser Server selbst zuständig ist, liefert er die Antwort als verbindliche („autoritative“) Antwort. Aber auch, wenn der Server für den Namensbereich einer Anfrage nicht zuständig ist, kann er die Antwort liefern. Er befragt dazu einen der Root-Server, der eine Referenz zu der gesuchten Subdomain liefert. Als nächstes kontaktiert der Nameserver, dem die Anfrage gestellt wurde, den referenzierten Nameserver, der für diese Subdomain zuständig ist. Dieser schickt entweder die gesuchte Adresse zurück oder eine weitere Referenz, die daraufhin kontaktiert wird. Dies wird solange wiederholt, bis die gesuchte Adresse zurückgeliefert wird. Hierbei ist zu beachten, dass nur der vom Resolver befragte Nameserver rekursiv arbeitet, nicht jedoch die weiteren Nameserver.

Das DNS hat ein definiertes Nachrichtenaustauschprotokoll für Anfragen und Antworten. Eine solche DNS-Nachricht besteht aus fünf Teilen: einem Header, einem Abschnitt für Anfragen (Question), einem für Antworten (Answer), einem Authority-Teil und einem Teil für zusätzliche Informationen (Additional). Der Header enthält Informationen über den Typ der Nachricht und darüber, welche anderen Abschnitte der Nachricht verwendet werden, der Question-Teil wird sowohl bei Anfragen als auch bei Antworten benutzt. Der Antwortteil enthält die RRs, die ein Resolver spezifisch angefragt hat. Der Authority-Teil ist speziellen RRs, wie SOA- und NS-Records, die zur Zone des Domain-Namens des Records im Antwortabschnitt gehören, vorbehalten. Zur Effizienzsteigerung des DNS werden die auf Anfragen erhaltenen RR in den Nameservern zwischengespeichert, womit die Antwortzeit bei erneuten Anfragen erheblich verkürzt werden kann. Damit die Daten im Cache nicht veralten, werden sie nach einer definierten Zeit (Time-to-Live, TTL) ungültig.

2.1.3 Datentypen

Der Domain-Name ist der Schlüssel für die im Verzeichnis gespeicherten Einträge, in denen dann die eigentlichen Adressinformationen oder auch weitere Daten abgelegt sind. Diese Daten werden in Resource Records (RR) gespeichert. Die RR einer Domain werden allgemein als Informationen einer Zone bezeichnet. Eine einzelne Zone kann entweder eine Forward-Zone, zur Auflösung von FQDNs in IP-Adressen, oder eine Reverse-Zone, zur Auflösung von IP-Adressen in FQDNs, sein. Das DNS erlaubt mehr als einen Nameserver pro Zone, aber nur einen primären Nameserver, an dem die tatsächliche Aktualisierung der Daten stattfindet. Weitere Nameserver einer Zone sind sekundäre Nameserver, die Kopien der Datenbank vorhalten. Eine Aktualisierung erfolgt in bestimmten Zeitabständen durch sogenannte Zonentransfers.

Die wichtigsten Resource-Record-Typen sind:

- A definiert eine IP-Adresse.

- PTR wird in der in-addr.arpa-Notation verwendet um Domain-Namen zu speichern.
- CNAME steht für „canonical name“ und ist ein Alias auf einen anderen Domain-Namen.
- NS definiert einen Nameserver.
- MX ist ein RR, in dem abgelegt ist, an welchen anderen Domain-Namen Mails weitergeleitet werden sollen, die an den Namen des RRs gerichtet sind.
- SOA ist ein RR, der den Beginn einer neuen Zone definiert und weiter zonenspezifische Informationen enthält.

Ein RR besitzt Felder für folgende Einträge:

- einen Record-Namen (Domain-Name),
- einen Gültigkeitszeitraum (TTL),
- eine Klasse (normalerweise „IN“ für Internet),
- einen numerischen Wert, der den Typ (z.B. „NS“) symbolisiert,
- ein Datenfeld („resource data“, „rdata“), in dem die eigentlichen Informationen abgelegt werden (z.B. eine IP-Adresse).

Zu jedem Domainnamen können mehrere Datensätze existieren (z.B. A und NS). Allerdings werden auch RRs mit gleichen Namen, gleicher Klasse und gleichem Typ, aber unterschiedlichen Einträgen im Datenfeld verwendet (z.B. ein Rechner mit mehreren Netzwerkkarten). Solche RRs werden Resource Record Sets (RRSets) [ElBu97] genannt.

2.1.4 Dynamic Update

Um DNS-Daten z.B. bei einer dynamischen Zuweisung von IP-Adressen an Endsysteme am Nameserver zu aktualisieren, wurden Möglichkeiten zu einem Dynamic Update geschaffen [VTRB97]. So können Clients Änderungen an den Nameserver-Daten vornehmen. In einem solchen Update können Records eingefügt oder gelöscht werden, in Abhängigkeit von definierbaren Bedingungen, die für ein Update gelten müssen. Eine Update-Nachricht besteht aus einem Header, einem Zonenfeld, in dem definiert ist, auf welche Zone das Update angewandt werden soll, einem Vorbedingungsabschnitt, in dem definiert ist, welche Bedingungen gelten müssen, damit dieses Update durchgeführt wird, einem Updatefeld, das die eigentlichen neuen Daten enthält und einem Feld für zusätzliche Informationen. Alle Update-Anforderungen müssen am primären Nameserver einer Zone durchgeführt werden, um die Datenkonsistenz bei einem Zonentransfer zu gewährleisten.

2.2 Sicherheitslücken des DNS

Da bei der Entwicklung des DNS Sicherheitsaspekte nicht im Vordergrund standen, existieren einige Sicherheitslücken. Das DNS wurde ohne Sicherungsfunktionen für die gespeicherten Daten oder für die Verbindungen entworfen. Daher können Antworten von herkömmlichen DNS-Servern nur bedingt für vertrauenswürdig gelten. Da aber ein großer Teil der im Internet angebotenen Dienste vom DNS abhängig sind, können Angriffe empfindliche Störungen zur Folge haben oder sogar als Grundlage für weitere Angriffe auf andere Dienste verwendet werden.

Das Abfangen und Verändern von DNS-Daten ist besonders einfach, da DNS zu Gunsten höherer Effizienz das verbindungslose User Data Protocol (UDP) verwendet. Ein weiterer Faktor, der zur Verwundbarkeit des DNS beiträgt, ist, dass das DNS als allgemeinzugängliche Datenbank entworfen wurde, in der Konzepte zur Einschränkung des Informationszugriffes absichtlich nicht Teil des Protokolls sind. Neuere Nameserver-Implementationen enthalten zwar Zugriffssteuerungen für Zonentransfers, diese Einschränkungen sind aber implementationsspezifisch und nicht standardisiert.

Die Verwendung von Protokollen, die auf die Korrektheit der DNS-Daten vertrauen, verlangt nach hoher Genauigkeit der in der DNS-Datenbank enthaltenen Informationen. Falsche Daten können sonst zu unerwarteten und möglicherweise gefährlichen Bedrohungen führen. Angriffe lassen sich in folgenden Szenarien kategorisieren: Cache-Poisoning, Client-Flooding, Manipulationen beim dynamischen Aktualisieren von DNS-Daten, Ausspionieren von Netzinformationen (Information Leakage) und Kompromittieren der zuständigen Datenbank des DNS-Servers.

Die Angriffskategorien werden im folgenden Abschnitt näher beschrieben:

- *Cache Poisoning:* Wenn ein Nameserver eine Anfrage eines Clients nicht selbst beantworten kann, muss er weitere Server in der DNS-Hierarchie befragen. Wenn in den befragten Nameservern absichtlich oder unabsichtlich falsche Informationen abgelegt sind, werden diese weitergegeben und im lokalen Cache zwischengespeichert. Dies bezeichnet man als Cache-Poisoning.

  Ältere Nameserverimplementationen ermöglichen sogar eine noch einfachere Variante der Manipulation. Nameserver speichern zusätzliche Informationen, die in Antworten auf Anfragen enthalten sind, zur Effizienzsteigerung zwischen. Ältere Versionen speichern allerdings auch Informationen, nach denen sie gar nicht gefragt haben. So lässt sich an die Antwort eines übergeordneten Nameservers ein gefälschter Eintrag anhängen, so dass der lokale Server bei späteren Anfragen nach diesem Namen dann ohne weitere Nachfrage an zuständige Server die falsche Adresse aus seinem Cache zurückliefert.

  Ein Angreifer wird sich des Cache-Poisonings bedienen, um entweder einen angebotenen Dienst zu stören (Denial of Service) oder um Anfragen eines Clients umzuleiten (Masquerading). Durch solches Masquerading lassen sich z.B. Zugriffe auf Internetdienste auf falsche IP-Adressen umleiten. Ist der Benutzer erst einmal unbemerkt auf einen falschen Dienstanbieter oder Server gelangt, bieten sich zahlreiche Möglichkeiten zum Missbrauch. Besonders gefährlich sind beispielsweise Angriffe, die auf das unbemerkte Vorspiegeln sicherheitskritischer Dienste (Banking, Austausch von Daten, etc.) abzielen. So könnte zum Beispiel ein Angreifer versuchen, Kunden einer Bank auf seinen eigenen Webserver umzuleiten, um sie dort durch die vorgetäuschten Webseiten der Bank dazu zu bringen, ihre persönlichen Identifikationsnummern einzugeben, die so dem Angreifer in die Hände fallen. Oftmals sind auch Zugriffskontrollen über die DNS-Namen implementiert worden, wie z.B. bei den so genannten r-Befehlen unter Unix, die einer Liste von Rechnernamen eine Anmeldung ohne Passwort erlauben. Bei einem Zugriff wird dann per Reverse Mapping der zur IP-Adresse gehörende Name gesucht und mit der Liste verglichen. Hier kann also über Manipulationen des DNS Zugriff auf Computersysteme erreicht werden.

- *Client Flooding:* Unter Client Flooding versteht man eine Manipulationsmethode, bei der ein Resolver auf eine Anfrage eine Vielzahl von vom Angreifer fingierten Antworten zusätzlich zur eigentlichen, richtigen Antwort erhält. Da im DNS-Protokoll keine sichere Authentifizierung definiert ist, kann der Resolver nicht zuverlässig die korrekte Antwort auswählen.

- *Manipulationen bei der dynamischen Aktualisierung von DNS-Daten:* Durch dynamische Aktualisierungen [VTRB97] lassen sich Änderungen an den DNS-Daten eines primären Nameserver von anderen Rechnern aus durchführen. Trotz der in [East97] definierten Zugriffskontrollmechanismen stellen Dynamic Updates ohne weitere Sicherheitsmaßnahmen ein Angriffspunkt dar, da Angreifer versuchen könnten, DNS-Daten durch dynamische Aktualisierungen zu verändern.

- *Information Leakage:* Ein Angreifer kann z.B. durch einen Zonentransfer Informationen über den Aufbau und die Struktur eines Netzwerks erhalten. Oftmals repräsentieren Rechnernamen bestimmte Projekte. Auch könnte ein Angreifer so eine freie IP-Adresse des internen Netzwerkes herausfinden und versuchen, durch Masquerading Zugriff auf interne Dienste zu erhalten.

- *Kompromittieren des DNS-Servers:* Eine weitere Bedrohung stellen Angreifer dar, die durch andere Attacken Administrationsrechte auf dem Nameserver erhalten. Durch zusätzliche Maßnahmen, wie sorgfältige Systemkonfigurationen oder Erniedrigung der Anzahl der Dienste die auf dem gleichen Rechner angeboten werden, lässt sich dieses Risiko senken.

Die hier aufgezeigten Sicherheitsprobleme resultieren daraus, dass im herkömmlichen DNS weder die Datensätze noch die Übertragung der Datensätze gesichert sind. Dies ermöglicht es Angreifern, Manipulationen vorzunehmen. Da diese Probleme seit langer Zeit bekannt sind, versuchen einige Dienste, selbst die Authentizität des Kommunikationspartners sicherzustellen (SSL, SSH). Dies erfolgt beispielsweise durch digitale Zertifikate, die von einer Certificate Authority (CA) erstellt werden. Um generell die Sicherheitsprobleme rund um das DNS zu beheben, hat die Internet Engineering Task Force (IETF) Sicherheitserweiterungen entwickelt.

3 DNS Security Extensions

1994 bildete die IETF eine Arbeitsgruppe, um Sicherheitserweiterungen für das Domain Name System zu entwickeln. Diese werden allgemein DNSSEC-Erweiterungen genannt. Um die Migration zu vereinfachen, musste ein hohes Maß an Kompatibilität gewährleistet werden. Die Arbeitsgruppe erreichte dies durch die Verwendung des Resource-Records-Datentyps, der bereits bei der Entwicklung des DNS auf Erweiterung ausgelegt war. So wurden einige neue RR-Typen definiert, die die Sicherheitsinformationen aufnehmen können. Da das DNS als öffentlicher Dienst konzipiert wurde, verzichtete man absichtlich auf Zugriffkontrollen für Anfragen und Geheimhaltung von DNS-Daten. Zusätzlich ist es möglich, DNSSEC als System zur Schlüsselverteilung für weitere Anwendungen wie Mail, FTP, SSH, etc. zu verwenden, die damit Verschlüsselung betreiben. Außerdem können digitale Zertifikate zur Verfügung gestellt werden.

Im Wesentlichen stellen die DNS Security Extensions (DNSSEC), wie in [East99a] spezifiziert, drei neue Funktionen zur Verfügung: Schlüsselverteilung, Authentifizierung der Datenherkunft und Integritätsprüfung der Daten von einem DNS-Server und Sicherung der Kommunikationsverbindung zwischen einem DNS-Server und einem Resolver, bzw. zwischen einem DNS-Server und einem weiteren DNS-Server.

Grundlage dieser Sicherheitserweiterungen ist die Verwendung kryptographischer digitaler Signaturen. Durch diese Signaturen ist es möglich, die Integrität und die Authentizität von Datensätzen (Records) an jeder Stelle im Netz zu prüfen. Die digitalen Signaturen werden in gesicherten Zonen des DNS als Resource Records gespeichert. Durch die Verwendung von Verschlüsselungsverfahren mit öffentlichen und privaten Schlüsseln werden zum Überprüfen

der durch Signaturen gesicherten Datensätze authentische öffentliche Schlüssel benötigt. Auch diese werden im DNS als Resoure Records abgelegt. Die abgelegten Schlüssel erlauben es Resolvern mit Sicherheitserweiterungen, zusätzlich zu anfänglich im Resolver konfigurierten Schlüsseln neue Authentifizierungsschlüssel einer Zone zu lernen und zu überprüfen.

3.1 Funktionsprinzip

Die Authentizität der Daten wird durch die kryptographische Verknüpfung von Resource Records des DNS mit digitalen Signaturen sichergestellt. Diese Signaturen werden als SIG Resource Records im DNS gespeichert. Jeder Eintrag in einer gesicherten Zone (z.B. A, MX, NS, etc.) erhält mindestens einen entsprechenden SIG Resource Record, der ihn mit der Zone und einem Gültigkeitsintervall verknüpft.

Zum Erzeugen der Signaturen werden Public-Key-Verschlüsselungsverfahren verwendet. Die öffentlichen Schlüssel werden als KEY Resource Records im DNS gespeichert. Für eine Zone gibt es üblicherweise nur einen privaten Schlüssel. Wenn ein Resolver mit Sicherheitserweiterungen den zugehörigen öffentlichen Schlüssel einer Zone zuverlässig erlernt hat, kann er damit signierte Daten der Zone authentifizieren.

Die benötigten Schlüssel sind entweder schon im Resolver konfiguriert oder können durch Auslesen aus dem DNS erhalten werden. Dazu muss der jeweilige Schlüssel allerdings mit einer Signatur gesichert sein, die der Resolver überprüfen kann. Da ein SIG RR mit dem KEY RR der Zone signiert ist, wird der Resolver diesen Schlüssel anfordern. Dies wiederholt sich solange, bis der Resolver einen Schlüssel erhält, dem er vertraut. Im schlechtesten Fall wird dies ein Schlüssel eines Root-Servers sein, da diese in allen Nameservern mit DNSSEC vorkonfiguriert sind. Ein Resolver muss also mit mindestens einem öffentlichen Schlüssel als Startpunkt konfiguriert sein, mit dem er weitere Schlüssel aus anderen Zonen prüfen kann. So kann er neue Schlüssel lernen, mit denen er schließlich die eigentlichen Daten der Zone verifizieren kann.

Erhält ein DNS-Server mit diesen Sicherheitserweiterungen eine Anfrage, liefert er zu den Resource Records der Antwort automatisch die passenden SIG Resource Records. Falls der Server diese Erweiterungen noch nicht implementiert, muss der Resolver die SIG Resource Records selbst anfordern und aus den gelieferten Records die passende Signatur herausfinden.

3.2 Schlüssel

Der KEY Resource Record wird dazu verwendet, öffentliche Schlüssel des DNS zu speichern. Dies können beispielsweise die öffentlichen Schlüssel einer Zone, eines Benutzers, eines Endsystems oder sonstige Schlüssel sein. Ein KEY Resource Record wird – wie jeder andere Resource Record auch – durch eine digitale Signatur in einem SIG Resource Record authentifiziert.

Ein KEY RR selbst enthält keine Angaben zu einer Gültigkeitsdauer. Das heißt allerdings nicht, dass er unbegrenzt gültig ist, da er ja durch einen SIG RR signiert sein muss, der eine Gültigkeitsperiode definiert.

Die einzelnen Felder haben folgende Bedeutung:

- *Flag-Feld:* In den ersten 16 Bit werden folgende Informationen dargestellt:
  - *A/C (Schlüsselart):* Mit diesen beiden Bits wird festgelegt, ob der Schlüssel zum Authentifizieren und bzw. oder zum Verschlüsseln verwendet werden darf. Sind beide Bits Null, ist kein Schlüssel im Public-Key-Feld gespeichert. Mit einem solchen RR kann nachprüfbar versichert werden, dass eine Zone nicht gesichert ist.

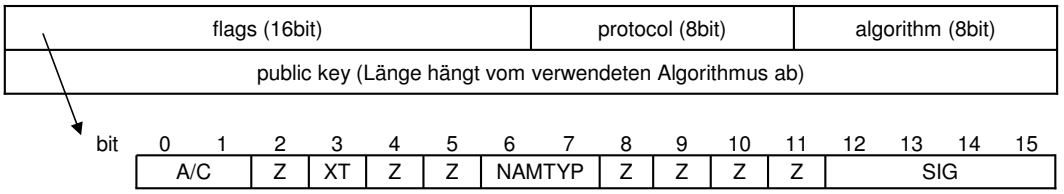


Abbildung 2: RDATA-Feld eines KEY Resource Records

  - *Z (Zero):* Felder, die mit Z gekennzeichnet sind, werden momentan nicht verwendet und sind für einen späteren Gebrauch reserviert. Sie müssen eine Null enthalten.
  - *XT (Erweiterungs-Bit):* Dieses Feld gibt an, ob ein weiteres Flag-Feld existiert. Wenn es auf Eins gesetzt ist, wird zwischen dem Algorithmus-Feld und dem Public-Key-Feld ein weiteres 16 bit großes Flag-Feld eingefügt, das als Erweiterung dienen kann, falls die Z-Felder zur Erweiterung nicht mehr ausreichen.
  - *NAMTYP (Typ des Eintrags):* Diese beiden Bits beschreiben die Verwendung des Schlüssels. Sind beide Bits Null, ist der Schlüssel einem Benutzer, also gewöhnlich einem Rechner, zugeordnet. Falls nur das zweite Bit auf Eins gesetzt ist, ist der Schlüssel einer Zone zugeordnet, ist nur das erste Bit auf Eins gesetzt, ist der Schlüssel weder einer Zone noch einem Benutzer zugeordnet.
  - *SIG (Signaturfeld):* Diese Bits werden für das dynamische Aktualisieren von DNS-Daten, wie in [East97] beschrieben, verwendet. Je nachdem welche Bits gesetzt sind, können mit diesem Schlüssel unterschiedliche Daten signiert werden. Dieses Verfahren wurde aber durch [Well00] abgelöst, mit dem durch Richtlinien für jeden einzelnen Benutzer oder für Benutzergruppen definiert werden kann, welche Daten aktualisiert werden dürfen.

- *Protokoll-Feld:* Da die Schlüssel, die im DNS gespeichert werden können, in Zukunft mit einer Vielzahl von Internet-Protokollen und somit Anwendungen verwendet werden können, kann mit dem Protokollfeld die Gültigkeit des Schlüssels auf ein einzelnes Protokoll beschränkt werden. Außer der Verwendung eines Schlüssels für DNSSEC, wurden bisher TLS, Email und IPSEC definiert.

- *Algorithmus-Feld:* Mit dem Algorithmusfeld wird der Verschlüsselungsalgorithmus spezifiziert, für den der Schlüssel geeignet ist. Bei einer Implementation ist die Verwendung des DSA-Algorithums vorgeschrieben [East99b], die des RSA/MD5-Algorithmus wird zusätzlich empfohlen [East99c]. Bisher ist außerdem der Einsatz des Diffie-Hellman-Algorithmus definiert [East99d]. Einige Nummern sind für die Verwendung mit indirekten Schlüsselalgorithmen, bei denen die Schlüssel an einer anderen Stelle hinterlegt sind, beziehungsweise für den privaten Gebrauch, vorgesehen.

3.3 Signaturen

Digitale Signaturen sind die Instrumente, die die kryptographische Authentifizierung ermöglichen. Eine digitale Signatur beweist, dass eine Nachricht von einer bestimmten Quelle kommt und dass diese Nachricht nicht verändert wurde. Sie sichern so die eigentlichen Daten des DNS, sonstige im DNS abgelegte Informationen und vor allem auch die KEY RRs. Sie werden in einem eigenen Resource Record, dem SIG Resource Record gespeichert. Er ist somit der Kern der Sicherheitserweiterungen, durch den die drei oben genannten Sicherheitsfunktionen des DNS gewährleistet werden.

Ein SIG RR unterschreibt also die Daten eines anderen Resource Records mit einer digitalen Signatur, indem er sie mit einem Zeitintervall und dem Domain-Namen des Signierenden verknüpft. Dies geschieht durch Verschlüsselung eines Hashwerts des zu signierenden Resource Records mit dem privaten Schlüssel des Signierenden. Bei DNS-Daten wird immer der private Schlüssel der Zone verwendet. Wenn ein Resolver also einer Zone vertraut, dann kann er auch den von der Zone signierten Datensätzen trauen. Über solche Vertrauensbeziehungen kann dann auch die Authentizität von nicht lokalen Resource Records geprüft werden.

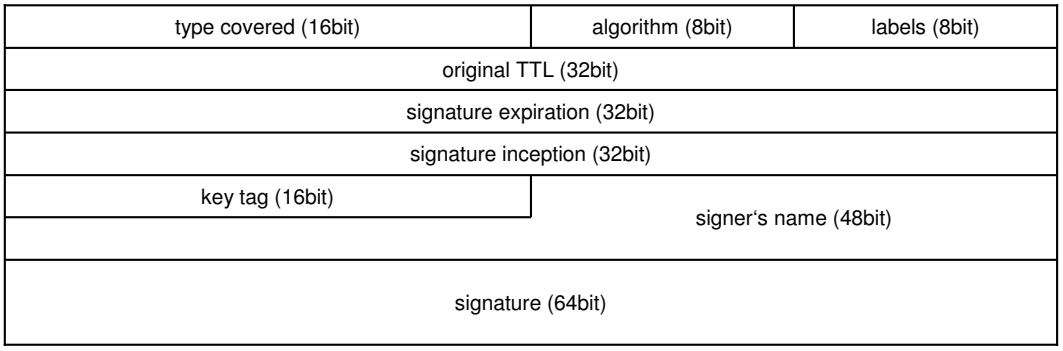


Abbildung 3: RDATA-Feld eines SIG Resource Records

Die einzelnen Felder haben folgende Bedeutung:

- *Type Covered:* Im Type-Covered-Feld ist definiert, welcher Resource-Record-Typ signiert worden ist.
- *Algorithmusfeld:* Das Algorithmusfeld ist wie beim KEY RR definiert.
- *Labelfeld:* In DNS-Daten können Wildcards in RRs verwendet werden. Der Nameserver generiert daraus dann bei einer passenden Anfrage on-the-fly einen vollständigen Eintrag. Zu einem solchen Eintrag kann natürlich nur eine Signatur existieren, die den ursprünglichen Datensatz mit Wildcard sichert. Möchte nun ein Resolver den generierten RR prüfen, muss er das Wildcard-Symbol an der richtigen Stelle einfügen. Hierzu existiert das Labelfeld in der Signatur, das beschreibt, an wievielter Stelle das Wildcard-Symbol im ursprünglichen RR ersetzt worden ist.
- *Original-TTL-Feld (Time to Live):* Da ein signierter Datensatz nach einer Dekrementierung des TTL-Feldes ungültig werden würde, wird im Original-TTL-Feld der ursprüngliche TTL-Wert gespeichert, der nicht verändert wird. Zum Prüfen der Signatur wird dann das TTL-Feld des RRs durch den Wert des Original-TTL-Feldes ersetzt.
- *Signature-Expiration-Feld und Signature-Inception-Feld:* Die Signatur ist gültig vom Zeitpunkt, der im Inception-Feld abgelegt ist, bis zum Wert im Expiration-Feld. Die Zeitpunkte werden als Anzahl der vergangenen Sekunden seit dem 1.1.1970 gespeichert.
- *Key-Tag-Feld:* Im Key-Tag-Feld wird eine Prüfsumme des zugehörigen KEY RR abgelegt, damit der zugehörige Schlüssel leichter identifiziert werden kann, falls mehrere Schlüssel verwendet werden.
- *Unterzeichner (signer's name):* In diesem Feld wird der Domainname des Signierenden, der den SIG RR erstellt hat, abgelegt.
- *Signaturfeld:* Dieses Feld enthält die eigentliche Signatur.

3.4 Existenz von Resource Records

Der SIG RR ermöglicht also eine strenge Authentifizierung von RR, die in einer Zone existieren. Allerdings kann damit noch nicht die Existenz eines Namens in einer Zone nachprüfbar verneint werden. Nameserver ohne Sicherheitserweiterungen liefern in einem solchen Fall den SOA RR und einen Fehlercode zurück. Ein DNSSEC-Nameserver könnte zusätzlich noch den SIG RR des SOA RR liefern. Dies bietet allerdings keine ausreichende Sicherheit, da so einem Angreifer eine Möglichkeit zu einem Angriff durch ein Wiedereinspielen dieser Records („Replay Attack“) geboten wird. Er könnte so existierende Einträge bzw. Rechner durch Spoofing verschwinden lassen, indem er einem Anfrager die vorher gespeicherte Fehlermeldung mit dem signierten SOA RR zuspielt.

Um dies zu unterbinden, wurde der NXT RR definiert. Das Fehlen eines Namens in einer Zone wird durch einen NXT RR für einen Namensbereich angezeigt. So erhält ein Anfrager einer gesicherten Zone immer einen signierten RR zurück, auch wenn zu seiner Anfrage kein entsprechender Eintrag existiert, ohne dass speziell für diese Anfrage ein RR erstellt und signiert werden müsste. Dies ist besonders wichtig, da so der private Schlüssel der Zone nicht online verfügbar sein muss.

Die NXT RR definieren eine Kette aller tatsächlich existierenden Namen in einer Zone, wodurch eine kanonische Reihenfolge aller Einträge einer Domain entsteht. Ein NXT RR für einen Eintrag zeigt dann auf den nächsten existierenden Eintrag innerhalb der Zone. Alle möglichen Namen in einer Zone sind so entweder im DNS als RR gespeichert, oder sie sind in einem Namensbereich enthalten, der von genau einem NXT RR abgedeckt wird.

Das folgende Beispiel stellt einen schematischen Ausschnitt aus den Datensätzen eines Nameservers in einer Domain mit zwei Rechnern dar.

```
domain.          NXT  a-host.domain.  NS SOA SIG KEY NXT
domain.          SIG  NXT             ...
a-host.domain.   NXT  m-host.domain.  A SIG NXT
a-host.domain.   SIG  NXT             ...
m-host.domain.   NXT  domain.         A SIG NXT
m-host.domain.   SIG  NXT             ...
```

Jede Zeile stellt hier einen RR dar. In der ersten Spalte steht der Domain-Namen, in der zweiten Spalte der Typ des RRs. Die beiden letzten Spalten beinhalten die wichtigsten Informationen aus dem Datenfeld. So beschreibt z.B. die erste Zeile einen NXT RR mit dem Namen `domain.`, unter dem auch noch NS, SOA, SIG, KEY und NXT RRs existieren. Der NXT RR verweist auf `a-host.domain.`. Die zweite Zeile beschreibt den SIG RR des Domain-Namens `domain.`, der den NXT RR signiert.

Falls nun ein Resolver nach einem MX Record, mit dem Namen `a-host.domain.` fragt bekommt er den NXT RR `a-host.domain.` zurückgeliefert, was zeigt, dass zwar RR mit diesem Namen existieren, darunter jedoch keine MX records. Falls ein Resolver nach einem A RR mit dem Namen `b-host.domain.` fragt, bekommt er auch den `a-host.domain.` RR zurückgeliefert. Dieser zeigt, dass keine RRs zwischen den Namen `a-host.domain.` und `m-host.domain.` existieren.

Allerdings lassen sich so durch sukzessive Anfragen eines Resolvers sämtliche Einträge einer Zone herausfinden, selbst wenn diese einen Zonentransfer verbietet. Eventuell wird deshalb die Funktionsweise des NXT RR in Zukunft noch einmal geändert.

Ein NXT RR besteht aus einem Domain-Namen und einem Bitmapfeld, in dem beschrieben wird, zu welchen RR-Typen Einträge unter diesem Domainnamen vorliegen.

3.5 Anfrage- und Transaktionsauthentifizierung

Die bisher vorgestellte Authentifizierung der Datenherkunft bietet noch keinen Schutz gegen Manipulationen der eigentlichen Nachricht oder des Nachrichtenkopfes. So könnte beispielsweise eine Anfrage manipuliert werden, oder die Verbindung übernommen werden („session hijacking“). Ein Resolver kann zwar herausfinden ob ein SIG Record in einer Nachricht fehlt, falls die Zone gesichert ist und alle RR durch einen SIG Record gesichert sein sollten, es fehlt allerdings z.B. eine Möglichkeit herauszufinden, ob in einer Nachricht ein RR mit zugehörigem SIG RR entfernt wurde.

Darum wurde in [East00a] eine Signatur definiert, die eine Nachricht mit einem weiteren SIG RR signiert, der am Schluss dieser Nachricht hinzugefügt wird. Ein solcher Signatur-Record wird oft SIG(0) genannt, da sein Type-Covered-Feld den Wert Null hat. Diese Signatur wird in der Regel mit dem Schlüssel des DNS-Servers signiert und nicht mit dem Schlüssel der Zone, damit auch sekundäre Nameserver Antworten selbständig signieren können. Problematisch bei der Authentifizierung der Tranksaktionen ist, dass jede Antwort während des Betriebs signiert werden muss. Dies nimmt nicht unerheblich Rechenzeit in Anspruch. Außerdem wird hierzu der private Schlüssel des DNS-Servers benötigt, so dass dieser ständig online verfügbar sein muss, was ein Sicherheitsrisiko darstellt.

3.6 Signieren und Prüfen von DNS-Daten

Ein DNSSEC-Nameserver muss außer dem Verwalten der Zoneninformationen, dem Cachen von DNS-Daten und dem Beantworten der Anfragen noch zusätzliche Funktionen erfüllen. So müssen die Signaturen erstellt werden, die dazugehörigen Schlüssel müssen zur Verfügung gestellt werden, und die NXT RRs müssen automatisch erzeugt werden. Da die privaten Schlüssel des Servers online auf dem Server vorgehalten werden müssen, besteht die Gefahr einer Manipulation. Die privaten Schlüssel der Zone, die zum Erstellen der Signaturen der Zoneninformation verwendet werden, werden daher offline gehalten und nur zum Signieren auf dem Server verwendet.

Records, die geprüft werden sollen, müssen durch mindestens einen SIG Record signiert sein. Diese SIG Records enthalten einen Namen, unter dem auch der zugehörige Schlüssel in Form eines KEY RR gespeichert ist. Jeder dieser Schlüssel ist durch eine Signatur gesichert unter deren Namen wieder der zugehörige KEY RR gespeichert ist und so weiter. Letztendlich erhält man eine Kette aus alternierenden SIG und KEY Records, deren erster SIG Record die Originaldaten sichert, die geprüft werden sollen, und deren letzter KEY Record einer der statisch konfigurierten Schlüssel ist.

Beim Prüfen von DNS-Daten durch einen Resolver oder einen Nameserver werden die RR in verschiedene Kategorien eingeteilt. Falls das Prüfen fehlschlägt, obwohl die Daten aus einer sicheren Zone stammen, gelten sie als *ungültig*. Daten gelten als *authentifiziert*, wenn sie durch eine Signatur gesichert sind, die durch eine oben beschriebene Kette geprüft wurde. Sie gelten als *schwebend*, falls sie noch nicht authentifiziert sind, aber noch mindestens eine nicht geprüfte Signatur vorliegt. Daten die weder als ungültig noch als authentifiziert eingestuft werden können, z.B. da sie aus einer nicht gesicherten Zone stammen, gelten als *ungesichert*.

Um für den Fall, dass bereits der Nameserver, der eine Anfrage eines Resolvers bearbeitet, die Überprüfung der Daten vornimmt, da der Resolver dazu z.B. nicht in der Lage ist, werden zwei vorher ungenutzte Bits im Flagteil des Headers der DNS-Nachrichten genutzt. Ein gesetztes AD-Bit (authentic data) in einer Antwort eines Nameservers zeigt an, dass alle enthaltenen RR bereits geprüft wurden. Bei Servern ohne Sicherheitserweiterungen ist dieses Bit Null, so dass deren gelieferte Daten als nicht geprüft gelten. Ist das CD-Bit (checking

disabled) in einer Anfrage gesetzt, akzeptiert der Resolver auch schwebende Daten vom Nameserver. So erhalten alte Resolver nur geprüfte Daten. Dies reicht zwar zur Gewährleistung der Sicherheit nicht aus, da Resolver ohne Sicherheitserweiterungen nicht prüfen können, ob sie dem Nameserver trauen, erschwert aber trotzdem Manipulationen. Resolver, die Antworten mit einem gesetzten AD-Bit erhalten, dürfen diesen erst trauen, wenn sie dem Nameserver trauen, der diese Daten gesendet hat. Tun sie dies nicht und möchten die Authentizität der Daten selbst prüfen oder möchten sie bewusst auf gesicherte Daten verzichten (z.B. aus Geschwindigkeits- oder Leistungsgründen), so sollten sie das CD-Bit setzen, um die Nameserver-Last zu reduzieren.

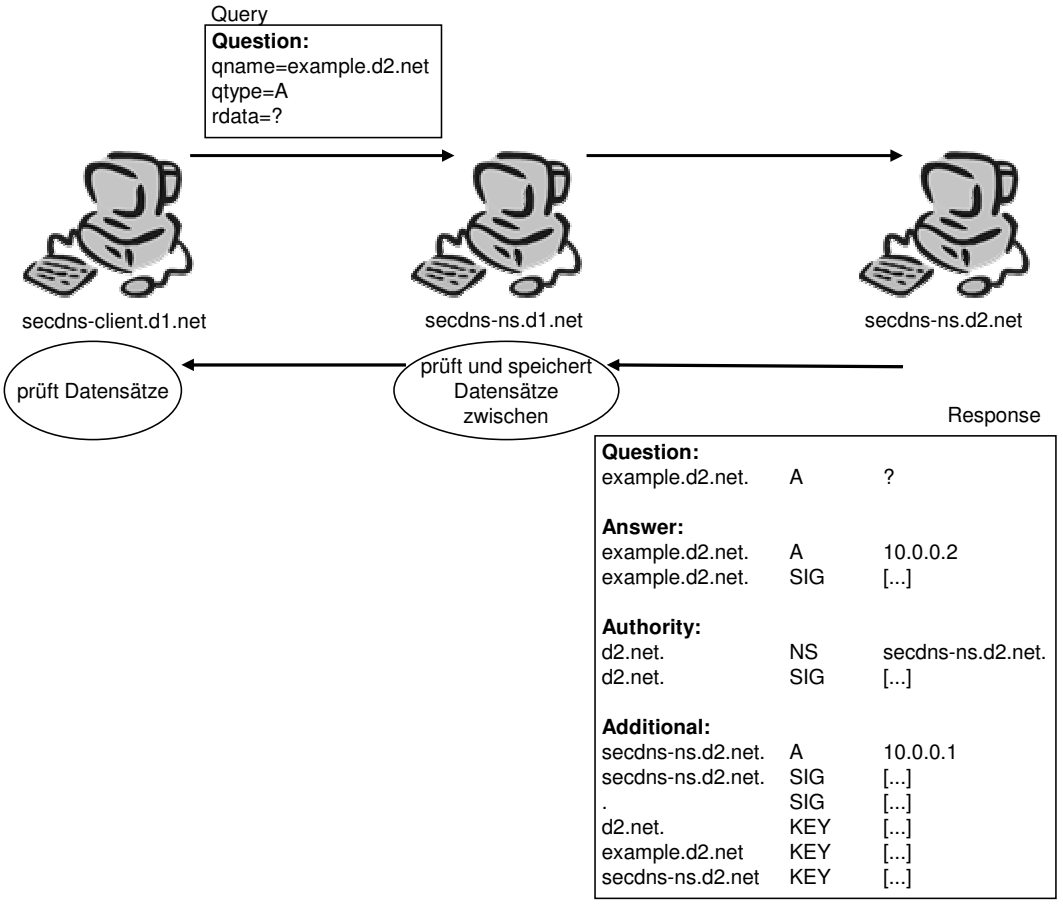


Abbildung 4: Beispiel zum Prüfen von Resource Records

In dem in Abbildung 4 dargestellten Beispiel stellt ein Resolver `secdns-client.d1.net` eine Anfrage nach dem A Record für den Namen `example.d2.net` an den Nameserver `secdns-ns.d1.net`. Dieser schickt die Anfrage an den zuständigen Nameserver `secdns-ns.d2.net`, der den A RR in einem Antwortpaket zurückschickt. Alle verwendeten Server und der Resolver benutzen die neuen Sicherheitserweiterungen. Zusätzlich wird hier angenommen, dass die 512-byte-Grenze für ein DNS-UDP-Paket hier nicht gilt. In Realität müssen größere Datenpakete geteilt werden, oder zusätzliche Informationen werden nicht automatisch mitgeschickt und müssen vom Resolver gesondert angefordert werden.

In dem Antwortfeld der Nachricht ist also der geforderte A RR enthalten. Zusätzlich wird der passende SIG RR mitgeschickt, der die Integrität und die Authentizität des A RR sichert. Im Authority-Teil werden der Nameserver-Eintrag und der dazugehörige SIG RR mitgesendet. In dem Teil für zusätzliche Informationen wird – wie bei normalen DNS-Antworten auch –

der A RR des Nameservers mitgesendet. Zusätzlich ist hier der passende SIG RR enthalten, der wie die anderen Signaturen auch mit dem privaten Schlüssel der Zone signiert wurde. Der letzte SIG Record ist eine Signatur der bisherigen DNS-Nachricht, die mit dem privaten Schlüssel des Nameservers erstellt wurde. Sie wird zur Authentifizierung der gesamten Nachricht verwendet (SIG(0)). Im Teil für zusätzliche Informationen werden die verwendeten Schlüssel mitgeschickt. In diesem Fall sind dies der Schlüssel der Zone `d2.net`, mit dem die ersten drei Signaturen geprüft werden können, der Schlüssel des Host `dnssec-ns.d2.net`, mit dem die Transaktionssignatur geprüft werden kann und zusätzlich den Schlüssel des Hosts `example.d2.net`.

In Realität wäre die hier gesendete Nachricht zu groß für ein UDP-Paket. Daher existieren genaue Prioritäten über die Wichtigkeit von RRs, die definieren, welche Daten beim Überschreiten der Paketgröße weggelassen werden. Allgemein haben KEY RRs eine niedrigere Priorität als SIG RRs, so dass diese zuerst weggelassen werden. Fehlt einem Resolver ein RR, so muss er diesen in einer neuen Anfrage anfordern.

3.7 Zertifikate

Zusätzlich zu den Sicherheitserweiterungen, die zwingend zur Absicherung benötigt wurden (KEY, SIG, NXT), lassen sich im DNS nun auch Zertifikate ablegen, die in kryptographischen Anwendungen dazu benutzt werden, um öffentliche Schlüssel und die Zugehörigkeit zu einer bestimmten Identität zu speichern. Hierfür wurde in [EaGu99] ein eigener Certificate Resource Record (CERT RR) spezifiziert, der es ermöglicht, bereits vorhandene Zertifikate zu integrieren. Momentan ist die Unterstützung der Zertifikattypen X.509/PKI, SPKI und PGP vorgesehen.

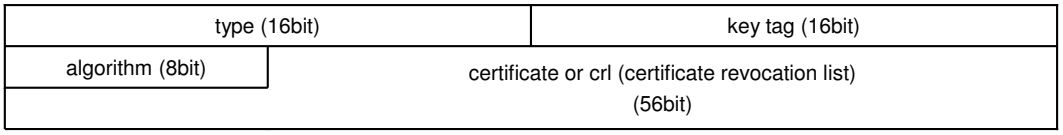


Abbildung 5: RDATA-Feld eines CERT Resource Records

Im Typfeld wird die Information darüber gespeichert, welche Art von Zertifikat in dem CERT Record abgelegt ist. Das Key-Tag-Feld und das Algorithmusfeld sind wie beim SIG RR definiert. Das Zertifikat wird im Certificate/CRL-Feld abgelegt und Base-64-codiert.

3.8 Authentifizierung mit symmetrischer Verschlüsselung

Ein Resolver ohne die hier besprochenen Sicherheitserweiterungen ist normalerweise eine einfache Bibliothek, die Anfragen erstellt, sendet und die Antworten empfängt. Dies funktioniert sehr schnell und erfordert wenig Rechenzeit. Außerdem werden die Ergebnisse nicht zwischengespeichert. Unter Berücksichtigung der damaligen Anforderungen war dies ein sehr sinnvolles Konzept, welches aber sehr ineffizient für DNSSEC ist. Da ein Resolver die erhaltenen Records nicht zwischenspeichert, muss er bei jeder Anfrage die ganze Kette aus KEY und SIG Records anfordern. Ein möglicher Ausweg hieraus wäre die Einführung eines Caches in den Resolvern gewesen, was allerdings auch deren Komplexität erhöht hätte.

Normalerweise kommunizieren Resolver nur mit einer sehr geringen Zahl von DNS-Servern im gleichen Netzwerk. Wenn ein Resolver dem lokalen DNS-Server vertraut, kann er den Vergleich der SIG und KEY RR dem Nameserver überlassen. Dazu muss allerdings die Kommunikation

zwischen lokalem Nameserver und dem Resolver gesichert werden. Da es in einer solchen Kommunikation nur zwei Beteiligte gibt, können diese ihre Kommunikation mit symmetrischen Verschlüsselungsverfahren schützen. Dies hat den Vorteil, dass die Ver- und Entschlüsselung weniger Rechenaufwand erfordert als bei asymmetrischen Verschlüsselungsverfahren, wodurch die Resolver auch auf einfachen Systemen gut funktionieren.

Der Resolver und der Server benötigen also einen gemeinsamen symmetrischen Schlüssel. Dieser Schlüssel kann mit einer beliebigen sicheren Methode erzeugt und übertragen werden. Hierfür wurde im September 2000 der TKEY RR eingeführt [East00b]. Dieser TKEY RR bietet Resolvern und Nameservern einen Standard, um sich auf einen gemeinsamen symmetrischen Schlüssel zu einigen, ohne dass ihre Kommunikation verschlüsselt sein muss. Hierzu werden verschiedene Verfahren verwendet; die Implementation des Diffie-Hellman-Schlüsselaustausch-Verfahrens ist vorgeschrieben, die anderen Verfahren wie z.B. die direkte Zuweisung durch Server oder Resolver können zusätzlich implementiert werden. Um diesen Schlüsselaustausch zu sichern, werden die TKEY RR durch asymmetrische Verfahren einem SIG(0) RR signiert.

Sobald Resolver und Server einen gemeinsamen Schlüssel besitzen, wird jeder Anfrage eines Resolvers eine Request Signature und jeder Antwort eines Servers eine Transaction Signature hinzugefügt. Diese Signaturen werden mit dem gemeinsamen symmetrischen Schlüssel erzeugt und vom Empfänger geprüft. Die Transaktionssignaturen werden in einem TSIG RR [VGEW00] abgelegt.

3.9 Dynamische Aktualisierung

Bei dynamischen Aktualisierungen sind Sicherheitsmaßnahmen besonders wichtig, da hier einem Client erlaubt wird, permanente Änderungen an den DNS-Daten vorzunehmen. Diese Sicherheitsmaßnahmen wurden erstmalig in [East97] definiert und später durch [Well00] ersetzt.

Eine Update-Nachricht muss immer durch einen TSIG oder durch einen SIG(0) RR signiert sein. Die Authentifizierung der Benutzer erfolgt dann über diese Signatur. Bei der Verwendung von SIG(0) wird die Identität des Senders über den Besitzer des KEY RR, der zum Signieren verwendet wurde, ermittelt. Bei statisch konfigurierten TSIG RRs kann die Identität über den gemeinsamen Schlüssel bestimmt werden, und bei dynamisch ausgehandelten TSIG RRs wird die Identität über den Schlüssel bestimmt, der zur Authentifizierung des TKEY RRs verwendet wurde.

Welche Benutzer hierbei welche Daten eines primären Nameservers aktualisieren dürfen, kann der Administrator einer Zone durch Richtlinien festlegen. In den Richtlinien ist bestimmt, ob eine Identität befugt ist, die erwünschte Änderung durchzuführen. So kann z.B. bestimmt werden, ob ein Benutzer Einträge hinzufügen, löschen oder ändern darf und in welchen Zonen diese Änderungen durchgeführt werden dürfen. Auch kann bestimmt werden, welche Typen von Records verändert werden dürfen, wodurch z.B. die Veränderung von SOA oder NS Records durch normale Benutzer verhindert werden kann.

Falls durch eine dynamische Aktualisierung Daten verändert worden sind, muss der Nameserver automatisch neue SOA RR und neue NXT RR erstellen und diese mit dem zugehörigen Zonenschlüssel signieren. Dadurch muss allerdings der private Zonenschlüssel online auf dem am Netzwerk angeschlossenen Nameserver zur Verfügung stehen, was ein Sicherheitsrisiko darstellt.

4 Stand der Technik

Die ersten Implementationen der Sicherheitserweiterungen wurden an der DNS-Implementation BIND (Berkeley Internet Name Domain) vorgenommen. Hierzu wurde die bereits recht alte Version 4.9.4 verwendet, so dass diese wohl nie im produktiven Betrieb eingesetzt wird. Diese Erweiterungen fanden Eingang in die 8er-Versionen von BIND (ab 8.1.2). Doch erst die komplett neu entwickelte Version 9 entspricht komplett den DNSSEC-Standards.

Mittlerweile fanden eine Reihe von Workshops zum Thema DNSSEC statt, auf denen praktische und theoretische Aspekte betrachtet wurden. So wurde beispielsweise die Machbarkeit der Signierung großer Zonen (.de, .nl, .se) untersucht, wobei keine Probleme bezüglich Arbeitsspeicher oder Rechengeschwindigkeit auftraten. Generelle Untersuchungen über die zusätzliche Last durch DNSSEC auf Nameservern im normalen Betrieb fehlen allerdings noch. Auch an der Verwendung von DNSSEC als Plattform zur Schlüsselverteilung wird gearbeitet, z.B. existieren schon erste SSH-Anwendungen, die öffentliche Schlüssel aus Nameservern auslesen können.

Die Entwicklung an der Authentifizierung mit TSIG RR ist schon so weit fortgeschritten, dass sie zumindest für Zonentransfers als anwendbar gilt. Andere Teile der Sicherheitserweiterungen sind aus praktischer Sicht noch nicht uneingeschränkt einsetzbar. So wird noch über die Verwendung der NXT RR entschieden, da diese ein Auslesen der kompletten Zoneninformationen erlauben. Hierzu existiert mittlerweile ein anderer Vorschlag bei dem dies nicht möglich ist (NO Resource Record). Die mittlerweile bei den Sicherheitserweiterungen gewonnenen praktischen Erfahrungen werden mit neuen Überlegungen daher noch in eine dritte Überarbeitung eingehen.

Weitere Informationen zum Stand der Technik finden sich in [East01].

5 Zusammenfassung

1987 ratifizierte die IETF den DNS-Standard, um das nicht skalierbare Namenssystem mit den `host.txt`-Dateien abzulösen. Seitdem hat der weite Gebrauch der Fähigkeiten des DNS, Rechnernamen in IP-Adressen umzuwandeln zu einer kritischen Komponente gemacht, da das DNS ohne Berücksichtigung von Sicherheitsaspekten entworfen worden ist. So ist es anfällig für vielerlei Manipulationen, was Folgen auch für andere Anwendungen und Dienste mit sich bringt.

Um diesen Missstand zu beheben, wurden von der IETF Sicherheitserweiterungen entworfen, die unter dem Namen DNSSEC bekannt sind. Sie unterstützen die Authentifizierung der Datenherkunft und bieten eine Integritätsprüfung von DNS-Daten, wodurch eine Absicherung gegen die meisten Manipulationsmöglichkeiten besteht. Cache-Poisoning-Angriffe und Client-Flooding-Attacks werden durch das Signieren von DNS-Daten verhindert. Die Schwächen der Dynamic-Update-Funktionen wurden durch neue Sicherheitsmaßnahmen mit Anfrage- und Transaktionsauthentifizierung behoben. Selbst das direkte Kompromittieren eines DNS-Servers kann durch einen offline gespeicherten Zonenschlüssel stark erschwert werden. Nur das Gewinnen von Informationen über die Struktur interner Netze (Information Leakage) lässt sich auch durch DNSSEC nicht verhindern. Da DNS allerdings als offenes System ohne Zugriffseinschränkungen entwickelt worden ist, wird dies allerdings auch nicht als direkte Aufgabe angesehen.

DNSSEC zeigt also viel versprechende Möglichkeiten, die Internet-Infrastruktur vor DNS-Angriffen zu schützen.

Literatur

[EaGu99] Donald E. Eastlake und Olafur Gudmundsson. *Storing Certificates in the Domain Name System (DNS)*. IETF, März 1999. RFC 2538.

[East97] Donald E. Eastlake. *Secure Domain Name System Dynamic Update*. IETF, April 1997. RFC 2137.

[East99a] Donald E. Eastlake. *Domain Name System Security Extensions*. IETF, März 1999. RFC 2535.

[East99b] Donald E. Eastlake. *DSA KEYs and SIGs in the Domain Name System (DNS)*. IETF, März 1999. RFC 2536.

[East99c] Donald E. Eastlake. *DSA RSA/MD5 KEYs and SIGs in the Domain Name System (DNS)*. IETF, März 1999. RFC 2537.

[East99d] Donald E. Eastlake. *Storage of Diffie-Hellman Keys in the Domain Name System (DNS)*. IETF, März 1999. RFC 2539.

[East00a] Donald E. Eastlake. *DNS Request and Transaction Signatures (SIG(0)s)*. IETF, September 2000. RFC 2931.

[East00b] Donald E. Eastlake. *Secret Key Establishment for DNS (TKEY RR)*. IETF, September 2000. RFC 2930.

[East01] Donald E. Eastlake. *Notes from the State-Of-The-Technology: DNSSEC*. IETF, Juni 2001. RFC 3130.

[ElBu97] R. Elz und R. Bush. *Clarifications to the DNS Specification*. IETF, Juli 1997. RFC 2181.

[Kuri96] Jürgen Kuri. Wenn der Postmann zweimal klingelt. Namen und Adressen im TCP/IP-Netzwerk und im Internet. *c't* (12), 1996, S. 334–347.

[Mart99] Kai Martius. Nachschlag. DNS gegen Mißbrauch schützen. *iX* (2), 1999, S. 108–113.

[Mock87a] P. Mockapetris. *Domain Names – Concepts and Facilities*. IETF, November 1987. RFC 1034.

[Mock87b] P. Mockapetris. *Domain Names – Implementation and Specification*. IETF, November 1987. RFC 1035.

[Stev94] W. Richard Stevens. *TCP/IP Illustrated, Volume 1 – The Protocols*, Kapitel 14, S. 187–208. Addison-Wesley Publishing Company, Reading/Mass. 1994.

[VGEW00] Paul Vixie, Olafur Gudmundsson, Donald E. Eastlake und Brian Wellington. *Secret Key Transaction Signatures for DNS (TSIG)*. IETF, Mai 2000. RFC 2845.

[VTRB97] Paul Vixie, Susan Thomson, Yakov Rekhter und Jim Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE)*. IETF, April 1997. RFC 2136.

[Well00] Brian Wellington. *Secure Domain Name System (DNS) Dynamic Update*. IETF, November 2000. RFC 3007.

Mobilitätsprofile in mobilen Ad-hoc-Netzen

Thomas Richter

Kurzfassung

Ziel dieser Arbeit ist, zwei Konzepte vorzustellen, mittels derer sich das Routing in Ad-hoc-Netzen verbessern lässt. Hierzu werden eingangs die wesentlichen Unterschiede zwischen mobilen drahtlosen Ad-hoc-Netzen und Netzen mit fester Kommunikationsinfrastruktur geschildert, und auf Probleme, die im Zusammenhang mit Routing in Ad-hoc-Netzen auftreten, hingewiesen. Anschließend werden einige bereits existierende übliche Routing-Protokolle vorgestellt und deren Aufbau wird erklärt. Die Frage, wie sich die Effizienz dieser Protokolle und somit die Leistung des Netzes erhöhen lässt, bildet den Inhalt des ersten Lösungsansatzes. Dieser macht sich das nicht unabhängige Bewegungsverhalten von Netzteilnehmern innerhalb einer Gruppe zu Nutze und stellt den Einfluss verschiedener Bewegungsmuster auf die Leistungsfähigkeit der eingeführten Routing-Protokolle heraus. Der zweite Ansatz integriert Mobilitätsinformation in Routing-Protokolle selbst und versucht somit, durch Voraussagen über wahrscheinliche zukünftige Netzwerktopologien Verbindungen zwischen Teilnehmern aufrecht zu erhalten. Die Frage, inwiefern die Genauigkeit der Vorhersage eine Rolle spielt, ist ebenfalls Thema der Arbeit. Abschließend werden die beiden Modelle einer Kritik und Schlussbetrachtung unterzogen.

1 Einleitung

In Umgebungen, in denen lediglich eine sehr eingeschränkte beziehungsweise keine Kommunikationsinfrastruktur existiert oder die vorhandene Infrastruktur ökonomisch gesehen zu teuer und wenig zweckmäßig ist, haben mobile Nutzer die Möglichkeit, durch die Formation eines Ad-hoc-Netzes über dieses miteinander zu kommunizieren. Mobile Ad-hoc-Netze stellen eine Ansammlung mehrerer beweglicher Teilnehmer mit drahtloser Netzwerkschnittstelle dar, welche dynamisch ein temporäres Netzwerk bilden, ohne hierfür eine bereits bestehende Infrastruktur oder zentrale Steuereinheit zu benötigen. Aufgrund der eingeschränkten Übertragungsreichweite der drahtlosen mobilen Geräte muss der Datenaustausch innerhalb des Netzwerkes meist über mehrere Teilstrecken (hops) geschehen. Dies bedeutet, dass in Ad-hoc-Netzen jeder Teilnehmer nicht unbedingt nur als Nutzer fungiert, welcher eigene Daten sendet und nur die für ihn bestimmten Datenpakete empfängt, sondern eventuell gleichzeitig auch als Router wirksam wird. Die Funktion Nutzer und/oder Router wird im Folgenden als Knoten bezeichnet. Stellen die Pfeile in Abbildung 1 die jeweils möglichen Verbindungen innerhalb eines Ad-hoc-Netzes dar, so fungieren beispielsweise J und I als Router und machen die Kommunikation zwischen K und H erst möglich.

Neben den oben erwähnten Merkmalen mobiler Ad-hoc-Netzwerke weisen die Autoren von [Netg97], [BMJH+98] ebenso auf die im folgenden dargestellte allgemeine Problematik des Routings in mobilen Ad-hoc-Netzen hin. Da Ad-hoc-Netze oft einem hohen Grad an Dynamik ausgesetzt sind, weil sich sowohl die Gruppe als Ganzes als auch die Mitglieder innerhalb der Gruppe ständig bewegen, kommt es häufig vor, dass bestimmte Datenübertragungswege zwischen verschiedenen Partnern nicht mehr zur Verfügung stehen und somit Verbindungen aufbrechen und teilweise Datenpakete verloren gehen. Die Frage, woher ein Nutzer bei der

Bildung eines Ad-hoc-Netzes überhaupt weiß, welche anderen Teilnehmer sich innerhalb des Netzes befinden beziehungsweise welche Netzdienste verfügbar sind und wie sich diese im Zeitverlauf ändern, stellt ebenfalls einen Teil der Problematik dar. Auch generelle Fragestellungen, wie Zutritts- beziehungsweise Zugriffsbeschränkungen, spielen bei der Formation eines Ad-hoc-Netzes eine Rolle.

Des Weiteren sind in Ad-hoc-Netzen typischerweise Informationen über die jeweiligen aktiven Verbindungen nicht verfügbar, und Verzögerungen sowie Kapazität einer Verbindung werden durch den Datenverkehr auf benachbarten Übertragungswegen beeinträchtigt. Betrachtet man bezogen auf Abbildung 1 die Datenströme von D nach G (D-E-F-G) und von H nach K (H-I-J-K), so wird sich bei gleichzeitiger Übertragung die jeweilige ursprüngliche Kapazität verringern. Hinzu kommen unumgängliche Einschränkungen wie eine begrenzte verfügbare Bandbreite in einer drahtlosen Umgebung sowie eine begrenzte Verfügbarkeit jedes mobilen Nutzers (zum Beispiel aufgrund begrenzter Akku-Kapazität).

In den meisten Netzwerken ist ein „minimum-hop routing“, also Übertragung der Daten über die Route mit der geringsten Anzahl an Teilstrecken, am effizientesten, da es die wenigsten Netz-Ressourcen beansprucht. Dieses Faktum trifft allerdings nicht für Ad-hoc-Netze zu, da sich, wie wir oben gesehen haben, Datenflüsse auf benachbarten Routen beeinträchtigen. In unserem Fall kann anstelle der vorher gewählten Wege D-E-F-G und H-I-J-K beispielsweise ein Routing über die Knoten D-A-B-C-G beziehungsweise H-L-M-N-K eine weitaus höhere Netz-Effizienz versprechen, da sich so die Datenflüsse gegenseitig nicht behindern, weniger Datenverluste und somit wenig wiederholtes Paketsenden stattfinden.

Die Tatsache, dass Routing in mobilen Ad-hoc-Netzen demnach eine Reihe von Problemen aufwirft, welche bei vorab installierter Kommunikationsinfrastruktur nicht zum Tragen kommen, stellt an den Entwurf der Routing-Protokolle für Ad-hoc-Netze die Forderung, sich in einer für den Endnutzer transparenten Weise an Topologieänderungen innerhalb des Netzwerkes anzupassen.

Abschnitt 2 dieser Arbeit führt einige gängige Routing-Protokolle für Ad-hoc-Netze ein und erklärt deren Aufbau. In Abschnitt 3 wird die Verwendung eines Mobilitätsmodells für Gruppen zur Bewertung der Leistungsfähigkeit dieser Routing-Protokolle vorgestellt. Einen Lösungsansatz zur Steigerung der Leistung mobiler Ad-hoc-Netzwerke bildet die in Abschnitt 4 beschriebene Einbeziehung von Mobilitätsinformation in Routing-Protokolle. Der letzte Abschnitt kritisiert beide Modelle und fasst die Ergebnisse dieser Arbeit zusammen.

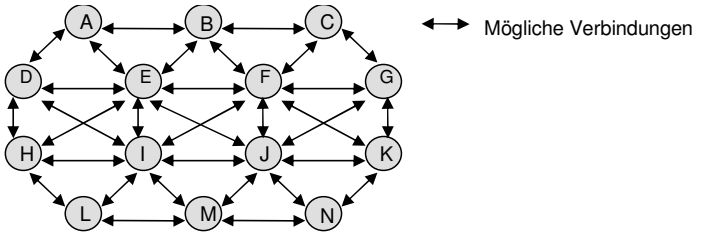


Abbildung 1: Graphische Darstellung eines Ad-hoc-Netzes

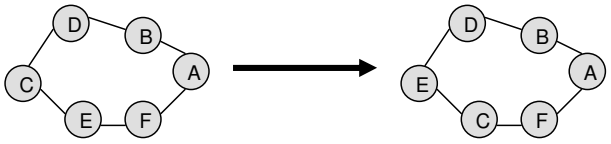
2 Routing-Protokolle für Ad-hoc-Netzwerke

Um das Routing in mobilen Ad-hoc-Netzen zu organisieren, existieren, wie von den Autoren in [Misr00], [PGHC99] beschrieben, bereits einige Protokolle, von denen drei im Folgenden vorgestellt werden. Während sich die ersten beiden Routing-Protokolle, DSDV und HSR,

dadurch auszeichnen, dass sie eine oder mehrere Routing-Tabellen, welche Informationen über alle anderen Systeme des Netzes beinhalten, aufbauen und ständig aktualisieren („table-driven“), handelt es sich bei dem dritten Verfahren, AODV, um ein Protokoll, welches erst im Bedarfsfall einen entsprechenden Weg für die Datenübertragung entwickelt („on-demand“).

2.1 Destination-Sequenced Distance Vector (DSDV) Protocol

Der DSDV-Routing-Algorithmus basiert auf der Idee des klassischen Distance Vector (DV) Routing-Algorithmus von Bellman-Ford. Jede mobile Station hält eine Routing-Tabelle, welche alle verfügbaren Bestimmungsorte, die Anzahl der Teilstrecken, die zu überwinden sind, um das Ziel zu erreichen, den für die Übertragung zu wählenden angrenzenden Knoten sowie eine vom Zielknoten zugewiesene Sequenznummer auflistet. Jede Station sendet periodisch, beziehungsweise wenn sich eine signifikante Änderung ergeben hat, ihre Routing-Tabelle an ihre direkten Nachbarn, das heißt, die Änderungsnachricht (Update) ist sowohl zeit- als auch ereignisabhängig. Bei sich schnell ändernder Netztopologie erfolgt das Update durch Übertragung der gesamten Routing-Tabelle („full dump“), bei hoher Stabilität des Netzwerkes werden lediglich die nach dem vorherigen Update geänderten Einträge weitergegeben („incremental update“). Der Weg mit der höchsten Sequenznummer wird für die Verbindung gewählt; besitzen zwei verschiedene Wege identische Sequenznummern, erfolgt die Übertragung der Daten über die Route, welche die geringere Anzahl von Teilstrecken bis zum Bestimmungsort aufweist (siehe Abbildung 2).



Originale Routing-Tabelle für A

Ziel	nächster Hop	Anzahl Hops	Sequenz Nummer
A	N/A	N/A	2
B	B	1	3
C	B	3	2
D	B	2	2
E	F	2	2
F	F	1	3

Routing-Tabelle erhalten von B

Ziel	nächster Hop	Anzahl Hops	Sequenz Nummer
A	A	1	2
B	N/A	N/A	4
C	D	3	3
D	D	1	3
E	D	2	3
F	A	2	3

Geänderte Routing-Tabelle für A

Ziel	nächster Hop	Anzahl Hops	Sequenz Nummer
A	N/A	N/A	2
B	B	1	4
C	B	4	3
D	B	2	3
E	B	3	3
F	F	1	3

Abbildung 2: Beispiel für ein Update einer Routing-Tabelle nach dem DSDV-Verfahren

2.2 Hierarchical State Routing (HSR) Protocol

HSR zeichnet sich durch eine hierarchische Netzwerkstruktur aus. Alle im Netz vorkommenden Systeme werden in Blöcke eingeteilt, innerhalb derer jeweils ein so genannter Clusterhead gemäß einem auf dieser Blockbildung basierenden Algorithmus bestimmt wird. Dieser Vorgang findet also zunächst auf der real existierenden Ebene statt, hier mit Ebene 0 bezeichnet. Die gewählten Clusterhead-Knoten werden gleichzeitig Mitglied der nächsthöheren Ebene (in diesem Fall Ebene 1) und organisieren sich untereinander wieder auf dieselbe Art und Weise in so genannten virtuellen Gruppen, wählen wieder Clusterheads aus, welche ebenfalls wieder Mitglieder der nächsthöheren Ebene werden und so weiter (siehe Abbildung 3). Die Knoten auf der untersten Ebene tauschen ihre gesamten Verbindungsinformationen innerhalb ihres Blocks aus. Die Informationen werden vom Clusterhead der Gruppe zusammengefasst und anschließend über den so genannten Gateway-Knoten zum benachbarten Clusterhead

gesendet. Diese Zusammenfassung sowie der Austausch von Verbindungsinformationen finden auf jeder Ebene statt. Sobald der Algorithmus eine neue Ebene erreicht, überflutet ein Knoten dieser Ebene die darunterliegende Ebene mit diesen Informationen. Jeder Knoten erhält eine hierarchische Adresse, welche notwendig ist, um ein Paket an einen beliebigen Bestimmungsort innerhalb des Netzes mit Hilfe der HSR-Routing-Tabellen zu senden.

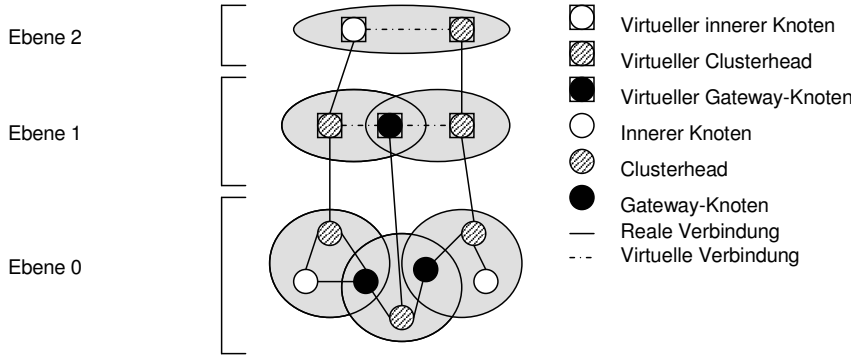


Abbildung 3: Beispiel einer hierarchischen Netzstrukturierung mittels des HSR-Protokolls

Zusätzlich wird beim HSR-Verfahren das Netz nicht nur geographisch, sondern auch logisch in mehrere Subnetze eingeteilt. Diese Eigenschaft ist besonders im Hinblick auf Bewegungsmodelle von entscheidender Bedeutung, da auch dort logische Beziehungen zwischen Mitgliedern einer Gruppe zum Tragen kommen. Jeder Knoten erhält hierbei eine logische Adresse des Typs <Subnetz, Host> und jedes Subnetz einen so genannten Location Management Server (LMS), welcher sowohl die hierarchische als auch die logische Adresse jedes Gruppenmitglieds kennt. Jedes Mitglied teilt hierfür dem LMS seine augenblickliche hierarchische Adresse mit. Die hierarchischen Adressen der LMS werden den höheren Ebenen mitgeteilt und von oben herab an alle LMS versandt. Sendet nun ein beliebiger Teilnehmer ein Paket an einen Empfänger, zieht er zunächst den Subnetz-Teil aus dessen logischer Adresse heraus, erfährt mit Hilfe seines LMS die hierarchische Adresse des LMS, dem der Empfänger angehört, und überträgt das Paket an diesen. Der empfangende LMS kennt die hierarchische Adresse des Bestimmungsortes und verteilt das Paket an sein endgültiges Ziel.

2.3 Ad-hoc On Demand Distance Vector (AODV) Protocol

Das AODV-Routing ist eine Weiterentwicklung des vorher dargestellten DSDV-Verfahrens. Es reduziert die Netzlast, indem es erst im Bedarfsfall Informationen über die Netztopologie und existierende Verbindungen anfordert, das heißt, Übertragungswege werden erst auf Anfrage entwickelt. Um eine Route zu einem Bestimmungsort zu finden, flutet der Sender das gesamte Netzwerk mit einem Anfragepaket, bis es einen Knoten erreicht, welcher einen entsprechenden Weg kennt, oder bis das Anfragepaket auf den Bestimmungsort selbst trifft. Jeder Knoten, der das Paket weiterleitet, kreiert für sich selbst eine Route, die zurück zum Sender führt, indem er sich den Knoten merkt, von dem er das Anfragepaket als Erstes bekommen hat. Wenn die Weganfrage einen Knoten erreicht, welcher den Weg zum Ziel kennt, generiert dieser ein Antwortpaket, welches die Anzahl der Teilstrecken, die nötig sind, um das Ziel zu erreichen, enthält. Zusätzlich wird das Paket mit einer Sequenznummer versehen, welche Schleifen verhindern und dafür sorgen soll, dass andere antwortende Knoten die aktuellsten Informationen besitzen. Wenn das Antwortpaket an den Sender zurückgeleitet wird, tragen alle sich auf diesem Weg befindenden Knoten denjenigen Knoten in ihre Routing-Tabelle ein, an den sie die zu übertragenden Nutzdaten weitersenden (siehe Abbildung 4).

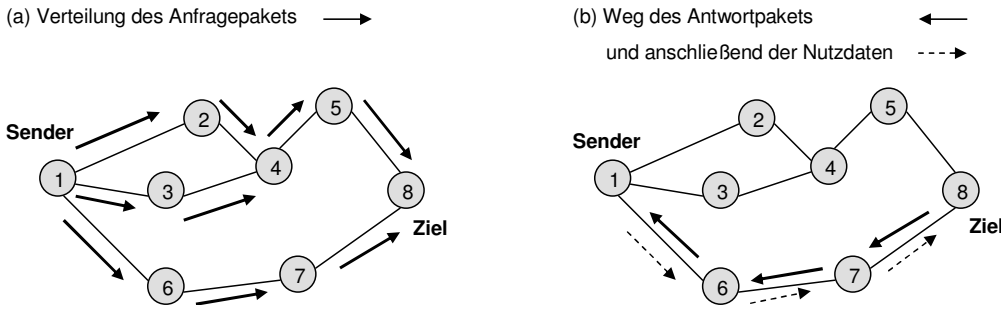


Abbildung 4: Herausfinden des Weges in AODV

3 Verwendung eines Mobilitätsmodells für Gruppen zur Bewertung der Leistungsfähigkeit der Routing-Protokolle für Ad-hoc-Netze

Wie wir gesehen haben, zeichnen sich Ad-hoc-Netze unter anderem dadurch aus, dass die Topologie sowie die Teilnahme der einzelnen Systeme ständig variiert. Die Netzteilnehmer innerhalb einer Gruppe bewegen sich entsprechend verschiedener Muster, was eine Analyse und eine realistische Simulation solcher Bewegungsmuster nahelegt, um deren Auswirkung auf die Stabilität des Netzes und die Effizienz verschiedener Routing-Protokolle herauszufinden. Bisherige Mobilitätsmodelle für Ad-hoc-Netze berücksichtigen lediglich das individuelle Bewegungsverhalten einer einzelnen Station, nehmen jedoch keinen Bezug auf abhängiges Verhalten zwischen Mitgliedern einer Gruppe. Das von den Autoren in [PGHC99], [HGPC99] vorgestellte Reference Point Group Mobility Model bietet eine Möglichkeit, die logischen Beziehungen zwischen mobilen Netzteilnehmern innerhalb einer Gruppe zu beschreiben. Dieses wurde benutzt, um verschiedene Netzwerkszenarien zu simulieren und dabei die Leistungsfähigkeit der bereits vorgestellten Routing-Rotokolle DSDV, HSR und AODV nachzuweisen.

3.1 Funktionsweise des Reference Point Group Mobility (RPGM) Model

Die Grundidee dieses Modells ist, dass sich die Gruppe als Ganzes bewegt, das Verhalten der Gruppen untereinander jedoch unabhängig sein darf. Jeder Gruppe wird ein logisches Zentrum zugewiesen, dessen Bewegung das Verhalten der gesamten Gruppe bezüglich Standort, Geschwindigkeit, Richtung, Beschleunigung, etc. vorgibt. Zusätzlich simuliert das RPGM ein willkürliches Bewegungsverhalten jedes einzelnen Knotens innerhalb seiner Gruppe.

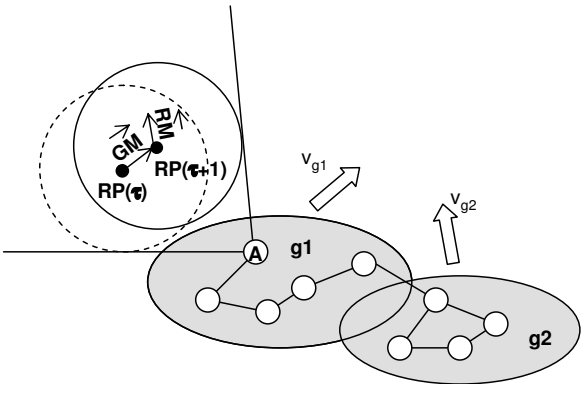


Abbildung 5: Reference Point Group Mobility Model

Jedem Knoten wird ein Referenzpunkt zugeteilt, welcher der Gruppenbewegung folgt und in dessen Umgebung der zugehörige Knoten nach jedem Schritt zufällig positioniert wird. In Abbildung 5 sind zwei Gruppen g1 und g2 dargestellt, welche die Bewegungsvektoren $\vec{v_{g1}}$ und $\vec{v_{g2}}$ besitzen. Greift man beispielsweise den Knoten A der Gruppe g1 zu den Zeitpunkten τ und τ+1 heraus, lässt sich der Standortwechsel folgendermaßen berechnen: Zunächst ändert sich die Position des Referenzpunktes von RP(τ) auf RP(τ+1) gemäß dem Gruppenbewegungsvektor $\vec{GM}$ ($\vec{GM} = \vec{v_{g1}}$). Die neue Position des Knotens A erhält man nun durch Addition eines Zufallsvektors $\vec{RM}$ auf den neuen Referenzpunkt RP(τ+1). Der Zufallsvektor simuliert das willkürliche Bewegungsverhalten des Knotens innerhalb der Gruppe und ist unabhängig von der vorherigen Position des Knotens.

3.2 Anwendungen des Reference Point Group Mobility (RPGM) Model

Durch Anwendung des RPGM-Modells und Festlegung der nötigen Parameter lassen sich nun einige realistische Mobilitäts-Szenarien darstellen:

In-Place Mobility Model: In diesem Modell teilt man ein Gebiet geographisch in mehrere Regionen ein, innerhalb derer sich jeweils eine Gruppe befindet. Diese Gruppen führen unterschiedliche Aufgaben aus und weisen deshalb verschiedenes Bewegungsverhalten auf. Ein Beispiel hierfür wäre eine groß angelegte Katastrophenbergung bestehend aus Feuerwehr-Teams, medizinischer Versorgung, Polizei, etc., welche ihre Arbeit in angrenzenden Nachbarschaften ausüben.

Overlap Mobility Model: In dieser Variante befinden sich mehrere Gruppen überlappend in der gleichen Region und führen dabei verschiedene Aufgaben durch. Die unterschiedlichen Ansprüche der einzelnen Gruppen bedeuten auch in diesem Fall jeweils andere Bewegungsmuster. Greifen wir auf das erwähnte Beispiel der Bergungsarbeiten in einem Katastrophengebiet zurück, erscheint es plausibel, dass sich beispielsweise das Ärzte-Team relativ zügig durch das Gebiet bewegt, während sich andere Betreuer und Helfer länger an einem bestimmten Ort aufhalten.

Convention Mobility Model: Im dritten Fall wird ein typisches Kongressverhalten modelliert. Eine Reihe von Ausstellern präsentieren ihre Forschungsprojekte in separaten, jedoch miteinander verbundenen Hallen. Eine Gruppe von Teilnehmern, beispielsweise desselben Unternehmens, bewegen sich gemeinsam durch die einzelnen Räume, verweilen zeitweise länger an einem bestimmten Stand, ein anderes Mal eilen sie durch einen Raum. Entscheidend ist hierbei die Interaktion zwischen Ausstellern und Teilnehmern.

Local Scope Mobility Model: Das Local-Scope-Mobilitätsmodell ist eine Variante des Convention-Mobilitätsmodells und unterscheidet sich von diesem nur dadurch, dass Datenverkehr nur innerhalb der Gruppe zugelassen ist. Das heißt in unserem Beispiel, die teilnehmende Gruppe gerät zwar mit anderen Gruppen und Teilnetzen in Kontakt, wodurch sich die Netztopologie und damit die Routing-Tabellen der inneren Knoten permanent ändern, Interaktion zwischen Mitgliedern der Gruppe und externen Knoten findet jedoch nicht statt.

3.3 Beurteilung der Leistungsfähigkeit der unterschiedlichen Routing-Protokolle unter verschiedenen Mobilitäts-Szenarien

In Ad-hoc-Netzen verursachen bereits geringfügige Standortänderungen von Teilnehmern entscheidende Änderungen in der Topologie des Netzes und wirken sich deutlich auf die Effizienz

der Protokolle in höheren Schichten aus, etwa auf den Durchsatz und die Verzögerung. Dieser Abschnitt stellt den Einfluss der zuvor erläuterten Bewegungsmodelle sowie steigender Mobilität auf die Leistung der Routing-Protokolle DSDV, HSR und AODV dar. Unter Mobilität ist im Folgenden von einer auf der durchschnittlichen Geschwindigkeit der Gruppe sowie mittleren Verschiebung der Knoten um ihren Referenzpunkt basierenden Größe auszugehen. Um die Leistungsfähigkeit dieser Protokolle zu bewerten, wurden, mittels der Simulation eines drahtlosen mobilen Ad-hoc-Netzes sowie der verschiedenen Bewegungsszenarien, einige Experimente durchgeführt und die folgenden Messgrößen betrachtet:

- *Verbindungsauf- und -abbaurate und Clusterhead-Wechsel:*

  Als erster Beurteilungsfaktor lässt sich die Häufigkeit des Verbindungsauf- und -abbaus heranziehen. Bewegen sich zwei Knoten, welche gerade gegenseitig Daten austauschen, weit auseinander, geht die Verbindung verloren, gelangen dagegen zwei Knoten in den Bereich ihrer Übertragungsreichweite, entsteht eine neue Verbindung. Ebenfalls als ein Indiz dafür, in welchem Maße Bewegungsverhalten die Leistungsfähigkeit der Protokolle beeinträchtigt, gilt die Häufigkeit der Clusterhead-Wechsel. Der Clusterhead fungiert als regionaler Broadcast-Knoten und Koordinator für Übertragungen innerhalb seines Bereichs. Wird sehr oft ein neuer Clusterhead gewählt, bedeutet dies eine instabile Netzwerkinfrastruktur für höhere Schichten.

  Da die Häufigkeit des Verbindungsauf- beziehungsweise -abbaus und gegebenenfalls des Clusterhead-Wechsels lediglich von dem zugrundeliegenden Bewegungsverhalten und nicht von der Wahl des Routing-Protokolls abhängt, war es ausreichend, sich in diesem ersten Versuch auf ein einfaches Bellman-Ford Routing-Protokoll zu beschränken. Als Vergleich zu den vier spezifizierten Bewegungsmodellen wird ein so genanntes Random-Modell eingeführt, bei dem sich alle Teilnehmer völlig willkürlich bewegen; eine Form der Gruppenabhängigkeit gibt es hier nicht.

  Aus den Experimenten ging hervor, dass bei zunehmender Mobilität alle Szenarien zu einer steigenden Verbindungsauf- und -abbaurate führen. Erwartungsgemäß ist die Rate beim Random-Modell aufgrund beliebiger Bewegung der Teilnehmer am höchsten, beim Convention-Modell durch die sich kaum bewegenden Ausstellergruppen dagegen am geringsten.

  Die Clusterhead-Wechselhäufigkeit verhält sich ähnlich und wächst bei allen Szenarien ebenfalls proportional mit der Mobilität, das Random-Modell hebt sich jedoch auch in dieser Kategorie deutlich von den übrigen Modellen ab und zeichnet sich durch eine hohe Anzahl an Wechseln der Clusterheads pro Sekunde aus. Am besten schneidet hier das In-Place-Modell ab, was daran liegt, dass sich die Mitglieder einer Gruppe nur innerhalb ihres begrenzten Bereiches bewegen dürfen und somit weniger Blockbildungsmöglichkeiten als beispielsweise beim Overlap-Modell bestehen, bei dem sich die Mitglieder aller Gruppen über das gesamte Areal verteilen.

- *Durchsatz:*

  Als nächste Größe betrachten wir den erzielten Durchsatz bei Verwendung verschiedener Routing-Protokolle in Abhängigkeit der fünf eingeführten Bewegungsmodelle und Steigerung der Mobilität.

  Zunächst wurde der Durchsatz, der mit Hilfe des DSDV-Protokolls erreicht wird, gemessen. Egal, welches Gruppenbewegungsszenario sich abspielt, sinkt die Rate der ankommenden Pakete bei zunehmender Mobilität der Teilnehmer drastisch und bleibt ab einem bestimmten Wert auf relativ niedrigem Niveau stehen. Den niedrigsten Wert erzielt das Random-Modell, unter allen anderen Bewegungsmustern unterscheiden sich

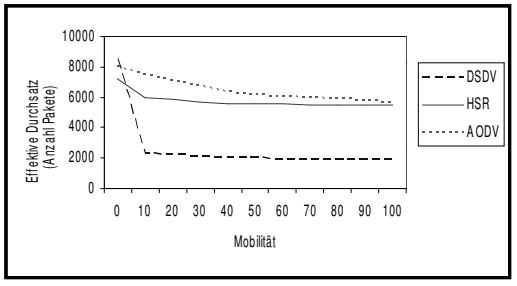
die einzelnen Werte kaum. Zurückzuführen ist dieses insgesamt schlechte Durchsatzresultat des DSDV-Protokolls, wie wir später sehen werden, auf den stark anwachsenden Kontroll-Overhead.

Die Verwendung des HSR-Verfahrens führt bei den Gruppenbewegungsmodellen zu einem deutlich besseren Resultat, was daran liegt, dass die hierarchische Struktur der innerhalb von HSR geformten Teilnetze in Einklang mit dem Bewegungsverhalten der Gruppen steht. Dies ist auch der Grund dafür, dass der Durchsatz im Falle des Random-Modells erneut an letzter Stelle liegt. Den höchsten Durchsatz erzielt HSR unter dem Local-Scope-Szenario, weil Datenverkehr nur innerhalb einer Gruppe erlaubt ist.

Der Durchsatz unter Verwendung des AODV-Protokolls variiert bei verschiedenen Mobilitäts-Szenarien stärker, als es bei den vorherigen Versuchen mit DSDV und HSR der Fall war. Befinden sich Sender und Empfänger innerhalb derselben sich fortbewegenden Gruppe, wie es beispielsweise im Local-Scope-Modell der Fall ist, liefert AODV sehr gute Ergebnisse. Dies ist unter anderem bedingt durch die Tatsache, dass die Übertragungswege hier aufgrund der am meisten eingeschränkten Bewegungsfreiheit länger als in anderen Fällen erhalten bleiben und die Effizienz aufgrund weniger fallengelassener Pakete steigt.

Abbildung 6(a) soll die tendenzielle Entwicklung in Bezug auf den Durchsatz, der bei steigender Mobilität durch die drei Routing-Protokolle DSDV, HSR und AODV gewährleistet wird, graphisch zum Ausdruck bringen. Repräsentativ wurde ein Mittelwert aus den unter verschiedenen Bewegungsszenarien für den Durchsatz erhaltenen Werten gebildet.

(a) Durchsatz bei steigender Mobilität



(b) Kontroll-Overhead bei steigender Mobilität

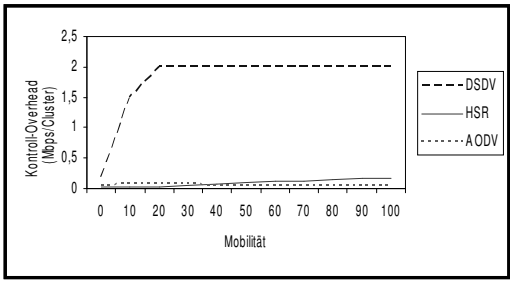


Abbildung 6: Überblick über die Leistungsfähigkeit der Routing-Protokolle

- *Kontroll-Overhead:*

  Als letzte der drei Bewertungsgrößen dient nun die Größe des Kontroll-Overheads in Abhängigkeit der einzelnen Bewegungsszenarien und wiederum steigender Mobilität.

  Das Verhalten des Kontroll-Overheads steht unter Verwendung des DSDV-Verfahrens in Einklang mit dem des Durchsatzes. Steigt die Mobilität der Teilnehmer, brechen häufiger Verbindungen auf, zahlreiche Änderungen der Routing-Tabellen werden notwendig, ein starkes Anwachsen des Kontroll-Overheads ist die Folge.

  Die Leistungsfähigkeit von HSR in diesem Zusammenhang ist um mehr als das zehnfache besser als die von DSDV. Insgesamt liegen die unter den jeweiligen Bewegungsmodellen erhaltenen Werte dicht beieinander, die einzige Ausnahme stellt das Local-Scope-Modell dar, welches sich durch einen noch geringeren Kontroll-Overhead von den übrigen Modellen abhebt.

  AODV liefert, was die Netzlast angeht, welche bei steigender Mobilität durch den Kontroll-Overhead entsteht, hervorragende Werte. Zwar ist bei geringer Mobilität bei je-

dem Bewegungsverhalten anfangs die Rate der übertragenen Kontrolldaten etwas höher als dies unter gleichen Bedingungen bei HSR der Fall war, jedoch steigt im Gegensatz zu HSR die Rate nicht kontinuierlich an, sondern bleibt auf einem sehr niedrigen Niveau stehen. Auch hier zeigt sich bezüglich der zugrundeliegenden Bewegungsmodelle eine genau umgekehrt proportionale Entwicklung des Kontroll-Overheads zu der des Durchsatzes.

In Abbildung 6(b) wird grob der Zusammenhang zwischen steigender Mobilität und beanspruchter Datenrate für den Kontroll-Overhead bei Verwendung der verschiedenen Protokolle dargestellt. Auch hier wurde ein Durchschnittswert der erzielten Ergebnisse unter den jeweiligen Szenarien für den Kontroll-Overhead gewählt.

3.4 Zusammenfassung der Ergebnisse

Das Gruppenmobilitätsmodell RPGM bietet eine Möglichkeit, Netzteilnehmer entsprechend ihrer logischen Beziehungen untereinander in Gruppen einzuteilen. Die auf Basis dieses Modells beschriebenen Bewegungsmuster dienten als Grundlage, um den Einfluss verschiedenen Bewegungsverhaltens auf die Leistungsfähigkeit unterschiedlicher Protokolle, hier DSDV, HSR und AODV zu zeigen.

Zu den aus den durchgeführten Experimenten resultierenden und zu erwartenden Ergebnissen zählt, dass, je mehr Dynamik, sei es steigende Mobilität oder geringer eingeschränkte Gruppenbewegung, ein Modell aufweist, sich sowohl die Verbindungen als auch die Clusterheads häufiger ändern.

Des weiteren geht aus den Versuchen hervor, dass die eingeführten Routing-Protokolle auf das jeweils zugrundeliegende Mobilitätsmodell -auch wenn teilweise nur geringfügig- unterschiedlich reagieren. Stimmt die Struktur des durch das jeweilige Routing-Protokoll aufgebauten Netzes mit bestimmten Eigenschaften der Gruppendynamik überein, wie es bei HSR und AODV bezüglich des Local-Scope-Modells der Fall ist, können Größen wie der Durchsatz erheblich bessere Werte erzielen. Maximal wird die Leistung des HSR-Routing-Protokolls, wenn die logischen Teilnetze den Gruppen des Mobilitätsmodells entsprechen. Minimal wird sie hingegen, wenn sich die Mitgliederzahl je Gruppe auf eins beläuft, jeder Nutzer also sein eigenes unabhängiges Bewegungsmuster besitzt.

Die Graphiken aus Abbildung 6 lassen zweifelsfrei erkennen, dass sich HSR und AODV bei hoher Mobilität der Gruppenmitglieder aufgrund ihres dynamischen beziehungsweise logischen Aufbaus eher bewähren als DSDV, welches nur bei sehr geringer Mobilität der Netzteilnehmer gute Resultate erzielt.

4 Einbeziehung von Mobilitätsinformation in Routing-Protokolle zur Verbesserung des Routings in mobilen Ad-hoc-Netzwerken

Die Autoren von [SuLG00b],[SuLG00a] beschreiben ein Verfahren, welches Mobilitätsinformation in Routing-Protokolle einbezieht und sich Mobility Prediction nennt. Das Grundprinzip der Mobilitäts-Vorhersage besteht darin, das nicht zufällige Bewegungsverhalten der mobilen Nutzer zu analysieren, mit Hilfe der gewonnenen Information die zukünftige Netzwerktopologie vorherzusagen und somit auch während des Änderungsprozesses der Netztopologie eine durchgängige Übertragung von Daten zu gewährleisten. Ähnlich wie ein Auto, das auf einer Strasse fährt, folgen auch die Teilnehmer von Ad-hoc-Netzen häufig in gewissem Maße gleichmäßig einem Weg und weichem von diesem nur geringfügig ab.

4.1 Funktionsweise der Mobilitäts-Vorhersage

Die mit Hilfe dieses Modells erhaltene Mobilitätsinformation fließt in die im Folgenden vorgestellten und getesteten Unicast- und Multicast-Protokolle ein und soll die Größe des Overheads, welcher bei der Rekonstruktion neuer Wege entsteht, minimieren. Während im vorherigen Abschnitt der Einfluss von Gruppenbewegungsverhalten auf die Leistungsfähigkeit einzelner Protokolle überprüft wurde, ist hier die Mobilitätsinformation direkter Bestandteil der Routing-Protokolle.

Um die einzelnen Knoten zu lokalisieren, bedient sich dieses Modell des GPS-Systems. Während einer Datenübertragung werden die GPS-Positionsdaten der jeweiligen Partner an die gesendeten Pakete angehängt und verwendet, um die so genannte Link Expiration Time (LET) zu bestimmen. Die LET gibt an, wie lange eine Verbindung zwischen zwei angrenzenden Knoten voraussichtlich noch bestehen bleibt. Ziel der Vorhersage ist, eine durchgängige Verbindung ohne Unterbrechungen zu garantieren, indem Wege neu konfiguriert werden, ehe sich Verbindungen auflösen und Datenpakete verloren gehen.

Wie sich die LET unter Kenntnis der Bewegungsparameter Geschwindigkeit, Richtung und Übertragungsweite berechnen lässt, wird im Folgenden beschrieben. Seien i und j zwei Knoten, welche sich innerhalb ihrer Übertragungsreichweite r befinden und (x_i,y_i) beziehungsweise (x_j,y_j) die zugehörigen Koordinaten der beiden Partner. Des Weiteren seien v_i und v_j deren Geschwindigkeit und θ_i und θ_j mit $0 \leq \theta_i$, $\theta_j < 2\pi$ die Bewegungsrichtung der Knoten i und j. Dann gilt für die LET zwischen den beiden Knoten:

$$LET = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2}$$

mit $a = v_i \cos\theta_i - v_j \cos\theta_j$,
$b = x_i - x_j$,
$c = v_i \sin\theta_i - v_j \sin\theta_j$,
$d = y_i - y_j$

Falls $v_i = v_j$ und $\theta_i = \theta_j$ erhält die LET den Wert ∞.

4.2 Anwendungen des Prinzips der Mobilitäts-Vorhersage

Dieser Abschnitt stellt drei in [SuLG00b],[SuLG00a] beschriebene Routing-Protokolle vor, welche sich den Mechanismus der Mobilitäts-Vorhersage zu Nutze machen. Das Hauptziel war, Routing-Protokolle robuster gegenüber der Mobilität der Teilnehmer und effizienter bezüglich der beanspruchten Bandbreite zu gestalten. Bei den ersten beiden Protokolltypen handelt es sich um Unicast-Protokolle, die sich aufgrund ihrer Routing-Philosophie („on-demand“ gegenüber „table-driven“) unterscheiden, bei dem dritten Protokolltyp handelt es sich um ein Multicast-Routing-Protokoll.

4.2.1 Flow Oriented Routing Protocol (FORP)

Das Flow Oriented Routing Protocol ist ein Routing-Verfahren ähnlich dem zuvor beschriebenen AODV-Protokoll, mit dem Unterschied, dass der Routing-Algorithmus die Mobilitätsinformation einbezieht. Das heißt, auch hier werden Übertragungswege nur im Bedarfsfall

konstruiert, eine ständige Aktualisierung von Routing-Tabellen ist nicht erforderlich. Die Besonderheit, durch die sich FORP gegenüber dem AODV-Verfahren auszeichnet, liegt in den geänderten Kriterien der Wegwahl. Der Empfänger erkennt bei FORP die Änderung der Netztopologie im Voraus und legt mit Hilfe der in den Datenpaketen enthaltenen Mobilitätsinformation fest, zu welchem Zeitpunkt ein neuer Weg für den Datenfluss gewählt werden muss. Es wird hierfür angenommen, dass jeder Knoten in der Lage ist, die Dauer, nach welcher die Verbindung zu seinen angrenzenden Knoten aufbricht, vorherzusagen. Lässt sich demnach die LET jeder Teilverbindung zwischen allen auf dem Übertragungsweg liegenden Knoten vorhersagen, kann hieraus die RET (Route Expiration Time) abgeleitet werden, welche das Minimum aller LETs entlang des Wegs darstellt. Die RET gibt also an, wie lange ein bestimmter Übertragungsweg zwischen Sender und endgültigem Empfänger zur Verfügung steht. Im Unterschied zu AODV wird nicht der Weg gewählt, welcher die wenigsten Zwischenstationen aufweist, sondern der, welcher am längsten bestehen bleibt.

Abbildung 7 verdeutlicht, wie die Wegwahl bei FORP beispielsweise aussieht. Quelle A sendet eine Anfrage an den Bestimmungsort G (siehe Abbildung 7(a)). Die übrigen Knoten leiten die Anfrage an G weiter und fügen der Nachricht ihre Knotenkennung sowie die LET der Teilverbindung, über welche sie die Anfrage Nachricht erhalten haben, hinzu. G empfängt schließlich zwei Anfragepakete. Eines beinhaltet den Weg <A,B,C,F,G> und die dazugehörigen LETs <6,2,4,5>, das andere den Weg <A,B,D,E,F,G> mit den LETs <6,4,3,3,5>. Somit erhält der Knoten F automatisch die RET der beiden Routen, indem er jeweils das Minimum der LETs beider Wege bestimmt. Da die RET von Route <A,B,D,E,F,G> 3 gegenüber 2 von Route <A,B,C,F,G> höher ist, ist der in Abbildung 7(b) gekennzeichnete Weg der stabilere und wird anschließend für die Übertragung der Nutzdaten von A nach G benutzt.

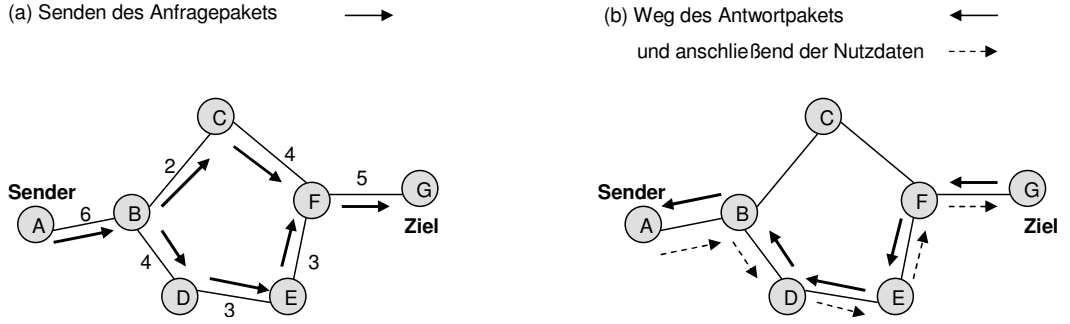


Abbildung 7: Wegwahl bei FORP

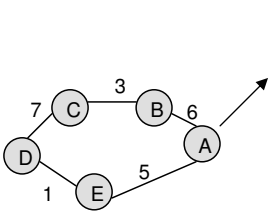
4.2.2 Distance Vector with Mobility Prediction (DV-MP)

Wie schon bei den Untersuchungen mit Hilfe des RPGM-Modells zu sehen war, reagieren typische Distanz-Vektor-Protokolle wie das DSDV-Protokoll sehr empfindlich auf steigende Mobilität, da Routing-Tabellen häufiger ausgetauscht und aktualisiert werden müssen, was wiederum ein starkes Anwachsen des Kontroll-Overheads zur Folge hat. Um diesem Problem entgegenzuwirken, versuchen die Autoren von [SuLG00b],[SuLG00a], Mobilitätsinformation in das Routing-Protokoll einfließen zu lassen, und nennen dieses weiterentwickelte Routing-Schema Distance Vector with Mobility Prediction (DV-MP). Die Spalte „Anzahl Hops“, die in den Routing-Tabellen des DSDV-Schemas als Wegauswahlkriterium herangezogen wird, ersetzt man beim DV-MP-Routing durch eine Spalte, welche die RETs bei Wegwahl über den entsprechenden angrenzenden Knoten enthält. Die ständige Vermittlung ereignisabhängiger Updates kann vermieden werden, da der Stabilitätsfaktor für die Wegwahl entscheidend ist. Ein weiterer Vorteil dieser Vorgehensweise ist, dass durch Auseinanderbewegung verursachte

Unterbrechungen vermindert werden, da eine Route mit einer höheren RET eine bestehende Verbindung ersetzt, bevor diese aufbricht. Besonders nützlich ist diese Eigenschaft im Hinblick auf die Übertragung von Echtzeitdaten wie beispielsweise Sprache und Video. Jeder Knoten führt dennoch regelmäßig ein Broadcast seiner Routing-Tabelle durch und fügt seinen Bewegungsvektor dem Update-Paket hinzu, damit die angrenzenden Knoten die aktuelle LET dieser Teilverbindung daraufhin bestimmen können. Wird ein Update durchgeführt, wird die dazugehörige Sequenznummer ausgegeben und anschließend nach jedem Broadcast der Routing-Tabelle erhöht. Das Update einer Routing-Tabelle für einen beliebigen Knoten wird nach folgenden Regeln durchgeführt:

- Empfängt der Knoten einen Eintrag für einen seiner Zielbestimmungsorte, welcher eine bessere RET aufweist, und ist die dazugehörige Sequenznummer größer beziehungsweise gleich der momentan in seiner Tabelle für dieses Ziel aufgeführten Sequenznummer, so aktualisiert der Knoten seinen Eintrag für dieses Ziel.

- Empfängt der Knoten einen Eintrag für einen seiner Zielbestimmungsorte, welcher eine höhere Sequenznummer als der momentan in seiner Tabelle für dieses Ziel vorhandene Eintrag hat, aktualisiert der Knoten seinen Eintrag für dieses Ziel.

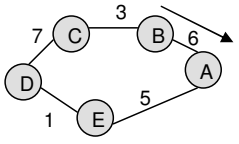


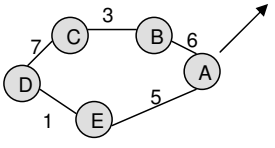
Originale Routing-Tabelle für A

Ziel	nächster Hop	RET	Sequenz Nummer
A	N/A	N/A	3
B	B	6	2
C	B	3	2
D	E	1	2
E	E	5	2

Routing-Tabelle erhalten von B

Ziel	nächster Hop	RET	Sequenz Nummer
A	A	6	2
B	N/A	N/A	3
C	C	3	3
D	C	3	3
E	A	5	2





Geänderte Routing-Tabelle für A

Ziel	nächster Hop	RET	Sequenz Nummer
A	N/A	N/A	3
B	B	6	3
C	B	3	3
D	B	3	3
E	E	5	2

Abbildung 8: Beispiel für ein Update einer Routing-Tabelle nach dem DV-MP-Verfahren

Wie die Aktualisierung der Routing-Tabelle abläuft, wird anhand Abbildung 8 veranschaulicht. Anfangs wählt Knoten A als nächsten Knoten E, um eine Verbindung zu D aufzubauen. Die RET bei Wegwahl über E beträgt hier 1. Nachdem A das Update-Paket von B erhalten hat, ändert der Knoten seine Route zu D und wählt als nächsten Knoten B, um gegebenenfalls Daten nach D zu übertragen. Kriterium für die neue Wegwahl ist die auf dieser Route höhere RET von 3 gegenüber der vorherigen RET von 1.

4.2.3 On Demand Multicast Routing Protocol (ODMRP)

Im Gegensatz zu den vorherigen beiden Protokollen handelt es sich beim ODMRP [LeGC99] um ein Multicast-Protokoll, welches Datenpakete an mehrere Empfänger sendet. Ähnlich wie beim AODV- und FORP-Routingverfahren werden Wege erst im Bedarfsfall entwickelt und nur aktive Verbindungen beibehalten. Durch die Anfrage- und Antwortphase werden Wege vom Sender zum Empfänger generiert und dabei ein Netz von Knoten, die so genannte Weiterleitungsgruppe, aufgebaut. Anhand Abbildung 9(a) kann das Konzept der Weiterleitungsgruppe näher erläutert werden. Diese Gruppe besteht aus einer Menge von Knoten (Multicast-Empfänger und Weiterleitungsknoten), welche die Aufgabe haben, Multicast-Pakete zu verteilen. Zu beachten ist hierbei, dass ein Multicast-Empfänger gleichzeitig die Rolle eines Weiterleitungsknotens, also eines Routers, einnehmen kann, wenn er sich auf dem Weg zwischen Quelle und einem weiteren Empfänger befindet.

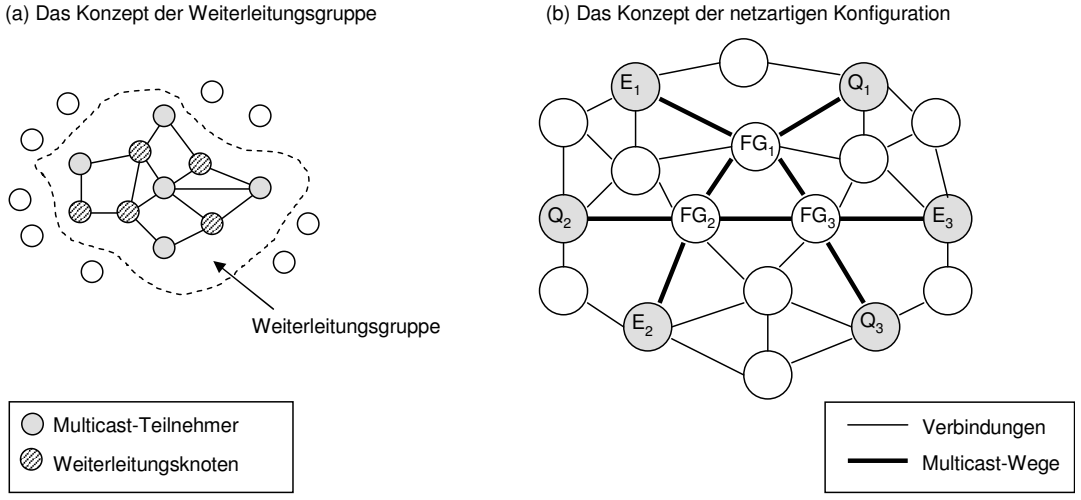


Abbildung 9: Konzepte des ODMRP

Die Konstruktion eines Netzes zur Verteilung der zu übertragenden Daten zeichnet sich gegenüber typischen Übertragungsbäumen dadurch aus, dass entlang der Multicast-Teilnehmer reichere Verbindungen gewährleistet werden und neue Konfigurationen nicht mehr permanent erforderlich sind. Eine anschauliche Darstellung liefert Abbildung 9(b), in der die Robustheit einer maschen- beziehungsweise netzartigen Konfiguration der erforderlichen Verbindungen verdeutlicht wird. Drei Quellen Q_1, Q_2, Q_3 senden Multicast-Pakete an drei Empfänger E_1, E_2, E_3 über die Weiterleitungsknoten FG_1, FG_2, FG_3. Angenommen, der Übertragungsweg von Q_2 zu E_3 wäre Q_2-FG_2-FG_3-E_3. Bricht nun die Verbindung zwischen FG_2 und FG_3 auf, würde der Datenverkehr bei Konfiguration entsprechend dem Baumschema zwischen Q_2 und E_3 solange unterbrochen bleiben, bis der Übertragungsbaum neu konfiguriert wäre. Bei Verwendung von ODMRP hingegen existiert bereits eine Ausweichroute über Q_2-FG_2-FG_1-FG_3-E_3 und diese ermöglicht einen durchgängigen, also unterbrechungsfreien Datentransfer. Auch wenn ständiges Fluten mit Änderungsnachrichten bei ODMRP nicht nötig ist, muss dies zumindest in regelmäßigen Abständen geschehen, um neue Wege zu konstruieren und alte zu aktualisieren. Die Schwierigkeit bei Verwendung des ODMRP-Verfahrens liegt darin, das optimale Zeitintervall, in dem man flutet, zu bestimmen, um einerseits die Aktualität der Routen zu wahren und andererseits dadurch entstehenden Stau und Kollisionen zu minimieren. Hier kommt jetzt der Nutzen der Mobilitäts-Vorhersage und somit Bestimmbarkeit der LETs zum Tragen, da man mit Hilfe dieser das gesuchte Intervall an das Bewegungsmuster und die Geschwindigkeit anpassen kann. Anfragepakete müssen erst gesendet werden, wenn das Aufbrechen aktiver Verbindungen bevorsteht.

4.3 Beurteilung der Leistungsfähigkeit der unterschiedlichen Routing-Protokolle bei Einbeziehung von Mobilitätsinformation

In diesem Kapitel werden Experimente beschrieben, die von den Autoren von [SuLG00b] und [SuLG00a] durchgeführt wurden, um zu untersuchen, in welchem Ausmaß sich das Routing in Ad-hoc-Netzen verbessert, wenn man Bewegungsinformation in Routing-Protokolle einbezieht, und inwiefern sich die Vorhersagegenauigkeit auf die Leistungsfähigkeit der Protokolle auswirkt. Da bisher von der Annahme ausgegangen wurde, dass weder plötzliche Richtungswechsel noch starke Geschwindigkeitsänderungen der Teilnehmer vorkommen, müssen ebenso die Folgen einer falschen Hervorsage betrachtet werden.

Für die Versuche wurde ein drahtloses mobiles Netzwerk simuliert; als messbare Größe fließt die Paketzustellungsrate bei Variation der Mobilitäts-Geschwindigkeit in die Bewertung ein. Die Paketzustellungsrate stellt die Anzahl der tatsächlich ankommenden Datenpakete im Verhältnis zu der vorgesehenen Anzahl von den Empfänger erreichenden Datenpaketen dar. Es wurde in diesen Tests die Leistungsfähigkeit der vorgestellten Unicast-Protokolle FORP und DV-MP, sowie des einfachen Distanz-Vektor-Protokolls für Ad-hoc-Netze WRP, welches keine Mobilitätsinformation einbezieht, beurteilt. Im Multicast-Bereich wurde einmal die Leistungsfähigkeit des gewöhnlichen On-demand Multicast Routing Protocol getestet und diese daraufhin mit der Leistungsfähigkeit von dessen Weiterentwicklung ODMRP-MP verglichen.

Zusätzlich wurden anschließend die Auswirkungen einer ungenauen Vorhersage auf die Leistungsfähigkeit der Bewegungsinformation einbeziehenden Protokolle betrachtet. Hierzu wurde die Rate der Richtungswechsel erhöht; das Bewegungsverhalten unterliegt somit einem steigenden Grad an Zufälligkeit. Diese Eigenschaft wurde anhand des Distanz-Vektor-Protokolls DV mit und ohne Mobilitäts-Vorhersage untersucht.

- *Paketzustellungsrate in Abhängigkeit von der Mobilitäts-Geschwindigkeit:*

  Im ersten Experiment wurde der Anteil der ankommenden Pakete in Abhängigkeit von der Mobilitäts-Geschwindigkeit gemessen. Beim Distanz-Vektor-Protokoll WRP ist bei steigender Geschwindigkeit ein deutlicher Abfall der Paketzustellungsrate auf unter 20 Prozent zu erkennen. Als Hauptursachen lassen sich nennen: Verbindungen brechen auf, Updates und deren Bestätigungen werden benötigt, Schleifen lassen sich wegen langsamer Verteilung der Updateinformationen im Netz nicht vermeiden, die Folge sind eine Fülle von zusätzlichen Paketen, welche zu Kollisionen, Stau und dem Verwerfen einzelner Pakete führen.

  Die anderen beiden Unicast-Protokolle DV-MP und FORP reagieren hingegen wesentlich unempfindlicher auf höhere Geschwindigkeiten und weisen eine Zustellungsrate von mindestens 90 Prozent auf. Dies liegt im Wesentlichen darin begründet, dass die Stabilität einer Verbindung als Hauptauswahlkriterium für die Wegwahl verwendet wird und neue Verbindungen bereits aufgebaut werden, bevor alte verloren gehen.

  Bei den Multicast-Protokollen ODMRP und ODMRP-MP verhält sich die Entwicklung der Zustellungsrate unter Zunahme der Geschwindigkeit ähnlich. ODMRP-MP verteilt über 90 Prozent der Daten an die jeweiligen Multicast-Empfänger, was ebenfalls aus den im Voraus konstruierten Routen resultiert. Ohne Einbeziehung von Mobilitätsinformation und somit ohne Anpassung des Intervalls für Anfrage- und Antwortpakete an hohe Geschwindigkeiten existieren Wege, die während der Anfragephase noch bestanden, zur Antwortphase schon nicht mehr. Deshalb sinkt die Rate der zugestellten Daten bei ODMRP vergleichsweise zu ODMRP-MP stärker.

  In Abbildung 10 wird der Zusammenhang zwischen Paketzustellungsrate und steigender Mobilitäts-Geschwindigkeit bei den erwähnten Protokollen tendenziell veranschaulicht.

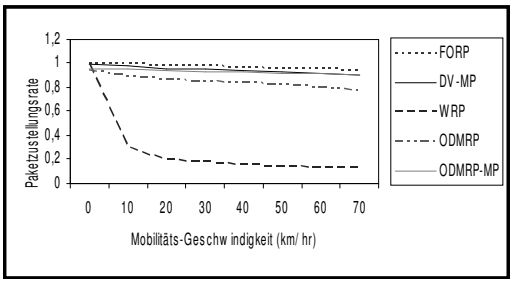


Abbildung 10: Paketzustellungsrate bei steigender Geschwindigkeit

- *Paketzustellungsrate in Abhängigkeit von der Häufigkeit der Richtungswechsel:*

  Der nun behandelte Zusammenhang soll als ein Indiz dafür gelten, wie sich die Einbeziehung einer ungenauen beziehungsweise fehlerhaften Vorhersage in Routing-Protokolle auf die Anzahl der ankommenden Datenpakete auswirkt. Wir vergleichen hierzu die Paketzustellungsraten der Protokolle DV-MP und DV.

  Der DV-Algorithmus verwendet in seiner Struktur keine Mobilitätsinformation und weist auch bei Zunahme der Richtungswechsel der Teilnehmer pro Sekunde keine Veränderung der Paketzustellungsrate auf. DV-MP dagegen zeigt eine gewisse Abhängigkeit von der Anzahl der Richtungswechsel pro Sekunde. Steigen diese, liegt nahe, dass die Vorhersage ungenauer wird, sich das Update-Intervall verkürzt, die Größe des Kontroll-Overheads wächst, und somit die Paketzustellungsrate im Fall der Verwendung von DV-MP sinkt.

  Dennoch ist, wie in Abbildung 11 grob dargestellt, die Rate des DV-MP-Protokolls im Vergleich zu der bei Verwendung des DV-Protokolls ohne Mobilitäts-Vorhersage deutlich höher.

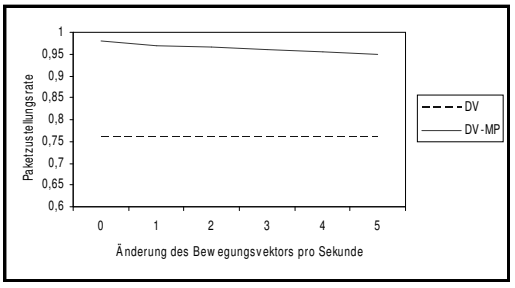


Abbildung 11: Zustellungsrate in Abhängigkeit von der Anzahl der Richtungswechsel pro Sekunde

4.4 Zusammenfassung und Ergebnisse

Wie in der Einleitung verdeutlicht wurde, treten in Ad-hoc-Netzen wegen der Dynamik der Netzteilnehmer eine Reihe von Problemen auf, die in Festnetzen nicht zum Tragen kommen. Probleme wie das effektive Zustellen von Dateneinheiten und die Minimierung der Anzahl aufgrund unterbrochener aktiver Übertragungswege verworfenen Pakete versucht man durch Einbeziehung von Mobilitätsinformation in Routing-Protokolle in den Griff zu bekommen. Die Funktionsweise der Mobilitäts-Vohersage besteht darin, eine Aussage über die zukünftige Netztopologie zu treffen und so durch Wahl des stabilsten Übertragungsweges eine Route neu zu konfigurieren, bevor die momentan genutzte Verbindung aufbricht. Angewendet wurde

dieses Prinzip auf drei repräsentative Protokolle der gängigsten Routing-Verfahren für Ad-hoc-Netze. Die zum Zweck der Leistungsbewertung durchgeführten Experimente zeigen, dass selbst bei ungenauer Vorhersage die Anzahl der ankommenden Pakete im Falle der Verwertung der Mobilitätsinformation durch die aufgeführten Routing-Protokolle höher ist als ohne Einbeziehung von Mobilitätsinformation.

5 Kritik und Schlussbetrachtung

Bezüglich des Ziel, Routing in Ad-hoc-Netzen zu verbessern, gelangt man zu dem eindeutigen Resultat, dass unter Zuhilfenahme so genannter Mobilitätsprofile Routing-Protokolle deutlich an Effizienz gewinnen können und somit eine Leistungssteigerung des Netzes bewirkt werden kann. Diese Beobachtung resultiert aus den zu diesem Zweck von den Autoren von [PGHC99], [HGPC99], [SuLG00b], [SuLG00a] durchgeführten Experimenten.

An den Untersuchungen mit Hilfe des RPGM-Modells gilt es zu kritisieren, dass verschiedenes Bewegungsverhalten zwar zu unterschiedlicher Leistungsfähigkeit der jeweiligen Protokolle führt, jedoch auch unter Berücksichtigung des zugrundeliegenden Mobilitätsmodells die Wahl des Routing-Protokolls eindeutig zu seien scheint. Gemeint ist damit, dass beispielsweise Protokoll x bei jedem Bewegungsmuster unterschiedliche Reaktionen zeigt, aber trotzdem immer ein besseres Ergebnis als Protokoll y und z liefert. Viel bedeutender erscheint hier die Erkenntnis, dass AODV und HSR bei den Experimenten ähnlich gut und deutlich besser als DSDV abschneiden. Man könnte sogar zusätzlich auf den Einfluss der Mobilität verzichten, und hier die Aussage treffen „Mit AODV fährt man immer gut“. Interessanter wäre demnach ein Ergebnis aus den Untersuchungen gewesen, welches folgendermaßen ausgesehen hätte: „Wähle bei zugrundeliegendem Bewegungsmodell x und Mobilität y am besten ein Routingverfahren nach dem Schema z.“ Verstärkt wird der Eindruck, dass die Nicht-Berücksichtigung des zugrundeliegenden Mobilitätsmodells sinnvoller wäre, dadurch, dass Bewegungsverhalten in der Realität sicher oft nur ungenau vorhersehbar beziehungsweise abgrenzbar ist. Da aber dennoch festzustellen war, dass etwa ein Bewegungsverhalten gemäß dem Random-Modell die niedrigste Leistungsfähigkeit aller Protokolle verursacht, sollten beim Entwurf neuer Routing-Protokolle und somit bei hierfür benötigten Tests und Simulationen realistische Bewegungsszenarien von Gruppen modelliert werden.

Des Weiteren sollte darauf hingewiesen werden, dass Distanz-Vektor-Protokolle trotz der schlechten Resultate in den durchgeführten Experimenten durchaus auch positive Eigenschaften besitzen: Die Autoren von [SuLG00a] stellen einerseits den Vorteil heraus, dass Wege bei einer Anfrage nicht erst entwickelt werden müssen, sondern bereits bestehen. Andererseits eignen sich DV-Protokolle für Netze mit einer hohen Teilnehmerzahl, da der Routing-Overhead unabhängig von der Anzahl der Sender ist, während bei „on-demand“-Protokollen das Fluten von Anfragepaketen bei steigender Senderanzahl häufiger vorkommt.

Um die Brauchbarkeit der DV-Protokolle auch bei hoher Mobilität zu gewährleisten, beziehen die Autoren von [SuLG00b],[SuLG00a] Mobilitätsinformation direkt in Routing-Protokolle ein und erzielen deutliche Verbesserungen was Größen wie den Durchsatz angeht. Auch die anderen vorgestellten Mobilitätsinformation einbeziehenden Protokolle FORP und ODMRP-MP zeichnen sich durch hohe Robustheit gegenüber Zunahme der Mobilität aus, sind beides jedoch „on-demand“-Routing-Protokolle. Die beste Leistungsfähigkeit bezüglich des Durchsatzes liefert in den durchgeführten Versuchen das FORP-Verfahren und wird besonders für die Übertragung von Echtzeitdaten als sehr tauglich angesehen; ODMRP-MP besitzt dafür sowohl die Unicast- als auch die Multicast-Fähigkeit.

Literatur

[BMJH+98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu und Jorjeta Jetcheva (Hrsg.). A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. Technischer Bericht, Computer Science Department Carnegie Mellon University, Pittsburgh, Oktober 1998.

[HGPC99] Xiaoyan Hong, Mario Gerla, Guangyu Pei und Ching-Chuan Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. Technischer Bericht, Computer Science Department University of California, Seattle, August 1999.

[LeGC99] Sung-Ju Lee, Mario Gerla und Ching-Chuan Chiang (Hrsg.). On-Demand Multicast Routing Protocol. Technischer Bericht, Wireless Adaptive Mobility Laboratory Computer Science Department University of California, September 1999.

[Misr00] Padmini Misra (Hrsg.). Routing Protocols for Ad Hoc Mobile Wireless Networks. Technischer Bericht, Ohio State University, Juli 2000.

[Netg97] Netgroup (Hrsg.). Routing Problems in Ad Hoc Networks and Work Done. http://netresearch.ics.uci.edu/agentos/related/routing/ad-hoc/ucla-dcnds-adhoc.pdf, Dept. of Information and Computer Science University of California, Irvine, Oktober 1997.

[PGHC99] Guangyu Pei, Mario Gerla, Xiaoyan Hong und Ching-Chuan Chiang (Hrsg.). A Wireless Hierarchical Routing Protocol with Group Mobility. Technischer Bericht, Computer Science Department University of California, September 1999.

[SuLG00a] William Su, Sung-Ju Lee und Mario Gerla (Hrsg.). Mobility Prediction and Routing in Ad Hoc Wireless Networks. Technischer Bericht, Wireless Adaptive Mobility Laboratory Computer Science Department University of California, 2000.

[SuLG00b] William Su, Sung-Ju Lee und Mario Gerla. Mobility Prediction in Wireless Networks. Technischer Bericht, Computer Science Department University of California, Los Angeles, Oktober 2000.

Watermarking: Copy Control for Multimedia

Ivonne Heinemann

Kurzfassung

Watermarking ist ein Ansatz, die Urheberrechte digitaler Daten zu schützen. Dabei werden Bits, die Informationen über den Besitzer oder eine Kopiererlaubnis enthalten, derart in die Rohdaten eingearbeitet, dass sie vom menschlichen audio-visuellen System nicht wahrgenommen und von Unberechtigten nicht verändert oder entfernt werden können. Dieser Bericht beginnt mit der Motivation der Notwendigkeit eines solchen Schutzes. Danach werden allgemeine Ziele wie Kopierschutz und Authentizitätsnachweis, allgemeine Ansätze zur Lösung dieser Ziele und Eigenschaften, die ein Wasserzeichen haben sollte, dargestellt. Die verschiedenen Methoden zur Wasserzeichen-Detektion sowie grundsätzlich abzuwehrende Angriffe werden kurz erläutert. In den Abschnitten 3 bis 5 werden dann bereits entwickelte Verfahren zur Einbettung von Wasserzeichen in Audio- und Video-Signale, einmal im Pixelbereich und einmal im Frequenzbereich, vorgestellt.

1 Motivierende Einleitung

Ein großer Vorteil digitaler Daten ist die Möglichkeit, unbegrenzt fortlaufende Kopien zu erstellen und zu verbreiten, ohne dabei Qualität einzubüßen, wie es bei analogen Daten der Fall wäre (man stelle sich nur einmal die Kopie einer Kopie einer Beatles-Kassette vor ...). Die Vervielfältigung digitaler Daten und damit auch das Erstellen sogenannter Raubkopien ist heutzutage besonders einfach und die dazu benötigte Hardware sehr billig, so dass dies auch für einen Normalverbraucher möglich ist. Durch das enorme Wachstum des Internets wird zudem die Verbreitung illegaler Kopien einfach, billig und vor allen Dingen schwer zu ahnden. Prinzipiell ist es unmöglich, unberechtigtes Kopieren und die illegale Verbreitung digitaler Daten zu verhindern. Allerdings kann man die Daten mit einem Kopie-Eigentümer-Profil (Fingerprinting) versehen. Treten illegale Kopien auf, kann man dem Profil dann entnehmen, wer seine eigene Kopie verbreitet hat, da dieses Profil bei jedem Kopiervorgang mitkopiert wird. In anderen Fällen möchte man die Authentizität eines Dokumentes, Bildes oder Videos feststellen. Dazu hängt man den Daten eine digitale Signatur an, die mit einem persönlichen Schlüssel des Authors erstellt wird. Man kann eine solche Signatur mit einer handschriftlichen Unterschrift vergleichen, die zusätzlich sicherstellt, dass die Daten nach der Signierung nicht mehr verändert wurden. Möchte man die Urheberrechte eines Dokumentes schützen, so bettet man in die Originaldaten ein Wasserzeichen ein.

2 Grundlagen

Im folgenden werden Ansätze wie digitale Wasserzeichen, digitale Signaturen und Fingerprinting sowie Anwendungen dieser Ansätze vorgestellt. Es folgt die Beschreibung allgemeiner Ziele. Dies sind die Feststellung der Authentizität, die Verteidigung der Urheberrechte, die kontrollierte Kopiererlaubnis, die Identifikation des Kopie-Besitzers und die versteckte

Kommunikation. In Abschnitt 2.3 werden Eigenschaften bezüglich der Wahrnehmbarkeit, der Sicherheit, der Größe und weiteren Aspekten aufgeführt, die ein digitales Wasserzeichen haben sollte. Am Ende dieses Kapitels werden dann die Grundtypen der Detektion und des Angriffs auf Wasserzeichen sowie Gegenmaßnahmen erläutert.

2.1 Ansätze

Ansätze, um digitale Daten vor unberechtigtem Kopieren oder Abändern zu schützen bzw. bei illegaler Verbreitung den Täter aufzuspüren, sind:

- Digitale Wasserzeichen:

  Dabei werden direkt in die Bild-, Video- oder Dokumentdaten bzw. den entsprechenden komprimierten Daten durch Modulation der Pixelwerte oder der Frequenzkoeffizienten bei transformierten Daten zusätzliche Informationsbits eingearbeitet. Nur mit Wissen über die Parameter, die bei der Einbettung des Wasserzeichens verwendet wurden, können diese Informationsbits später wieder extrahiert werden. Die Informationen, die in digitale Daten eingebettet werden, sollen meist den Besitzer des Originals identifizieren, die Echtheit der Daten garantieren (etwa bei Bildern oder Videos, die vor Gericht als Beweismittel dienen sollen) oder z. B. dem DVD-Player signalisieren, ob das Video kopiert werden darf oder nicht. Enthält die DVD ein „nicht kopieren“-Wasserzeichen, so verweigert der DVD-Player jeglichen Kopiervorgang.

- Digitale Signaturen:

  Für das Dokument wird zusammen mit einem persönlichen Schlüssel eine Art „Quersumme“ berechnet und angehängt. Werden die Daten manipuliert, stimmt die Signatur nicht mehr und die Änderung wird bemerkt. Ein Unberechtigter kann ohne den passenden (und selbstverständlich geheimen) persönlichen Schlüssel des Dokumentbesitzers nicht die richtige Quersumme für die manipulierten Daten berechnen und statt der alten Signatur anhängen. Beispiel für eine Anwendung ist die El-Gamal-Signatur.

- Fingerprinting:

  Beim Fingerprinting bettet man in die digitalen Daten eine Bitsequenz ein, die den Eigentümer der Kopie identifiziert. Vervielfältigt und verbreitet dieser seine Kopie, kann man ihn durch jede dieser illegalen Kopien identifizieren, da die eingebetteten Daten mitkopiert wurden (Tracing).

2.2 Anwendungen

Anwendungen der oben genannten Ansätze sind die Feststellung der Authentizität der Daten, die Verteidigung der Urheberrechte des Authors, die Möglichkeit der kontrollierten Kopiererlaubnis, die Identifikation des Kopie-Eigentümers, der seine Kopie illegal vervielfältigt und verbreitet hat, und die versteckte Kommunikation (Steganographie). Diese Anwendungen sollen im Folgenden vorgestellt werden.

Die *Feststellung der Authentizität der Daten* ist z. B. wichtig, wenn digitale Daten als Beweismittel vor Gericht verwendet werden sollen. Da man digitale Daten einfach und nicht nachweisbar verändern kann, können sie ohne Authentizitätsnachweis nicht als Beweismittel vor Gericht anerkannt werden.

Für die *Verteidigung der Urheberrechte* markiert der Author sein Original mit einem Wasserzeichen, das ihn als rechtmäßigen Besitzer identifiziert, bevor er es für andere zugänglich

macht. Behauptet dann ein anderer, er sei der Besitzer, kann der Author seinen Anspruch leicht durch Nachweis des Wasserzeichens behaupten. Ein Problem kann entstehen, wenn der falsche Besitzer einfach ein zweites Wasserzeichen einfügt, das ihn als rechtmäßigen Besitzer bestätigt. Man muss also zusätzlich auch die zeitliche Abfolge der Wasserzeichen rekonstruieren können.

Eine weitere Anwendung ist die *Bereitstellung der Möglichkeit zur kontrollierten Kopiererlaubnis* (der Besitzer erlaubt kein/einmaliges/mehrfaches Kopieren). Am häufigsten wird der Author dem Kopiebesitzer keine einzige Kopie gestatten. Es kann aber auch vorkommen, dass z. B. eine oder gar mehrere Kopien erlaubt werden sollen. Handelt es sich um die Erlaubnis, nur eine Kopie anzufertigen, so muss der Detektor auch ein Gerät zur Einbettung eines Wasserzeichens sein, denn sowohl die Kopiervorlage als auch die eine erlaubte Kopie sollen hinterher ein „keine Kopie gestattet"-Wasserzeichen anstelle des „einmal kopieren gestattet"-Wasserzeichens enthalten.

Die *Identifikation des Kopie-Eigentümers* ist ebenfalls eine wichtige Anwendung (für Tracing, durch Fingerprinting). Beispiel: Ein Video-on-demand-Server markiert jeden Videostrom mit Daten, die den Empfänger dieses Videos identifizieren. Verbreitet der Empfänger illegale Kopien des Videos, enthalten diese immer noch jene Daten und der Täter kann idetifiziert und zur Verantwortung gezogen werden.

Als letzte Anwendung sei hier die *Versteckte Kommunikation (Steganographie)* vorgestellt. Mit Steganographie bezeichnet man das Verbergen der eigentlichen Nachricht in einem harmlosen Trägermedium, so dass nicht einmal die Kommunikation selbst von Unberechtigten bemerkt wird. Steganographie ist schon viel älter als digitale Daten. So wurde vor rund 2500 Jahren versucht, unauffällig Nachrichten auszutauschen: Scheinbar ungenutzte Wachs-Schreibtafeln beispielsweise trugen die Nachricht auf dem Holz unter der Wachsschicht. Auch heutzutage ist Steganographie noch wichtig, da oft schon allein das Vorhandensein von Kommunikation selbst wichtige Rückschlüsse auf den Inhalt zulässt, z. B. bei Unternehmen, die fusionieren möchten. Dann reicht es also nicht aus, nur den Inhalt der Kommunikation zu verschlüsseln, sondern man muss die verschlüsselten Daten auch noch heimlich und unbemerkt übertragen. Näheres in [Wesf01].

2.3 Eigenschaften eines Wasserzeichens

Im weiteren Verlauf dieses Beitrages wird als ein Ansatz zur Verwirklichung kontrollierter Kopiererlaubnis das digitale Wasserzeichen betrachtet.
In diesem Abschnitt werden Eigenschaften dargestellt, die Wasserzeichen besitzen sollten, um ihre Ziele zu erfüllen. Diese Eigenschaften beziehen sich auf die Wahrnehmbarkeit, die Sicherheit, die Möglichkeit zur späteren Entfernung, die Größe und weitere Aspekte.

Bezüglich der *Wahrnehmbarkeit (imperceptibility)* soll ein eingebettetes Wasserzeichen selbstverständlich mit blossem Auge/Ohr nicht hörbar bzw. nicht sichtbar oder wenigstens kaum wahrnehmbar sein und die Qualität der Daten nicht verschlechtern. Schließlich soll es für den Nutzer keinen Unterschied machen, ob seine CD oder seine DVD eine Markierung enthält oder nicht.

Die *Sicherheit* ist ein wichtiger Aspekt. Das Verfahren zur Einbettung sollte so entworfen werden, dass unauthorisiertes Entfernen oder Abändern des Wasserzeichens ohne Kenntnis der exakten Einbettungsparameter unmöglich ist. Dies soll selbst dann noch gewährleistet sein, wenn das grundsätzliche Verfahren zur Einbettung bekannt ist.

Für die *Möglichkeit zur späteren Entfernung (removeability) oder Modifikation* sollte der Besitzer dahingegen auch später noch in der Lage sein, das Wasserzeichen zu modifizieren oder zu

entfernen. Wer sein Video anfangs mit einem „keine Kopie gestattet“-Wasserzeichen markiert hat, entscheidet vielleicht später, dass er doch lieber ein „eine Kopie gestattet“-Wasserzeichen einbetten möchte.

Die *Größe (data payload)* der Orginaldatei sollte nicht zunehmen. Nicht nur deshalb, weil dann z. B. mehr Daten gespeichert/übertragen werden müssen, sondern vor allen Dingen, damit auch schon die Anwesenheit eines Wasserzeichen unentdeckt bleibt. Würde das Wasserzeichen die Größe der digitalen Datei beeinträchtigen, so könnte man die Bitzahl N der eingebetteten Information herausfinden, wodurch dann nur noch genau $2^N + 1$ mögliche Markierungen zu untersuchen wären (keine Markierung sei hier auch als Möglichkeit in Betracht gezogen).

Ein weiterer Aspekt ist der *Berechnungsaufwand (computational cost)*. Je nach Anwendung ist es wichtig, dass die Einbettung oder das Detektieren des Wasserzeichens komplex ist oder in Realzeit durchgeführt werden kann. Bei einer DVD wird man darauf achten, dass das Einbetten der Kopierrechte sehr aufwendig ist und teure Hardware erfordert, da ein Einbettungsgerät meist auch zur Entfernung der Markierung genutzt werden kann. Auf der anderen Seite soll der Detektor das Abspielen der DVD nicht verzögern und billig sein, damit der DVD-Player, in den dieser integriert ist, nicht zu teuer wird und marktfähig bleibt.

Ein Verfahren zur Einbettung von digitalen Wasserzeichen sollte auch die *Bearbeitung komprimierter Daten* zulassen. Für Broadcast-Anwendungen ist es z. B. nicht praktikabel, die Daten erst zu dekomprimieren, dann zu markieren und sie danach wieder zu komprimieren.

Wasserzeichen-Detektion sollte *standardisierbar* sein. In manchen Anwendungen benötigt man unbedingt Standards, die den weltweiten Gebrauch ermöglichen. Ein solcher Bereich ist die DVD-Industrie. Jeder DVD-Player sollte Wasserzeichen nach demselben Schema detektieren, ganz gleich ob er in China oder in Amerika gebaut wurde. Allerdings bedeutet das nicht, dass bei standardisierter Detektion auch die Einbettungsmethode standardisiert sein muss.

Bezüglich der *Auffindbarkeit* sollte das Wasserzeichen zwei gegensätzliche Eigenschaften haben: Zum einen sollte es für Kontrollgeräte leicht auffindbar sein, zum anderen aber sollte es für Unberechtigte schwer sein, die Anwesenheit oder gar die Lokalität der eingebetteten Information ausfindig zu machen.

Die *„false positive rate“* eines Detektiersystems bezeichnet die Wahrscheinlichkeit, dass unmarkierte Daten als markiert erkannt werden. Für manche Anwendungen kann dies einen schwerwiegenden Fehler darstellen. Beispielsweise verweigern DVD-Player das Abspielen einer DVD, wenn das Video ein „keine Kopie gestattet“-Wasserzeichen enthält, aber nicht auf einer „Werks-DVD“ gespeichert ist, da es sich dann höchstwahrscheinlich um eine Raubkopie handelt. Ein privates Hochzeitsvideo hingegen, das gewöhnlich auch nicht auf einer „Werks-DVD“ gespeichert ist, aber kein Wasserzeichen enthält, sollte er ohne Probleme abspielen. Bei einem Detektiersystem mit hoher false positive rate kann es dann aber passieren, dass der DVD-Player ständig das Abspielen des Videos verweigert, weil er fälschlicherweise ein „keine Kopie gestattet“-Wasserzeichen auf einer „Nicht-Werks-DVD“ detektiert.

Ein besonders wichtiger Aspekt für Wasserzeichen ist die *Robustheit*. Ein Wassereichen sollte sowohl gegen gewöhnliche Signalverzerrungen als auch gegen digital-zu-analog und analog-zu-digital Konvertierungen, erneutes Abtasten (resampling) und verlustbehaftete Kompression (z. B. JPEG) robust sein. Für Wasserzeichen in Bildern und Videodaten ist es zusätzlich wichtig, geometrische Verzerrungen wie Verschiebung, Rotation, Skalierung und Ausschnittsbildung zu überleben. Da bei jeder Datenübertragung ein Rauschen auftritt, sollte ein Wasserzeichen auch nach Addition eines solchen noch vorhanden und detektierbar sein. Man beachte, dass Robustheit zwei Aspekte betrifft: Zum einen muss das Wasserzeichen nach einer Verzerrung noch in den Daten enthalten sein und zum anderen muss der Detektor es noch erkennen können. Nach einer geometrischen Verzerrung wie der Skalierung beispielsweise ist das Was-

serzeichen immer noch in den Daten enthalten, aber eventuell schon nicht mehr detektierbar, ohne dass die Verzerrung vorher rückgängig gemacht wurde.

Als letztes sei die Eigenschaft bezüglich der *Zerbrechlichkeit (fragility)* beschrieben. Unter diesem Aspekt soll ein Wasserzeichen alles andere als robust sein: Bei jeglichen unauthorisierten Versuchen, das Wasserzeichen zu verändern oder zu entfernen, soll das Bild/Video etc. unbrauchbar werden, d.h. auf dem Video oder dem Bild soll dann nichts mehr zu erkennen sein.

2.4 Nachweis/Detektion

Es gibt zwei Grundtypen der Wasserzeichen-Detektion: Einmal benötigt der Detektor das Original, das andere Mal benötigt er nur die markierte Version. Beide Fälle seien kurz erläutert:

- Der Detektor benötigt ein Orginal.

  Dabei wird zur Ermittlung des Wasserzeichens einfach das Original von der markierten Version subtrahiert. Es ist offensichtlich, dass dies für große zu markierende Datenmengen wie z. B. für ein Video zu einem sehr großen Overhead führt. Ein Angreifer müsste sich nicht einmal die Mühe machen, das Wasserzeichen aus der markierten Version zu entfernen, sondern könnte sich leicht Zugang zu dem unmarkierten Original verschaffen, da dieses ja jedem Detektor zur Verfügung stehen muss.

- Der Detektor benötigt nur die markierte Version.

  Hier wird meist die Autokorrelation der markierten Daten berechnet, wodurch das Wasserzeichen extrahiert werden kann bzw. seine Anwesenheit durch einen hohen Autokorrelationswert angezeigt wird. Die Korrelation ist ein spezielles Integral zweier Funktionen mit der Eigenschaft, dass sich für Funktionen, die sehr ähnliche Funktionsverläufe haben, ein hoher Wert ergibt. Bei der Autokorrelation wird die Funktion mit sich selbst korreliert. Daher gibt der Autokorrelationswert an, wieviel die Funktion „mit sich selbst zu tun hat“.

2.5 Angriffe

Je nach Ausrüstung und Ziel des Angreifers gibt es verschiedene Arten von Angriffen auf Wasserzeichen. Je nachdem, ob dem Angreifer ein Detektor zur Verfügung steht oder nicht und ob er das Wasserzeichen entfernen, verändern oder ein weiteres einfügen möchte, ergeben sich nachfolgend beschriebene Szenarien.

Man betrachte den Fall, dass dem Angreifer ein Detektor zur Verfügung steht. Dient das Wasserzeichen dem Kopierschutz, so kann man davon ausgehen, dass dem Angreifer ein Detektor zur Verfügung steht. Bei DVDs z. B. ist er in den DVD-Player integriert, der bei entsprechendem Wasserzeichen das Abspielen der DVD verweigert. Der Angriff zur Entfernung des Wasserzeichens gestaltet sich dann derart, dass die markierten Daten solange Stückchen für Stückchen (für den Menschen nicht wahrnehmbar) moduliert werden, bis der Detektor kein Wasserzeichen mehr findet und der Inhalt der DVD immer noch gute Qualität hat.

Nun betrachte man den anderen Fall: dem Angreifer steht kein Detektor zur Verfügung. Hier braucht der Angreifer für einen erfolgreichen Angriff Daten über das Verfahren zur Einbettung des Wasserzeichens sowie der verwendeten Parameter, z. B. das verwendete Rauschen, das mit dem Wasserzeichen moduliert wurde.

Falls der Angreifer das Wasserzeichen entfernen, verändern oder weitere Wasserzeichen einfügen möchte, könnte er das Signal verfälschen, so dass das Wasserzeichen nicht mehr detektierbar ist. Um es zu verändern, muss der Angreifer Kenntnisse über das Verfahren der Einbettung haben, denn er muss das alte Wasserzeichen entfernen und dann sein Wasserzeichen einfügen. Verfolgt der Angreifer beispielsweise das Ziel, dem Author die Urheberrechte streitig zu machen, so kann er unter Kenntnis des Verfahrens zur Einbettung und regulärer Parameter ein zweites Wasserzeichen einfügen, das ihn als Author identifiziert.

Zur Verfälschung des markierten Signals mit dem Ziel der Nichtdetektierbarkeit des Wasserzeichens, stehen dem Angreifer folgende Methoden zu Verfügung:

- Datenkompression.

  Oft überleben Wasserzeichen eine Kompression nicht, da die Informationsbits wegen der Nichtwahrnehmbarkeit meist in Bereiche der Originaldaten eingebettet werden, die bei der Kompression aus demselben Grund „wegkomprimiert“ werden.

- Tiefpassfilterung.

  Andere Wasserzeichen werden in die hohen Frequenzen eingebettet und überleben daher eine Tiefpassfilterung nicht.

- Farbreduzierungen.

  Falls das Wasserzeichen durch nicht wahrnehmbare Modulation der Färbung kodiert wurde, kann man durch Farbreduktion erreichen, dass der Detektor kein Wasserzeichen mehr erkennt.

- Ausschnittsbildung.

  Wasserzeichen werden oft über mehrere Pixel verteilt, um bei der Modulation einzelner Pixel noch mit höherer Wahrscheinlichkeit zu überleben. Gibt man dem Detektor dann immer nur einen kleinen Ausschnitt, so enthält dieser nur noch wenige der Pixel, in denen das Wasserzeichen-Bit eingebettet ist, und er wird das Wasserzeichen nicht mehr detektieren.

- Skalierung.

  Bei der Skalierung bleibt das Wasserzeichen zwar erhalten, aber die relative Energie nimmt ab und damit auch die Detektionsleistung.

- Erschließung des unmarkierten Originals durch Kombination von mehreren, verschieden markierten Versionen (collusion attack).

  Schließen sich mehrere Besitzer verschieden markierter Versionen zusammen oder stehen einem Angreifer mehrere markierte Versionen zur Verfügung, muss er nur den „Durchschnitt“ von ihnen berechnen. Dieser enthält dann das Original und den Durchschnitt aller Wasserzeichen, der selbst kein Wasserzeichen mehr, sondern nur noch unerhebliches Rauschen darstellen dürfte.

2.6 Gegenmaßnahmen

Um den vielfältigen Angriffsmöglichkeiten entgegenzuwirken, wird man die Wasserzeichen-Bits mehrfach (redundant) und in Raum und Zeit verteilt (spread) einbetten und sie in „wichtige Informationsteile“ einbauen (gegen verlustbehaftete Kompression). Gegen den Angriff, ein weiteres einzufügen, wird man Wasserzeichen kombiniert mit einem Zeitstempel und gegebenenfalls einer Verwaltungszentrale (trust center) entwerfen müssen.

3 Beispiel: Wasserzeichen in Audiosignalen

Im folgenden wird das Verfahren aus der Quelle [BoTH96] vorgestellt, das die Urheberrechte von Audiosignalen durch Einbettung von Wasserzeichen schützt. Das Besondere bei diesem Ansatz ist, dass Pseudo-Rauschen (PN-sequence) durch Filterung den charakteristischen Frequenzen des menschlichen Hörsystems (human auditary system, HAS) angepasst und dann in der Zeitdomäne gewichtet wird, bevor es mit den Informationsbits versehen und als Wasserzeichen in ein Audiosignal eingebettet wird.

3.1 Entwurf des Wasserzeichens

Ein Ansatz zur Erzeugung von Wasserzeichen ist es, die N größten Frequenzkomponenten der Audiodaten mit Gauss'schem Rauschen zu verändern. Dies verändert jedoch nur einen Teil der vorkommenden Frequenzen und lässt die Eigenschaften des HAS außer acht. Der Ansatz in [BoTH96] verwendet ein Modell, welches das HAS maskiert. Zusätzlich wird hier die Energie des Wasserzeichen-Signals im Rahmen der Nichtwahrnehmbarkeit erhöht. Dadurch wird die Leistung des Detektors, die mit der Energie des zu detektierenden Signals zunimmt, verbessert.

3.1.1 Erster Schritt zur Audiomarkierung

Man beginnt mit einer PN-Sequenz. Wie bei zufälligen Binärsequenzen treten bei PN-Sequenzen die Zahlen 0 und 1 mit gleicher Wahrscheinlichkeit auf. Die Autokorrelationsfunktion ACF einer solchen Sequenz hat die Periode N. Durch diese Periodizität der ACF ist die PN-Sequenz selbsttaktend, was man für die spätere Detektion gut ausnutzen kann. Das ist wichtig, falls die markierten Daten Angriffen wie Abschneiden (cropping) oder Neuabtastung (resampling) unterlagen. Zuerst wird die Maskierschwelle des Signals mit Hilfe des MPEG Psychoacoustic Models 1, [ISO/93], berechnet. Diese Schwelle wird jeweils für aufeinanderfolgende Audiosegmente von 512 Abtastwerten (samples) ermittelt. Jedes Segment wird

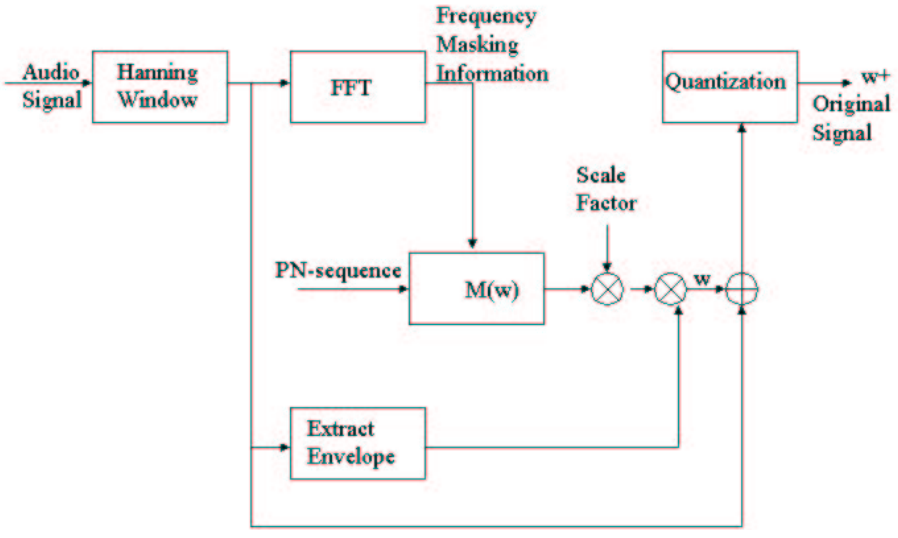


Abbildung 1: Erste Stufe der Audiomarkierung.

mit einem Hanningfenster (hanning window) gewichtet, aufeinanderfolgende Blöcke überlappen zu 50%. Die PN-Sequenz wird mit einem passenden Filter $M(w)$ gefiltert, so dass sie unterhalb des Schwellwertes liegt. Da sich das Spektrum eines Audiosignals mit der Zeit ändert, wären selbst Wasserzeichen, die mit derselben PN-Sequenz erzeugt wurden, im allgemeinen verschieden. Es ist aber empfehlenswert, für jeden Block eine andere PN-Sequenz zu verwenden, um unauthorisierten Benutzern eine statistische Detektion zu erschweren. Die Berechnungen für die Maskierung im Frequenzbereich basieren auf der Fourierzerlegung. Eine FFT (Fast Fouriertransformation) mit fester Länge bietet leider keine gute zeitliche Lokalisation. Die modulierten Frequenzkomponenten können in der Zeit weit verteilt sein, so dass die Modulation hörbar wird (z. B. durch Vor-Echos). Um dem entgegenzuwirken, wird hier das Wasserzeichen zusätzlich im Zeitbereich mit der relativen Energie des Audiosignals gewichtet. Diese Gewichtung im Zeitbereich schwächt die Energie des Wasserzeichens ab. Die so erzeugten Wasserzeichen haben typischerweise kleinere Amplituden als die Größe eines Quantisierungsschrittes, was zum Verlust des Wasserzeichens während der Quantisierung führen und die Detektionsleistung aus oben genanntem Grund wieder abnehmen würde. Um dies zu umgehen, wird das Wasserzeichen vor der Gewichtung im Zeitbereich um 40 dB (scale factor) verstärkt. Experimente haben bestätigt, dass diese Verstärkung die Wahrnehmbarkeit des Wasserzeichens wegen des Dämpfungseffekts der Gewichtung im Zeitbereich nicht beeinflusst. Bild 1 veranschaulicht das Verfahren.

3.1.2 Das gesamte Verfahren

Wie bereits erwähnt, muss ein Verfahren für die Einbettung von Wasserzeichen gegen Kodierungsoperationen robust sein. Algorithmen zur Audiokodierung mit niedrigen Bitraten lassen meist nur die niederfrequenten Informationsanteile übrig. Daher muss die meiste Energie des Wasserzeichens in den niedrigen Frequenzen liegen. Ein niederfrequentes Wasserzeichen erzeugt man am besten als Differenz von einem mit niedriger Bitrate kodierten/dekodierten markierten Signal und dem mit derselben Bitrate kodierten/dekodierten Original. Das Wasserzeichen für diese erste Markierung wird mit der ersten Stufe der Audiomarkierung (watermarking generator first stage) wie in Abb. 1 erzeugt. Man benutzt dabei als niedrige Bitrate die kleinste bekannte Bitrate, die für Signale mit der Abtastrate des Originals verwendet werden kann. Das so erzeugte niederfrequente Wasserzeichen wird in Abb. 2 mit w_{br} bezeichnet, wobei br für die Bitrate steht. Für bessere Detektionsleistungen bei höheren Bitraten muss die Information des Wasserzeichens auch noch in die höheren Frequenzen eingebettet werden. Dafür wird ein Wasserzeichen w_{err} für den Kodierungsfehler erzeugt. Der Kodierungsfehler ist die Differenz zwischen dem Original-Audiosignal und seiner mit niedriger Bitrate kodierten Version. Mit diesem Kodierungsfehler erzeugt man wie in Abb. 1 (watermark generator first stage) dargestellt das Wasserzeichen w_{err}. Das endgültige Wasserzeichen ist dann die Summe von dem niederfrequenten Wasserzeichen w_{br} und dem Kodierungsfehler-Wasserzeichen w_{err}. Abb. 2 zeigt in der oberen Hälfte die Konstruktion von w_{err} und in der unteren Hälfte die Konstruktion des Wasserzeichens w_{br}.

3.1.3 Hörbarkeit des Wasserzeichens

Das vorgestellte Verfahren [BoTH96] wurde an Musikstücken getestet, die für Vor-Echos oder längere Abschnitte der Stille bekannt sind. Bei den informellen Test hörten sich acht Personen aus unterschiedlichen Bereichen jeweils das Original und die markierte Version an. Es wurden keine hörbare Verzerrungen und keine Vor-Echos bemerkt. Auch die Abschnitte der Stille verblieben still.

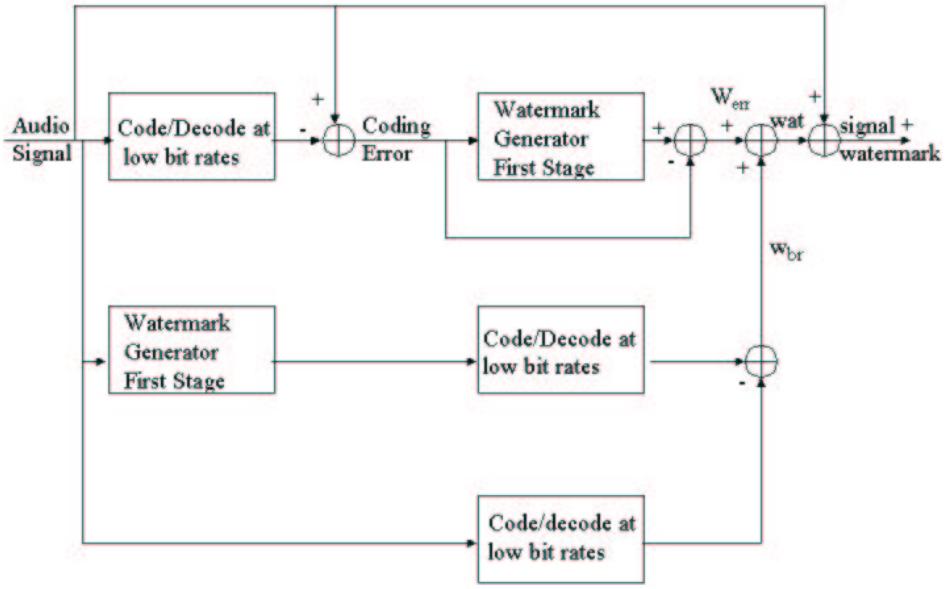


Abbildung 2: Gesamtverfahren zur Wasserzeichen-Einbettung in Audiosignale.

3.2 Detektion des Wasserzeichens

Es wird angenommen, dass der Author Zugang zum Original und zu der PN-Sequenz hat, die er zur Markierung verwendete. Um zu entscheiden, ob ein gegebenes Audiosignal $r(k)$ markiert ist oder nicht, zieht der Author die kodierte Version s_{br} des Originals $s(k)$ ab. s_{br} wird erzeugt, indem $s(k)$ mit der Bitrate, die vermutlicherweise für $r(k)$ verwendet wurde, mit der MPEG-Kodierungs-Prozedur kodiert wird. Man beachte, dass $r(k)$ bereits mit einem anderen Kodierungsalgorithmus kodiert sein könnte. Der Unterschied zwischen der Kodierung von $s(k)$ mit der geschätzten Bitrate und der tatsächlichen Kodierung, die das Audiosignal $r(k)$ erfahren hat, tritt als zusätzliches Rauschen auf. Als nächstes prüft der Author folgendes Hypothesensystem:

- $H_0 : x(k) = r(k) - s_{br} = n(k)$
- $H_1 : x(k) = r(k) - s_{br} = w'(k) + n(k)$,

wobei $n(k)$ für zusätzliches Rauschen steht, das sowohl durch die Verwendung unterschiedlicher Kodierungsalgorithmen als auch durch Signalmanipulation und Übertragungsrauschen entstanden sein kann. $w'(k)$ bezeichnet das modifizierte Wasserzeichen. Da $n(k)$ nicht genau bekannt ist, löst man das Hypothesenproblem durch Korrelation von $x(k)$ mit $w'(k)$ und einem Vergleich mit einem Schwellwert. Bild 3 zeigt die Korrelation von einem Wasserzeichen mit sich selbst, mit einem gestörten (jammed) Wasserzeichen und mit Rauschen. Das Bild zeigt, dass eine zuverlässige Detektion realisierbar ist.

3.2.1 Erzeugung des zusätzlichen Rauschens

Eine Approximation der schlimmsten zu erwartenden zusätzlichen Verzerrung (distortion) des Wasserzeichens ist ein Rauschen, das dieselben spektralen Eigenschaften hat wie die Maskierungsschwelle. Diese Art von Verzerrung ist ein gutes worst case Modell für Verzerrungen

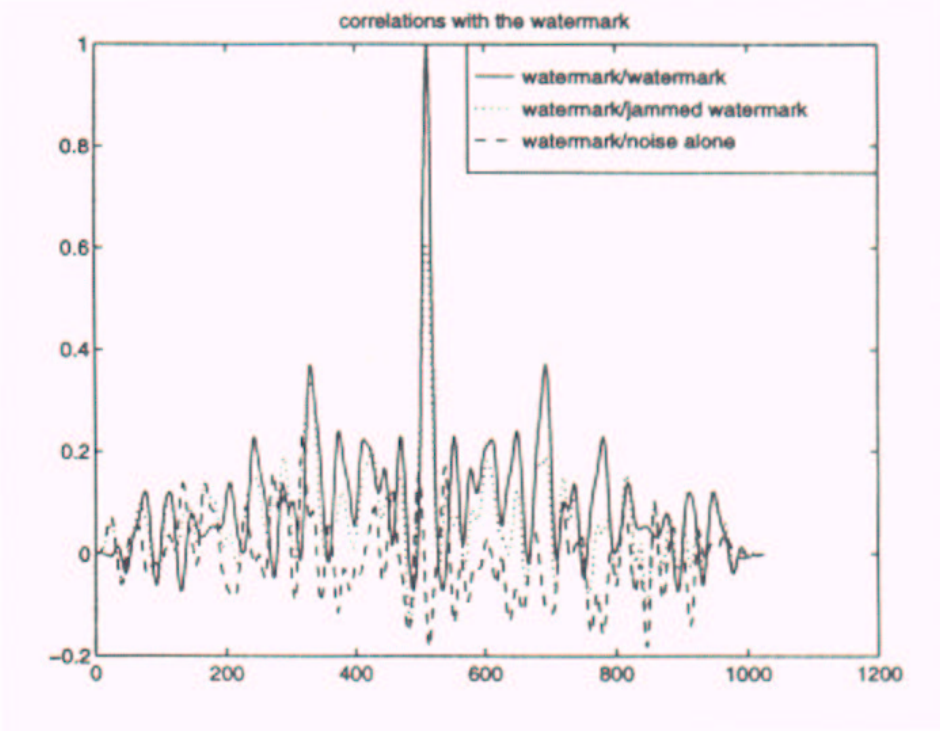


Abbildung 3: Korrelation eines Wasserzeichens mit sich selbst, dem gestörten Wasserzeichen und mit Rauschen.

durch beabsichtigtes Stören (jamming) mit unhörbaren Signalen und Unstimmigkeiten zwischen dem geschätzten und dem tatsächlichen Kodierungsalgorithmus. Das Rauschen, das bei den Tests verwendet wurde, wurde auf dieselbe Weise wie das Wasserzeichen erzeugt. Die Maskierungsschwelle wurde erst um +40dB versetzt und dann mit der diskreten Fouriertransformierten eines weißen Gauss'schem Rauschen multipliziert. Das resultierende Rauschen wurde dann in der Zeit mit der relativen Energie des Signals gewichtet. Nach einer Quantisierung wurde das erzeugte Rauschen mit der Maskierungsschwelle gefiltert und zurückquantisiert. Das Ergebnis ist nahezu unhörbar und eine gute Approximation des größtmöglichen Rauschens, das noch unterhalb der Maskierungsschwelle liegt.

3.2.2 Detektionsresultate und Robustheit

Um die Robustheit dieses Ansatzes zu testen, wurde Rauschen zu verschiedenen markierten und unmarkierten Audiosignalen addiert und das Ergebnis mit einer Implementation des ISO/MPEG-1 Audio Layer III Kodierers kodiert. Die Wasserzeichen wurden mit verschiedenen PN-Sequenzen erzeugt und waren so gut wie nicht hörbar. Es zeigte sich, dass die Wahrscheinlichkeit der Detektion fast 1 und die des falschen Alarms fast 0 ist und das vorgestellte Verfahren somit gute Resultate vorweisen kann (siehe [BoTH96]).

4 Beispiel: Digitale Wasserzeichen im Pixelbereich eines uncodierten Videos

Für die Einbettung eines Wasserzeichens in ein Video gelten andere Bedingungen als für die Einbettung in Audiosignale. Die Bedingungen sind vielmehr denen der Einbettung von Wasserzeichen in Bilder ähnlich, da das Auge schlechtere Rekomstruktionsfähigkeiten als das Ohr

hat. Aus diesem Grund sollte das Wasserzeichen für die Einbettung in Bilder oder Videos zuerst verteilt (spread) werden, da es sonst schnell sichtbar wird.
Im folgenden wird die Einbettung und die Wiedergewinnung eines Wasserzeichens in digitale Videodaten kurz vorgestellt. Das Verfahren [HaGi97] verteilt die Informationsbits des Wasserzeichens erst auf mehrere Videopixel und moduliert es vor der Einbettung mit Pseudorauschen.

4.1 Einbettung des Wasserzeichens

Abbildung 4 zeigt das Grundprinzip der Einbettung des Wasserzeichens im Pixelbereich (Quelle: [HaGi97] Kapitel 4). Die Informationsbits $a_i \epsilon \{-1, 1\}$, die eingebettet werden sollen, werden zuerst mit einem großen Faktor *cr* verteilt (spread). Dieser Faktor heißt in Anlehnung an die spektral verteilte Kommunikation (spread spectrum communication) „chip rate“. Der Zweck dieser Ausbreitung ist es, einzelne Informationsbits auf mehrere, genau gesagt auf *cr* viele Videopixel zu verteilen und Redundanz hinzuzufügen. Die verteilten Informationsbits

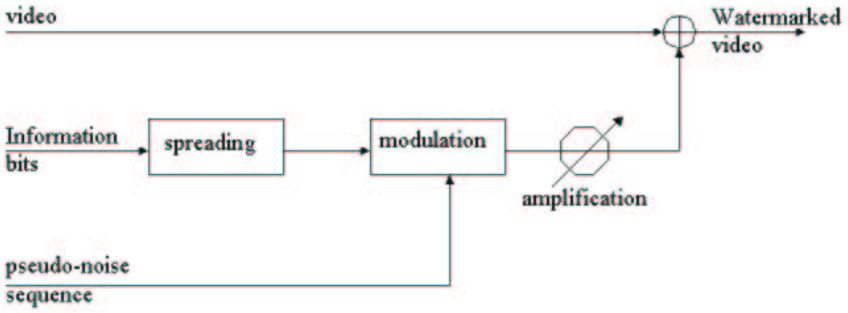


Abbildung 4: Einbettung eines Wasserzeichens im Pixelbereich.

werden dann mit Pseudorauschen moduliert. Das resultierende Wasserzeichen-Signal wird eventuell noch verstärkt (amplification), bevor es schließlich auf die Pixel der Videosequenz aufaddiert wird. Der Verstärkungsfaktor kann den lokalen Eigenschaften des Bildes angepasst werden und dazu dienen, die räumlichen und zeitlichen Maskierungseffekte des menschlichen visuellen Systems (HVS) auszunutzen.

4.2 Wiedergewinnung des Wasserzeichens

Die Wiedergewinnung der versteckten Information im Detektor ist einfach zu realisieren durch eine Korrelation des markierten Videos mit dem bei der Einbettung verwendeten Pseudorauschen (siehe Abb. 5). Die Breite des Korrelationsfensters ist dann gerade die chip rate. Liefert die Korrelation einen positiven Peak, dann ist das Wasserzeichen-Bit eine +1, ist der Peak negativ, so handelt es sich um eine −1. Das bedeutet, dass der Empfänger selbst dann, wenn er das Basisschema der Einbettung kennt, nur mit Kenntnis des verwendeten Pseudorauschens

die Information aus dem markierten Video extrahieren kann. Man beachte ausserdem, dass der Detektor nur die markierte Version und nicht das Original benötigt.

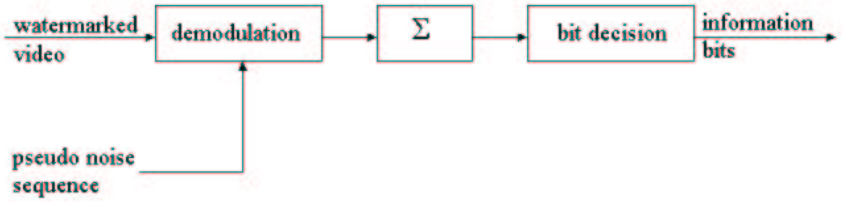


Abbildung 5: Schema zur Wiedergewinnung der Informationsbits des Wasserzeichens.

5 Beispiel: Digitale Wasserzeichen im Bitstrom-Bereich eines MPEG-2 kodierten Videos

In diesem Abschnitt wird ein Verfahren [HaGi97] vorgestellt, das Wasserzeichen in den Bitstrombereich eines komprimierten Videos einbettet. Das Grundprinzip der Wasserzeichen-Einbettung sowie Maßnahmen zur Drift-Kompensation dieses Verfahrens werden kurz erläutert.

Eine offensichtliche Idee zur Einbettung eines Wasserzeichens in ein MPEG-2-komprimiertes Video wäre es, die Wasserzeichen-Bits den Randinformationen des MPEG-Bitstroms hinzuzufügen. Aber ein Wasserzeichen muss auch nach der Dekodierung noch im Video enthalten sein und die Bitrate des komprimierten Videos soll durch die Markierung nicht vergrößert werden. Die oben erwähnte Idee kann diese Anforderungen aber nicht erfüllen. Also wird man wieder dazu übergehen, das Wasserzeichen in das Videosignal selbst einzubetten. Der in [HaGi97], Kapitel 5, vorgestellte Ansatz geht aber nicht den Umweg über die Dekodierung (kodierten Bitstrom dekodieren, markieren und das Ergebnis wieder kodieren), sondern fügt das Wasserzeichen direkt in den kodierten Bitstrom ein.

5.1 Grundprinzip

Das Prinzip der MPEG-2 Videokompression ist bewegungskompensierte Hybridkodierung. Dabei werden aus vorangegangenen Rahmen Bewegungsvorhersagen getroffen und dafür verwendet, den aktuellen Rahmen zu rekonstruieren. Zur Einbettung eines Wasserzeichens in ein auf diese Weise komprimiertes Video werden die I-Frames zunächst in Blöcke der Größe 8×8 Pixel aufgeteilt. Danach werden die Blöcke mit der Diskreten Cosinustransformation (DCT)

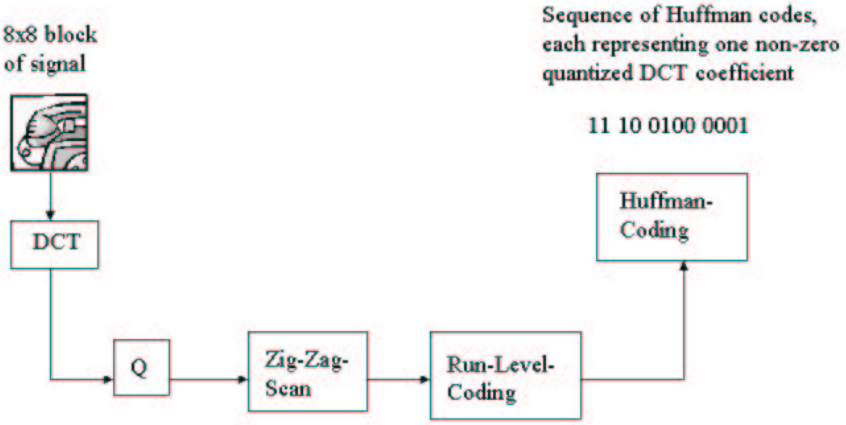


Abbildung 6: Kodierung eines 8×8-Pixel Blocks

komprimiert, quantisiert (Q), zick-zack-gescannt, run-level und Entropie-kodiert (Huffman Coding). Abb. 6 veranschaulicht das Schema.

Die P- und B-Rahmen sind bewegungskompensiert. Für jeden kodierten 8×8 Block des Videos wird der entsprechende Block des Wasserzeichens bestimmt, mit der DCT transformiert und zum transformierten Videoblock dazuaddiert (siehe Abb. 7).

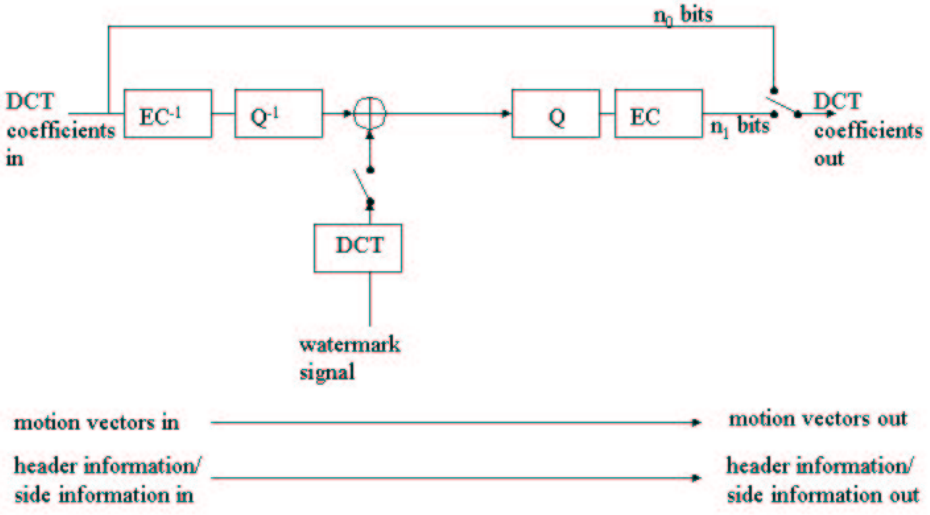


Abbildung 7: Schema zur Markierung eines komprimierten Videos.

Der ankommende MPEG-2 Bitstrom wird in Kopf- und Randinformation, Bewegungsvektoren (motion vectors) und DCT kodierte Signalblöcke unterteilt. Nur der kodierte Signalteil wird verändert, die beiden anderen Teile werden zum markierten MPEG-2 Bitstrom kopiert. Die kodierten Signalblöcke werden durch Huffmancodes repräsentiert, die jeweils einen nicht

nullwertigen DCT-Koeffizienten darstellen. Hier wird nun zuerst jede ankommende Huffman-Code-Sequenz dekodiert (in Bild 7 $\boxed{EC^{-1}}$) und zurückquantisiert ($\boxed{Q^{-1}}$), d.h. der Quantisierungswert wird duch den tatsächlichen Wert ersetzt. Danach wird zu jedem DCT-Koeffizient der entsprechende DCT-Koeffizient des kodierten Wasserzeichens addiert und das Ergebnis wiederum quantisiert und Huffman-kodiert. Danach vergleicht man die Anzahl n_0 der Huffman-Bits vor und die Anzahl n_1 nach der Markierung. Nur wenn sich die Bitzahl nicht vergrößert hat ($n_1 \leq n_0$), wird der markierte Bitstrom übertragen, ansonsten wird der unmarkierte übertragen. Da man nur wenige Wasserzeichen-Bits in viele DCT-Koeffizienten einbetten möchte, macht es nichts aus, wenn man wenige DCT-Koeffizienten nicht markieren kann, solange genügend viele übrig bleiben, die ohne Erhöhung der Bitzahl markiert werden können. Je nach Anspruch an die Robustheit oder die Datenrate des Wasserzeichens erhöht oder erniedrigt man die chip rate. Man beachte folgendes:

- Nur nicht-nullwertige DCT-Koeffizienten des Inputstroms können zur Markierung mit Wasserzeichen verwendet werden. Das heißt, das eingebettete Wasserzeichen hängt vom Bildsignal ab.
- Von den nicht-nullwertigen DCT-Koeffizienten werden nur solche markiert, die dadurch nicht die Bitrate erhöhen.

Je nach Struktur der dargestellten Szene werden gewöhnlich $15-30\%$ der DCT-Koeffizienten verändert.

5.2 Drift-Kompensation

MPEG-2 verwendet ein bewegungskompensierendes Hybridkodierungsverfahren, bei dem aus den vorangegangenen Rahmen Bewegungsvorhersagen gemacht werden und dafür verwendet werden, den aktuellen Rahmen zu rekonstruieren. Dieser Rahmen kann selbst wiederum Grundlage für spätere Vorhersagen sein usw. Jede Veränderung, wie z. B. das Hinzufügen eines Wasserzeichens, würde sich somit in Zeit und Raum verbreiten. Ein Wasserzeichen könnte sich mit denen aus vorangegangenen und folgenden Rahmen akkumulieren und zu einer sichtbaren Verzerrung führen. Wenn man also einen MPEG-2 Bitstrom verändert, entsteht Drift durch die verschiedenen Vorhersagen, die im MPEG-2-Kodierer und im MPEG-2-Dekoder getroffen werden. Möchte man diesen Drift kompensieren, so muss man genau die Differenz der beiden Vorhersagen hinzuaddieren. Abb. 8 zeigt das Schema zur Einbettung eines Wasserzeichens mit der Addition eines entsprechenden Drift-Kompensierungssignals.

Man muss also die Differenz zwischen den Vorhersagen für den unmarkierten Bitstrom und den Vorhersagen für den markierten Bitstrom berechnen und zum markierten MPEG-2-Bitstrom dazuaddieren . Wird kein Wasserzeichen eingebettet, so sind die beiden Vorhersagen jeweils gleich und die Differenz (und damit auch das Drift-kompensierende Signal) ist 0. Abb. 9 zeigt das vollständige Verfahren mit Berechnung des Drift-kompensierenden Signals. Der linke $\boxed{\text{MC}}$-Block (motion compensation) berechnet die Vorhersage für den unmarkierten Bitstrom, der rechte $\boxed{\text{MC}}$-Block die Vorhersage für den markierten Bitstrom.

5.2.1 Ergebnis

Die Komplexität der Implememtation [HaGi97] ist wesentlich niedriger als die Komplexität eines Dekodierprozesses gefolgt von der Einbettung eines Wasserzeichens im Pixelbereich und einer anschließenden Komprimierung. Es wird bereits an schnelleren Implementationen gearbeitet, deren Komplexitäten nahe der eines Dekoders liegen werden. Typische Parameter

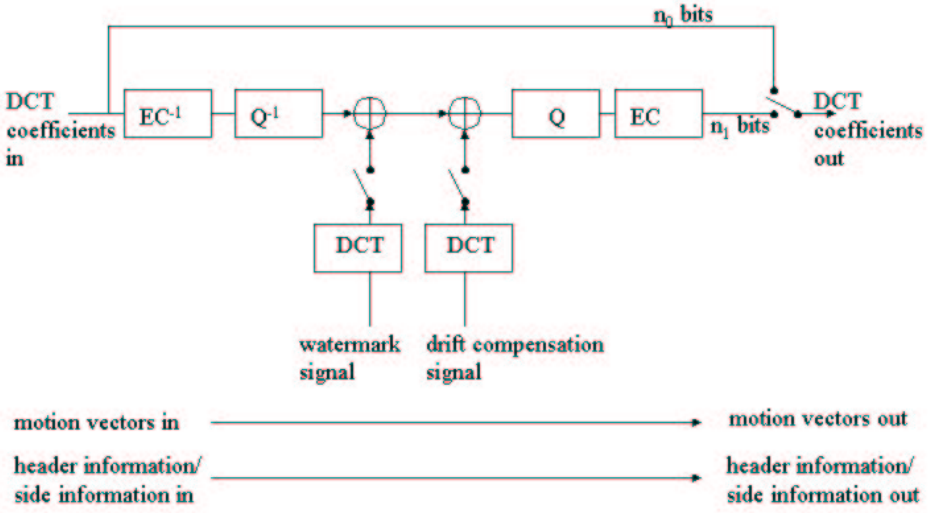


Abbildung 8: Markierung eines komprimierten Videos mit Drift-Kompensation.

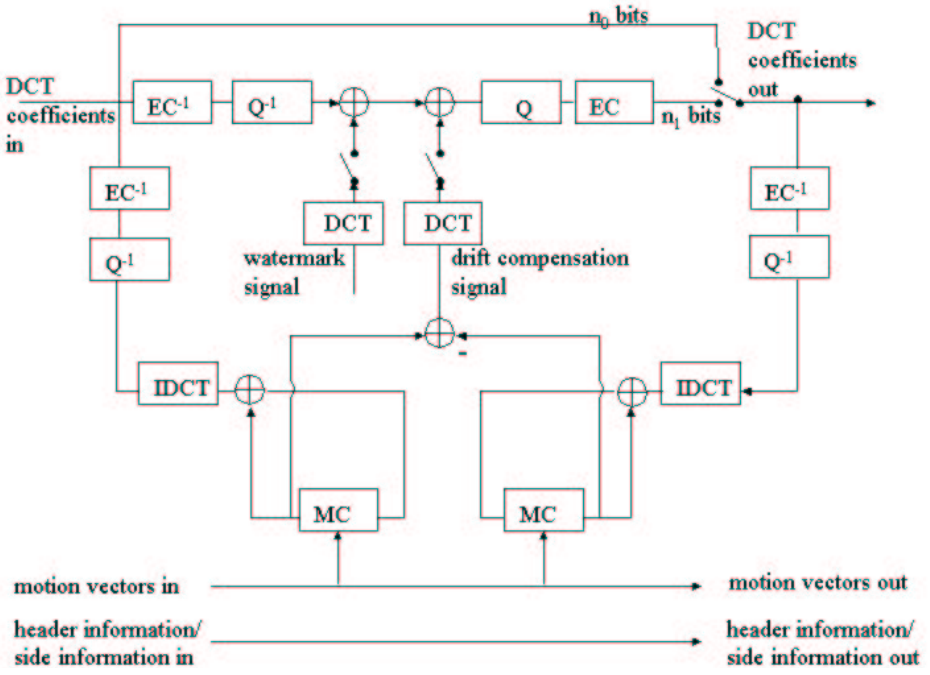


Abbildung 9: Vollständiges Schema zur Markierung eines MPEG-kodierten Videos mit Drift-Kompensation.

sind $a = 1..5$ für die Wasserzeichen-Amplitude und $cr = 10000..1000000$ für die chip rate. Die eingebetteten Wasserzeichen sind robust gegen lineare und nichtlineare Operationen wie cropping, Filtern, Quantisieren im Pixel- oder im Frequenzbereich und andere Angriffe [HaGi97].

6 Zusammenfassung

Diese Ausarbeitung motiviert erst die Notwendigkeit von Techniken zur Verwirklichung von Kopierschutz, Schutz der Urheberrechte und Nachweisbarkeit der Authentizität digitaler Daten. Danach werden Ansätze solcher Techniken wie Wasserzeichen, Fingerprinting und digitale Signaturen sowie deren Anwendungsgebiete vorgestellt. In Kapitel 2.3 werden die wichtigsten Eigenschaften eines Wasserzeichens aufgezählt und kurz erläutert. Es folgt die Beschreibung der Detektion und möglicher Angriffe auf ein Wasserzeichen sowie der entsprechenden Gegenmaßnahmen. Abschließend werden in den Kapiteln 3 bis 5 bereits entworfene Verfahren zur Einbettung von Wasserzeichen in Audiosignale, in den Pixelbereich eines Videos und in den Frequenzbereich eines MPEG-2-kodierten Videos vorgestellt.

7 Fazit

Es existieren bereits fortgeschrittene Ansätze zur Markierung von digitalen Audio- und Video-Daten sowie von Maschinencode (siehe [SHKQ99]). Allerdings sind die Ansprüche an solche Markierungsverfahren sehr widersprüchlich: Zum einen sollen die Informationen „unsichtbar“ und so in die Daten eingearbeitet werden, dass sie die Größe der Datei nicht verändern. Dies spricht für Wasserzeichen mit wenig „Informationsdichte“ und geringer Energie. Zum anderen sollen sie sowohl robust gegen jegliche Entfernungs- sowie Störungsversuche als auch zuverlässig detektierbar sein, was wiederum für Wasserzeichen mit viel Redundanz und viel Energie spricht. Es bleibt also fraglich, ob es überhaupt möglich ist, tatsächlich sichere Verfahren zur Einbettung von Wasserzeichen zu entwickeln.

Literatur

[BoTH96] L. Boney, A. H. Tewfik und K. N. Hamdy. Digital Watermarks for Audio Signals. In *Proceedings of the 1996 International Conference on Multimedia Computing and Systems (ICMCS '96)*. IEEE Press, 1996.

[CKLS95] I. J. Cox, J. Kilian, T. Leighton und T. Shamoon (Hrsg.). Secure Spread Spectrum Watermarking for Multimedia. Technischer Bericht, NEC Research Institute, Oktober 1995.

[HaEG98] F. Hartung, P. Eisert und B. Girod. Digital Watermarking of MPEG-4 Facial Animation Parameters. *Computers & Graphics* 22(3), 1998.

[HaGi97] F. Hartung und B. Girod. *Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video*, Band 1242, S. 423–436. Springer Lecture Notes in Computer Science. 1997.

[HaRR00] F. Hartung, F. Ramme und E. Research. Digital Rights Management and Watermarking of Multimedia Content for M-Commerce Applications. *IEEE Communications Magazine*, November 2000, S. 78–84.

[HaSG99] F. Hartung, J. K. Su und B Girod. Spread Spectrum Watermarking: Malicious Attacks and Counterattacks. *Proceedings of SPIE* Band 3657, Januar 1999.

[HHJQ+00] Gael Hachez, Laurent Den Hollander, Mehrdad Jalali, Jean-Jacques Quisquater und Christophe Vasserot. Towards a Practical Secure Framework for Mobile Code Commerce. In *ISW*, 2000, S. 164–178.

[ISO/93] ISO/CEI. Codage de l'image animee et du son associe pour les supports de stockage numerique jusqu'a environ 1,5 mbit/s, 1993. ISO/CEI 11172.

[Mait98] H. Maitre. Image Watermarking-Why is watermarking a hard proplem. Korea-France Workshop on Multimedia, Juli 1998.

[PaND99] K.K. Parhi, T. Nishitani und M. Dekker. *A review of watermarking principles and practices*, Kapitel 18, S. 461–485. 1999.

[PeAK99] F. A. P. Petitcolas, R. J. Anderson und M. G. Kuhn. Information Hiding – A Survey. *Proceedings of the IEEE* 87(7), Juli 1999.

[SHKQ99] Julien P. Stern, Gael Hachez, Francois Koeune und Jean-Jacques Quisquater. Robust Object Watermarking: Application to Code. 1999, S. 368–378.

[Stei99] R. Steinmetz. *Multimedia-Technologie*, Kapitel 19. Springer. 1999.

[Wesf01] A. Wesfeld. Steganographie: Unsichtbare Botschaften. *c't Magazin für Computer und Technik* (9), 2001, S. 170–181.

Internet Telephony: Technical Challenges and Solutions

Jens Deidersen

Kurzfassung

Die starke Verbreitung des Internet hat ein großes Interesse daran ausgelöst, Telefonie über das Internet zu realisieren. Das Internet ist jedoch nicht für Real-Time-Dienste in der Datenkommunikation ausgelegt, so dass die Forderung nach Internettelefonie eine große technische Herausforderung darstellt. Dieser Seminarbeitrag soll die technischen Probleme aufzeigen, die auf das Internet mit der Telefonie zukommen, sowie aktuelle Lösungen präsentieren. Zuerst werden die technischen Herausforderungen konkretisiert. Danach wird der H.323-Protokollstack kurz vorgestellt. Dieser ist integraler Bestandteil des letzten Abschnitts *Real-Time Multimedia over ATM (RMOA)*.

1 Einleitung

Die weltweite Akzeptanz und das Wachstum des Internets in den letzten Jahren hat großes Interesse an der Übermittlung von Sprache über dieses Netz hervorgerufen. Neben einer Kostenreduktion ermöglicht *Voice over IP (VoIP)* auch die Integration von Daten- und Sprachdiensten. Probleme bei der Realisierung von Internettelefonie bereiten vorallendingen die mangelnde Unterstützung der IP-Netze für Datenübertragung in Echtzeit. Im Gegensatz zu einem leitungsvermittelnden Netz, wie dem herkömmlichen Telefonnetz (*public switched telephone system (PSTN)*), können im Internet bisher keine Garantien bezüglich Dienstgüteparametern, wie etwa benötigte Bandbreite oder maximale Verzögerung, gegeben werden. Bei der Übertragung von Telefongesprächen über das Internet treten somit der Verlust von Paketen, eine zu große Verzögerung und *Jitter* auf. Um bei der Internettelefonie dieselbe Qualität wie bei der herkömmlichen Telefonie zu erreichen, müssen diese technischen Probleme gelöst werden.

Dieser Seminarbeitrag soll eine Übersicht der technischen Herausforderungen und aktueller Techniken zu deren Lösung geben. Danach soll als Beispiel einer möglichen Implementation der Internettelefonie *Real-Time Multimedia over ATM (RMOA)* vorgestellt werden. Bei diesem Ansatz werden die oben genannten Probleme durch den Einsatz von ATM als Internetbackbone weitgehend vermieden.

1.1 Vorteile der Internettelefonie

An der Weiterentwicklung der Internettelefonie trotz der angesprochenen technischen Probleme, erkennt man das große Interesse der Telekommunikationsunternehmen an dieser Technik und die Vorteile für die Nutzer. Als Sprachdienst, der auf bereits bestehenden Datendiensten aufbaut, kann die Internettelefonie über die Möglichkeiten der reinen Sprachübertragung hinaus die Einführung neuer und Integration bereits bestehender Sprach-, Fax- und Datendienste ermöglichen. Im folgenden sollen kurz einige der neuen Möglichkeiten und Vorteile aufgezeigt werden.

1.1.1 Kostenreduzierung

Da Internettelefonie auf dem paketvermittelnden IP-Netz aufsetzt, muss bei einem Anruf keine dedizierte Leitung vom Netzbetreiber vergeben werden, vielmehr teilen sich alle momentan vermittelten Anrufe die vorhandenen Netzwerkressourcen. Durch diese effizientere Nutzung der Ressourcen sinken die Kosten des Betreibers, denn während bei einem leitungsvermittelnden Netz pro Verbindung 64 kb/s benötigt werden, reichen für einen Internettelefonieanruf 6-8 kb/s aus. Durch die Vereinheitlichung der Netzinfrastruktur lassen sich im Vergleich zum Parallelbetrieb zweier verschiedener Netzarchitekturen weitere Kosten bei der Anschaffung und Administration einsparen.

Mit der Internettelefonie kann der Heimnutzer dieselbe Leitung parallel zum Telefonieren und für Datendienste nutzen. Eine zweite kostenpflichtige Telefonleitung für Internetdienste ist nicht mehr notwendig.

1.1.2 Unified Messaging

Der Unified Messaging Dienst ermöglicht die Kontrolle der über verschiedene Medien eingehenden Nachrichten an einer zentralen Stelle. Mittlerweile sind die meisten Menschen über mehrere Kontaktmöglichkeiten erreichbar. Dies sind im allgemeinen eine Emailadresse, Telefon- und Faxnummer am Arbeitsplatz und zu Hause, sowie eine Mobilfunknummer. Da der Nutzer meist nicht an allen diesen Kontaktpunkten gleichzeitig erreichbar ist, könnten Nachrichten verloren gehen. Bei der Nutzung einer Unified Messaging Box könnte die Nachricht weitergeleitet oder gespeichert und über das Internet jederzeit abgerufen werden. Die weitreichenden Konfigurationsmöglichkeiten dieses Dienstes erlauben die Weiterleitung von Anrufen, Emails oder Faxnachrichten an einen bestimmten Anschluss, wodurch letztlich nur eine einzige Telefonnummer benötigt werden würde.

1.1.3 Videokonferenzen und Teleworking

Als Sprachdienst auf einem Datennetz kann die Internettelefonie so erweitert werden, dass sie für den einfachen und kostengünstigen Einsatz von Videokonferenzen eingesetzt werden kann. Auch der Einsatz geteilter Anwendungen und der Austausch von Daten parallel zu der Konferenz wird möglich sein.

Die Nutzung von Internettelefonie erweitert somit auch die Möglichkeiten der Telearbeiter. Sie können über eine Internetverbindung auf das Firmennetz zugreifen und dort die Sprach- und Datendienste nutzen.

1.1.4 Call-Center im Internet

Um potentielle Kunden auf den eigenen Webseiten besser betreuen zu können, eignet sich für Unternehmen der Einsatz von webbasierten Call-Centern. Diese Betreuung kann für den Kunden parallel zu seinen Online-Aktivitäten direkt von der Webseite des Unternehmens erfolgen (*Click-to-Dial*), d.h. die Internetverbindung muss nicht beendet werden, und erlaubt somit eine direkte Weiterleitung zu den relevanten Informationen durch den Call-Center. Zusätzlich kann das Produkt auch ohne weiteres direkt telefonisch bestellt werden.

1.1.5 Abrechungsinformationen

Bislang musste der Nutzer bis zum Monatsende warten, um den fälligen Rechnungsbetrag zu erfahren. Bei normalen Telefongeräten ist mangels Ein- und Ausgabemöglichkeiten das Abrufen aktuellerer Rechungsinformationen nicht möglich. Bei Einsatz von Internettelefonie werden die Abrechnungsdaten über das Internet in Echtzeit verarbeitet und sind jederzeit über die Webseite des Telekommunikationsanbieters abrufbar, sofern dieser sie freigibt.

1.1.6 Sprachqualität

Das herkömmliche Telefonnetz unterstützt nur eine Tonstufe (4 kHz) und ist damit nicht für HiFi-Stereo und Surroundsound geeignet. Ist genug Bandbreite vorhanden, kann die Sprachqualität diesbezüglich durch Internettelefonie gesteigert werden.

2 Technische Herausforderungen

Das Internet setzt sich aus einer Ansammlung von kleinen und großen paketvermittelnden IP-Netzen zusammen. Diese Netze bieten bislang nur eine einzige, für Datendienste ausreichende, *best-effort*-Dienstgüteklasse an. Bei der Implementation von Internettelefonie führt dieser Mangel an Dienstgütezusagen zu technischen Problemen, die die Sprachqualität beeinträchtigen. In dem folgenden Abschnitt werden die Probleme Paketverlust, Verzögerung und Jitter kurz erklärt und einige Lösungen aufgezeigt.

2.1 Paketverlust

In den paketvermittelnden IP-Netzen des Internets werden die Pakete vor der Weiterleitung über eine Übertragungsstrecke in eine der Warteschlangen eines Routers eingereiht. Die Warteschlange wird dann durch den Versand der einzelnen Pakete vom Anfang der Warteschlange abgebaut (*FIFO-Prinzip*). Sollte ein Paket auf eine bereits komplett gefüllte Warteschlange eines Routers treffen, wird es von dem betreffenden Router verworfen.

Da die Anzahl der Internetnutzer stark gewachsen ist, kommen die Router vorallendingen zu Stoßzeiten mit der Vermittlung der Pakete nicht nach. Dadurch steigt die Anzahl der verworfenen Pakete. In IP-Netzen ist der Verlust eines Paketes somit nichts ungewöhnliches.

Ein Paket enthält etwa 40-80 ms Sprachdaten, deren Verlust die Qualität mindert. Während wenige verlorene Sprachfragmente von unserem Gehirn rekonstruiert werden können, machen zu viele verlorene Pakete die Sprache unverständlich.

Um die Auswirkungen des Verlustes von Paketen auf die Internettelefonie zu begrenzen, wurden verschiedene Verfahren entwickelt. Einige dieser Verfahren versuchen, die Anzahl der verworfenen Pakete direkt zu mindern, während sich andere darauf konzentrieren, den Schaden zu beheben, der durch verlorene Pakete entsteht.

2.1.1 Aufrüsten der Netzinfrastruktur

Der Verlust eines Paketes ist das Resultat mangelnder Bandbreite der Übertragungsstrecken oder zu geringer Verarbeitungsgeschwindigkeit des Routers. Durch Aufrüsten der Netzinfrastruktur kann die Anzahl der verworfenen Pakete gemindert werden. Einige technische Entwicklungen der letzten Jahre können die Übertragungskapazität der IP Backbones und Router wesentlich steigern. Hochgeschwindigkeitsübertragungtechnologien, etwa *Asynchronous Transfer Mode (ATM)*, *Synchronous Optical Network (SONET)* und

Wavelength-Division Multiplexing (WDM) ermöglichen Übertragungsgeschwindigkeiten bis in den Terrabit/Sekunde-Bereich. Um diese Bandbreite nutzen zu können, wurden Routertechologien wie *Multiprotocoll Label Switching (MPLS)* entwickelt, bei der mehrere Millionen Pakete pro Sekunde verarbeit werden können. Das Aufrüsten der Netzwerkinfrastruktur reduziert den Paketverlust, stellt aber eine kostenintensive und langfristige Lösung dar. Andere Lösungen konzentrieren sich darauf, den Schaden an der Sprachqualität durch verworfene Pakete zu beheben.

2.1.2 Noise oder Silence Substitution

Nach der Ankunft der Pakete beim Empfänger wird deren Inhalt in ein Audiosignal umgewandelt und abgespielt. Wenn ein Paket im Netzwerk verworfen wurde, kann der Inhalt nicht wiedergegeben werden. Einige Internettelefonie-Systeme ersetzen das Zeitintervall des verlorengegangenen Inhaltes durch Stille (*Silence Substitution*). Dies erlaubt eine unterbrechungsfreie, aber gleichzeitig abgehackt wirkende und auf Dauer unverständliche Sprachwiedergabe beim Empfänger [Hard95].

Das Ersetzen des Inhaltes von verlorenen Paketen durch weißes Rauschen (*Noise Substitution)* erzielt bessere Ergebnisse als die Silence Substitution. Dies liegt an der Fähigkeit des menschlichen Gehirns, die erhaltene Nachricht besser mit Hintergrundlärm rekonstruieren zu können als mit Stille (Phonemic Restoration) [PeHH98].

2.1.3 Paket Wiederholung oder Interpolation

Um den Inhalt eines verlorengegangen Paketes zu ersetzen, kann der Inhalt des zuletzt abgespielten Paketes wiederholt werden. Um eine bessere Qualität zu erreichen, wird empfohlen, das wiederholte Signal gedämpft oder abklingend wiederzugeben, z.B. bei *Global System for Mobile Communications (GSM).*

Die Reparatur durch Interpolation benutzt zur Substitution die Charakteristiken der Pakete in der Umgebung eines verlorenen Paketes. Dies erlaubt eine Ersetzung, die den Charakteristiken des gesamten Sprachstromes folgt. Studien haben gezeigt, dass die Interpolation eine bessere Sprachqualität als die Silence Substitution oder die Paketwiederholung ermöglicht [GoLW86].

2.1.4 Frame Interleaving

Bei *Frame Interleaving (Rahmenverzahnung)* werden die ursprünglich aufeinanderfolgenden Sprachrahmen in einer anderen Reihenfolge auf die Pakete verteilt und verschickt. Somit verursacht der Verlust eines Paketes nicht eine lange Unterbrechung an einer Stelle, sondern nur eine Reihe kurzer Unterbrechungen an verschiedenen Stellen, welche durch Noise Substitution (vgl. 2.1.2) oder Paketinterpolation (vgl. 2.1.3) kompensiert werden können, so dass sie beim Empfänger kaum bemerkt werden. Das Interleaving reduziert die Auswirkungen eines verworfenes Paketes auf die Sprachqualität, erzeugt aber eine weitere Verzögerungskomponente (vgl. 2.2).

2.1.5 Vorwärtsfehlerkorrektur

Bei dem Einsatz dieses Verfahrens werden Teile der Informationen eines Paketes in darauffolgenden Paketen erneut übertragen. Sollte ein Paket verworfen werden, kann dessen Inhalt aus

den Daten der folgenden Pakete rekonstruiert werden. Die Informationen können entweder unabhängig vom Datenstrom sein oder die Eigenschaften des Datenstroms zur Rekonstruktion nutzen [Stei99].

2.2 Verzögerung

Eine der größten technischen Herausforderungen bei der Internettelefonie stellt die Verzögerung dar. Eine zu große Verzögerung kann die Kommunikation der Teilnehmer in verschiedenster Weise beeinträchtigen, z.B. könnten sich die Gesprächsteilnehmer gegenseitig ins Wort fallen, wenn sie die Dauer der Verzögerung als Gesprächspause der anderen Partei interpretieren. Zusätzlich verschlechtert eine große Verzögerung das Echo durch Reflexion des übertragenen Signals beim Empfänger. Um dies zu verhindern, sollte die Verzögerung einen oberen Grenzwert von 150 ms nicht überschreiten. Dennoch sind Verzögerungen zwischen 150-400 ms für Fernverbindungen akzeptabel, da die Nutzer mental auf eine lange Verbindung eingestellt sind. Eine Gesamtverzögerung von über 400 ms ist nicht mehr akzeptabel.

In einem IP-Netzwerk wird die Gesamtverzögerung durch verschiedene Faktoren beeinflusst (vgl. Abbildung 1). Während einige Verzögerungsarten mit deren Dauer bekannt sind, gibt es andere, deren Dauer variabel und unberechenbar ist (vgl. insbesondere 2.2.2). Der kritische Grenzwert von 400 ms kann leicht überschritten werden.

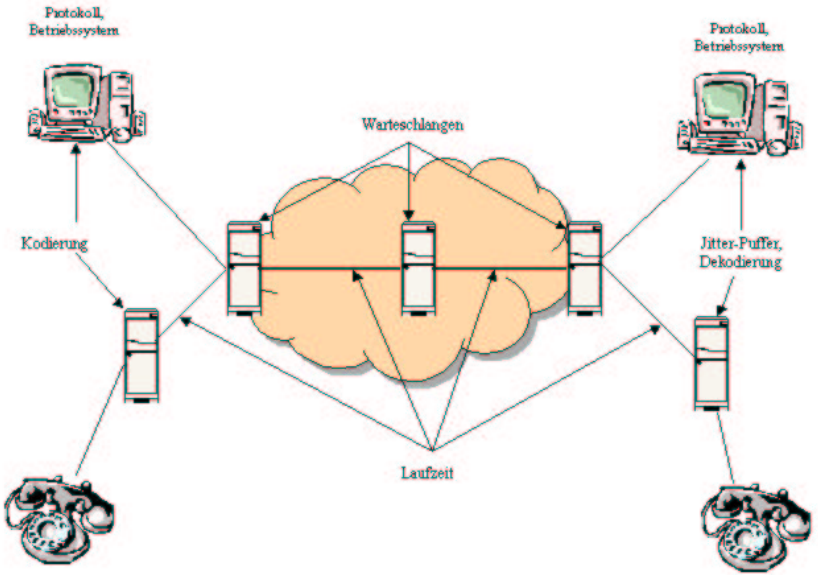


Abbildung 1: Verzögerungen

Im folgenden sollen die Ursachen sowie mögliche Schritte zur Reduzierung der Kodierungsverzögerung, der Wartezeit und der Signalverzögerung erläutert werden.

2.2.1 Kodierungsverzögerung

Kodierungsverfahren wandeln die analoge Sprache in digitale Daten und komprimieren sie, um den Bandbreitenbedarf für die Internettelefonieverbindung zu senken. Die Wandlung und Sprachkomprimierung sind mit einer Verzögerung durch die Kodierung verbunden, wobei eine höhere Kompression auch eine grössere Verzögerung impliziert.

Die gesamte Kodierungsverzögerung setzt sich aus der Rahmenerzeugungszeit, der Rahmenbearbeitungszeit und der Vorausschau zusammen. Die Rahmenerzeugungszeit gibt die zur Erzeugung eines Sprachrahmens benötigte Zeit an, d.h. die Dauer der Sprachdaten in einem Rahmen. Die Dauer der Vorausschau entsteht durch die Berücksichtigung von Korrelation aufeinanderfolgender Pakete zur Komprimierung durch den Kodierer. Die Dekodierungsverzögerung beim Empfänger beträgt typischerweise die Hälfte der Kodierungsverzögerung beim Sender. Drei von der ITU standardisierte Kodierungsverfahren werden in Abbildung 2 mit

Codec	G.723.1	G.729	G.729A
Bitrate	5.3/6.4 kb/s	8 kb/s	8 kb/s
Framesize	30 ms	10 ms	10 ms
Processing delay	30 ms	10 ms	10 ms
Lookahead delay	7.5 ms	5 ms	5 ms
Frame length	20/24 bytes	10 bytes	10 bytes
DSP MIPS	16	20	10.5
RAM	2200	3000	2000

Abbildung 2: Kodierverfahren

ihren Verzögerungseigenschaften aufgelistet.

2.2.2 Wartezeit

Die Wartezeit tritt an den verschiedenen Vermittlungs- und Übertragungspunkten auf, etwa bei Routern und Gateways eines Netzwerkes. Dort warten die Sprachpakete hinter anderen Paketen auf die Übertragung über denselben Ausgang. Die Sprachpakete werden mit derselben Priorität wie Datenpakete weitergeleitet, obwohl bei diesen eine größere Verzögerung akzeptabel sein kann. Da die Anzahl der wartenden Pakete in der Warteschlange stochastisch und somit nicht berechenbar ist, kann die Wartezeit im Internet signifikant von Paket zu Paket schwanken. Eine Reduzierung dieser Verzögerung ist etwa durch schnellere Anbindungen möglich. Die Internet Engineering Task Force (IETF) arbeitet an Mechanismen wie *Integrated Services (IntServ)* oder *Differentiated Services (DiffServ)*, um den Sprachpaketen eine höhere Priorität bei der Vermittlung gegenüber den reinen Datenpaketen geben zu können, damit deren Wartezeit reduziert werden kann.

2.2.3 Ausbreitungsverzögerung

Die Zeit, die ein Signal zur Überwindung einer Strecke benötigt, wird als *Ausbreitungsverzögerung (propagation delay)* bezeichnet und ist proportional zur Lichtgeschwindigkeit. Durch die direkte Abhängigkeit von der Länge der Strecke, wird sie besonders bei langen Distanzen wichtig, wie z.B. bei geostationären Satellitenstrecken. Sie stellt auch für das herkömmliche Telefonnetz (PSTN) eine unvermeidbare Verzögerugskomponente dar.

2.3 Jitter

Auf ihrem Weg zum Empfänger durch ein IP-Netz müssen IP-Pakete desselben Datenstromes nicht zwangsläufig denselben Weg durchlaufen. Oft werden sie über komplett verschiedene Wege weitergeleitet, und erfahren dadurch unterschiedliche Wartezeiten in den Routern.

Auch die streckenabhängigen Ausbreitungsverzögerungen der einzelnen Pakete schwanken. Dies führt zu einer Varianz der Paket-Ankunkftszeiten beim Empfänger. Das Schwanken der Ankunftszeiten bezeichnet man als *Jitter*. Die Auswirkungen des Jitters auf die Sprachqualität stellt ein großes Problem für die Internettelefonie dar. Sollte ein IP-Paket im Netz zu stark verzögert werden, dann wird es beim Empfänger als verloren betrachtet. Sollten zu viele Pakete verworfen werden, so leidet die Sprachqualität erheblich (siehe 2.1). Um eine gewisse, unvermeidbare Ankunfszeitenvarianz zu erlauben und trotzdem einen kontinuierlichen Paketstrom zum Abspielen gewährleisten zu können, werden die erhaltenen Pakete beim Empfänger vor dem Abspielen in einem *Jitter-Puffer* zwischengespeichert. Die Einrichtung dieses Puffers erzeugt eine weitere Verzögerungskomponente, die zu der Gesamtverzögerung beiträgt (vgl. 2.2). Daher steigt bei einem großen Jitter-Puffer die Gesamtverzögerung, auch wenn die durchschnittliche Verzögerung durch das Netz bis zum Empfänger gering ist. Ein Jitterpuffer optimaler Größe muss sowohl den Jitter kompensieren, als auch die Verzögerung auf tolerierbare Grenzen beschränken. Im Idealfall sollte die Größe des Jitter-Puffers dynamisch an die Netzbedingungen angepasst werden.

3 Real-Time Multimedia over ATM (RMOA)

Um den in Abschnitt 2 aufgeführten technischen Problemen entgegenzuwirken, müssen Dienstgüteparameter wie Bandbreite und Verzögerung eingehalten werden. Um in IP-Netzen diese Garantien ermöglichen zu können, wurden mehrere Ansätze durch die IETF verfolgt, *Internet Integrated Services (IntServ)* und *Differentiated Services (DiffServ)*. Diese Ansätze konnten sich bislang noch nicht durchsetzen, so dass nicht klar ist, ob die versprochene Dienstgüte tatsächlich geliefert werden kann [Brad94, Blak98].

Weit verbreitet mit erprobten Dienstgütemechanismen ist *Asynchronous Transfer Mode (ATM)*. Durch den Einsatz dieser Mechanismen können die technischen Probleme der Internettelefonie teilweise reduziert werden. Deswegen wurde *Real-Time Multimedia over ATM* für die Vermittlung von H.323-Internettelefonie über ATM-basierte Internetbackbones entworfen.

Da bei diesem Ansatz Internettelefonie über H.323 realisiert wird, sei hier erst ein Anruf nach dem H.323 Standard vorgestellt. Danach wird das RMOA-Einsatzszenario und der in diesem Szenario zur Abwicklung eines H.323-Anrufes benötigte H.323-H.323 Gateway erklärt.

3.1 Aufbau eines H.323 Anrufes

Die *International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)* hat zur Multimediakommunikation über das Internet den H.323-Standard zur Beschreibung der Systemkomponenten, der Anrufmodelle und der Signalisierungsprozeduren geschaffen. H.323 wurde so konzipiert, dass die traditionellen leitungsvermittelten Dienste auf paketvermittelnde Netze erweitert werden können. Ein primäres Ziel von H.323 ist die Interoperabilität mit den leitungsvermittelnden Netzen (PSTN und ISDN). Die Basiselemente der H.323-Architektur sind Endgeräte, Gateways, Gatekeeper und Multipoint Control Units (MCU). Endgeräte, Gateways und MCU's werden hier als *Endpunkte* bezeichnet.

Ein *Endgerät* ist ein Endnutzergerät, etwa ein einfaches Telefon oder ein PC. Seine Hauptaufgabe ist die Initiierung und das Entgegennehmen von Anrufen und die Teilnahme an Konferenzen über H.323.

Ein *Gateway* ist ein vermittelndes Gerät, welches Interoperabilität zwischen H.323-fähigen und nicht-H.323-fähigen Geräten herstellt. Seine Hauptaufgaben sind Signalübersetzung, Medienkodierung und der Versand der erstellten Pakete.

Ein *Gatekeeper* verwaltet eine Menge an registrierten Endpunkten, welche als *Zone* bezeichnet wird. Seine Hauptaufgaben sind Anrufzugangskontrolle, Adressauflösung und andere Managementfunktionen. Jeder Endpunkt muss sich bei dem für die Zone zuständigen Gatekeeper registrieren, erst dann kann er einen Anruf oder eine Konferenz initiieren. Der Gatekeeper ermittelt dann die richtige Transportadresse zu dem Zielempfänger. Anfrufanfragen werden von dem Gatekeeper anhand der Netzparameter, z.B. verfügbare Bandbreite, akzeptiert oder abgelehnt.

Eine *MCU* ermöglicht die Kontrolle über Multiparty Videokonferenzen. Sie enthält zwei logische Komponenten, einen *multipoint controller (MC)* zur Anrufkontrollkoordination und einen *multipoint processor (MP)*, der die Audio- und Videodaten koordiniert.

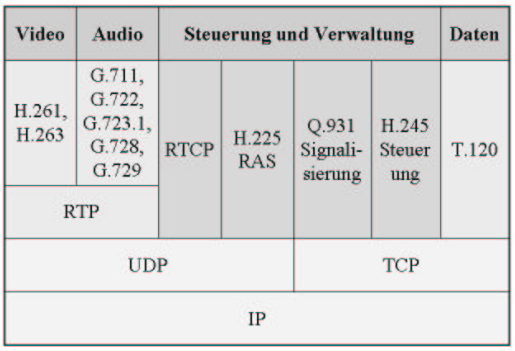
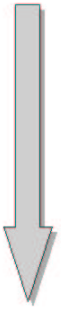


Abbildung 3: H.323 Protokolle

Die Hauptprotokolle in der Anrufvorbereitung sind das *Registration Admission Status (RAS) Protokoll*, das *Q.931 Signalisierungsprotokoll* und das *H.245 Medien- und Konferenzkontroll Protokoll*. Das RAS Protokoll ist verantwortlich für die Registrierung von Endpunkten beim zuständigen Gatekeeper. Neben der Registrierung ermöglicht RAS auch die Kontrolle der Endpunkte einer Zone durch den Gatekeeper und die Verwaltung der Ressourcen einer Zone. Das H.245 Medien und Konferenzkontroll Protokoll wird nach dem Verbindungsaufbau durch das Signalisierungsprotokoll Q.931 zwischen zwei Teilnehmern zum Austausch verschiedenster Informationen bezüglich deren Kommunikation eingesetzt. Dazu gehören der Nachrichtentyp (Audio, Video oder Daten), das Aushandeln der verwendeten Kodierung oder das Öffnen logischer Kanäle.

Von den H.323 Endgeräten wird das *Real-Time Transport Protokoll (RTP)* als Transportprotokoll für Multimedia über UDP/IP verwendet. Hauptaufgabe dieses Protokolls ist der Transport der Audio- und Videodaten von Echtzeitdiensten über ein IP-Netzwerk. RTP enthält die Identifikation des transportierten Inhaltes, Sequenznummern und Zeitstempel, damit die Pakete wieder geordnet und Verluste entdeckt werden können. Kontrolliert wird RTP über das *Real-Time Control Protocol (RTCP)*. Es wird an dieselben Teilnehmer, aber an einen anderen Port verschickt. RCTP-Pakete ermöglichen viele Dienste, z.B. die Nutzeridentifikation oder Qualitätsrückmeldungen, damit die Übertragungsrate oder die Kodierung dynamisch verändert werden kann.

In Abbildung 4 werden die H.323 Protokollphasen dargestellt. RAS wird in den Phasen 0,1 und 6 zur Registrierung und zum Abbau der Verbindung eingesetzt. Das Signalisierungsprotokoll wird in den Phasen 2,5 und 6, H.245 in den Phasen 3 und 5 eingesetzt. Mediendaten werden über RTP und RCTP in Phase 4 ausgetauscht. Der H.323 Standard stellt keine Dienstgüteanforderungen an das Netz, über das der H.323-Anruf vermittelt werden sollen.



0	Initialisierung	- Registrierung beim Gatekeeper
1	Gatekeeper admission	- Anruferlaubnis durch den Gatekeeper - Adressauflösung
2	Signalisierung	-Verbindungsaufbau - Anrufinitiierung und Akzeptierung/Ablehnung
3	Verhandlung	- Verhandlung der Beziehungen - Einigung auf den Operationsmodus
4	Datenaustausch	- Konfigurierung und Öffnung logischer Kanäle - Übertragung und Empfang der Daten
5	Verhandlung	- Änderung der Parameter, z.B. Der Teilnehmer
6	Abbau	- Beendigung des Anrufes - Abmelden beim Gatekeeper

Abbildung 4: H.323 Ablauf

3.2 Das Szenario zu RMOA

Um den in Abschnitt 2 aufgeführten technischen Problemen entgegenzuwirken, wurde *Real-Time Multimedia over ATM* für den Transport von H.323 Internettelefonie über ATM-basierte Internetbackbones entworfen. ATM kommt zunehmend im Kern der Telekommunikationsanbieternetze zum Einsatz und bietet einfache Dienstgütemechanismen an, etwa eine *constant bit rate (CBR)* oder *real-time variable bit rate (rt-VBR)* für garantiere Bandbreite. Um diese Dienstgütemechanismen für die Internettelefonie nutzbar zu machen, hat das ATM Forum eine effizente und skalierbare Möglichkeit zum Transport von H.323 Anrufen über ATM entwickelt.

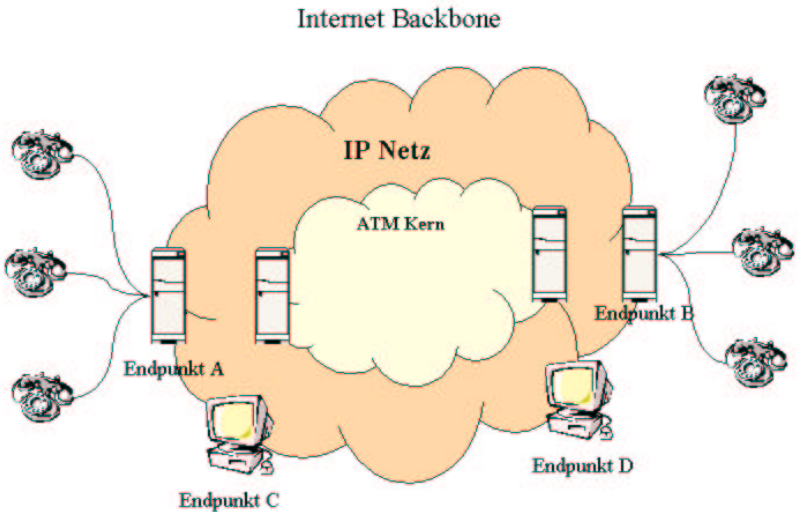


Abbildung 5: Internet Backbone mit ATM-Kern

Abbildung 5 stellt ein Internetbackbone über ATM, sowie die zwei Typen von H.323-Endpunkten, Gateways (Endpunkte A und B) und Terminals (Endpunkte C und D) dar. H.323-Terminals sind in der Lage, einen H.323-Anruf zu initiieren und zu empfangen, während H.323-Gateways dieses für andere nicht-H.323-Endgeräte, etwa normale Telefone, durchführen.

Die H.323-Endpunkte in Abbildung 5 befinden sich in einem IP-Netz, welches der H.323-Anruf bis zu den Gateways am ATM-Kern durchläuft. Sollte dieses IP-Netz nicht korrekt konfiguriert sein, so kann auch die garantierte Dienstgüte über den ATM-Kern die Sprachqualität nicht

verbessern. Um H.323 Anrufe mit der Dienstgüte von ATM zu transportieren, wurde ein neues *H.323-H.323 Gateway (Gateway for H.323 Media Transport over ATM)* konzipiert. Diese Gateways an den Rändern des ATM-Kerns sind in der Lage, H.323 Anrufe zu empfangen und über eine dedizierte virtuelle Leitung (*Virtual Circuit (VC)*) in den ATM-Kern zu schicken. Ein effizienter Transport dieser Anrufe ist über ATM möglich, da die UDP- und IP-Header nicht über den ATM-Kern transportiert werden müssen. Durch die Komprimierung des RTP-Headers wird dieser Ansatz noch effizienter.

3.3 Der H.323 - H.323 Gateway

Abbildung 6 beschreibt das Szenario aus 3.2 bzw. Abbildung 5, wobei hier ein H.323 Anruf zwischen den Endpunkten A und B dargestellt wird.

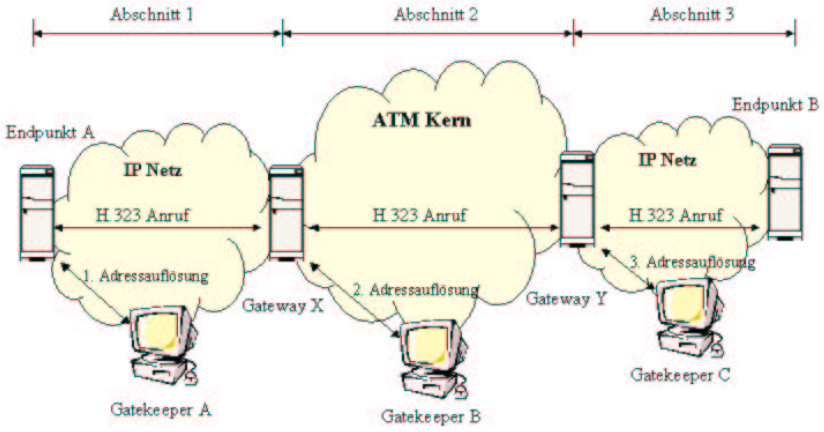


Abbildung 6: Ein H.323-Anruf über Real-Time Multimedia over ATM

Es werden drei Gatekeeper mit den von ihnen verwalteten Zonen dargestellt. Der Anruf von Endpunkt A zu Endpunkt B wird durch die Gatekeeper in drei Abschnitte geteilt. Diese Aufteilung des Anrufes in verschiedene Abschnitte wird durch die Medien- und Kontrollterminierung an den H.323-H.323 Gateways ([Foru99]) am Rand des ATM-Kerns erzwungen. Die Kontrollterminierung erlaubt es diesem Gateway, einen dedizierten VC für jeden H.323 Anruf aufzubauen, während die Medientermination eine effiziente Paketweiterleitung ermöglicht.

Wie in Abschnitt 3.1 beschrieben, bietet ein H.323-Gatekeeper die Dienste Registrierung, Zugangskontrolle und Status (registration, admission and status (RAS)) an. Dies beinhaltet auch die Adressauflösung. Um einen Anruf zu Endpunkt B zu tätigen, wird Endpunkt A über RAS den Gatekeeper A nutzen, da er bei diesem registriert ist. Gatekeeper A übersetzt die von Endpunkt A übermittelte Telefonnummer von Endpunkt B in eine IP-Adresse und eine Portnummer. In dem Szenario aus Abbildung 6 teilt Gatekeeper A dem Endpunkt A mit, dass für Anrufe zu Endpunkt B das Gateway zuständig ist. Die zum Aufbau eines H.323 Anrufes zu Endpunkt B notwendigen H.225.0 Nachrichten schickt Endpunkt A an diese IP-Adresse. Gateway X kann als eigener H.323-Endpunkt betrachtet werden, der einen Anruf zu Endpunkt B tätigen will. Die Adressauflösung für den nächsten Abschnitt des Anrufes durch Gatekeeper B ergibt als Ziel für den Anruf zu Endpunkt B das Gateway Y. Dieses richtet über Gatekeeper C den letzten Abschnitt des Anrufes ein. Das Unterteilen des Anrufes in Abschnitte durch die

Gatekeeper ist für die beiden Endpunkte transparent. Die Signalisierungsnachrichten werden an die H.323-H.323 Gateways adressiert.

Die H.225.0 Nachrichten bauen den H.323-Anruf durch die Öffnung eines H.245 Kontrollkanals auf. Die H.225.0 Nachrichten werden durch die Unterteilung an das jeweilige Gateway auf dem Weg von Endpunkt A zu Endpunkt B geschickt. Diese H.225.0 Nachrichten enthalten die für den H.245-Kanal notwendigen Adressen zum Austausch der H.245-Kontrollnachrichten. Diese Adressen werden von den Gateways ersetzt durch die für den nächsten Abschnitt gültigen lokalen Adressen. Durch diese Adressersetzung wird der H.245-Kontrollkanal für den H.323-Anruf in 3 Abschnitte unterteilt. Dadurch ist es für die H.323-H.323 Gateways an den Enden eines Abschnittes des Anrufes einfacher, die zur Einrichtung des Virtual Circuits notwendigen Informationen der H.245-Nachrichten zu identifizieren.

3.4 Aubau logischer Verbindungen über Virtual Circuits

Nach der Einrichtungs des H.245 Kontrollkanals können die Endpunkte die gewünschten Audio- und Video-RTP-Datenströme öffnen. Endpunkt A schickt dazu eine H.245 Kontrollnachricht um alle Kanäle vorzuschlagen, die dieser Endpunkt zu Endpunkt B öffnen möchte. Auch werden verschiedene Kodierungsverfahren (vgl. Abbildung 2) für jeden Kanal vorgeschlagen. Das verwendete Kodierverfahren gibt dabei die benötigten Ressourcen beim Einrichten des VC's im ATM-Netz vor. Um die benötigte Dienstgüte zu erfüllen, sollte entweder CBR oder rt-VBR als Dienstkategorie verwendet werden. Endpunkt B teilt Endpunkt A dann durch eine weitere H.245 Kontrollnachricht mit, welche Kanäle er annimmt und welche Kodierverfahren jeweils benutzt werden sollen. Für die Kanäle von Endpunkt B zu Endpunkt A werden von Endpunkt B dieselben Nachrichten verschickt.

Durch den Austausch der Transportadressen, wie in 3.3 beschrieben, werden die H.245-Nachrichten in Wirklichkeit an die H.323-H.323-Gateways gesendet. Deswegen können die Gateways diese Nachrichten vor der Weiterleitung inspizieren und gegebenenfalls modifizieren.

Während die H.245 Prozedur unidirektionale RTP-Kanäle für bidirektionale Kommunikation aufbaut, wird nur ein einziger VC zum Transport des bidirektionalen Flusses eingerichtet.

3.5 H.323-Anrufvermittlung über den ATM-Kern

Wie in Abschnitt 3.3 beschrieben, werden die H.245 Kontrollnachrichten zur Einrichtung der Virtual Circuits an das H.323-H.323 Gateway adressiert. Dies stellt eine Kontrollterminierung an den Gateways dar. Die Einheiten, die diese Nachrichten verschicken, teilen die Adressen mit, auf denen sie den Empfang der B-zu-A RTCP Pakete erwarten. Auch die antwortenden Einheiten teilen die Adressen mit, auf denen sie die RTP und RTCP Pakete empfangen wollen.

Durch die Unterteilung des Anrufes in drei Abschnitte werden die RTP Pakete jeweils nur an lokal gültige IP-Adressen verschickt, so dass nicht nur der Kontrollkanal, sondern auch der Medienkanal in dieselben Abschnitte unterteilt wird.

Für die Weiterleitung des H.323-Anrufes über den ATM-Kern werden die IP und UDP Header der bei Gateway X eintreffenden Pakete nicht benötigt. Nach der Einrichtung des Virtual Circuits verwaltet Gateway X eine Abbildung der Transportadressen auf die VC virtual path/connection identifier (VPI/VCI). Gateway Y verwaltet die Abbildung der VPI/VCI zurück auf Transportadressen. Nachdem Gateway Y die Pakete über den Virtual Circuit empfangen hat, fügt es diesen neue IP und UDP-Header hinzu und leitet sie an Endpunkt B weiter.

Dass die IP und UDP Header nicht über den ATM-Kern verschickt werden, ist für den Sprachverkehr wichtig. Denn einige Kodierverfahren erzeugen 10 Byte große Rahmen (vgl. Abbildung 2). Der Versand dieser Rahmen über IP ergibt einen gesamten Protokolloverhead von mindestens weiteren 20 Byte für den IP-Header, 8 Byte für den UDP-Header und 12 Byte für den RTP-Header. Dieser Protokolloverhead wird nicht über den ATM-Kern transportiert und damit die benötigte Bandbreite reduziert.

Durch die Komprimierung des RTP-Headers kann die benötige Bandbreite weiter gesenkt werden. Da die meisten Felder des RTP-Headers über die gesamte Lebensdauer einer Sprachverbindung konstant bleiben, muss dieser nur einmal komplett übertragen werden. Danach reicht die Übermittlung der geänderten Felder aus (vgl. [PaKM00]). Dies macht den RMOA-Ansatz sehr effizient.

4 Fazit

Nur durch die Lösung der in Abschnitt 2 aufgeführten technischen Probleme kann die Internettelefonie zu einem erfolgreichen Produkt mit weitreichender Akzeptanz werden. Die Akzeptanz hängt von der gebotenen Qualität der Internettelefonie ab, welche mindestens der Qualtität der herkömmlichen Telefonie entsprechen sollte. Wegen der enormen Größe und Undurchschaubarkeit des Internets sind Lösungen schwierig zu implementieren. Dazu müssen neue Protokolle und Techniken in das Internet eingebaut werden, um so dem Endanwender echte Dienstgüte anbieten zu können. Die IETF entwickelt neue Protokolle und Techniken um Dienstgüte in das Internet der nächsten Generation integrieren zu können, z.B. *Integrated Services* und *Differentiated Services*. Der in Abschnitt 3 beschriebene Ansatz kann durch die Vermittlung der Anrufe über ATM einige der Probleme umgehen und so Internettelefonie mit einer angemessenen Qualität anbieten.

Literatur

[BeKe00] D. Bergmark und S. Keshav. Building Blocks for IP Telephony. *IEEE Communications Magazine* Band April 2000, 2000, S. 88–94.

[Blak98] S. Blake. An Architecture for Differentiated Services. RFC 2475, Dezember 1998.

[Brad94] R. Braden. Integrated Services in the Internet Architecture. RFC 1633, Juni 1994.

[Foru99] ATM Forum. Gateway for H.323 Media Transport over ATM, Juli 1999. AF-SAA-0124.000.

[GoLW86] D. Goodman, O. Lockhart und W. Wong. Waveform Substitution Techniques for Recovering Missing Speech Segments in Paket Voice Communications. *IEEE Trans. Acoustics Speech and Sig. Processing* Band ASSP-34 no. 6, 1986, S. 1440–48.

[HaNA00] M. Hassan, A. Nayandoro und M. Atiquzzaman. Internet Telephony: Services, Technical Challenges, and Products. *IEEE Communications Magazine* Band April 2000, 2000, S. 96–103.

[Hard95] V. Hardman. Reliable Audio for Use over the Internet. *Proc. INET '95*, 1995.

[IKTO00] K. Iida, K. Kawahara, T. Takine und Y. Oie. Performance Evaluation of the Architecture for End-to-End Quality-of-Service Provisioning. *IEEE Communications Magazine* Band April 2000, 2000, S. 76–81.

[ITUT00] International Telecommunication Union Telecomunications Sector. Re. H.225.0 Media Stream Packetization and Synchronization for Visual Telephone Systems on Non-Guaranteed Quality of Service LANs, November 2000.

[KBSS+98] T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec und J. Mahler. Real-Time Voice Over Packet-Switched Networks. *IEEE Network* Band January/February 2000, 1998, S. 18–27.

[Kuri99] Jürgen Kuri. Sprache in Päckchen. *ct magazin für computer technik*, April 1999, S. 220–229.

[LHHJ+00] B. Li, M. Hamdi, Y. Hou, D. Jiang und X. Cao. QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges. *IEEE Communications Magazine* Band April 2000, 2000, S. 54–61.

[PaKM00] C. Pazos, M. Kotelba und A. Mails. Real-Time Multimedia over ATM: RMOA. *IEEE Communications Magazine* Band April 2000, 2000, S. 82–87.

[PeHH98] C. Perkins, O. Hodson und V. Hardman. A Survey of Paket-Loss Recovery for Streaming Audio. *IEEE Network* Band 12, 1998, S. 40–48.

[RaSa00] A. Rayes und K. Sage. Integrated Management Architecture for IP-Based Networks. *IEEE Communications Magazine* Band April 2000, 2000, S. 48–53.

[Stei99] R. Steinmetz. *Multimedia-Technologie*. Springer. 1999.

Das Stream Control Transmission Protocol SCTP

Georgios Papadopoulos

Kurzfassung

Der schnelle und zuverlässige Transport von Signalisierungsmeldungen über IP-basierte Netze hat für viele Internet-Anwendungen und insbesondere für moderne Telekommunikationsnetze in letzter Zeit sehr viel an Bedeutung gewonnen. Zur Zeit wird dieser Austausch üblicherweise durch UDP oder TCP bewerkstelligt, aber keine der beiden Transportprotokolle kann die Anforderung nach einem leistungsfähigen Signalisierungssystem erfüllen. Aus diesem Grund wurde im Rahmen der Internet Engineering Task Force (IETF) ein neues Protokoll zum Transport von Signalisierungsmeldungen über IP-Netze definiert: das Stream Control Transmission Protocol (SCTP). Dieser Beitrag beschreibt SCTP und die Mechanismen, die es für den sicheren und effizienten Transport von Signalisierungsmeldungen den klassischen Protokollen TCP und UDP überlegen machen.

1 Einleitung

Die Dienstgüte, also die Qualitätsmerkmale eines Kommunikationsnetzes aus der Sicht der Benutzer eines betrachteten Dienstes, wird durch die Leistungsfähigkeit des unterliegenden Signalisierungssystems entscheidend beeinflusst. Zur Zeit werden in öffentlichen Telekommukationsnetzen die für die Signalisierung genutzten Meldungen vorwiegend mittels des Signalisierungssystems Nummer 7 übertragen. Bestehende Dienste und Anwendungen nutzen die hohe Effizienz vom SS7, die im Wesentlichen von der guten Fehlerbehandlung in der Sicherungsschicht sowie von den signalisierungsspezifischen Netzmanagement-Prozeduren in der Vermittlungsschicht herrührt. Ein SS7-Signalisierungsnetz stellt allerdings ein logisch eigenständiges Netz dar, welches eine gesonderte Infrastruktur erfordert und nur auf der Bitübertragungsschicht Ressourcen mit dem Nutzdatenverkehr teilt. Den Wunsch nach einem genau so guten und sicheren Protokoll basierend auf IP-Netzen will in der Zukunft SCTP verwirklichen.

2 Eigenschaften von SCTP

Das Stream Control Transmission Protocol ist ein verbindungsorientiertes, nachrichtenorientiertes Protokoll, das Flusssteuerung und Fehlersicherung Ende-zu-Ende durchführt. Es garantiert also die bestätigte, fehlerfreie und nicht duplizierte Übertragung von Nachrichten (z.B. Signalisierungsmeldungen) zwischen genau zwei Endpunkten und wiederholt den Vorgang, falls dieser nicht richtig stattgefunden hat. Die Verbindung ist in SCTP voll-duplex, bleibt erhalten bis alle Daten richtig übertragen wurden und hat letztendlich als Ziel, die gemeinsame Übertragung von Nutz- und Signalisierungsverkehr über ein einheitliches IP-basiertes Kernnetz zu ermöglichen.

2.1 Nachteile von UDP und TCP

Im Vergleich dazu ist UDP auch ein nachrichtenbasiertes Protokoll, welches einen schnellen, verbindungslosen Dienst zur Verfügung stellt. So sind die Meldungen, die schnell übertragen werden müssen, sehr gut bedient, aber UDP bietet keinen zuverlässigen Transportdienst. Deshalb muss die jeweilige Anwendung für die Behebung der Übertragungsfehler, wie z.B. duplizierte Nachrichten, durch Reihenfolgesicherung und wiederholte Übertragung von verlorengegangenen Paketen, selbst sorgen.

Vielmehr kann man SCTP als eine Erweiterung von TCP sehen, oder besser gesagt, SCTP wurde so konstruiert, daß die Schwächen von TCP im Punkt Übertragung von Signalisierungsmeldungen gelöst wurden. Fehlersicherung und Flusssteuerung sind Eigenschaften die auch TCP garantiert, aber TCP ist Bytestrom-orientiert. Das bedeutet, dass keine Nachrichten, sondern ein Bytestrom übertragen wird. Falls eine Anwendung mittels Nachrichten kommunizieren will, muss sie selber Nachrichtengrenzen einfügen (Record-Marking). Die Anwendung muss auch selbst, mittels des Push-Mechanismus, für das sofortige Absenden von vollständigen Nachrichten sorgen. In diesem Byte-Strom liefert TCP dann die Daten in strikter Reihenfolge aus, auch wenn mehrere Signalisierungsverbindungen in eine TCP-Verbindung gemultiplext werden. Viele Anwendungen erfordern aber lediglich eine teilweise Reihenfolgesicherung von Meldungen, z.B. bei Signalisierungsnachrichten, die zum gleichen Anruf gehören. Durch diesen Aufbau von TCP können unnötigerweise Blockierungen bereits angekommener Datenpakete durch fehlende Teile von Meldungen anderer Anrufe oder Transaktionen auftreten, was wiederum unnötige Verzögerung hervorruft. Ein weiterer Nachteil von TCP ist, dass die Verbindung durch ein Paar von sogenannten Sockets (Kombination von IP-Adressen und Portnummern) bestimmt ist. Dadurch ist es nicht möglich, einen Host unter mehr als einer IP-Adresse zu erreichen. Letzteres wird aber aus Gründen der Fehlertoleranz benötigt. Letztes könnte man die Empfindlichkeit von TCP gegen Angriffen, wie z.B. SYN-Angriff, zu den Nachteilen dazuzählen.

2.2 Lösungen von SCTP

SCTP erweitert die Funktionalität von TCP und bietet einen sicheren Nachrichtentransport ohne die Beschränkungen von TCP. So kann SCTP nicht nur für den bestätigten, fehlerfreien und nicht-duplizierten Transfer von Nachrichten garantieren. Durch SCTP erfolgt die Segmentierung von Datenpaketen gemäß festgestellter Maximum Transmission Unit des Pfades (Path-MTU). Die Path-MTU ist die größte Paketlänge, die Ende-zu-Ende ohne Segmentierung übertragen werden kann, was die Leistung des Protokolls optimiert. Parallel dazu wird für die reihenfolgesichere Übertragung gesorgt, und zwar in mehreren unabhängigen SCTP-Streams mit der Option, Daten auch gemäß der Reihenfolge des Eintreffens – und nicht unbedingt in der Reihenfolge des Absendens – zuzustellen. SCTP bietet auch das Multiplexen mehrerer Nachrichten in ein SCTP-Datagramm und bietet mehr Fehlertoleranz auf Netzebene durch Unterstützung von Multi-Homing auf einer oder beiden Seiten einer Assoziation. Jedoch ist das Konzept der Assoziation weiter gefasst als bei TCP. Eine SCTP-Assoziation umfasst die gesamte Kommunikation zwischen zwei Signalisierungspunkten, d.h. zwischen SCTP-Knoten mit möglicherweise mehreren Transportadressen.

In einem Anwendungsszenario für die Übertragung von Signalisierungsmeldungen geschieht der Verbindungsauf- und abbau eher selten. SCTP-Assoziationen werden für gewöhnlich eine – im Vergleich zur Dauer eines Verbindungsaufbaus – sehr lange Lebensdauer haben. Als letztes beinhaltet das Design von SCTP Maßnahmen gegen die in jüngster Vergangenheit bekannt gewordenen „Denial of Service- (DoS-)“Angriffe, die als Ziel haben, die Verfügbarkeit eines Dienstes zu blockieren.

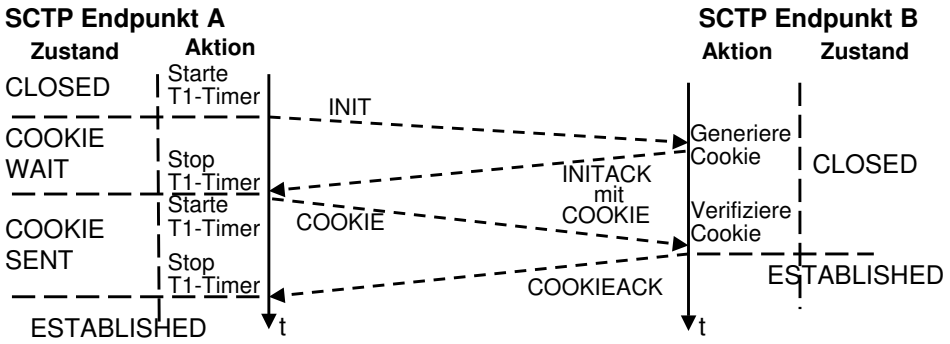


Abbildung 1: Eine SCTP Assoziation.

2.3 Architektur von SCTP

2.3.1 Verbindungsauf- und -abbau

Der Verbindungsaufbau einer SCTP-Assoziation wird mittels der INIT-Primitive von einem Endpunkt eingeleitet. Ähnlich wie bei TCP werden eine Reihe von Zuständen auf beiden Seiten einer Assoziation durchlaufen, bis eine Verbindung aufgebaut ist. Im Gegensatz zu TCP werden bei SCTP jedoch vier Kontrollnachrichten (Four-Way Handshake) ausgetauscht, wobei auf der passiven Seite erst Ressourcen belegt werden, wenn die dritte dieser Meldungen (COOKIE) angekommen ist.

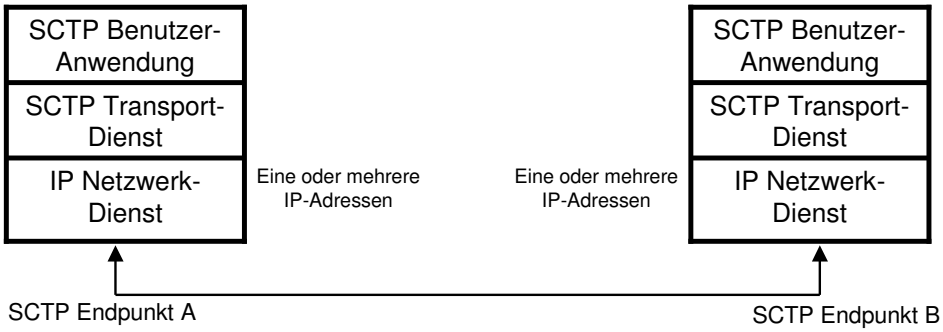


Abbildung 2: Nachrichtenflussdiagramm eines normalen Verbindungsaufbaus.

Bei einem normalen Verbindungsaufbau (vgl. Abb. 2), bekommt der Host, der den Aufbau einleitete, vom Server die Bestätigung des Aufbauwunsches. Diese Bestätigung wird mit einer kryptographischen Checksumme (z.B. MD-5) gesichert und wird als ein sogenanntes Cookie an den Server zurückgeschickt. Das Cookie enthält alle notwendigen Zustandsdaten, die der Server im Folgenden benötigt, um die Assoziation einzurichten. Damit wird weitgehend ausgeschlossen, dass Denial of Service-Angriffe durchgeführt werden, oder dass ein Host mit einer gefälschten IP-Adresse eine Verbindung aufbauen kann (IP-Spoofing). Die aktive Seite durchläuft im Normalfall die Zustände Closed, CookieWait, CookieSent und Established. Beim Verbindungsaufbau darf die jeweils zweite gesendete Nachricht einer jeden Seite bereits Nutzdatenpakete enthalten, die zusammen mit den SCTP-Kontrollmeldungen verschickt werden. Während des Aufbaus werden auch die Listen mit den IP-Adressen, unter denen die zwei Endpunkte zu erreichen sind, ausgetauscht (Multi-Homing).

Wie bei TCP gibt es auch in SCTP zwei Wege, eine Assoziation zu beenden. Den Verbindungsabbau (Shutdown) und den Verbindungsabbruch (Abort). Bei dem normalen Verbindungsabbau meldet der Host dem Server den Abbauwunsch mit der SHUTDOWN-Primitive.

Die Verbindung wird aber nicht sofort abgetrennt, sondern es werden zuerst alle, bis zu dem Punkt verlangten Daten, ganz normal übertragen und erst dann wird die Bestätigung für den Abbau (SHUTDOWN-ACK-Primitiv) zum Host geschickt und die Verbindung getrennt. Im Gegensatz zu TCP unterstützt SCTP kein „half-open“ Zustand, wo eine Seite der Verbindung nach einem Abbauwunsch weiter Daten schickt, während für die andere Seite die Verbindung beendet hat. Eine ABORT-Nachricht, die entweder vom Host geschickt wurde oder wegen einer Störung in der Verbindung entstanden ist, wird vom Server hingegen anders behandelt. In diesem Fall werden keine weiteren Daten geschickt und die Assoziation wird sofort beendet, ohne Rücksicht auf Nachrichten, die verloren gehen könnten.

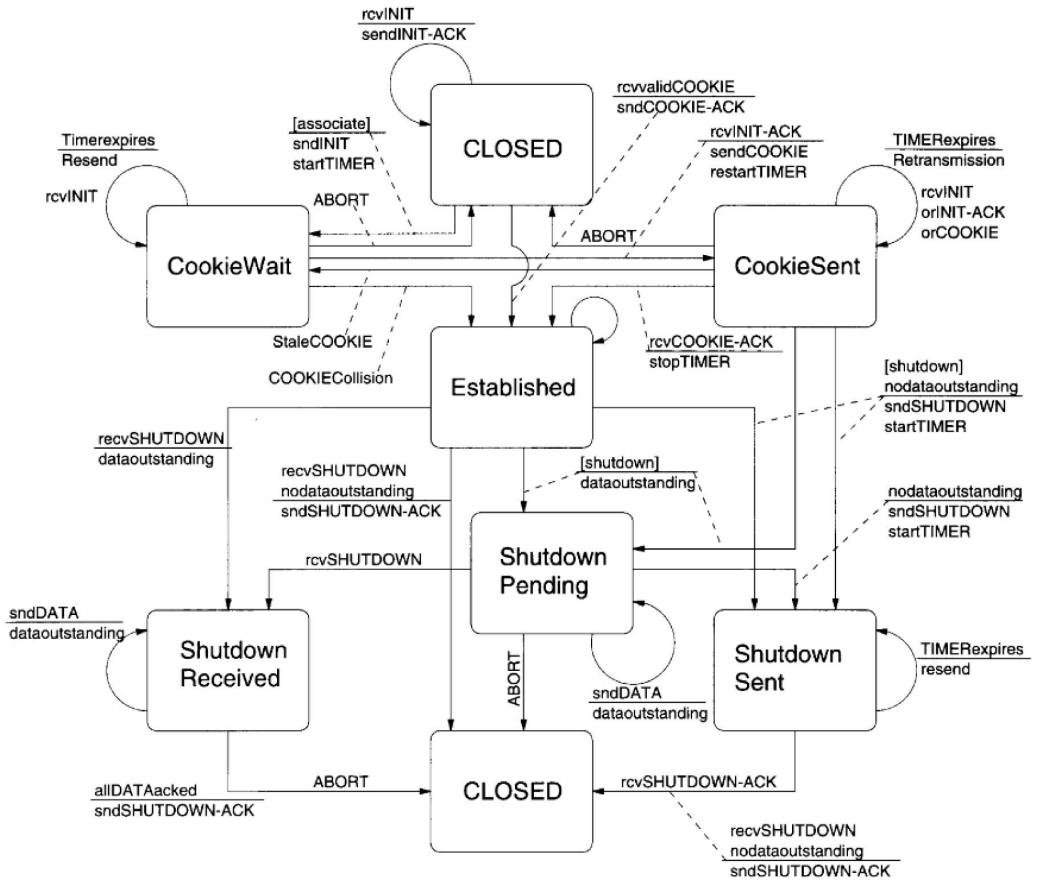


Abbildung 3: Zustandsautomat des SCTP.

Abbildung 3 zeigt alle möglichen Zustände eines SCTP-Endpunkts und alle Primitive, die den Übergang zu den verschiedenen Zustände einleiten.

2.3.2 Reihenfolgesichere Übertragung in Streams

Wie schon erwähnt, ist SCTP ein Protokoll, das mehrere Streams unabhängig voneinander verwalten kann. Die Anwendung, die SCTP benutzt (vgl. Abb. 1 SCTP Benutzer-Anwendung), kann während des Verbindungsaufbaus die Anzahl vom Streams, die von der Assoziation unterstützt werden, mit dem anderen Endpunkt verhandeln. Solange beide Enden sich auf eine feste Anzahl von Streams geeinigt haben, sorgt SCTP dafür, dass die Nachrichten in den voneinander unabhängig Streams, reihenfolgesicher übertragen werden. Das geschieht folgendermaßen: Jede Benutzernachricht wird einem Stream zugewiesen, bekommt die Nummer die diesen Stream eindeutig identifiziert sowie noch eine Sequenznummer welche die Nachricht innerhalb eines Streams eindeutig bezeichnet. So kann der Empfänger die

Nachrichten zuerst dem jeweiligen Stream zuordnen und nach der Reihenfolge überprüfen. Dennoch kann SCTP die anderen Streams weiter übertragen, falls ein Stream blockiert ist, weil eine Nachricht verloren ging,was beim Byte-orientierten TCP nicht der Fall war. Optional kann in SCTP auch die reihenfolgesichere Übertragung in den Streams außer Kraft gesetzt werden.

2.3.3 Daten-Fragmentierung

Die Fragmentierung von Anwenderdaten ist manchmal nötig, damit SCTP sich zu dem jeweiligen, zu Verfügung stehendem Pfad optimal anpassen kann. Deshalb werden die Daten in die schon erwähnten Path-MTU's zerstückelt. Dann werden beim Empfänger die Fragmente wieder zusammengesetzt und zur Anwendung weitergegeben.

2.3.4 Bestätigung vom Empfang und Staukontrolle

SCTP gibt allen fragmentierten und unfragmentierten Anwenderdaten eine bestimmte Nummer, die sogenannte TSN (Transmission Sequence Number), welche unabhängig von der Stream-Sequenznummer ist. Beim Empfänger werden alle TSN's von den Nachrichten die angekommen sind, bestätigt, auch wenn es Lücken in der Reihenfolge gibt. So wird die Funktionalität der Bestätigung garantiert, ohne auf die strenge Reihenfolgesicherung in den Streams aufpassen zu müssen. Die Staukontrolle-Funktion ist dafür zuständig, um Nachrichten die nicht bestätigt wurden, nach einer bestimmten Zeit wiederzusenden ohne dass sie einen Stau verursachen. Diese Funktion soll gleich sein wie bei TCP.

2.3.5 Bündelung von Steuerung- und Anwender-Daten

Das SCTP-Paket besteht aus dem Paketkopf und einem oder mehreren großen Stücken, den sogenannten „Chunks“, bis die Path-MTU-Größe erreicht wird. Die Chunks können Steuerung- oder Anwenderdaten behalten.

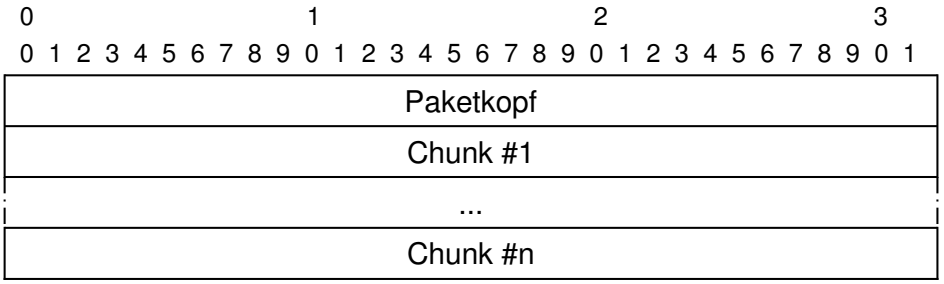


Abbildung 4: Allgemeines Paketformat von SCTP.

SCTP bietet dem Benutzer die Funktionalität an, mehrere Nachrichten in einem Paket zu bündeln und sorgt auch für die Zusammensetzung beim Sender und für die Trennung beim Empfänger. Manchmal, wenn Staufälle auftreten oder wenn Daten wiederzusenden sind, werden die Daten zusammengefasst, auch wenn der Benutzer sich keine Bündelung wünscht. Ein Benutzer kann durch Deaktivierung der Bündelungsfunktion die kleine Verzögerung sparen, die das System braucht um für die optimale Bündelung zu berechnen und zu sorgen.

2.3.6 Gültigkeitsprüfung der Pakete

Im Paketkopf benutzt SCTP zwei Felder zur Gültigkeitsprüfung der Pakete. Das Verifikations-Feld und ein 32-Bit Prüfsummen-Feld (s. Abb. 5).

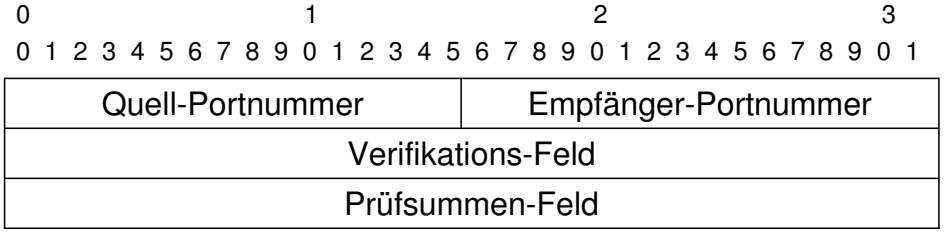


Abbildung 5: Paketkopf von SCTP (Common Header).

Der Verifikationstag wird von den Assoziationsenden beim Verbindungsaufbau gewählt. Alle mit einem falschen Verifikationstag empfangenen Pakete, werden nicht weiter verarbeitet und einfach weggeworfen. Diese Funktion schützt SCTP vor maskierten Angriffen und vor Paketen die von einer vorherigen Assoziation stammen. Mit dem 32-Bit Prüfsummen-Feld (Adler 32 Prüfsummenalgorithmus), was vom jeweiligen Sender gesetzt wird, will SCTP mehr Sicherheit im Punkt Datenverfälschung im Netzwerk anbieten. Auch Pakete, die mit einem ungültigen Prüfsummen-Feld ankommen, werden weggeworfen.

2.3.7 Pfadverwaltung

Beim gewöhnlichen Datenaustausch wird bei der Initialisierung einer Assoziation ein Hauptpfad gewählt durch den die Übertragungen von Nachrichten erfolgen. Wie aber schon erwähnt, werden beim Verbindungsaufbau zwischen den Endpunkten der Assoziation mehrere IP-Adressen ausgetauscht unter denen die Endpunkte erreichbar sind. Die Pfadverwaltungsfunktion sorgt für diesen Austausch und kann während einer Übertragung, falls die Situation es erfordert, z.B. wenn ein Packet wiederholt übertragen werden muss, die Ziel-Adressen von Paketen beliebig ändern. Die IP-Adresse, die gewählt wird, hängt von den Instruktionen, die SCTP von der Anwendung bekommt, und von der Erreichbarkeit des anderen Endes mit dieser bestimmten Adresse ab. Die Erreichbarkeit des Ziel-Endpunkts durch die verschiedenen IP's, wird mit so genannten Heartbeats kontrolliert, falls die anderen Pakete diese Informationen nicht liefern können. Natürlich werden diese nützliche Daten der SCTP-Anwendung mitgeteilt. Mit dieser Multi-Homing Funktion, die SCTP unterstützt, wird eine höhere Überlebenswahrscheinlichkeit der Verbindung garantiert. Ein LAN, das auf dem Hauptpfad liegt, könnte z.B. für eine bestimmte Periode ausfallen und würde bei einer TCP-Verbindung zu erheblichen Verzögerungen führen, bis das IP-Routing-Protokoll einen alternativen Weg findet. Bei SCTP können mehrere redundante LAN's benutzt werden um den lokalen Zugriff zu garantieren und IP-Adressen mit verschiedenen Präfixen können dazu führen, verschiedene Routern zu nutzen, um den Zielpunkt zu erreichen. Beim Empfänger kontrolliert die Pfadverwaltung aus Sicherheitsgründen, ob das Paket zu einer SCTP-Assoziation gehört und falls ja, wird es weiter verarbeitet.

2.4 Paketformat

Wie schon erwähnt, besteht ein SCTP Paket aus dem Paketkopf und einer endlichen Anzahl von Chunks (vgl. Abb. 4). Der Paketkopf enthält die 16-Bit Quell-Portnummer und die 16-Bit Empfänger-Portnummer. Die sind zusammen mit der IP-Adresse für das Zuordnen jedes

ID Value	Chunk Type
0	Payload Data (Data)
1	Initiation (INIT)
2	Initiation Acknowledgement (INIT ACK
3	Selective Acknowledgment (SACK)
4	Heartbeat Request (HEARTBEAT)
5	Heartbeat Acknowledgement (HEARTBEAT ACK)
6	Abort (ABORT)
7	Shutdown (SHUTDOWN)
8	Shutdown Acknowledgement (SHUTDOWN ACK)
9	Operation Error (ERROR)
10	State Cookie (COOKIE ECHO)
11	Cookie Acknowledgement (COOKIE ACK)
12	Reserved for Explicit Congestion Notification Echo (ECNE)
13	Reserved for Congestion Window Reduced (CWR)
14	Shutdown Complete (SHUTDOWN COMPLETE)
15 to 62	reserved by IETF
63	IETF-defined Chunk Extensions
64 to 126	reserved by IETF
127	IETF-defined Chunk Extensions
128 to 190	reserved by IETF
191	IETF-defined Chunk Extensions
192 to 254	reserved by IETF
255	IETF-defined Chunk Extensions

Tabelle 1: Chunk-Typ-Werte

Pakets zu der richtigen Anwendung zuständig. Die Funktionalität von den Verifikations- und Prüfsummen-Felder wurde schon erklärt.

Ein Chunk-Feld besteht aus einem Chunk-Typ-Feld und dem Flag-Feld, dem Chunk-Länge-Feld und letztendlich dem Daten-Feld (vgl. Abb. 6).

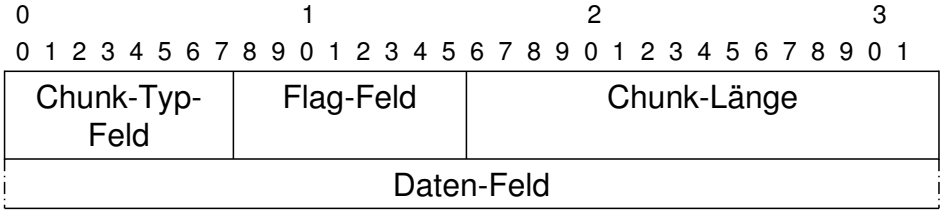


Abbildung 6: Chunk-Feld von SCTP.

Das 8 Bit Chunk-Typ-Feld beschreibt die Art von Informationen die sich im Daten-Feld befinden und kann die Werte von 0 bis 254 annehmen. Der Wert 255 ist für zukünftige Anwendungen als Erweiterung reserviert. Die Werte und was sie bedeuten sind in Tabelle 1 zu finden.

Die Chunk-Typ-Werte sind so codiert, dass falls der Empfänger den Typ nicht identifizieren kann, die zwei höchsten Bits vom Feld bestimmen, wie mit dem Chunk und mit dem gesamten Paket weiter verfahren werden soll. Die vier Möglichkeiten sind folgende:

- 00: Die Verarbeitung des Pakets wird beendet und es werden auch keine andere Chunks in diesem Paket behandelt.

- 01: Wie bei 00 und eswird ein Bericht gemacht, dass der Typ nicht erkannt wurde.
- 10: Dieser Chunk wird übersprungen und es wird mit den anderen Chunks in diesem Paket fortgefahren.
- 11: Wie bei 10 und es wird ein Bericht gemacht, dass der Typ nicht erkannt wurde.

Das Flag-Feld, das auch aus 8 Bit besteht, hängt vom Chunk-Typ ab und wird normalerweise vom Sender-Endpunkt mit Nullen gesetzt und vom Empfänger ignoriert. Die Größe des Chunks wird vom 16-Bit Längen-Feld repräsentiert und beinhaltet das Chunk-Typ-Feld, das Chunk-Typ-Feld, das Chunk-Längen-Feld und das Daten-Feld. Deshalb ist die Länge eines Chunks 4 Bytes falls das Daten-Feld leer ist. Das Daten-Feld ist das Feld, das die eigentlichen Informationen die zu übertragen sind, beinhaltet. Letztendlich muss die Chunk-Länge ein Vielfaches von 4 sein. Falls das nicht der Fall ist, muss der Sender die Bytes die noch fehlen (maximal 3) mit Nullen auffüllen. Diese Nullen zählen beim Längen-Feld nicht mit.

Ein Beispiel eines sehr wichtigen und häufig benutzten Chunk-Typs ist der 0-Typ d. h. Nutzdaten (vgl. Abb. 7). Die interessanten Felder sind in diesem Fall das Längen-Feld, das TSN-Feld,

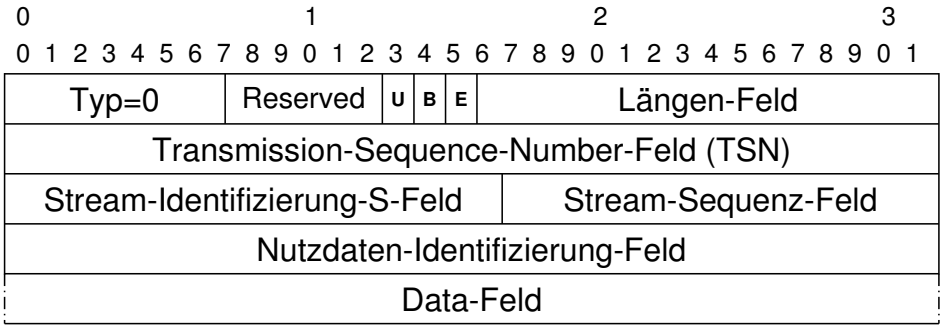


Abbildung 7: Payload Data.

das Stream-Identifizierung-S-Feld, das Stream-Sequenz-Feld, das Nutzdaten-Identifizierung-Feld und das Data-Feld. Das Längen-Feld repräsentiert die Größe des Data-Chunks in Bytes (Minimum in diesem Fall 16 Bytes). Das TSN-Feld behält die schon vorgestellte TSN-Nummer die zwischen 0 und 4294967295 liegen kann. Das Stream-Identifizierung-S-Feld besagt in welchem Stream S diese Anwender-Daten gehören und das Stream-Sequenz-Feld wo genau in diesem Stream S. Das Nutzdaten-Identifizierung-Feld wird von den Anwendungen gesetzt, damit sie den Typ der Informationen in diesem Chunk identifizieren können und wird von SCTP nicht benutzt. Als Letztes folgt das Data-Feld, das die eigentliche Anwender-Daten behält.

Die Chunk-Typen 4 und 5, Heartbeat Request bzw. Heartbeat Acknowlegment, sind sehr wichtig für die Multi-Homing-Funktion und werden hier kurz beschrieben. Damit die Erreichbarkeit der verschiedenen IP-Adressen kontrolliert wird, werden regelmäßig von jedem Endpunkt zum jeweils anderen Heartbeat-Requests geschickt. Das Format von diesen Anfragen wird in Abbildung 8 gezeigt.

Im Heartbeat-Informations-Feld steht wann dieses Paket gesendet wurde und die Ziel-Adresse des anderen Endpunkts. Auf der Empfänger-Seite, wenn so ein Request empfangen wird, muss sofort mit einem Acknowlegment beantwortet werden. Das Format der Bestätigungen ist identisch mit dem der Anfragen und das Heartbeat-Informations-Feld der Anfrage wird sogar einfach in das jeweiligen Feld der Bestätigung kopiert. So ist der Sender der Anfragen, solange er die Bestätigungen bekommt, in der Lage, die vergangene Zeit zu den jeweiligen IP-Adressen zu berechnen und so ist ihm die Erreichbarkeit durch alle diese Adressen jederzeit bekannt.

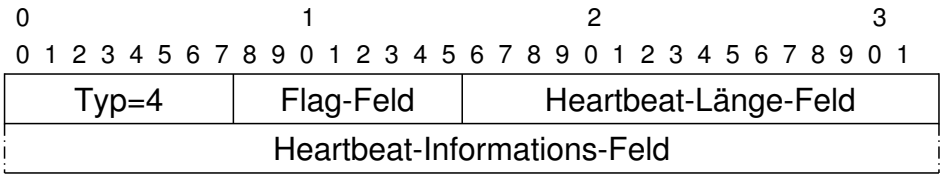


Abbildung 8: Heartbeat-Reaquest Format.

Mit dieser Methode, falls Probleme beim Hauptpfad auftreten, kann SCTP die Pakete mit einer anderen IP-Adresse versehen und so den anderen Endpunkt erreichen.

3 Ausblick

Das neue Protokoll erfüllt die Anforderungen an die Übermittlung von sensitiven Signalisierungsdaten über IP-basierte Netze besser als andere bereits etablierte Protokolle und hat darüber hinaus das Potential, als generisches Transferprotokoll auch in anderen Anwendungsgebieten Fuß zu fassen. Die Frage ob sich der Umstieg von TCP zu SCTP lohnt wird, nach der Betrachtung der vielen Vorteile, die SCTP für zahlreiche Anwendungen bietet, mit einem klaren ja beantwortet. Außerdem gibt es Anwendungen, die sehr viel von SCTP profitieren aber auch andere die vom neuen Protokol weniger begünstigt werden. Es gibt aber keine Anwendung, die etwas zu verlieren hat, falls SCTP sich durchsetzt. SCTP ist zur Zeit ein angehender Internet Standard und Implementierungen sind schon öffentlich verfügbar. Gleichzeitig werden diese Implementierungen von verschiedenen Internet- und Telekommunikationsfirmen weiter entwickelt. Bei diesen Implementierungen wurde aufgepasst, dass die Flusskontrolle von SCTP ähnlich zu der bei TCP ist. Grund hierfür ist die Tatsache, dass SCTP und TCP sich kooperativ verhalten sollen, damit der Umstieg einfacher wird. Allerdings ist noch nicht hinreichend nachgewiesen, dass SCTP unter realen Betriebsbedingungen – d.h. in IP-Netzen, in denen es mit anderen Protokollen um die Netzressourcen konkurriert – wirklich die sehr harten Dienstgüte- und Echtzeitanforderungen erfüllt, die in den SS7-Spezifikationen festgelegt wurden. In diesem Bereich sind weitere Arbeiten notwendig, die einerseits im Labor unter Verwendung der vorhandenen Implementierung und anderseits mit Hilfe simulativer Methoden durchgeführt werden müssen.

Literatur

[Jung00] A. Jungmaier. Stream Control Transmission Protocol for beginners. http://tdrwww.exp-math.uni-essen.de/pages/forschung/sctp_fb/, 2000.

[JuST00a] A. Jungmaier, M. Schopp und M. Tüxen. Das Simple Control Transmission Protocol(SCTP) – ein neues Protokoll zum Transport von Signalisierungsmeldungen über IP-basierte Netze. *Elektrotechnik und Informationstechnik, Zeitschrift des Österreichischen Verbandes für Elektrotechnik* 117(6), 2000.

[JuST00b] A. Jungmaier, M. Schopp und M. Tüxen. Performance Evaluation of the Stream Control Transmission Protocol(SCTP). http://tdrwww.exp-math.uni-essen.de/pages/forschung/atm2000.pdf, Juni 2000. ATM 2000 Conference Presentation, Heidelberg.

[SXMS+00] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang und V. Paxson. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard), Oktober 2000.

[TVCL+01] M. Tüxen, G. Verwimp, L. Coene, J. Loughney, R.R. Stewart, Qiaobing Xie, M. Holdrege, M.C. Belinchon und A. Jungmayer. Stream Control Transmission Protocol Applicability Statement. Internet-Draft draft-ietf-sigtran-sctp-applicability-05.txt, Januar 2001.

Multicast - Empowering the Next-Generation Internet

Anselm Kreuzer

Kurzfassung

Multicast : zwar in aller Munde, aber noch lange nicht richtig einsatzfähig. Obwohl mit IP-Multicast ein erster Multicast-Standard verabschiedet ist und es genügend Andwendungsmöglichkeiten für Gruppenkommunikation-unterstützenden Service gibt, werden immer noch Protokolle entwickelt, Verfahren angepasst, verworfen und erneuert. In dieser Seminararbeit wird die aktuelle IP-Multicast-Technologie vorgestellt sowie deren Mechanismen zur Adressierung, zur Gruppenverwaltung und zum Routing betrachtet. Darauf aufbauend werden mit dem Active Error Recovery-Protokoll und dem Nominee Congestion Avoidance-Protokoll zwei Verfahren präsentiert, welche Multicast skalierbarer und fairer in Bezug auf die Netzauslastung machen (reliable Multicast). Desweiteren werden Ansätze und Überlegungen zu sicherer Multicast-Kommunikation vorgestellt und anhand konkreter Verfahren erörtert.

1 Was ist Multicast?

Multicast ist eine spezielle Kommunikationsform, bei der ein Sender Daten an eine ausgewählte Gruppe von Empfängern übertragt [WiZi99]. Dabei werden alle Empfänger durch eine Adresse, die sogenannte Multicast-Adresse, angesprochen. Der Sender benötigt grundsätzlich keine Informationen über die Größe der Gruppe oder die Standorte der einzelnen Gruppenmitglieder; die Auslieferung der Daten wird von multicast-fähigen Netzzwischensystemen (den Multicast-Routern, im Folgenden MRouter genannt) übernommen. Großer Vorteil von Multicast ist die Einschränkung des Datenaufkommens und die damit verbundene Einsparung von Ressourcen (Bandbreite), da Pakete an die Gruppe vom Sender nur einmal gesendet werden müssen.

Abb.1 zeigt eine durch Unicast simulierte Multicast-Kommunikation. Auch wenn die Empfänger E1 bis E4 zeitgleich dieselben Daten vom Sender beziehen, so wird jedes Datenpaket einzeln an jeden Empfänger adressiert und versendet.

Abb.2 zeigt eine Multicast-Kommunikation. E1 bis E4 bilden hierbei eine Multicast-Gruppe, welche durch eine Multicast-Adresse eindeutig gekennzeichnet ist. Der Sender sendet jedes Datenpaket grundsätzlich nur einmal; die MRouter müssen bei Erhalt einer Multicast-Dateneinheit überprüfen, ob sich in den Teilbäumen unter ihnen evtl. Gruppenmitglieder befinden und gegebenenfalls eine Kopie des Paketes dorthin leiten.

Überlegungen zu der Punkt-zu-Mehrpunkt-Kommunikation werden schon seit 1988 gemacht; doch hat sich Multicast noch nicht auf breiter Basis durchgesetzt, obwohl bereits eine Vielzahl an Anwendungen, die auf eine funktionierende, skalierbare Multicast-Technologie aufsetzen könnte, existiert (Open Distance Learning, Videokonferenzen, Interaktive und Realzeit-Applikationen, Push-Technologien usw.). Gründe hierfür sind die Komplexität und die limitierte Skalierbarkeit (in Bezug auf dynamische Netztopologie, Link-Geschwindingkeiten und Empfängergruppen) der aktuellen Verfahren [DLLK+00]. Im Folgenden wird das „aktuelle“ Multicast-Modell (IP-Multicast) beschrieben und auf seine Mängel und Nachteile eingegangen.

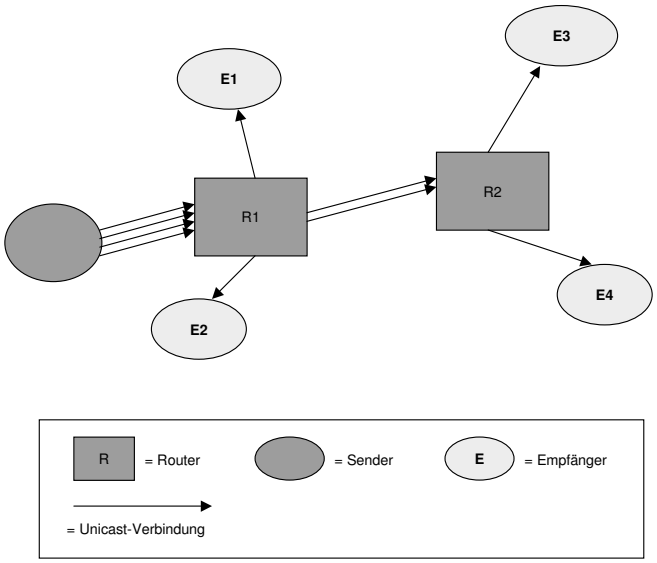


Abbildung 1: Unicast-Kommunikation

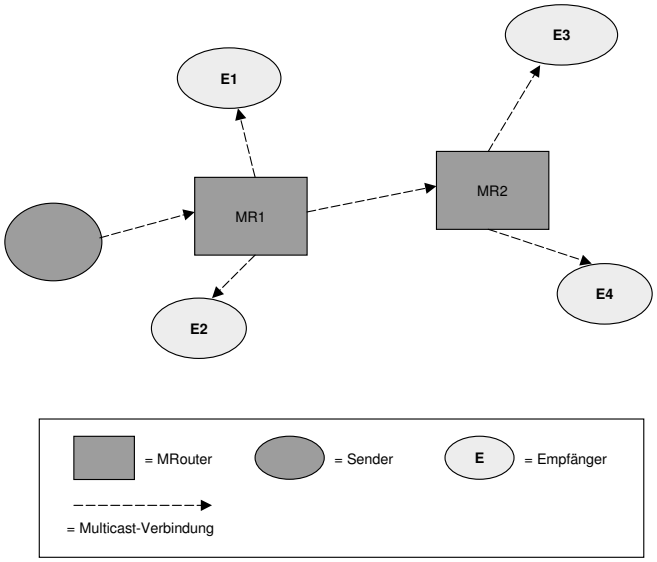


Abbildung 2: Multicast-Kommunikation

2 IP-Multicast

Vorangetrieben durch den Wunsch, Audio- und Videodaten von Konferenzen der Internet Engineering Task Force (IETF) effizient über das Internet zu übertragen, entwickelte Steve Deering von der Stanford University 1988 eine Liste von Erweiterungen des Internet Protokolls (IP), um damit Multicastdaten übertragen zu können (die erste Übertragung wurde im Jahr 1992 realisiert). Diese Sendeform wird IP-Multicast genannt. Das heute aktuelle IP-Multicast, welches als Architektur für das Multicast Backbone on the Internet (MBone), ein weltweites virtuelles Multicast-Forschungsnetzwerk, dient und damit praxisnahe Entwicklung erfährt, hat folgende Charakteristika [Alme00] :

1. offene und dynamische Gruppen : die Empfänger werden zu einer Hostgruppe zusammengefasst, die mit einer eindeutigen IP-Adresse angesprochen wird. Die Mitgliedschaft in einer Hostgruppe ist dynamisch, d. h. einzelne Hosts können jederzeit einer Gruppe beitreten und sie auch wieder verlassen. Gleichzeitig kann ein Host auch Mitglied in

mehreren Gruppen sein. Um an eine Gruppe Daten zu senden, muss der Sender nicht zwingend Mitglied dieser Gruppe sein (offene Gruppe). Mittels IP-Multicast können sowohl permanente als auch transiente Hostgruppen gebildet werden. Im Gegensatz zu transienten Gruppen hängt die Existenz von permanenten Gruppen nicht davon ab, ob die Gruppe momentan Mitglieder hat oder nicht. Diese Gruppen bestehen immer und müssen eine feste Adresse zugewiesen bekommen. Für die Verwaltung der Gruppenzugehörigkeit ist das Internet Group Management Protokoll (IGMP) verantwortlich.

2. IP-basierte Semantik : durch die Versendung der Daten mit IP gelten für die Multicastübertragung alle Eigenschaften, die IP charakterisieren. Es wird ein lediglich verbindungsloser unzuverlässiger best-effort-Dienst bereitgestellt. Von IP wird weder eine Fehlerkontrolle noch eine Flusskontrolle bereitgestellt; diese Kontrollmechanismen müssen von überliegenden Schichten übernommen werden (siehe Abschnitt 3).

2.1 IP-Multicast-Adressierung

IP verwendet 4 Klassen zur Adressierung (Klasse A bis D). Für die Adressierung von Multicastgruppen werden die Klasse D-Adressen verwendet. Ein Teil der Adressen wird von der Internet Assigned Numbers Authority (IANA) auf Antrag an permanente Gruppen zugewiesen, die wie eine Telefonnummer ständig an diese Gruppen gebunden sind (Beispiele für dauerhafte Gruppen sind die Gruppe aller Hosts innerhalb eines Subnetzes (224.0.0.1) und die Gruppe aller MRouter innerhalb eines Subnetzes (224.0.0.2) - diese beiden Gruppen dienen nur dem Nachrichtenaustausch. Die Konferenzen der IETF werden beispielsweise über die Adressen 224.0.1.11 und 224.0.1.12 übertragen. Beide zusammen bilden einen Kanal, wobei die erste Gruppe für die Audioübertragung und die zweite Gruppe für die Videoübertragung vorgesehen ist). Nicht an permanente Gruppen vergebene Adressen werden dynamisch verwaltet, d. h. eine neue Gruppe wählt zufällig eine Adresse aus. Um zu testen, ob diese Adresse bereits von einer anderen Gruppe verwendet wird, wird eine Kontrollnachricht an diese Adresse gesendet. Falls keine Antwort erfolgt, wird davon ausgegangen, dass die Adresse unbenutzt ist [WiZi99].

2.2 Das Internet Group Management Protocol (IGMP)

Zur Erweiterung des Funktionsumfanges sind in IP-Multicast weitere Protokolle eingebettet. Beispielsweise dient das Internet Control Message Protokoll (ICMP) der Versendung von Kontrollnachrichten für die Netzwerksteuerung bei unerwarteten Ereignissen. Hier sind etwa ein Dutzend Nachrichtentypen definiert; z.B. die Nachricht „Time Exceeded“, welche generiert wird, sobald das TTL-Feld eines Paketes den Wert Null hat (ein Anzeichen dafür, dass entweder Pakete kreisen oder eine Überlastung vorliegt). ICMP-Nachrichten werden in IP-Paketen verkapselt [Tane00].

Auf der gleichen Ebene ist das Internet Group Management Protocol (IGMP) angesiedelt. Es ermöglicht einem MRouter, Gruppenzugehörigkeiten einzelner Subnetze und Rechner abzufragen, um gegebenenfalls Multicast-Dateneinheiten an diese weiterzuleiten. Dabei senden die MRouter periodisch sogenannte Host Membership Querys an die Gruppe aller Hosts im Subnetz, wobei der TTL-Wert im IP-Header auf den Wert 1 gesetzt ist (so kann die Query das Subnetz nicht verlassen). Nach Erhalt dieser Query antwortet jeder Host für jede Gruppe, in der er Mitglied ist, mit einem Host Membership Report; er gibt also dem MRouter bekannt, dass er Mitglied dieser Gruppe ist. Zur Vermeidung redundanter Report-Einheiten verzögert jeder Host die Antwort um eine zufällige Zeit. Antwortet währenddessen ein anderes Mitglied, so unterdrücken die restlichen Hosts im Subnetz ihren Report. In den Versionen 2 (aktuell) und 3 wurde IGMP um einige nützliche Eigenschaften erweitert : unter anderem

um eine Funktion, die einem Host ermöglicht, seinem übergeordneten MRouter das Verlassen einer Gruppe anzuzeigen; außerdem kann der Empfang von Multicastdaten nicht nur auf eine bestimmte Gruppe eingeschränkt werden, sondern man kann nun auch innerhalb einer Gruppe einen Sender spezifizieren, von dem man ausschließlich Daten dieser Gruppe empfangen möchte. [DLLK+00].

2.3 IP-Multicast Routing

Wie schon erwähnt, sind auch bei Multicast-Gruppenkommunikation die Netzzwischensysteme (MRouter) für das Weiterleiten der Pakete an die einzelnen Empfänger verantwortlich. Die für das Routing eingesetzen Verfahren müssen auf jeden Fall genau, einfach, robust und stabil arbeiten. Die meisten in der Praxis eingesetzten Multicast-Routingverfahren arbeiten dynamisch und basieren entweder auf Distanz-Vektor-Routing oder Link-State-Routing. Da es eine große Anzahl von Multicast-Routingverfahren gibt und deren detaillierte Betrachtung den Rahmen dieser Seminararbeit sprengen würde, wird hier nur das Distanz Vektor Multicast Routing Protokoll (DVMRP), welches das verbreitetste Multicast-Routingprotokoll für intradomänes Multicast-Routing im Internet, vorgestellt.

DVRMP ist ein Distanz-Vektor-Protokoll, d.h. jedes System kennt die Distanz zu allen anderen MRoutern in seiner Domäne; Informationen über Distanzen und Distanzänderungen werden periodisch über sogenannte Distanzvektoren ausgetauscht. In der aktuellen Version 3.5 verwendet DVMRP den effizienten Reverse-Path Multicasting-Algorithmus, welcher für jede Gruppe und für jeden Sender innerhalb der Gruppe einen eigenen Multicast-Baum aufspannt. Viele MRouter unterstützen aber nur frühere Versionen von DVMRP, die mittels Truncated Reverse Path Broadcasting (TRPB) die Datenpakete durchs Internet leiten. Damit wird keine Rücksicht auf den Bedarf der Daten genommen und evtl. Multicast-Pakete verbreitet, die eigentlich niemand benötigt. Eine Möglichkeit, die Reichweite der Datenpakete einzuschränken, wird durch das Setzen des Time-to-Live-Parameters (TTL) des IP-Headers gegeben. Der Sender initialisiert die Pakete mit einem bestimmten Wert, der von jedem MRouter vor dem Weiterleiten um eins erniedrigt wird. Ist der TTL-Wert bei Null angekommen, wird das Paket verworfen. Um nun die Pakete innerhalb bestimmter Regionen zu halten, werden die Ausgangsleitungen eines MRouters, die zu anderen Kontinenten oder anderen Subnetzen führen, mit Grenzwerten belegt. Es werden also nur Pakete mit einem TTL-Wert über diese Leitung weitergeleitet, der höher als dieser Grenzwert liegt (initialisierter TTL-Wert 1 bedeutet z.B. auf gleiches Subnetz beschränkt, TTL 64 auf gleiche Region, TTL 128 auf gleichen Kontinent) [SaMu00].

DVMRP-Router nutzen zwei unabhängige Routingprotokolle, z.B. RIP oder OSPF für Unicast-Routing und DVMRP für Multicast. Im Internet sind aber heutzutage die wenigsten Router multicastfähig, so dass häufig Daten von einem MRouter zum anderen über einen oder mehrere nicht multicastfähige Router geleitet werden müssen. Ohne Veränderung der Pakete ist dies nicht möglich, weil ein „normaler“ Router Datenpakete mit einer Klasse D Adresse als Empfänger sofort verwerfen würde. Um Knoten aus nicht multicastfähigen Routern zu überwinden, werden sogenannte Tunnel eingesetzt. Ein Tunnel ist eine virtuelle Verbindung zwischen zwei MRoutern, welche das Senden von Multicast-Datenpaketen über nicht multicastfähige Router ermöglicht. Falls ein MRouter ein Datenpaket mit einer Gruppenadresse empfängt und es auf einer Leitung weitergeben möchte, auf der als nächstes ein Unicast-Router liegt, erweitert er den IP-Header um ein Optionsfeld, in das er die ursprüngliche Quell- und Zieladresse speichert. Die eigene Adresse wird als Quelladresse eingetragen und mit dem auf der betroffenen Leitung nächstgelegenen MRouter als Empfänger adressiert. Durch diesen Austausch der Adressen ist das Paket ein Unicast Paket geworden, das jeder Router versteht und zum richtigen MRouter weiterleiten kann. Dieser führt einen Rücktausch

der Adressen durch, womit der Urzustand des Pakets wieder hergestellt ist und die kritische Strecke überwunden wurde.

Neben DVMRP werden für intradomänes Multicasting in einigen MRoutern auch die Protokolle MOSPF und PIM (Sparse-Mode) eingesetzt. MOSPF steht für das um Multicastfunktionalität erweiterte Open Shortest Path First Protokoll (OSPF). Protocol Independent Multicast (PIM) ist momentan noch in der Entwicklungsphase der IETF. Für aufkommendes hierarchisches Routing (hier wird versucht, das Internet in mehrere Regionen aufzuteilen; Routing-Informationen müssen dann nur noch für die eigene Region gespeichert werden. Das Routing zwischen den einzelnen Regionen wird auf einer höheren Ebene von sogenannten Grenzroutern übernommen. Vorteile sind kleinere Routingtabellen und die Möglichkeit, in verschiedenen Regionen unterschiedliche Routingprotokolle einzusetzen - der Testeinsatz eines neu entwickelten Protokolls wird somit möglich, ohne Auswirkungen auf andere Regionen in Kauf nehmen zu müssen) wird das Multicast Source Discovery Protokoll eingesetzt; Verfahren wie das Border Gateway Multicast Protokoll (BGMP) befinden sich noch in der Entwicklungsphase [SaMu00].

Das momentan größte Problem im Internet stellt die stetig wachsende Anzahl an MRoutern und der damit verbundene Anstieg des Verwaltungsaufwandes dar. Bei einer Anzahl von mittlerweile über 1700 MRoutern (Stand von 1999) werden die Routingtabellen immens groß. Dies erhöht den Datenverkehr für den Austausch von Routinginformationen und verlängert die Bearbeitungszeit der Tabellen. Zu erwarten ist auch, dass die Speicherkapazität für Routing Tabellen nicht mehr lange ausreicht. Lösungsansätze hierfür sind das hierarchische (interdomänes) Multicast Routing und die neue IP-Version 6. Hierarchisches Routing befindet sich noch in der Entwicklungsphase; ein domänenübergreifender Einführungszeitpunkt von IPv6 ist noch nicht festgelegt (in einzelnen Domänen wird IPv6 teilweise schon eingesetzt). Durch die Einführung von IPv6 entfällt die Unterscheidung von MRoutern und Unicast-Routern; Router und Host unterstüzen dann sowohl Unicast- als auch Multicast-Verbindungen standardmäßig. Desweiteren wird der Funktionsumfang von IGMP in ICMP integriert und durch die Erweiterung der Adresslänge auf 128 Bit stehen für zukünftige Anforderungen genügend Adressen zur Verfügung.

3 Reliable Multicast

3.1 Einführung

Bei einer großen Anzahl von Anwendungen macht der Einsatz von Multicast Sinn - viele dieser Anwendungen haben aber an die unter ihnen liegenden Schichten weitergehende Anforderungen, welche von IP-Multicast nicht unterstützt werden. IP-Multicast basiert auf dem User Datagramm Protokoll (UDP) und bietet damit nur verbindungslose, unzuverlässige Paketübertragung ohne gesicherte Übertragungsrate an. So kann es geschehen, dass einige Empfänger unbemerkt nicht alle gesendeten Pakete erhalten. Außerdem können Multicast-Übertragungen mit unkontrollierten Datenraten Netzressourcen überlasten und so Stau verursachen [KBKK$^+$00]. Im folgenden werden Ansätze aufgezeigt, die das bisher abgehandelte Multicast skalierbarer, zuverlässiger und fairer in Bezug auf die Netzauslastung machen, in dem sie Lösungen für Probleme wie Feedback-Imposion, Behandlung von wiederholter Übertragung und Staukontrolle bieten. Grundsätzlich ist die Definiton von Zuverlässigkeit im Bereich von Multicast-Anwendungen ein schwammiger Begriff (es gibt mehere Zuverlässigkeitsstufen; hier wird im folgenden nur die totale Zuverlässigkeit -alle Pakete müssen bei allen Empfängern ankommen- betrachtet), welches sich auch in der Vielzahl und der Unterschiedlichkeit der entwickelten Protokolle zeigt.

Es werden zwei neue Verfahren vorgestellt, welche in Zusammenarbeit zuverlässiges Multicast bieten : Das Active Error Recovery Protocol (AER) ist ein Protokoll für das Handling von Übertragungswiederholungen; das Nominee Congestion Avoidance Protocol (NCA) kümmert sich um Fluss- und Staukontrolle. Beiden Verfahren ist gemein, dass sie auf aktive Netzzwischensysteme zurückgreifen (sogenannte aktive Netze, deren Knotenpunkte auch andere Aufgaben als das Routing übernehmen). Vor den Funktionsbeschreibungen dieser Protokolle zunächst noch die Klärung einiger wichtiger Begriffe :

3.1.1 Vermeidung von Feedback-Implosion

Um analog zu dem Unicast-Transportprokoll TCP auch bei Multicast-Verbindungen gewährleisten zu können, dass alle Pakete vollständig und in richtiger Reihenfolge beim Empfänger angekommen sind, ist es unabdingbar, dass die Empfänger dem Sender Feedback-Informationen über den Erhalt der Daten zukommen lassen. Dieses können sie entweder durch ACKs (acknowledgements - es wird der korrekte Empfang eines Paketes bestätigt) oder durch NACKs (negative acknowledgements - es wird ein nicht erhaltenes Paket gemeldet) tun. Das nachfolgend beschriebene AER arbeitet mit NACKs: zum einen, weil hier weniger Datenverkehr zu erwarten ist; zum anderen, weil die Aufgabe der Fehlererkennung hier an die Empfänger übergeht. Sowohl bei sender-basierter (ACK) als auch bei empfänger-basierter Fehlererkennung (NACK) können bei schlechtem Durchsatz und hoher Paketverlustwahrscheinlichkeit Feedback-Implosionen auftreten; diese gefährden die Performance des Senders und müssen auf jeden Fall vermieden werden.

3.1.2 Retransmission Scoping

Enthält ein Empfänger ein Paket nicht, so muss in einer zuverlässigen Multicast-Umgebung dafür gesorgt werden, dass eine wiederholte Übertragung dieses Paketes an den entsprechenden Empfänger stattfindet. Bei großen Gruppen und hoher Netzauslastung ist es leicht möglich, dass oft mehrere identische Wiederholungen an verschiedene Empfänger anstehen; hier sind einzelne Unicast-Wiederholungen nicht mehr effektiv. Wird man aber entsprechende Pakete per Multicast versenden, so werden diese auch von Gruppenmitgliedern empfangen, welche die wiederholte Übertragung nicht angefordert haben - auch dieses erzeugt unnötige Netzauslastung. Durch Retransmission Scoping wird nun versucht, Wiederholungen so effektiv wie möglich zu versenden (am besten so, dass nur diejenigen Gruppenmitglieder erreicht werden, die vom Paketverlust betroffen sind).

3.1.3 Ausgangsort der wiederholten Übertragung

Stehen oft wiederholte Übertragungen an, so können beim Sender und in seiner Nähe schnell Flaschenhals-Situationen entstehen (durch Latenz und Beanspruchung von Bandbreite). Deshalb wurden verschiedene Konzepte entwickelt, wiederholte Übertragungen nicht allein dem ursprünglichen Sender zuzuweisen, sondern diese Aufgabe auf das gesamte Netz zu verteilen - zum Beispiel durch mit Cache (Zwischenspeicher) ausgestattete Repair-Server, welche an verschiedende Netzknotenpunkte (MRouter) gekoppelt sind und beim Erhalten eines NACKs dieses nicht an den ursprünglichen Sender weiterleiten, sondern dem Empfänger das angezeigte Paket selbst aus ihrem Cache zukommen lassen.

3.1.4 Fluss- und Staukontrolle

Bei einer senderbasierten Fluss- und Staukontrolle erhält der Sender durch die ACKs/NACKs Informationen über den Zustand des Netzes und trifft davon abhängig Entscheidungen über

sein weiteres Sendeverhalten. Weitergehend soll aber auch darauf darauf geachtet werden, dass sich Multicast-Verbindungen die verfügbare Bandbreite fair mit Unicast-Verbindungen teilen. TCP ist und bleibt das mit großem Abstand gebräuchlichste Transportprotokoll - eine Multicast-Lösung, welche die zu Verfügung stehende Bandbreite nicht mindestens fair mit TCP-Verbindungen teilt, wäre höchst inakzeptabel.

3.2 AER - Das Active Error Recovery Protocol

Das Packet Loss Recovery-Protokoll AER bleibt inaktiv, solange alle Pakete von allen Gruppenmitgliedern korrekt empfangen werden (einzelne Bitfehler können von der darunterliegenden Schicht des IP-Multicast korrigiert werden). Sobald irgendwo im Multicast-Baum ein Paketverlust auftritt, sendet der betroffene Empfänger (oder Repair-Server) ein NACK per Unicast-Verbindung an seinen übergestellten Repair-Server (dieses muss nicht automatisch der MRouter auf dem rückwertigem Multicast-Pfad sein; da auch Router ohne angeschlossene Repair-Server im Multicast-Baum sein können oder die Route asymmetrisch sein kann. AER benutzt ein eigenes Signalisierungssystem, um jedem Gruppenmitglied eindeutig einen Repair-Server zuzuordnen). Hat dieser Repair-Server das gewünschte Paket zwischengespeichert, so sendet er dieses per Multicast an die Gruppenmitglieder, welche im Multicast-Pfad unter dem ihm angeschlossenen MRouter liegen (von mindestens einem dieser Empfänger kommt das NACK). Hat er das gewünschte Paket nicht zwischengespeichert, so merkt er sich dessen Kennung und sendet das NACK gleichauf weiter an seinen Repair-Server. Irgendwann landet dieses NACK bei einem Repair-Server, der das betreffende Paket aus dem Cache holen kann (oder gar dem Sender selbst), und dieses wird nochmal per Multicast „abwärts“ gesendet. Dabei senden aber nur die MRouter der Repair-Server, die den betreffenden Paketverlust gespeichert haben, dieses Paket weiter - so wird unnötige Resourcenbelegung vermieden. Der Signalisierungsmechanismus von AER ist dabei nicht nur für die Lokalisierung der Repair-Server zuständig, sondern behandelt auch Routenänderungen, asymmetrische Routen und den dynamischen Eintritt/Austritt von Repair-Servern.

Von den MRoutern im Multicast-Baum wird erwartet, dass sie zwischen gewöhnlichen Paketen (welche einfach nur weiterzuvermitteln sind) und NACKs, Repairs und Meldungen des Signalisierungssystems unterscheiden können. Deshalb werden bei besonderen Paketen bestimmte Flags im Header der IP-Pakete (IP options) gesetzt; so können MRouter erkennen, ob ein bestimmtes Paket für den Repair Server bestimmt ist (sogenannter Router Alert), oder ob es sich um ein wiederholt übertragenes Paket handelt, welches besondere Aufmerksamkeit verdient (sog. Repair NACK).

Verschiedene Untersuchungen und Testläufe im ABone (Active Backbone Network, ein Testnetz des MIT) haben ergeben, das AER entscheidende Verbesserungen im Bereich der Netzauslastung erwirkt (je nach Anzahl Gruppenmitglieder und Paketverlustrate 24-72 Prozent); auch erreichen Paketwiederholungen den Empfänger nun deutlich schneller, als wenn dieser die Paketwiederholung direkt beim Sender anfordern würde. AER ist noch in der Versuchsphase und wird zur Zeit auf anderen Plattformen getestet; sein Einsatz in näherer Zukunft ist aufgrund der benötigten aktiven Netze nicht wahrscheinlich [KBKK+00].

3.3 Flusskontrolle mit dem Nominee Congestion Avoidance Protocol (NCA)

NCA wurde von derselben Forschungsgruppe, die auch AER entworfen hat, vorgestellt. Es arbeitet gut mit AER zusammen; kann aber auch mit anderen Packet-Loss-Recovery-Protokollen betrieben werden. Zentrales Ziel bei der Entwicklung war die Vermeidung der Überlastung einzelner Empfänger durch einen schnellen Sender, gleichzeitig soll der durch

Multicast-Kommunikation entstehende Paketverkehr kontrollierbar und damit skalierbar in Bezug auf TCP-Verbindungen, welche über die gleichen physischen Kanäle laufen, sein.

NCA richtet sein Übertragungsverhalten nach der Paketverlustrate eines einzelnen Gruppenmitgliedes, dem sogenannten Nominee, aus. Dieser Nominee sendet dabei einzelne Paketbestätigungen per Unicast an den NCA-Sender. Der Nominee ist dasjenige Gruppenmitglied, welches am Ende desjenigen Pfades sitzt, der TCP-Sessions die wenigste Bandbreite zugesteht (sogenannter worst path). Die Bestimmung des Nominees läuft wie folgt ab : jeder Empfänger schätzt seine Paket-Verlustwahrscheinlichkeit p ab und sendet diese zusammen mit der taxierten Round-Trip-Time T (die Zeit, die ein Paket vom Sender bis zum Enpfänger unterwegs war) in einer Congestion Status Message (CSM) an seinen Repair-Server. Dieser Repair-Server berechnet aus allen erhaltenen CSMs diejenige mit dem größten Wert g, wobei sich wie folgt berechnet : $g(p, T) = T * \sqrt{p}$. Diese CSM kommt vom worst path und wird per Unicast an den nächsthöheren Repair-Server gesendet. Die CSMs mit dem größten Werten g landen dann beim Sender, welcher analog das Gruppenmitglied am Ende des worst path bestimmt. Dieses wird per Unicast zum Nominee erklärt und hat nun paketweise Bestätigungen (ACKs) an den Sender zu leisten. Diese Prozedur wird in bestimmten Zeitabständen wiederholt, so dass der Nominee auch wechseln kann.

Der Algorithmus, der nun abhängig von den eintreffenden ACKs des Nominees die Senderate steuert, benutzt ein Slow-Start/Sliding Windows-Verfahren ähnlich der TCP-New-Reno. Mit einem Steuerfenster der Größe W und eine Slow-Start-Schwellenwert-Variable t verfährt der Algorithmus wie folgt :

- Bei Erhalt eines ACKs : if (W < t) setze $W := W + 1$; sonst $W := W + 1/W$; (Anstieg der Fenstergröße verläuft bestenfalls linear)

- Bei Auftreten eines Paketverlustes : setze $t := W/2$ und $W := W/2$; (Halbierung des Schwellenwertes und der Fenstergröße zur Stauvermeidung)

- Bei einem Timeout : setze $t := W/2$ und $W := 1$; (neuer Slow-Start)

Ein Paketverlust wir dann angenommen, wenn der Sender ein Paket nicht, aber die drei darauffolgenden Pakete bestätigt bekommt. Ein Timeout tritt dann ein, wenn der Sender innerhalb einer bestimmten Zeit überhaupt keine ACKs des Nominees erhält. Zusätzlich hat NCA noch eine Sicherheitsfunktion, die NACKs bei einem auftretendem Burst einfach ignoriert, um die Leistung des Senders sicherzustellen. Verschiedene Simualtionen in unterschiedlichen Szenarien haben ergeben, dass sich unter Einsatz von NCA einerseits schnell ein effektiver Multicast-Datenvekehr einstellt, anderseits wird aber auch die verfügbare Bandbreite mit TCP-Sessions, welche über dieselben physischen Kanäle laufen, fair geteilt.

3.4 Fazit zu Reliable Multicast

Wesentliches Problem bei der Entwicklung eines zuverlässigen Multicast-Protokolls ist neben der Implementierung eines zuverlässigen Dienstes auch die Skalierbarkeit der Protokolle. Kritisch anzumerken ist, dass Multicast-Routingprotokolle allgemein noch schlecht skalieren und fehlerhaft arbeiten, sobald die Größe des Netzes (im Bereich der Weitverkehrsnetze) zunimmt. Es besteht also durchaus noch Entwicklungsbedarf an zuverlässigen Multicast-Protokollen, die in Weiterverkehrsnetzen gut skalieren und dennoch einen zuverlässigen Dienst anbieten.

4 Multicast und Sicherheit

4.1 Einführung

Neben Zuverlässigkeit spielt auch die „Sicherheit“ in der Gruppenkommunikation und in der Entwicklung von zukünftigen Multicast-Anwendungen eine Schlüsselrolle. Die meisten Protokolle für zuverlässiges Multicast sind zwar robust gegen gewöhnliche Fehler wie Paketverlust oder plötzliches Ausscheiden eines oder mehrerer Gruppenmitglieder, sind aber nicht abhörsicher und bieten keinen Schutz vor Angriffen (und sind damit nicht für vertrauliche oder geheime Kommunikation geeignet). Sicherheit bedeutet die Gewährleistung, dass nur authifizierte Mitglieder einer Gruppe Nachrichten senden/empfangen dürfen (Authentizität) und das diese Daten nicht von unbefugten Dritten gelesen oder gar verändert werden dürfen (Intregrität). Die folgenden Abschnitte beschäftigen sich mit einigen grundsätzlichen Überlegungen zur Integriät und Authentizität, desweiteren werden einige Verfahren zur Wahrung von Integriät und Authentizität in Bezug auf Gruppenkommunikation vorgestellt.

4.2 Integrität durch verschlüsselte Kommunikation/Schlüsselmanagement innerhalb von dynamischen Gruppen

Will man die Integrität einer Gruppenkommunikation wahren, müssen die Dateneinheiten auf ihrem Weg durch das Netz verschlüsselt werden. Gruppenkommunikation mit Hilfe von kryptografischen Verfahren erfordert aufgrund der Vielzahl der Teilnehmer ein effizientes System zur Schlüsselverwaltung. Bei einer sicheren Unicast-Verbindung wird am Anfang einer Sitzung mittels eines Schlüsselübermittlungsverfahrens der Sitzungsschlüssel festgelegt und für die Dauer der gesamten Sitzung verwendet. Nach Beenden der Sitzung wird dieser Schlüssel dann verworfen [Wobs98]. Wenn man dieses Schema auf eine dynamische Multicast-Gruppe übertragen will, so ergeben sich eine Reihe von neuen Problemen : sobald ein Mitglied die Gruppe verlässt/ein neues Mitglied hinzukommt, muss ein neuer Gruppenschlüssel festgelegt und an alle Mitglieder der Gruppe weitergegeben werden - andernfalls könnte ein Nichtmitglied weiter an der Gruppenkommunikation partizipieren oder ein neues Mitglied (unbefugt) alte Daten anfordern (sogenannte „Join“- bzw. „Leave“-Secrecy). Ein zentrales, naives Schlüsselmanagement müsste bei jedem Join oder Leave eines Mitgliedes n Nachrichten an n Gruppenmitglieder versenden; es werden im folgenden auch drei weitere Verfahren vorgestellt, welche mit weitaus weniger Nachrichten zurechtkommen und deshalb bessere Skalierbarkeit für dynamische Gruppen bieten.

4.2.1 Das naive Schlüsselmanagement

Das naive Schlüsselmanagement basiert auf einer zweischichtigen Architektur mit einer zentralen Kontrolleinheit und vielen Gruppenmitgliedern [MoRR00]. Jedes neue Gruppenmitglied bekommt dabei 2 Schlüssel zugewiesen - den Gruppenschlüssel zum Teilnehmen an der Gruppenkommunikation und einen privaten Schlüssel, den außer dem jeweiligen Client nur die Kontrollinstanz kennt. Bei jedem Hinzukommen oder Verlassen eines Gruppenmitgliedes generiert die Kontrolleinheit einen neuen Gruppenschlüssel und sendet diesen jedem einzelnen Client per Unicast-Verbindung zu, wobei der Gruppenschlüssel vorher mit dem privaten Schlüssel des jeweiligen Clients verschlüsselt wird.

Sowohl Vorteile als auch Nachteile dieser unkomplizierten Methode liegen auf der Hand : Sie ist einerseits einfach zu implementieren, garantiert Join- und Leave-Secrecy und stellt keine Bedingungungen an die darunterliegende Architektur. Andererseits weist sie schlechte Skalierbarbeit in Bezug auf die Anzahl der Gruppenmitglieder und die Dynamik der Gruppe auf :

sowohl der Nachrichtenverkehr als auch der Arbeitsaufwand der Kontrolleinheit (Generierung und Speicherung der Schlüssel, Nachrichtenerzeugung und -versand) wachsen linear mit der Größe der Gruppe. Daher ist das naive Schlüsselmanagement allenfalls als Lösung für sehr kleine und wenig dynamische Gruppen interessant.

4.2.2 Das Tree-Based-Schüsselmanagement

Im Juni 1999 wurde nachfolgendes Verfahren zu der Schlüsselmanagement-Problematik von Mitarbeitern der NSA (National Security Agency) veröffentlicht [MoRR00]. Das grundsätzliche Merkmal dieses Verfahrens ist die Verwendung von zusätzlichen Hilfsschlüsseln, welche hierarchisch in den Knoten eines logischen k-fachen Baumes (ein Baum, dessen Wurzel k Söhne hat, welche jeweils wiederum k Söhne haben usw. - ein Binärbaum ist ein Spezialfall eines k-fachen Baumes) abgelegt sind. Dieser Baum ist in seinem ganzen Umfang nur der zentralen Kontrolleinheit bekannt.

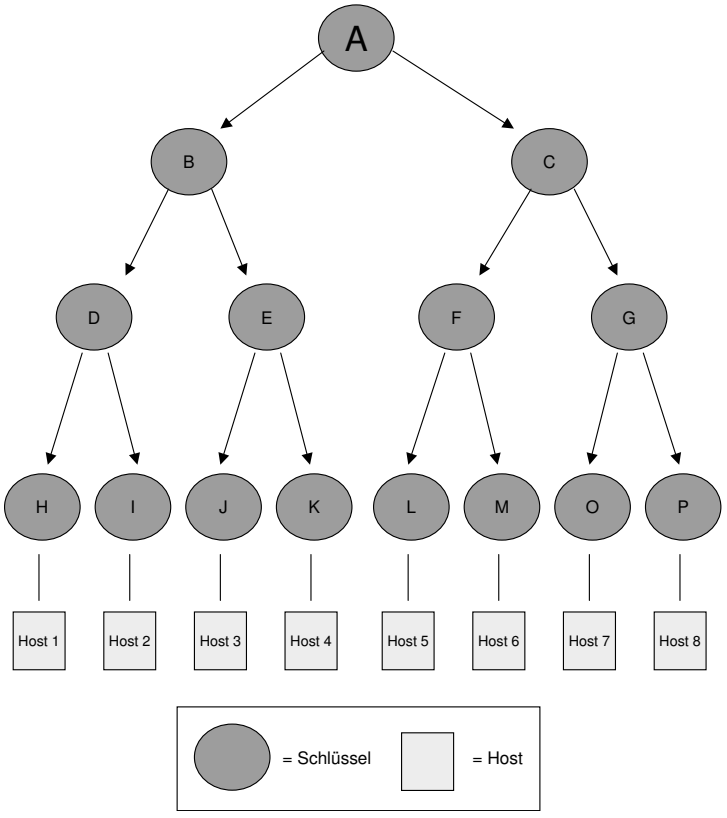


Abbildung 3: Tree-Based-Schlüsselmanagement

In Abb.3 gibt es acht Gruppenmitglieder. Die Knoten A bis P entsprechen keinen Hosts, sondern dienen - wie bereits erwähnt - nur der Schlüsselverteilung. In den Endknoten des Baumes sind die jeweiligen privaten Schlüssel der einzelnen Mitglieder gespeichert, welche bei Initialisierung per Unicast-Verbindung von der zentralen Kontrolleinheit übermittelt werden. Jedem Gruppenmitglied sind dabei alle Schlüssel der Knoten, welche auf dem Pfad von dem ihm übergeordneten Endknoten bis hinauf zur Wurzel liegen, bekannt. Host 4 kennt z.B. die Schlüssel K, E, B und A, wobei der in der Wurzel A abgelegte Schlüssel der Gruppenschlüssel für die gesamte Multicastgruppe ist. Verlässt nun ein Host die Multicast-Gruppe, so muss sichergestellt werden, dass er keine weiteren Nachrichten der Gruppenkommunikation entschlüsseln kann. Das bedeutet, dass alle Schlüssel, die dem Host bekannt waren, gewechselt werden müssen. Dies erfolgt im Baum „von unten nach oben“. Soweit möglich, werden

Multicast-Nachrichten versendet. Ein Beispiel : Angenommen, Host 4 verlässt die Multicast-Gruppe. Host 4 kennt (neben seinem privaten Schlüssel K für die Kommunikation mit der Kontrolleinheit) die Schlüssel E, B und A. Zunächst wird der kompromittierte Schlüssel E durch E_{neu} ersetzt. Der Kontrolleinheit erzeugt einen neuen Schlüssel und sendet diesen unter Verwendung von Schlüssel J per Unicast an Host 3. Nun besitzen alle priviligierten Mitglieder des Teilbaums unter E den neuen Schlüssel. Dies kann zur Verteilung des Schlüssels B_{neu} ausgenutzt werden: Dieser wird mit Schlüssel D bzw. mit Schlüssel E_{neu} verschlüsselt und per Multicast versendet. Nur die betroffenen Hosts können die Nachricht entschlüsseln und erhalten so den neuen Schlüssel. Auf dieselbe Weise wird der neue Gruppenschlüssel A_{neu} durch Verschlüsselung mit B_{neu} bzw. C verteilt.

Das Verfahren ist unabhängig vom verwendeten Routingverfahren; allerdings basiert es auf zuverlässigem Multicast - es muss gewährleistet sein, dass alle Gruppenmitglieder zuverlässig über Schlüsselaktualisierungen informiert werden können. Der Nachrichtenaufwand (Versorgung aller aktuellen Gruppenmitglieder mit neuen Schlüsseln bei Join/Leave) steigt nur mit der Tiefe des Baumes, so sind bei balancierten k-fachen Bäumen mit n Gruppenmitgliedern $k * log_k(n)$ Nachrichten notwendig. Selbiges gilt auch für den Speicherbedarf in den Hosts. Einzig die Kontrolleinheit hat Schlüssel für sämtliche Gruppenteilnehmer zu kennen - der Speicherbedarf für die Schlüssel wächst hier linear mit der Größe der Gruppe. Das Tree-Based-Schüsselmanagement-Verfahren beschreibt zwar einen effizienteren Weg für das Aktualisieren des Gruppenschlüssels; bei jedem Join/Leave müssen aber immer alle Gruppenmitglieder über diese Aktualisierung informiert werden. Es eignet sich daher ebenfalls besser als Schlüsselmanagement für Subgruppen von großen Gruppen.

4.2.3 One-Way-Function-Trees

Eine Weiterentwicklung der Tree-Based-Schlüsselmanagement-Verfahren sind die One-Way-Function-Tree-Verfahren, welche mit Binärbäumen und Einwegfunktionen arbeiten und mit weniger Update-Messages bei Aktualisierung des Gruppenschlüssels auskommen [MoRR00]. Wie auch in Standard-Tree-Based-Verfahren existiert eine Schlüsselmanager-Kontrolleinheit. Die Wurzel des Baumes entspricht dabei dem Gruppenschlüssel; die Endknoten jeweils speichern die privaten Schlüssel der Hosts für die Kommunikation mit der Kontrolleinheit. Jeder Knoten X des Baumes speichert aber nicht nur einen, sondern zwei Schlüssel : einmal einen unverborgenen Schlüssel k_X und außerdem einen verborgenen Schlüssel k_X^g, welcher mit Hilfe einer nicht geheimen Einwegfunktion g aus k_X generiert wird : $k_X^g = g(k_X)$. Für die inneren Knoten Y wird der unverborgene Schlüssel k_X aus den verborgenen Schlüsseln der beiden Söhnen Z^1 und Z^2 mittels einer einfachen Funktion f (z.B. mit XOR) generiert - die Initialisierung erfolgt also von unten nach oben.

Jedes Gruppenmitglied M kennt dabei alle unverborgenen Schlüssel der Knoten auf dem Weg durch den Baum von unten nach oben zur Wurzel (analog zu Tree-Based Keymanagement) und außerdem alle verborgenen Schlüssel, die in den zweiten Sohnknoten der Knoten auf dem Weg von unten zur Wurzel gespeichert sind (siehe Abb.4). Idee dieses Verfahrens ist es, den Schlüsselmanager bei einem Join/Leave nicht Nachrichten an alle Gruppenmitglieder versenden lassen zu müssen - es werden nur bestimmte Informationen versendet, die Gruppenmitglieder können dann aufgrund dieser Informationen und den ihnen bekannten unverborgenen und verborgenen Schlüsseln selbst den neuen Gruppenschlüssel berechnen.

Wie gehabt muss die Schlüsselmanager-Kontrolleinheit bei einem Join/Leave eines Mitgliedes M alle Schlüssel, die in den Knoten von der Wurzel bis hin zu dem mit M korrespondierenden Endknoten gespeichert sind, austauschen. Da aber alle Schlüssel in den inneren Knoten von unteren Schlüsseln abgeleitet sind, braucht die Kontrolleinheit nur den unverborgenen Schlüssel im bestimmten Endknoten explizit generieren - alle übergeordneten Schlüssel bestimmen sich nach oben beschriebenem Muster neu.

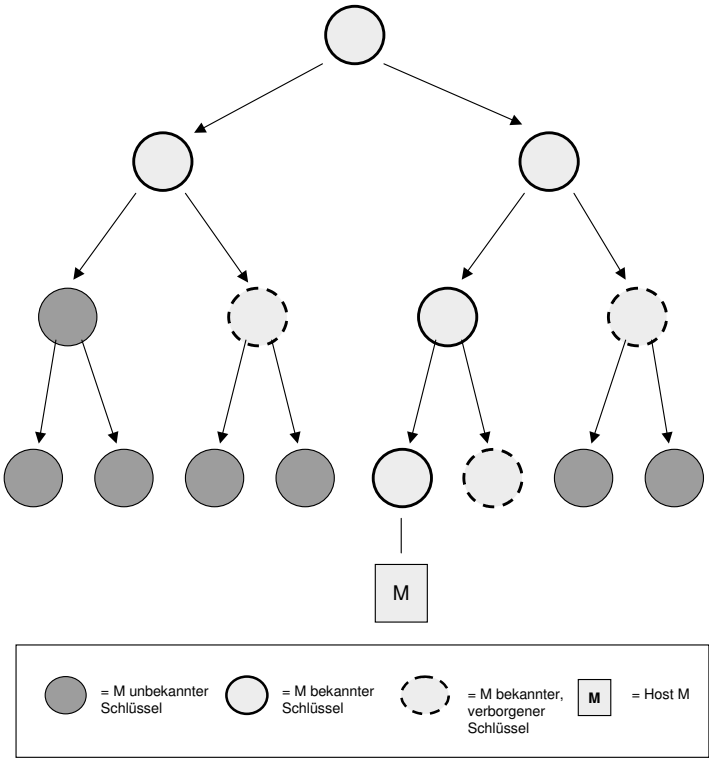


Abbildung 4: One-Way-Function-Trees

Die Anzahl der zu versendenden Key-Update-Nachrichten wurde bei diesem Verfahren auf $log_2(n)$ Nachrichten verbessert, ansonsten hat es dieselben Stärken und Schwächen wie das gewöhnliche Tree-Based-Verfahren.

4.2.4 SRM Key Management

Das SRM Toolkit ist ein in Java implementierter Prototyp für zuverlässiges und sicheres Multicast [MoRR00]. Hier wird wie bei den Tree-Based-Verfahren auch mit mehreren Schlüsseln gearbeitet; allerdings sind diese nicht in einer Baumstruktur hierarchisch geordnet. Bei einer Gruppengröße von n Mitgliedern werden vom SRM Toolkit $(2 * log_2(n)) + 1$ Schlüssel in der Kontrolleinheit gespeichert (bei 32 Mitgliedern also 10 Schlüssel und der zentrale Gruppenschlüssel). Jedem einzelnen Gruppenmitglied sind aber nur $log_2(n)+1$ Schlüssel bekannt (also bei 32 Mitgliedern 5 Schlüssel und der Gruppenschlüssel), also bis auf den Gruppenschlüssel gerade die Hälfte aller Schlüssel. Dabei wird darauf geachtet, dass keinem Mitglied dieselben 5 Schlüssel bekannt sind.

Verlässt nun Mitglied M die Gruppe, so sendet die Kontrolleinheit den aktualisierten Gruppenschlüssel an alle verbleibenden Mitglieder, indem er unter allen Schlüsseln, welche M nicht bekannt sind, chiffriert wird. Dieses impliziert die Generierung von $log_2(n)$ Nachrichten (bei 32 Mitgliedern also 5 Nachrichten), welche einzeln per Multicast an die Gruppe gesendet werden. Jedes verbleibende Mitglied kennt wenigstens einen der 5 verwendeten Schlüssel und kann so mindestens eine Nachricht entschlüsseln. M kann die Nachrichten nicht entschlüsseln und ist damit nicht mehr im Besitz des aktuellen Gruppenschlüssels.

4.2.5 Fazit Schlüsselmanagement

Bei den drei bisher betrachteten Verfahren wächst die Anzahl der von der Kontrolleinheit zu speichernden Schlüssel linear mit der Größe der Gruppe; das SRM Toolkit kommt bei

Kriterium	Naives SM	Tree-Based	One-Way-Fkt.	SRM
Zentrale Architektur	Ja	Ja	Ja	Ja
# Keys Kontrolleinheit	$n+1$	$\frac{kn-1}{k-1}$	$2n-1$	$(2 * log_2(n)) + 1$
# Keys Empfänger	2	$klog_k(n)$	$log_2(n)$	$(log_2(n) + 1$
Updates bei Join/Leave	n	$k * log_k(n)$	$log_2(n)$	$log_2(n)$
Join/Leave-Secrecy	Ja	Ja	Ja	Nein
Anfällig gegen Kollusion	Nein	Nein	Nein	Ja

Abbildung 5: Vergleich Schlüsselmangagement-Verfahren (n = # Gruppenmitglieder, k = Weite des k-fachen-Baumes)

einer Gruppengröße von n Mitgliedern mit weniger als n Schlüsseln aus. Allerdings hat es entscheidenen Nachteil, da es keine Join/Leave-Secrecy garantiert : jedes Mitglied kennt mindestens einen Schlüssel, den ein anderes Mitglied nicht kennt. Tun sich mehrere Mitglieder zu einer Subgruppe zusammen (Kollusion), so kann ein ausscheidendes Mitglied weiter an der Kommunikation partizipieren (indem es sich der Schlüssel der Subgruppe bedient), obwohl es offiziell von der Kontrolleinheit abgetrennt wurde.

Abb.5 fasst noch einmal die wichtigsten Fakten der 4 betrachteten Verfahren zusammen. Zentrale Gemeinsamkeit aller Ansätze ist, dass bei einem Join/Leave eines Gruppenmitgliedes allen aktuellen Gruppenmitgliedern immer aktualisierte Gruppenschlüssel bekanntgemacht werden müssen - deshalb eignen sich diese Verfahren alle für den Einsatz in Subgruppen von großen, dynamischen Gruppen. Im Folgenden werden deshalb verschiedene Methoden, große Multicast-Gruppen in kleinere Untergruppen aufzuteilen, betrachtet.

4.3 Architektur von sicheren Multicast-Gruppen

4.3.1 Iolus

Iolus [Mitt97] ist eine Infrastruktur für sicheres Multicasting, welche 1997 an der Universität Stanford entwickelt wurde. Iolus ist als Standalone-Service für sicheres Multicasting, aber auch als Security-Modul für bestehende Multicast-Applikationen einsetzbar. Iolus teilt eine große Multicast-Gruppe in mehrere Subgruppen auf, welche jeweils eine eigene Multicast-Adresse haben und eigene Schlüssel zur Sicherung der Kommunikation verwenden.

Dabei wird eine Baumstruktur mit sogenannten GSA´s (Group Security Agents) erzeugt, welche die Kommunikation (Routing und Sicherheits-Management) zwischen den Gruppen regeln und autark existieren. Die Wurzel des Baumes bildet der sogenannte GSC (Group Security Controller), welche als Top-Level-Kontrolleinheit fungiert; alle anderen GSA´s werden GSI´s (Group Security Intermediates) genannt und sind für das Management in den Subgruppen verantwortlich.

Will nun ein Gruppenmitglied eine Nachricht an die gesamte Gruppe senden, so können die anderen Mitglieder derselben Subgruppe diese Nachricht sofort entschlüsseln, da sie über den Subgruppen-Schlüssel verfügen. Die GSI dieser Gruppe ist nun dafür verantwortlich, die Nachricht zu entschlüsseln, mit dem Schlüsseln der ihr übergeordneten GSI wieder zu verschlüsseln und die Nachricht an diese weiterzuleiten. Diese GSI macht die Nachricht nun Ihren Subgruppen-Mitgliedern zugänglich und reicht sie evtl. an die ihr übergelagerte Instanz weiter (Abb. 6). So kann es geschehen, dass eine Nachricht bis hinauf zum GSC gereicht wird, von wo sie dann wieder nach unten an andere Subgruppen weitergereicht wird.

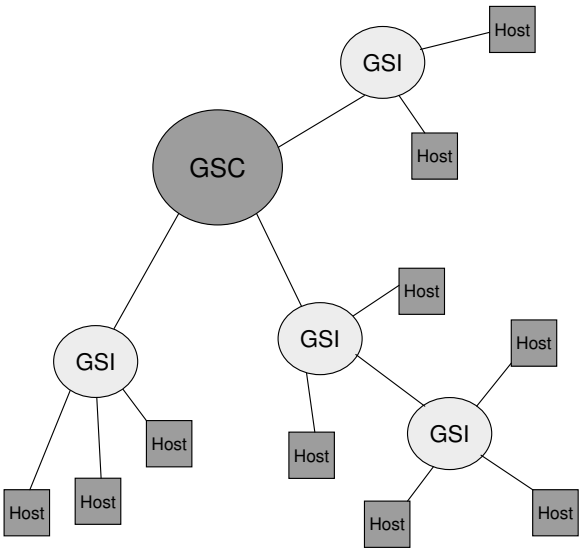


Abbildung 6: Architektur von Iolus

Die Vorteile dieser Architektur sind die Verteiltheit durch die Baumstruktur - erstens werden die Gruppen so klein und skalierbar gehalten, da bei einem Join/Leave eines Gruppenmitgliedes nur die Schlüssel einer Subgruppe a ktualisiert werden müssen, zweitens werden bei auftretenden Fehlern nur ein Teil der Gruppe von der Gruppenkommunikation getrennt; drittens wird bei auftretenden Störungen (solange dies nicht die GSC betrifft) nur ein Zweig der Gruppe von der Kommunikation getrennt. Wenn aber die GSC ausfällt, werden viele Subgruppen voneinander getrennt. Außerdem wird ein Großteil des Paketverkehrs über die GSC laufen; hier kann also leicht ein Datenstau entstehen (Flaschenhals-Gefahr).

4.3.2 Die Architektur des SRM Toolkits

Im Gegensatz zu der Multilevel-Architektur bei Iolus setzt das SRM Toolkit auf eine zweischichtige Hierarchie auf. Eine Multicast-Gruppe wird in mehrere Domains aufgeteilt; ihnen übergeordnet ist der sogenannte Master Controller. Seine Aufgabe ist die Verwaltung der Domains und die Authifizierung von neuen Mitgliedern anhand einer Zugriffs-Kontrollliste. Innerhalb der einzelnen Domains sind die Domain Controller für das Schlüsselmanagement (s.o.) und das interdomäne Routing verantwortlich - analog zu den GSI´s in Iolus. Aufgrund der einfacheren Archikektur ist das SRM Toolkit im Einsatz besser zu handhaben, andererseits ist aber auch nicht so skalierbar wie Iolus [MoRR00].

4.4 Authentizität durch digitale Signaturen / Packet Source Authentication

Für eine sichere Multicast-Gruppenkommunikation ist es erforderlich, dass dem Empfänger einer Dateneinheit gewährleistet ist, das diese Dateneinheit

1. von einem anderen Gruppenmitglied, welches nicht eindeutig identifiziert werden kann
2. von einem als Sender registrierten Gruppenmitglied, welches nicht eindeutig identfiziert werden kann
3. von einem als Sender registrierten Gruppenmitglied, welches eindeutig identifiziert werden kann

gesendet worden ist (um sogenannten Man-in-the-Middle-Angriffen vorzubeugen). Für eine einfache Lösung dieser drei Fälle scheinen sich asymmetrische Kryptografieverfahren anzubieten; hier wird eine Nachricht mit einem öffentlichen Verschlüsselungsschlüssel chiffriert und mit einem privaten Entschlüsselungsschlüssel wieder dechiffriert. Die Datenpakete bekommen also eine digitale Signatur, indem sie mit einem verschlüsselten Hashwert versehen werden, welcher mittels einem nur der Gruppe bekannten Entschlüsselungs-Schlüssel entschlüsselbar ist. Ist nur Gruppenauthentizität (Fall 1) gefordert, so wird der Hashwert vom jeweiligen Sender mit einem allgemeinen Verschlüsselungs-Schlüssel verschlüsselt; will man individuelle Senderauthentizität erreichen, so bekommt jeder Sender einen privaten Verschlüsselungs-Schlüssel. Großer Nachteil dieses naiven Ansatzes ist die Tatsache, dass noch keine wirklich praxistauglichen asymmetrischen Algorithmen existieren (der Rechenaufwand für die Erzeugung „sicherer“ Schlüsselpaare und die Verschüsselung der Hashwerte mit großen Schlüsseln wäre zu hoch).

4.4.1 Der Multiple MACs-Ansatz

Das Multiple-MACs-Schema (MAC = message authentication code) versucht die Authentifizierung effizienter zu gestalten, indem es Einschränkungen an die Sicherheits-Forderungen macht. Ein MAC-Kontrollwert ist ein Wert, der sich nur mit einem geheimen Schlüssel verschlüsseln und wieder entschlüsseln lässt. Das Verfahren legt dabei eine große Anzahl von Schlüsseln fest; dem Sender sind zwar alle Schlüssel bekannt, die Empfänger aber kennen jeweils nur eine Teilmenge der Schlüssel. Will ein Sender nun eine Nachricht versenden, so konstruiert er mit jedem seiner Schlüssel einen MAC-Hashwert und hängt alle MACs an die Nachricht an. Die Empfänger können nun alle MACs überprüfen, zu denen sie die Schlüssel besitzen - sind alle MACs korrekt, so ist die Nachricht authentisch. Je nach Art und Weise, wie die Schlüssel verteilt sind, ist hier ein Angriff durch Kollusion möglich: sobald sich eine Subgruppe von Teilnehmern zusammenfindet, welche alle Schlüssel besitzt, die ein anderes einzelnes Gruppenmitglied M auch kennt, kann diese Gruppe eine Nachricht konstruiereren, die M als gültig erkennt. Kennt diese Subgruppe alle Schlüssel, so bricht das Verfahren vollständig zusammen. Daher ist das Multiple-MACs-Schema eher auf kleine Gruppen anwendbar, bei denen Kollusion keine Gefahr bedeutet [MoRR00].

4.4.2 The Stream Signing Solution

Kann man zuverlässiges Multicast zugrundelegen, so bietet sich das Stream-Signing-Verfahren an : nur das erste Paket einer Datenstroms wird mit einer digitalen Signatur versehen, Die nachfolgenden Pakete enthalten entweder einen verschlüsselten Hashwert des nächsten Paketes oder einen öffentlichen Einweg-Schlüssel, mit dem die Einweg-Private-Key-Signatur des nächsten Paketes verifiziert werden kann. Hierbei muss natürlich gewährleistet sein, das auch alle Pakete ankommen - und dies in der richtigen Reihenfolge [MoRR00].

4.5 Fazit sichere Multicast-Kommunikation

Ansätze, Verfahren und Architekturen für sicheres Multicast befinden sich noch in der Entwicklung. Sobald sich Multicast als Kommunikationsform etabliert, wird auch ihnen ein hoher Stellenwert zuteil - zumal die kommerzielle Nutzung des Internets zukünfig noch weiter zunimmt und die „elektronischen Märkte“ eine sichere und die Privatsphäre schützende Kommunikation erfordern.

5 Schlussbemerkung

Multicast ist die zukunftsweisende Technologie, wenn es darum geht, vernünftig mit vorhandenen Ressourcen umzugehen. Viele Anwendungen, die eigentlich gut in das Schema der Gruppenkommunikation passen, werden heutzutage noch durch Unicast-Kommunikation simuliert. Zum einen Teil ist daran das noch nicht ausgereifte IP-Multicast verantwortlich; welches zwar grundsätzliche Möglichkeiten von Multicast aufzeigt, für einen weltweiten, kommerziellen Einsatz aber nicht skalierbar genug ist und nicht genügend Funktionalität und Dienstgüte bietet. Anderseits sind IP-Multicast-Erweiterungen und andere Multicast-Verfahren noch in der Entwicklung, basieren auf aktiven Netzen, müssen gegeneinander abgewogen werden. Unbestreitbar ist aber, dass Multicast eine prägende Kommunikationsform im Next-Generation-Internet sein wird. Wann dessen Zeit anbricht und welche Verfahren sich hier durchsetzen werden, bleibt abzuwarten.

Literatur

[Alme00] Almeroth (Hrsg.). The Evolution of Multicast. White Paper, The IEEE Communications Society, Februar 2000.

[DLLK+00] Diot, Leevine, Lyles, Kassem und Balensiefen (Hrsg.). Deployment Issues for the IPMulticast Service and Architecture. White Paper, The IEEE Communications Society, Februar 2000.

[KBKK+00] Kasera, Bhattacharyya, Keaton, Kiwior, Kurose und Townsley (Hrsg.). Scalable Fair Active Multicast. White Paper, The IEEE Communications Society, Februar 2000.

[Mitt97] Mittra (Hrsg.). Iolus, Framework for Scalable Fair Multicasting. White Paper, SIGCOMM France, Cannes, 1997.

[MoRR00] Moyer, Rao und Rothagi (Hrsg.). Security Issues in Multicast Communications. White Paper, The IEEE Communications Society, Februar 2000.

[SaMu00] Sahasrabuddhe und Mukherjee (Hrsg.). Multicast Routing Tutorial. White Paper, The IEEE Communications Society, Februar 2000.

[Tane00] Andrew Tanenbaum. *Computernetzwerke*. Pearson Studium. 2000.

[WiZi99] Wittmann und Zitterbart. *Multicast - Protokolle und Anwendungen*. dpunkt Verlag. 1999.

[Wobs98] Reinhard Wobst. *Abenteuer Kryptologie*. Addison Wesley. 1998.