Echtzeitplattformen für das Internet

DATACOM Akademie

Pearson Education

ADDISON-WESLEY

Kai-Oliver Detken

Echtzeitplattformen für das Internet

Grundlagen, Lösungsansätze der sicheren Kommunikation mit QoS und VoIP

ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei
Der Deutschen Bibliothek erhältlich.

10 9 8 7 6 5 4 3 2 1

04 03 02

ISBN 3-8273-1914-5

Pearson Education Deutschland GmbH,
Martin-Kollar-Straße 10–12, D-81829 München/Germany

Einbandgestaltung: bicom, Agentur für
Unternehmenskommunikation GmbH, Frechen
Lektorat: Irmgard Wagner, Planegg, Irmgard.Wagner@munich.netsurf.de
Korrektorat: Andrea Stumpf, München
Satz: reemers publishing services gmbh, Krefeld
Druck und Verarbeitung: freiburger grafische betriebe
Printed in Germany

Inhaltsverzeichnis

„Es ist schwieriger, eine
vorgefertigte Meinung zu zertrümmern
als ein Atom“

Albert Einstein

Vorwort

Wír schreiben das Jahr 2010. Herr Schmidt macht es sich gerade zu Hause vor dem Fernseher gemütlich, als ihm einfällt, dass er noch ein Fahrrad für seine Tochter, die am nächsten Tag Geburtstag hat, besorgen wollte. Er schaltet den Fernseher an und sucht im Internet nach einem günstigen Angebot für Kinderfahrräder. Wegen der unübersichtlichen Menge an Angeboten sucht er sich ein Portal aus, auf dem viele Anbieter vertreten sind. Da er nicht genau weiß, was er eigentlich haben möchte, schaltet er die Hilfefunktion ein. Ein weiblicher Avatar erscheint und fragt mit einer warmen und einfühlsamen Stimme, was sie für ihn tun könne. Herr Schmidt beschreibt sein Problem, woraufhin der Avatar ihn durch einige weltweite Anbieterseiten führt, die seinen Produktwünschen am ehesten entsprechen. Sie zeigt ihm ferner, wie er zu diesen Seiten gelangt und wie er sein Wunschprodukt zusammenstellen kann. Herr Schmidt bedankt sich und der Avatar verschwindet wieder im Hintergrund. Herr Schmidt entscheidet sich nun für einen Anbieter. Er wählt die erforderlichen Parameter wie Farbe, Lenker, Radgröße, Gabel und Pedale aus und betrachtet das Ergebnis seiner Zusammenstellung in einer 3-D-Ansicht. Während er spielerisch immer neue Kombinationen ausprobiert, weist ihn die Software höflich darauf hin, wenn bestimmte Parameter nicht zusammenpassen. Das muss er natürlich berücksichtigen. Anschließend gibt er noch die Körpermaße seiner Tochter ein.

Daraufhin erscheint ein weiterer Avatar in der entsprechenden Größe und nimmt auf dem 3-D-Fahrrad Platz. Die Echtzeitbewegung in virtueller Realität hilft Herrn Schmidt festzustellen, ob das Fahrrad für seine Tochter geeignet ist. Er möchte das Fahrrad kaufen und befördert es in den Einkaufskorb. Nach dem Buchungsvorgang bedankt sich der Avatar des Anbieters bei Herrn Schmidt für seine Bestellung und wünscht ihm noch einen schönen Abend. Herr Schmidt hat allerdings noch eine Frage bezüglich der Lieferzeit, da seine Tochter doch schon am nächsten Tag das Geschenk erhalten soll. Er wählt eine Direktverbindung, worauf sich automatisch eine Videokonferenz öffnet und eine Mitarbeiterin der Firma sich in der ausgewählten Sprache nach seinem Anliegen erkundigt. Er beschreibt ihr die Umstände und sie bestätigt ihm, dass am nächsten Morgen das Fahrrad eingepackt vor seiner Tür stehen wird. Sie weist ihn auch

noch darauf hin, dass er den Bestellvorgang sowie die gesamte Logistikkette über das Internet nachprüfen und verfolgen kann. Er bedankt sich für die Auskunft und die beiden verabschieden sich.

Dieses oder ein ähnliches Szenarium ist nicht mehr weit von der Realität entfernt. Die rasante Verbreitung des Internets, das als weltweites Computernetzwerk für jedermann erreichbar ist, ermöglicht solche Szenarien und wird einen großen Einfluss auf alle Lebensbereiche haben. Das betrifft momentan stark den beruflichen Bereich und wird sich immer mehr in Richtung des privaten Konsums ausweiten. Der tägliche Einkauf und die Behördengänge können mit dem Internet in Zukunft sehr viel leichter erledigt werden. Informationen zum Beruf oder Hobby sind erhältlich sowie Zeitschriften und Bücher online abrufbar. Musik und Filme werden aus dem Internet, nach Bezahlen einer kleinen Nutzungsgebühr, heruntergeladen, um sie dann auf dem Computer oder dem Fernseher im Heimkino anzusehen. Es lassen sich auch viele Berufe von zu Hause aus ausüben und es werden virtuelle Unternehmen mit Telearbeitsplätzen entstehen.

Diese Visionen haben mich dazu gebracht, meine Forschungsschwerpunkte auf die Bereiche Security, Quality-of-Service, Traffic Engineering und Voice-over-IP zu erweitern, da das Internet heute de facto nicht in der Lage ist, ausreichende Qualitäten bereitzustellen und Echtzeitanforderungen gerecht zu werden. Das vorliegende Buch ist das Ergebnis einer mehrjährigen Forschungsarbeit, die durch das europäische Projekt INTELLECT[1] untermauert wurde. Das Thema E-Commerce, das in den letzten Jahren stark an Bedeutung gewonnen hat, wurde bislang nur aus der Anwendungssicht betrachtet. Was fehlte – und diese Lücke will das vorliegende Buch schließen –, war die Betrachtung des ganzen Systems, das heißt die Einbeziehung der Infrastruktur (IT und Telekommunikation) und der Sicherheit des Gesamtsystems. Dadurch werden Echtzeittransaktionen über eine sichere IT-Plattform mit einer garantierten Qualität über das Internet ermöglicht. Das erfordert grundsätzliche Änderungen des bisherigen Konzepts des Internets, die aber erforderlich sind, da sich sonst E-Business- und E-Commerce-Anwendungen sowie Echtzeitapplikationen nicht durchsetzen werden. Das Buch basiert dabei auf meiner eigenen Promotion, die um Grundlagen, bereits existierende Standards und Technologien erweitert wurde. Der Leser erhält somit nicht nur einen Einblick in ein komplexes Thema, sondern wird auch durch neue Ansätze hindurch praxisorientiert begleitet.

1 Intelligent Online Configuration of Products by Customers of Electronic Shop Systems

Hiermit möchte ich mich bei allen Mitarbeitern, Kollegen und Freunden und bei meiner Familie bedanken – sie haben mich während meiner gesamten Promotionszeit unterstützt und wertvolle Anregungen geliefert. Auch die fachliche Diskussion mit Vertretern von Firmen, Herstellern und Hochschulen haben viel zu meinen Überlegungen beigetragen. Ganz besonders aber möchte ich meiner Frau Astrid danken. Die Gründung eines eigenes Unternehmens haben meine Zeit für die Familie nicht gerade anwachsen lassen. Ohne ihre Unterstützung wäre dieses Projekt nicht möglich gewesen.

Kai-Oliver Detken *Grasberg bei Bremen, den 10. Januar 2002*

1

Einführung

Dieses vorliegende Buch beschäftigt sich mit der Erstellung von IP-basierten Echtzeitplattformen unter Berücksichtigung der Sicherheit. Dementsprechend lassen sich vier Hauptthemen herausstellen, die für die Realisierung eines solchen Ziels entscheidend sind:

1. **Security:** Die Sicherheit spielt eine entscheidende Rolle bei der Akzeptanz heutiger IP-Lösungen. Ohne Vertraulichkeit, Integrität, Authentifizierung und Verschlüsselung werden E-Commerce- und E-Business- sowie Kommunikationslösungen im Internet sich nicht durchsetzen.
2. **Quality-of-Service (QoS):** Daten- und Echtzeitdienste müssen mit einer bestimmten Qualität angeboten werden können. Bislang war dies nicht entscheidend. Auf dem Weg in die Kommerzialisierung ist eine garantierte Dienstgüte aber eine wichtige Voraussetzung. Das Internet bietet bislang nur Best-effort.
3. **Traffic Engineering (TE):** Um eine Dienstgüte garantieren zu können, sind TE-Mechanismen im Kernnetz notwendig. Bislang besitzt das Internet keinerlei solcher Mechanismen.
4. **Voice-over-IP (VoIP):** Echtzeitapplikationen setzen kurze und konstante Verzögerungszeiten voraus. Das Gleiche kann für Datenanwendungen gelten. VoIP ist eine sensible Anwendung, die eine hohe Qualität des Netzes voraussetzt, weshalb sie hier kritisch betrachtet wird.

Aus diesen Gründen wird sich dieses Buch auch hauptsächlich um diese vier Themen drehen. Dabei werden die Grundlagen genauso betrachtet wie die Realisierungsmöglichkeiten. Am Ende wird anhand von Messungen die Theorie mit der Praxis verglichen.

1.1 Security

Sicherheit und Geschäftsprozesse sind eigentlich eng miteinander verknüpft, nur wird das anscheinend im Zeitalter des Internets häufig nicht beachtet. Aber die Betrachtung von Sicherheitsaspekten ist im Grunde eine Art Risikomanagement und sollte daher Teil jedes Internetprojekts sein, um mögliche Schäden

für das Unternehmen abzuwenden. Jedes Geschäft beinhaltet gewisse Werte, die es zu schützen gilt. Im Internet sind dies zumeist Informationswerte, aber z. B. auch die Integrität der Inhalte von Webseiten, Erreichbarkeit der Seiten, Geheimhaltung von persönlichen Daten, Vertrauen und Image. Nicht immer müssen diese Werte physikalisch sein; der Schaden kann z.B. gerade bei Imageverlust durch gehackte Seiten mit diskreditierenden Inhalten oder das Nichterreichen eines elektronischen Shops sehr viel größer sein. Dabei lassen sich für dieses Buch folgende Punkte herausstellen:

- Sicherheitskonzept
- Sicherheit der Web-Server
- Sicheres Webdesign
- Sicherheit der Applikationen für eingeschränkte Benutzergruppen
- Sichere Datenübertragung von sensiblen Daten
- Schutz der Privatsphäre der Benutzer

1.1.1 Gefahren durch das Internet

Die Anforderungen von Unternehmen und Institutionen an die Sicherheit unterscheiden sich stark, da unterschiedliche Organisationen verschiedene Sicherheitsanforderungen besitzen. Um das Risiko abschätzen zu können, müssen im ersten Schritt die Gefahren analysiert werden. Es gibt eine Vielzahl von Gefahren, wie die folgende, sicherlich nicht vollzählige Aufzählung zeigt:

1. Eine Auswahl programmierter Gefahren

- **Viren** (**Viruses**) befallen „normale“ Programme und verbreiten sich über diese weiter, indem sie meist den ausführbaren Code des Wirtsprogramms modifizieren. Wird das infizierte Programm ausgeführt, versucht das Virus, weitere Programme zu infizieren.
- **Würmer** (**Worms**) breiten sich in einem Netz selbstständig von Knoten zu Knoten aus, ohne jedoch andere Programme zu infizieren, und richten im Allgemeinen keinen Schaden an, außer einem erhöhten Verbrauch der Ressourcen.
- **Trojanische Pferde** (**Trojan Horses**) sind Programme, die von Benutzern ausgeführt werden und dabei an Stelle der gewünschten Aktion andere, unbeabsichtigte Nebeneffekte hervorrufen.
- **Logische Bomben** (**Logic Bombs**) werden meist in anderen ausführbaren Programmen versteckt und durch bestimmte Bedingungen ausgelöst, beispielsweise an einem bestimmten Tag oder wenn ein Mitarbeiter nicht mehr auf der Gehaltsliste steht. Meistens zerstören sie dann Daten oder setzen Viren frei.
- **Hintertüren** (**Trapdoors**) sind Programmteile, mit deren Hilfe ein Zugriff auf das System unter Umgehung der Authentisierungsverfahren oder mit erhöhten Privilegien ermöglicht wird.

2. Eine Auswahl von Manipulationen

- **Denial-of-Service-Attacken** vermindern die Verfügbarkeit von Servern und Applikationen bis hin zum totalen Ausfall durch den Missbrauch von Diensten.
- **Sniffing:** Abhören des Datenverkehrs zwischen Server und Client. Ist es einem Hacker gelungen, Nachrichten und somit auch Logon-Informationen eines Benutzers abzuhören, kann er diese Informationen verwenden, um sich als dieser User auszugeben (Spoofing).
- **IP-Spoofing:** Beim IP-Spoofing wird eine IP-Adresse vorgetäuscht bzw. gefälscht. Applikationen, die IP-Adressen-basierende Authentifizierung durchführen, sind damit verwundbar. Dadurch ist es möglich, sich für einen anderen Teilnehmer auszugeben, indem man ihm die Adresse vorher entwendet hat.
- **Buffer Overflow:** Mangelhafte Längenabfragen bei den an ein Programm übergebenen Daten. Hat ein Programm eine bestimmte Anzahl von Bytes für beispielsweise die Annahme eines Passworts oder einer URL reserviert, so führt die Übergabe eines überlangen Strings zu einer Schutzverletzung in der Speicherverwaltung des Servers/Clients, die es normalerweise ermöglichen, dass Programme ausgeführt werden können.
- **Exploits:** kleine Programme, die bekannte Schwächen in Betriebssystemen ausnutzen (wie z.B. Buffer Overflow).

3. Hacker und Cracker

Neben den programmierten Gefahren gibt es vor allem Probleme, die durch die direkte Beteiligung von Personen entstehen. In solchen Fällen sollte man die Einschätzung der Vorfälle nach der Motivation des Angreifers vornehmen. An dieser Stelle wird deutlich zwischen Crackern und Hackern unterschieden. Hacker versuchen, in ein System einzudringen, da sie sich für die Umgehung der Sicherheitsmechanismen interessieren. Sie zerstören dabei keine Daten und setzen keine Viren frei. So wie sie die Hintertür eines Intranets betreten haben, so verlassen sie das Netz auch wieder. Cracker hegen hingegen von Anfang an kriminelle Absichten. Sie versuchen, in ein Netzwerk einzudringen, um sich persönliche Vorteile zu verschaffen und eventuell Daten zu zerstören.

Die Unterschiede in den Motiven bestimmen im Allgemeinen auch das Gefährdungspotenzial, das von solchen Vorfällen ausgeht. Beispielsweise wird der Hacker meistens keinen Datenverlust auslösen, es sei denn durch unbedarfte Vorgehensweise im fremden Netzwerk. Größere Probleme entstehen durch ambitionierte Cracker, die sich vorgenommen haben, einen wirklichen Schaden im Netzwerk anzurichten und dieses Ziel auch mit einer gewissen Hartnäckigkeit verfolgen. Dabei hat der Bereich der Industriespionage bzw. der kriminellen Angriffsbemühungen deutlich zugenommen. Tatsache ist auch,

dass sich erfahrene Hacker mit immer raffinierteren Methoden Zugang zu Systemen verschaffen. Vor solchen Attacken kann man sich nur mit extrem gut gesicherten Zugangskontrollsystemen schützen.

1.1.2 Sicherheitskonzept

Das Internet hat sich zum weltweit größten und mächtigsten globalen Informations- und Kommunikationsmedium entwickelt und stellt die Quelle der wichtigsten Innovationen nicht zuletzt auf dem Gebiet der Sicherheit dar. Durch die Anbindung an Netzwerke ergeben sich für ein Unternehmen und dessen Mitarbeiter eine Vielzahl zusätzlicher Kommunikationsmöglichkeiten und nutzbarer Dienstleistungen. Dabei sind alle Unternehmen, die direkt an ein WAN wie das Internet angeschlossen sind, durch diese Verbindung Angriffen auf das eigene Intranet ausgesetzt. Den Systemverwaltern der firmeninternen, lokalen Netzwerke (LANs) obliegt es, betriebsspezifische, programmtechnische und persönliche Daten vor dem externen Zugriff durch Unbefugte zu schützen. Im gleichen Maße müssen Daten der verschiedenen Dienstleistungen das lokale Netzwerk aber auch verlassen dürfen. Zusätzlich sollen sie auch nur diejenigen Adressaten erreichen, für die die Daten letztendlich bestimmt sind. Sicherheitsrelevante Themen sind ebenfalls ein wesentliches Kriterium für Internet-, Intranet- und Extranetlösungen. Die Definition der Sicherheitsstrategie steht dabei im Vordergrund, welche folgende Schritte beinhalten sollte:

1. Analyse der Gefahren
2. Analyse der Anforderungen und Sicherheitsbedürfnisse
3. Risikobewertung
4. Definition von Organisation und Verantwortlichkeiten für bestimmte Teilaspekte des Systems
5. Ausarbeitung eines Sicherheitskonzepts auch für spätere Ausbaustufen

Der empfindlichste Punkt ist die Anbindung des lokalen Netzes an das WAN und der damit jederzeit mögliche, externe Zugriff auf jeden einzelnen Computer in dem lokalen Netzwerk. Der beste Schutz wäre demnach die Trennung der beiden Netze. Eine komplette physikalische Trennung scheidet aus, denn sie würde den notwendigen Informationsfluss verhindern. An dieser Stelle muss zuerst eine Risikoanalyse unter den Anforderungen des Netzbetreibers erfolgen, um mögliche Gefahren erkennen zu können und ihnen entgegenzuwirken. Anschließend muss ein Sicherheitskonzept erarbeitet werden, um nicht nur das interne Netz vom Internet zu schützen, sondern auch die Kommunikation und den Zugang von externen Mitarbeitern und Geschäftspartnern. Es entsteht ein Gesamtkonzept, welches in der Firmenphilosophie verankert werden muss, damit der Sicherheitsgrad so hoch wie möglich gesetzt werden kann.

Ein unternehmensweites bzw. plattformübergreifendes Sicherheitskonzept für die Informationsverarbeitung ist als entscheidende Basis anzusehen, um

einen möglichst hohen Sicherheitsgrad zu erzeugen. Die wichtigsten Aufgaben des IT-Sicherheitskonzepts sind die Gewährleistung der Vertraulichkeit der Daten, d.h. die Geheimhaltung von geschäfts- und personenbezogenen Daten:

- **Integrität der Daten:** Manipulation von Daten nur durch berechtigte Personen und Prozesse
- **Verfügbarkeit der Daten:** Uneingeschränkte Verfügbarkeit der Daten und Ressourcen für berechtigte Personen und Prozesse

Aber auch an die Umsetzung von Sicherheitskonzepten muss mit ökonomischem Sachverstand herangegangen werden. Die häufig gestellte Frage nach den Kosten der Sicherheit kann nur mit der Frage, was der Eintritt eines als Risiko bezeichneten Ereignisses kostet, begegnet werden. Die Umsetzung eines Konzepts erfordert individuell für jedes Unternehmen eine Reihe von technischen und organisatorischen Maßnahmen. Die Thematik Internetsicherheit ist dabei nur ein Bestandteil eines IT-Sicherheitskonzepts.

1.1.3 Web-Server

Web-Server sind besonderen Gefahren ausgesetzt, da ihr Zweck darin besteht, Informationen und Dienste der Öffentlichkeit zugänglich zu machen, weshalb sie auch für alle Anwender im Internet erreichbar sein müssen. Angriffe, die besonders Web-Servern Probleme bereiten können, basieren auf so genannten Denial-of-Service-Attacken. Das sind Programme, die immer neue Anfragen an die Web-Server richten, bis diese aufgrund der Vielzahl der Anfragen nicht mehr in der Lage sind zu antworten. Dann kann niemand mehr auf diese Web-Seiten zugreifen. Der Ruf des Unternehmens ist unwiderruflich beschädigt. Noch schlimmer sind verteilte Denial-of-Service-Attacken, bei denen richtige Client-Server-Anwendungen gestartet werden. Der Angreifer richtet Agenten auf bereits angegriffenen Rechnern ein, die er von einem Server aus startet, um dann das Opfer von verschiedenen Punkten aus anzugreifen. Dadurch werden noch mehr Anfragen gleichzeitig gestartet und die Rückverfolgbarkeit solcher Angriffe ist schwierig, wenn nicht sogar unmöglich. Für diese Art von Attacken sind auch noch frei erhältliche Beispielprogramme im Internet zu finden. Dabei kann man den Opfern noch nicht einmal Versäumnisse ankreiden. Es werden keine Einbrüche in die Rechner vorgenommen, sondern die Systeme werden mehr als das Zwanzigfache der normalen Systemlast belastet. Solche Reserven in der Performance sieht kaum ein ökonomisch denkendes Unternehmen vor. Das Problem sind die schlecht geschützten Rechner anderer, denn um einen Web-Server, der meist mit reichlich Performance ausgerüstet ist, bis zum Zusammenbruch zu überlasten, benötigt der Angreifer viele gute Internetanbindungen von anderen Systemen.

Es gibt zwar einige Möglichkeiten, um Gegenmaßnahmen einzuleiten, einen hunderprozentigen Schutz kann es dabei allerdings nicht geben, da die

Server offen für Anfragen aus dem Internet sind. Hinzu kommt, dass zusätzliche Sicherheitsmaßnahmen, um ankommende Anfragen zu überprüfen, auch immer größeren Rechenaufwand erfordern und somit die Server verlangsamen. Genau dies begünstigt wiederum Attacken.

1.1.4 Webdesign

Sicherheit bedeutet nicht nur, sich gegen Denial-of-Service-Attacken zu schützen. Es reicht daher nicht aus, einen Web-Server hinter einer Firewall zu platzieren, die in einer demilitarisierten Zone (DMZ) steht. Neben der Netzwerk- oder Systemebene muss man zusätzlich das Webdesign mit in die Betrachtung einbeziehen. Das heißt, die Software und Entwicklungsprogramme müssen ebenfalls einer Sicherheitsüberprüfung standhalten. Traditionell erstellen Entwickler Programme, die auf internen Systemen laufen und somit keinen Gefahren durch Unbekannte ausgesetzt sind. Selbst Web-Seiten, die nicht mit internen Systemen verbunden sind, bieten ein ähnliches Szenario: Sicherheit wird auf Systemebene abgehandelt, und zwar durch die Installation des entsprechenden Betriebssystems. Die korrekte Konfiguration und das Setzen der korrekten Zugriffsrechte auf dem Web-Server sind die einzigen Schutzmaßnahmen. Sollte es gelingen, in die Seite einzubrechen (hacken), dann wird das System vom Netz genommen, der Schaden behoben und die Sicherheitsmaßnahmen verschärft. Dies geschieht alles auf der Ebene der Systemverwaltung.

Allerdings wird heute das Internet immer mehr für kommerzielle Zwecke genutzt, wodurch die Web-Seite zu einer Applikation wird, die nicht mehr nur die reine Darstellung von Inhalten übernimmt. Die Kenntnisse in der Programmierung von Web-Seiten und verteilten Anwendungen sind in den meisten Fällen vorhanden, allerdings ohne eine tiefere Betrachtung von Sicherheitsaspekten. Bekannte Themen sind u.a. Secure Socket Layer (SSL), digitale Signatur, Zertifikate oder Cookies. Die Frage, die sich stellt, ist aber, gegen welche Gefahren der eigentliche Programmiercode anfällig ist und wie der eigene Code sicherer umgesetzt werden kann. Für die Informationssicherheit müssen ebenfalls alle Aspekte beachtet werden. Die folgenden Punkte betreffen daher sowohl die Bereiche auf Netzwerk- und Systemebene als auch die auf der Programmierebene:

- Authentifizierung
- Autorisierung
- Schutz der Privatsphäre
- Non-Reputation
- Integrität der Daten
- Erkennen und Kontrollieren von unerlaubten Aktivitäten
- Rechtliche Aspekte bezüglich Schutz und Reaktion

Das Erkennen von unerlaubten Aktivitäten kann z.B. am einfachsten auf Netzwerk- oder Systemebene durchgeführt werden (Firewalls, Verschlüsselung und Zugriffsrechte). Rechtliche Themen müssen aber auch außerhalb der elektronischen Systeme geregelt werden. Daher gehören beispielsweise auch organisatorische Maßnahmen zur Informationssicherheit. Erst alle Maßnahmen zusammen ergeben umfassende Sicherheit. Jedoch muss bei der Entwicklung sicherer Webanwendungen gleich von vornherein an Security gedacht werden, damit es Teil der Anwendung wird, was man heute oft nicht beachtet. Sonst kann es durchaus passieren, dass bei Nichtbeachtung von Sicherheitsmechanismen die Software neu entwickelt werden muss.

Eine Anforderung für die Programmierung von Anwendungen ist die Einbeziehung der Rechte der jeweiligen Benutzergruppen. Es sollte vermieden werden, mit anonymen Benutzern zu arbeiten, gerade bei unterschiedlichen Benutzergruppen, vom Shop-Kunden bis hin zu Zulieferern und Geschäftspartnern. Zudem sollte es verschiedene Möglichkeiten der Authentifizierung geben, z.B. Smardcards oder biometrische Verfahren. Bei der Übertragung sensibler Daten sollte nicht nur die Verbindung verschlüsselt werden, sondern auch die einzelnen Dateninhalte, da sie sonst z.B. durch Aufzeichnungen in Protokolldateien auf Server ungeschützt archiviert werden könnten. Diese Themen werden heute in den seltensten Fällen beachtet, wodurch immer wieder Sicherheitslücken entstehen, die ausgenutzt werden können.

1.1.5 Schutz der Privatsphäre

Ein sehr wichtiger Aspekt bei der Behandlung von Sicherheit ist der Schutz der Privatsphäre des Kunden. Das Vertrauen des Kunden muss gewonnen werden, da er sonst von E-Commerce-Lösungen abgeschreckt wird. Deshalb sollte man alles unterlassen, was diese gefährden könnte, wie z.B. unerlaubt Kundendaten weiterzugeben. Gleichzeitig muss man vertrauensbildende Maßnahmen ergreifen, z.B. sichere Verschlüsselung der Kundendaten, nur Informationen auf dem Kundenrechner speichern, die auch benötigt werden, und den Kunden darüber informieren und selbst entscheiden lassen, ob er dies möchte (z.B. in Form von Cookies).

Ein weiterer wichtiger Punkt ist die Sicherheit bei der elektronischen Zahlungsabwicklung. Um diese zu gewährleisten, sind einige Sicherheitsmaßnahmen nötig. Zunächst muss die Identität der Kommunikationspartner sichergestellt werden. Weiterhin sollten nur Berechtigte den Inhalt der Transaktion lesen dürfen. Damit beide Partner die Transaktion als rechtsgültig ansehen, muss die Authentizität sichergestellt werden. Auch die Nicht-Abstreitbarkeit muss geregelt werden, sodass keine der Parteien im nachhinein Geschäftsprozesse eigenwillig rückgängig machen oder leugnen kann. Letztendlich muss man ebenfalls die Integrität sicherstellen, sodass am Inhalt der Transaktion nicht manipuliert werden kann.

Da es sich bei einer Zahlungstransaktion übers Internet um einen umfangreichen Prozess handelt, müssen in der Praxis folgende Aufgaben erfüllt werden, um die Sicherheit zu gewährleisten:

- Sicherung der Datenbestände
- Sicherung der Datenübertragungswege
- Sicherung der Transaktionsdaten
- Sicherstellung des Zahlungseinzugs

Elektronische Sicherheit konzentriert sich auf zwei Bereiche: den ungewollten Zugriff auf interne Daten verhindern und die Übertragung von Informationen sicherstellen. Sichere Zugriffstechnologien sind für Onlinehändler ebenso wichtig wie für Banken und Kreditkartenorganisationen. Sie sind Grundvoraussetzung für den praktischen Betrieb. Das schließt Passwortschutz und Firewall-Mechanismen ein, um unautorisiertem Zugriff vorzubeugen, und endet mit Angriffssimulatoren, welche die Zuverlässigkeit von Schutzsystemen testen. Bei der Datenübertragung können die Informationen durch Verschlüsselungstechniken geschützt werden und durch Authentifizierungssoftware kann der Absender festgestellt werden. Beide Verfahren verhindern, dass nichtautorisierte Personen die Daten manipulieren können. [KRAU98]

1.1.6 Problemstellung

Die Sicherheit ist im Internet nicht von Anfang an ein Thema gewesen und dementsprechend nicht adressiert worden. Erst spätere Spezifikationen im Rahmen von IPv6[1], die Mitte der 90er Jahre in den Grundzügen definiert wurde, haben sich dieser Problematik angenommen. Aus diesen Spezifikationen ging IPsec hervor, welcher umfangreiche Sicherheitsmechanismen anbietet. Trotzdem sind heutige Internetprotokolle alles andere als sicher, da sie nicht angepasst wurden und kaum Sicherheitsimplementierungen vorhanden sind. Zudem stellt IPsec nicht die einzige Möglichkeit dar, um neue Sicherheitsmerkmale zu integrieren. Weitere Verfahren sind vorhanden, die sich für unterschiedliche Anwendungsfälle durchgesetzt haben. Die Kompatibilität und Interoperabilität zwischen diesen Ansätzen ist allerdings nicht gegeben, sodass hier weitere Anpassungen vorgenommen werden müssten, was meistens nicht geschieht.

Zusammenfassend lassen sich folgende Probleme für die Sicherheit im Internet auflisten:

- Der Teilnehmer kann sich nicht sicher sein, ob Nachrichten, die er empfängt, vom angegebenen Absender stammen und ob sie unterwegs manipuliert oder mitgelesen wurden. Außerdem können die Nachrichten Viren enthalten, die seinen Computer beschädigen können.

1 Version 6 des Internet Protocol (IP), die auch als Next Generation bezeichnet wird.

- Beim Einkaufen im Internet werden zunehmend Kreditkarten- oder Kontoinformationen über das Internet weitergegeben. Dabei besteht die Gefahr, dass diese Daten in den Besitz Unbefugter gelangen oder durch die Empfänger selbst missbraucht werden.
- Beim Surfen in den globalen Netzwerken wird fremde Software zwischen den Rechnern ausgetauscht, deren Aktivitäten der Anwender nur begrenzt kontrollieren kann.
- Von Kunden erworbene, nicht geschützte digitale Güter können ohne Qualitätsverlust und ohne Beachtung des Urheberrechts beliebig oft kopiert und weiterverteilt werden.
- Digitale Verträge oder Zahlungsanweisungen können manipuliert werden.
- Sprachübertragung über das Internet oder allgemein paketbasierte Netze ist unverschlüsselt und jedermann zugänglich.
- Hacker starten Angriffe auf Internetseiten.

Gewünscht werden daher Verfahren, mit denen die Herkunft von Daten zweifelsfrei festgestellt und somit die Urheberschaft des Eigentums bewiesen werden kann (z.B. digitale Wasserzeichen), Intranets und Extranets, die Unternehmen gegenüber nicht authentisierten Personen absichern sowie Möglichkeiten der sicheren Sprach- und Datenübertragung über das Internet.

1.1.7 Lösungsansätze

Um die erwähnten Sicherheitslücken aus den unterschiedlichen Themengebieten zu kompensieren und eine geeignete Plattform für E-Commerce und E-Business zu schaffen, wird der Aufbau eines Extranets angestrebt. Durch die Öffnung zu Kunden und Partnern sowie Lieferanten und die gleichzeitige Abschottung gegenüber nicht authentisierten Personen wird so eine sichere Geschäftsplattform geschaffen, auf die die anderen Bereiche QoS und VoIP aufsetzen können.

Dabei ist ein Extranet im Grunde ein Intranet, welches nach außen, über die eigenen Unternehmensgrenzen hinweg operiert bzw. geöffnet ist. Ein Extranet ist somit als logisches Netz zu definieren, welches man für eine geschlossene Benutzergruppe etabliert, während die Dienstleistungen über ein öffentliches Netz erbracht werden. Der Anwender betrachtet in jedem Fall die Verbindungen als sein privates Netz. Extranets werden heute oftmals mit Virtual Private Networks (VPN) gleichgesetzt und deshalb vornehmlich als die Realisierungsform von Corporate Networks (CNs) großer Unternehmen angesehen. Dabei gibt es unterschiedliche Möglichkeiten, ein VPN zu etablieren. Über Leased Lines oder Festverbindungen im analogen oder digitalen Telefonnetz lassen sich eigene VPNs aufbauen. Dabei steht meistens die Nutzung von Leistungsmerkmalen nicht im Vordergrund, sondern Einsparungsmöglichkeiten und Verfüg-

barkeit. Zusätzlich lassen sich VPNs im Mobilfunkbereich genauso realisieren wie im Festnetzbereich. Somit lassen sich unterschiedliche VPNs für unterschiedliche Kundenanforderungen umsetzen.

Die Umsetzung auf ein Extranet wird zu einer Konvergenz von Netz- und Dienstplattformen führen. Somit werden bekannte Dienste auf alternativen Plattformen eingesetzt und neue, kombinierte (Multimedia-)Dienste wie Videokonferenzen, E-Mail, Handy, IP-Telefonie, LAN-LAN-Verbindungen, Homebanking, Distant Learning, Multimedia Service, Shopping werden verfügbar sein. Für die Verbindung der einzelnen Außenstellen werden so genannte Tunnelverfahren eingesetzt, mit deren Hilfe sichere, private Verbindungen für Netzapplikationen über ein öffentliches oder ein unsicheres Medium zwischen abgesetzten Netzwerken und/oder einzelnen PC-Arbeitsplätzen zu einem zentralen Datennetz aufgebaut werden können.

Dabei müssen die Anforderungen an ein solches Extranet beachtet werden, welches sich mit traditionellen Netzen messen muss. Dementsprechend sind folgende Eigenschaften zu integrieren:

- Verfügbarkeit
- Sicherheit
- Skalierbarkeit
- Quality-of-Service
- Mobilität
- Netzwerkmanagement
- Accounting & Billing
- Migrationsfähigkeit/Integrierbarkeit

Diese Anforderungskriterien für ein Extranet muss jedes Unternehmen berücksichtigen, wenn es effektiv, kostensparend, sicher und leistungsfähig eingesetzt werden soll.

Die Kapitel 2 und 6 befassen sich mit dem Aufbau einer solchen Sicherheitsplattform, wobei vorhandene Ansätze und Standards vorgestellt, getestet und bewertet werden. Es werden die Verschlüsselungsarten (asymmetrisch/symmetrisch) in Kapitel 2 vorgestellt. Schwerpunktmäßig wird auf die Implementierungen der Protokolle IP Security (IPsec), Secure Socket Layer (SSL) und Secure-HTTP (S-HTTP) in Kapitel 6 eingegangen. Der Aufbau einer Public Key Infrastructure (PKI) sowie die Handhabung des Key Management ist ebenfalls ein zentrales Thema. Beim Aufbau einer Echtzeitplattform in Kapitel 6 werden diese Erkenntnisse gebündelt und die zugrunde liegenden Algorithmen untersucht und evaluiert. Dies wird abschließend in Kapitel 7 durch Messungen untermauert. Es werden in Kapitel 8 eine Übersicht über das Machbare und Realisierbare und wertvolle Tipps für eine Umsetzung gegeben.

1.2 Quality-of-Service

Das Hauptziel bei der Entwicklung von Rechnernetzen war und ist es, alle bisherigen Rechnernetze und Spezialnetze mit ihren Diensten in ein gemeinsames Kommunikationsnetz münden zu lassen. Dabei geht es nicht allein um die Erhöhung der Bandbreite, wie oft falsch angenommen wird, sondern auch um die Einbeziehung neuer Dienste und die optimale Auslastung eines Netzes. Zukünftig müssen Netze im LAN[2] und WAN[3] in der Lage sein, die gleichzeitige Übertragung von verschiedenen Datenformaten wie Audio, Video und Bilddaten über dasselbe Medium vornehmen zu können. Dies beinhaltet die Übertragung zeitkritischer Daten in Echtzeit sowie die optimale Auslastung der zur Verfügung stehenden Bandbreite bei gleichzeitiger Zuteilung gesicherter Datenraten an die einzelnen Anwendungen. Das hierfür geeignete Übertragungsverfahren muss also in der Lage sein, mögliche Datenformate bezüglich ihrer Anforderungen in Klassen zu gruppieren und eine entsprechende Dienstgüte bei der Übertragung zu gewährleisten.

Anfang der 90er Jahre wurde das World Wide Web (WWW) entwickelt und es veränderte den Charakter der im Internet übertragenen Daten. Die Web-Seiten beinhalteten auf einmal Bilder, Musikstücke und Filme. Außerdem bekam das Internet immer mehr Relevanz im kommerziellen Bereich. Damit entwickelten sich neue Dienste im Internet wie web-basierte Datenbanken, Audio- und Video-Streaming-Dienste. Diese Entwicklung erhöhte rapide den Bedarf an Übertragungsbandbreite. Aber allein mit der Erhöhung der verfügbaren Bandbreite können auch hier die neuen Anforderungen an das Internet nicht erfüllt werden. Das dem Internet zugrunde liegende Protokoll ist im Grunde genommen unsicher: jedes Datenpaket, welches nicht weitergeleitet werden kann, wird, ohne den Sender zu benachrichtigen, verworfen. Dabei werden Ressourcen in den Routern sowie auf den Übertragungsstrecken zwischen allen Teilnehmern geteilt. Nutzen viele Anwender das Netz gleichzeitig, so steigen die Verzögerungszeiten in den Routern, es kommt zu Warteschlangenüberläufen und folglich zu Paketverlusten.

Anders werden die Ressourcen im herkömmlichen Telefonnetz verwaltet: Es werden entlang der ganzen Strecke zwischen der Quelle und Sinke explizit Ressourcen reserviert und dem Nutzer zur exklusiven Verfügung gestellt. Sind die angeforderten Ressourcen im Netz nicht verfügbar, so wird die Anforderung abgewiesen (z.B. mit einem Besetztzeichen im Telefon). Hält man eine Videokonferenz oder ein Telefonat über ein IP-Netz, so möchte man die gleiche Qualität haben, die man aus dem Telefonnetz kennt. Dabei ist es allerdings möglich, anders als im Telefonnetz, die momentan nicht benötigten Netzressourcen für

2 Local Area Network
3 Wide Area Network

andere Teilnehmer verfügbar zu machen. Möchte man also multimediale Dienste im Internet anbieten bzw. nutzen, so stellen sich folgende Forderungen an das Netz:

- hohe Verfügbarkeit der Netzressourcen
- Management dieser Ressourcen (z.B. der Warteschlangen in den Routern usw.), um die gewünschte Dienstgüte zu gewährleisten.

Die zweite Anforderung steht im Gegensatz zu dem vorhandenen Best-effort-Prinzip des Internets. Seit 1993 beschäftigt sich die Internet Engineering Task Force (IETF)[4] mit der Erweiterung der IP-Protokollfamilie um Protokolle, mit deren Hilfe eine vorhersagbare Dienstgüte im Internet garantiert werden kann. Es ist hierbei zu beachten, dass das Prinzip der Gleichberechtigung der Teilnehmer im Internet verletzt wird. Bei kommerziellen Internet Service Providern (ISPs) würde sie bedeuten, dass unterschiedliche Dienstarten (z.B. Bulk-Service, Premium-Service) angeboten und unterschiedlich abgerechnet werden. [SIEM99]

1.2.1 Definition und Standardisierungsgremien

Die einzelnen Merkmale eines Dienstes, die ein Teilnehmer fühlen oder messen kann, werden mit dem Sammelbegriff Quality-of-Service (QoS) bezeichnet. Dabei sind folgende Punkte zu beachten:

- QoS bezieht sich auf die Sicht des Anwenders, welcher den geforderten Dienst Ende-zu-Ende nutzen möchte und ihn wahrnimmt. Die Fähigkeiten bzw. Eigenschaften eines einzelnen Netzelements tragen zur Dienstgüte bei, können aber nicht separat als QoS-Parameter betrachtet werden.
- QoS beschränkt sich nur auf messbare technische Aspekte.
- In Abhängigkeit von der Zielsetzung kann ein Teil der QoS-Parameter hervorgehoben und separat betrachtet werden. Es müssen dabei nicht immer alle Parameter berücksichtigt werden.
- QoS ist ein neuer Begriff, der keine spezifizierte Definition besitzt. Dadurch sind mehrere Begrifflichkeiten entstanden, die miteinander konkurrieren.

Das Standardisierungsgremium der International Telecommunication Union (ITU) stellt QoS als ein hierarchisches Diagramm vor, welches in der Abb. 1.1 zu sehen ist. In diesem Ansatz werden nicht nur technische Aspekte eines Dienstes betrachtet. So sind beispielsweise die Bedienung von Endgeräten oder die Verfügbarkeit für einen Dienst weitere wichtige Themen. Aus diesen Anforderungen heraus hat die ITU Dienstklassen spezifiziert, die sich auf ATM als Basistechnologie stützen. QoS hat in Abhängigkeit von dem Definitionszusam-

4 Die IETF ist eine Organisation, die die Entwicklung der Internet-Protokollfamilie koordiniert und vorantreibt.

menhang sehr unterschiedliche Bedeutungen. QoS beschreibt im Grunde die Zusicherung eines bestimmten Dienstes, welcher einem Benutzer angeboten wird. Der Benutzer kann dabei aus einer Person oder einer Protokollschicht bestehen. Das heißt, der angebotene Dienst kann ein High-Layer-Service sein, wie das bei Videokonferenzanwendungen der Fall ist, oder ein Service der unteren Schichten nach dem OSI-Referenzmodell.

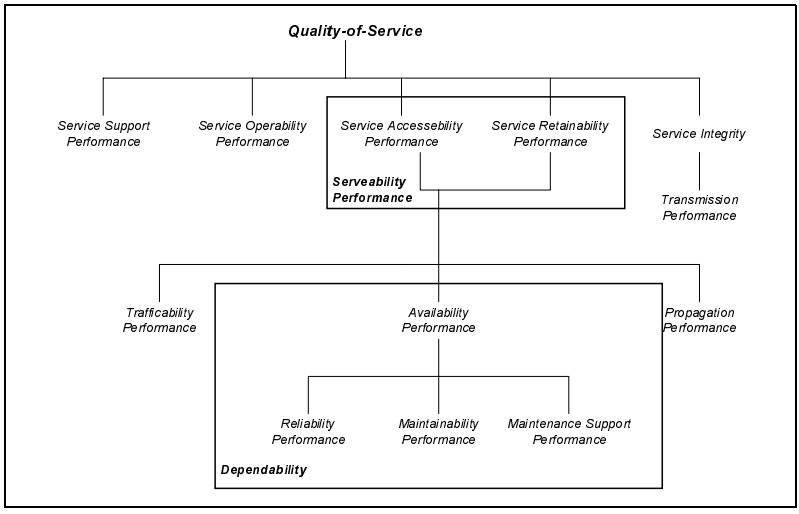


Abb. 1.1
QoS-Modell der ITU

Das ATM-Forum hat sich ebenfalls sehr früh mit dem Thema QoS beschäftigt. In Anlehnung an die Arbeiten von der ITU ist eine Dienstgüte definiert worden, die auf einer virtuellen ATM-Verbindung für einen bestimmten Dienst angeboten werden könnte. Um dies zu erreichen, sind Dienstklassen spezifiziert worden. Die Hauptmotivation, um ATM einzuführen und weiterzuentwickeln, ist die Flexibilität von ATM sowie die Unterstützung existierender und neuer Dienste. Untersuchungen in europäischen Pilotprojekten haben jedoch bewiesen, dass die Verkehrskontrollfunktionen auf den Bedarf der speziellen Anwendungskategorien und ihren unterschiedlichen QoS-Anforderungen angepasst werden müssen, um einen effizienten Transport der Daten gewährleisten zu können. Damit die vorhandene Bandbreite intelligent ausgenutzt werden kann, hat das ATM-Forum deshalb Dienstkategorien entwickelt, die verschiedene Verkehrsarten beinhalten und unterschiedliche Anforderungen an das Netz stellen, wie Tab. 1.1 zeigt. Diese ermöglichen eine Auslastung eines ATM-Netzes von 90 bis 100%. [DEER01]

Tab. 1.1 QoS-Klassenzuordnung nach ITU-T und ATM-Forum

Dienstklasse	Anwendungen	ATM-Forum QoS-Klassen
CBR, Rt-VBR	Standleitung, Video	Klasse 1
Nrt-VBR, ABR	Paketierte Audio/Video-Verbindungen	Klasse 2
Nrt-VBR, ABR	Verbindungsorientierte Datendienste (z.B. Frame Relay)	Klasse 3
UBR	Verbindungslose Datendienste (z.B. IP)	Klasse 4
-	Übertragung ohne definierte Parameter	-

Die Internet Engineering Task Force (IETF) hat sich erst relativ spät dem Thema QoS angenommen, da sich das Internet immer durch seine Einfachheit, Flexibilität und Heterogenität ausgezeichnet hat. Erst als Anfang der 90er Jahre der Ruf nach Netzqualitäten und Echtzeitapplikationen sowie nach mehr Bandbreite laut wurde, setzte man erste Arbeitsgruppen an das Thema. Die neue Protokollversion IPv6, die in den Grundzügen bereits 1995 fertig standardisiert war, sollte bereits eine erhöhte Qualität aufweisen. Dafür wurde im IP-Header das Feld Flow Label definiert, welches in der Lage ist, bestimmte Pakete zu kennzeichnen und dadurch mit einer erhöhten Priorität zu versehen. Dadurch wird letztendlich eine einfachere Unterstützung des Datenflusses gewährleistet, als dies bei IPv4 (der heutigen Internetprotokollversion) der Fall ist. Allerdings hat man aufgrund der Verzögerungen bezüglich des Einsatzes von IPv6 bereits reagiert und einen Ansatz (Differentiated Services) für die bessere Verwendung des Type-of-Service (TOS) Feldes vorgenommen. Somit ist auch IPv4 bereits in der Lage, eine Priorisierung des Datenstroms vorzunehmen. Die IETF fasst die folgenden für Sie wichtigen technisch messbaren Parameter als QoS-Parameter zusammen:

- **Datendurchsatz (Throughput):** Performance einer Verbindung anhand der Menge der übertragenden Bits pro Sekunde
- **Paketverzögerung:** Gesamtverzögerung bei der Übertragung vom Sender bis zum Empfänger (Latency)
- **Jitter:** Betrag der Differenz der Übertragungszeit zweier benachbarter Datenpakete. Werden zwei Datenpakete direkt nacheinander vom Sender zu unterschiedlichen Zeitpunkten gesendet, so werden danach diese beim Empfänger entsprechend nacheinander empfangen.
- **Paketverlustrate:** Das Verhältnis der verworfenen Pakete zur Gesamtanzahl der gesendeten Datenpakete (Packet Loss Rate)

1.2.2 Keine Dienstgütegarantien im Internet

Möchte man eine garantierte Dienstgüte im Internet anbieten, wird man momentan allerdings mit größeren Problemen konfrontiert, da es hier noch keine ganzheitlichen Lösungen gibt. Aus diesem Grund besitzt das Internet auch heute nur einen Best-effort. Eine globale Einführung neuer Technologien ist mit dem Umstieg auf neue Hard- bzw. Software verbunden. Da es aber nicht möglich ist, alle Router weltweit kurzfristig auszutauschen, müssen andere Möglichkeiten geschaffen werden, um einen QoS zu ermöglichen. Zwei unterschiedliche Ansätze sind dafür im Internet denkbar:

- Bildung von QoS-Domänen, innerhalb welcher neue Technik zum Einsatz kommt, wodurch die Unterstützung von QoS-Diensten ermöglicht wird
- QoS-Datenströme werden auf Best-effort-Strecken transparent übertragen. Dabei muss man sicherstellen, dass Best-effort-Abschnitte über ausreichend Kapazität verfügen, um nicht in Leistungsengpässe zu geraten

Wenn das Internet nicht als einzelnes Netz, sondern aus einem Verbund gleichberechtigter Netze betrachtet wird, müssen nicht nur technische Vorkehrungen beim Übergang in andere Teilnetze getroffen werden, sondern auch gegenseitige Abkommen bezüglich der Einhaltung der Dienstgüte im eigenen Netz. Hinzu kommt, dass die Tarifierung der angebotenen QoS-Dienste und die Authentifizierung übergreifend sichergestellt werden muss. Eine Authentifizierung muss im Allgemeinen sowohl am Netzeingang als auch an den Netzübergängen stattfinden. Dies ist aber auch im Internet ein offenes Thema.

Als wichtige Vermittlungseinheiten im Internet agieren Router. Verkehrsströme einzelner Rechner (Hosts) kommen am Eingang des Routers in eine Warteschlange. Der Router analysiert den Paketkopf und entscheidet anhand dessen, zu welchem Ausgang das Paket weitergereicht wird. In der Regel werden am Eingangspunkt des Netzes mehrere kleinere Datenströme zu einem großen zusammengefasst. Da die einzelnen Hosts im Allgemeinem unabhängig voneinander das Netz benutzen, wird auch die Auslastung der Ressourcen statistisch verteilt. Diesen Zugriffsmechanismus nennt man statistisches Multiplexen. Folgende Parameter sind dabei wichtig, die die Verkehrscharakteristik beeinflussen können:

- Die (momentane) Menge der ankommenden Datenpakete
- Verarbeitungsgeschwindigkeit des Routers
- Die verfügbare Bandbreite zwischen den Routern im Backbone

Weiterhin ist anzumerken, dass IP ein Protokoll der Vermittlungsschicht (Schicht 3) ist und nicht den Zugriff auf ein physikalisches Medium beschreibt. Dies ermöglicht zwar, dass das Internet Protocol (IP) unabhängig von der Übertragungstechnologie arbeiten kann, Mechanismen zwischen den Layer-2-Protokollen und IP müssen aber vorhanden sein, um beispielsweise QoS, Prio-

risierung oder Ressourcenreservierung nutzen zu können. Dafür sind aber nicht alle Layer-2-Protokolle geeignet.

Momentan werden zwei Ansätze diskutiert, um Class-of-Service (CoS) einzuführen: Integrated Services (IntServ) und Differentiated Services (DiffServ). Beide Ansätze können aber ebenfalls keine absoluten Garantien geben und gehen als Grundlage von ausreichend Bandbreite (Überkapazität) aus. [DETK00a]

1.2.3 Unterschiede zwischen QoS und CoS

Zwei unterschiedliche Ansätze werden momentan favorisiert, wenn es um die Bereitstellung einer ausreichenden Dienstqualität im Netz geht:

1. Überkapazitäten
   - Aufbau eines redundanten und geswitchten Netzwerks
   - Keine Überschreitung der Auslastung über 40%[5]
   - Class-of-Services-Ansätze können eingesetzt werden (IntServ, DiffServ, IEEE802.1D).
2. QoS-Netzwerk
   - Definition einer angemessenen Dienstgüte
   - Hohe Verfügbarkeit der Netze, Server sowie Datenintegrität
   - Hohe Auslastung der Endsysteme, des Backbone, der Server und Speicher
   - QoS-Parameter von ATM werden eingesetzt

Unter Quality-of-Service (QoS) wird die erreichte Dienstgüte verstanden, die sich je einzelner Session über bestimmte Messwerte (garantierte Bandbreite, Verzögerung, Jitter, Bandbreitenschwankungen, Prioritäten) definiert. Zusätzlich muss Verfügbarkeit, Fehlertoleranz, Redundanz, Effizienz und Sicherheit einbezogen werden. Unter Class-of-Service (CoS) wird hingegen die Zusammenfassung von „irgendwie“ gleichartigen Datenströmen zu einer gemeinsamen Klasse verstanden, die dann eine gleichgeartete Dienstgüte vom Netzwerk erhält, auch als aggregierte Dienstgüte oder aggregriertes Leistungsverhalten bezeichnet. Eine einzelne Session erhält keine individuelle Dienstgüte mehr.

Die Einteilung in CoS-Prioritäten ist nicht mit einer garantierten Dienstgüte, wie bei ATM, verbunden. Es wird kein Traffic Contract aufgesetzt und es werden auch keine Garantien vergeben. Zusätzlich können Jitter und Verzögerungszeiten nicht einbezogen werden. Man geht einfach davon aus, dass das Netz isochrone Datenströme optimal unterstützt, sodass es zu keinen Störungen kommt. Somit ist CoS mit QoS nicht identisch und sollte in jedem Fall unterschieden werden. [DEER01]

5 Praxiswert, der durch verschiedene Projekte und Austausch mit anderen Experten ermittelt wurde.

LAN-Topologie	Beschreibung	Art der Garantie
Shared Topologie ohne Traffic Classes	Ethernet/802.3 ohne 802.1D, keine Unterscheidung zwischen Datenflüssen möglich	Keine Garantien
Shared Topologie mit Traffic Classes	Token Ring/FDDI/Ethernet mit 802.1D	Nur statistische Garantien für Shared Ethernet, bessere Garantien für TR und FDDI
Switched Half Duplex ohne Traffic Classes	Ethernet/802.3 ohne 802.1D	Keine Unterscheidung möglich, wodurch keine Garantien vorhanden sind
Switched Half Duplex mit Traffic Classes	Token Ring/FDDI/Ethernet mit 802.1D, nur zwei Sender in Konkurrenz zueinander	Bessere statistische Garantien
Switched Full Duplex ohne Traffic Classes	Switched Ethernet und Token Ring, keine Unterscheidung zwischen Datenflüssen	Keine Garantien möglich
Switched Full Duplex mit Traffic Classes	Datenflüsse können unterschieden werden	Garantien sind fast möglich geworden und besser als bei den anderen Möglichkeiten

Tab. 1.2 LAN-Topologien mit und ohne CoS [DETK99a]

1.2.4 Einführung von Policies

Neue Anforderungen an Netzwerk-Infrastrukturen werden auf der einen Seite durch neue Applikationen wie Videokonferenzen, IP-Telefonie oder Voice-over-IP (VoIP) geschaffen, auf der anderen Seite durch Integration und Konvergenz heutiger Netze. Die Anforderungen Verfügbarkeit, steigende Datenrate, niedrige Verzögerungen und stabiles Verhalten lassen sich daraus ableiten. Im Betrieb müssen also die Ressourcen zukünftig dynamisch angepasst bereitgestellt werden. Somit muss es eine Vorhersagbarkeit und Beherrschbarkeit sowie die Einhaltung von Service Level Agreements (SLAs) geben. Um die Kosten gering zu halten, muss es weiterhin ein optimales Verhältnis der Dienstgüte, Vermeidung von Überkapazitäten sowie eine effektive Skalierbarkeit im Netzwerk geben. Zusätzlich muss die Personalkapazität möglichst gering gehalten werden. Aus diesem Grund entstand der Bedarf nach Policy-basierten Netzwerken, die eine eindeutige QoS bieten können.

Um End-to-end QoS bereitstellen zu können, muss allerdings ein Mapping zwischen den verschiedenen Verfahren erfolgen. Zudem müssen die Applikationen die QoS-Parameter unterstützen. Differentiated Service (DS) und IEEE802.1D definiert nur eine Hop-by-Hop-Dienstgüte, die im schlechtesten

Fall bei Switches und Routern manuell konfiguriert werden muss. Dies ist allerdings mittelfristig im Normalbetrieb größerer Netzwerke nicht handhabbar. Die Konfiguration von Dienstgüte-Parametern muss daher zentral gesteuert und automatisiert werden. Service Provider und Betreiber privater Netze streben deshalb die zuverlässige Umsetzung von SLAs über Policy-based Networks an. Dadurch kann der Zugriff auf Ressourcen und Applikationen logisch zentral gesteuert werden. Genauso kann die Verwaltung zentral durchgeführt werden. Zusätzlich werden Regeln für die QoS-Steuerung über eine zentrale Benutzerschnittstelle aufgestellt.

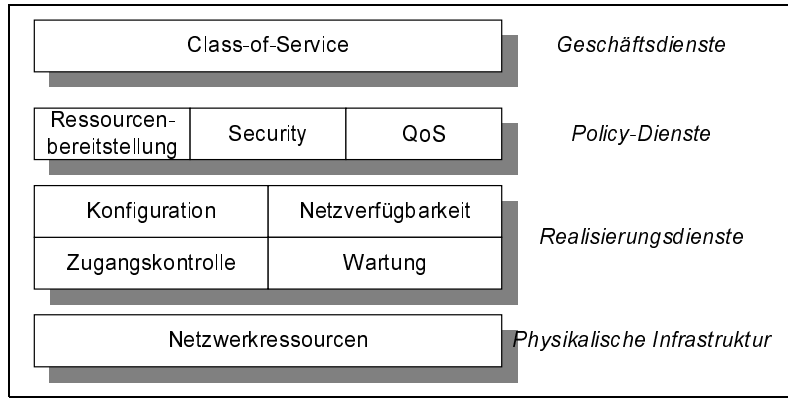


Abb. 1.2 Einordnung der Policy-Dienste

Das mögliche Einsatzszenario der Abb. 1.3 für ein Policy-basiertes Netzwerk enthält folgende Komponenten:

- **Policy Server/Interpreter:** arbeitet als Policy Decision Point (PDP). Es werden Policies in spezielle Konfigurationsinformationen für die Netzkomponenten umgesetzt, die dann auf die Komponenten heruntergeladen werden. Anschließend lässt man einen Policy Client Request entweder zu oder lehnt diesen ab. Zusätzlich könnte er eine Anfrage mittels Zugriff auf einen zentralen Verzeichnisdienst überprüfen.
- **Policy Client:** Endgeräte oder Netzkomponenten arbeiten als Policy Enforcement Point (PEP) und beantragen für beispielsweise eine Session oder eine Applikation eine bestimmte Dienstgüte.
- **Datenbank/Verzeichnis:** enthält Konfigurationsinformationen aller Benutzer. Server und Netzressourcen sowie Policy-Regeln dienen zur Verwaltung und zur Kontrollabfrage der notwendigen Parameter (Directory Services).
- **Policy Informationen:** Policy Server enthalten diese Informationen (Benutzer, Logs, Time etc.), die dynamisch über LDAP[6] aus einem Gesamtverzeichnis heruntergeladen oder ergänzt werden.

6 Lightweight Directory Access Protocol

Die Arbeitsweise sieht zwischen den Komponenten dabei so aus, dass beim Session-Aufbau ein Policy Server einen Request empfängt (z.B. über COPS[7]), ob der jeweilige Benutzer den Dienst mit den jeweils gewünschten Parametern für die Dienstgüte aktivieren darf. Zur Absicherung der konfigurierten Policies muss eine Überwachung stattfinden, die die Einhaltung der zugesicherten Service Levels garantiert. In bestimmten Fällen muss die Überwachung eine Änderung der Policy-Regeln als Folge haben. Die Überwachung kann über Probes, Accounting Tools und Management-Agenten erfolgen. Falls zeitgesteuerte Policies eingesetzt werden sollen, kann ein zusätzlicher Time Server eingesetzt werden.

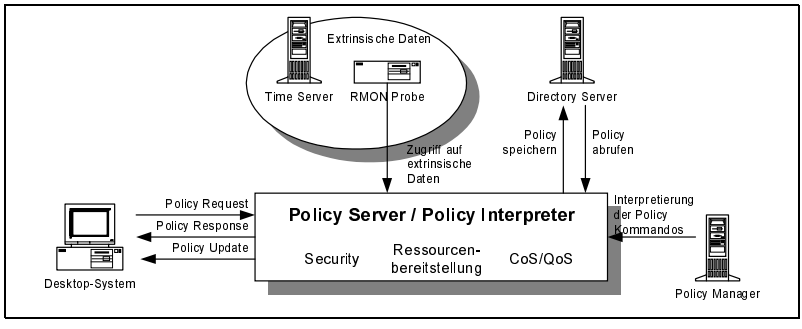


Abb. 1.3 Mögliches Einsatzszenario für Policy-basierte Netzsteuerung

1.2.5 Problemstellung

ATM und QoS waren lange Jahre ein einheitlicher Begriff, sodass die These aufkam, dass zum einen Anwendungen QoS verlangen und andererseits QoS nur in ATM-Netzen realisiert werden kann. ATM QoS ist verbindungsorientiert und legt für einzelne Datenflüsse die Verkehrs- und QoS-Parameter fest, um so eine Dienstgüte wirklich zu garantieren. Das Verfahren arbeitet dabei Folgendes genau in dieser Reihenfolge ab:

1. **Parameter bestimmen:** Verkehrs- (PCR[8], SCR[9], MCR[10], MCTD[11]) und QoS-Parameter (CDV[12], CLR[13], CER[14], CTR[15], CMR[16])
2. **Verbindungen aufbauen:** Wegewahl und Ressourcen-Reservierung
3. **Daten austauschen:** Verkehr wird überwacht/Policing, Garantien werden sichergestellt

7 Common Open Policy Server
8 Peak Cell Rate = Spitzenzellrate
9 Sustainable Cell Rate = mittlere Zellrate
10 Mean Cell Rate = durchschnittliche Zellrate
11 Mean Cell Transfer Delay = durchschnittliche Zellverzögerung
12 Cell Delay Variation = Zellverzögerungsschwankung
13 Cell Loss Rate = Zellverlustrate
14 Cell Error Ratio = Zellfehlerverhältnis
15 Cell Transfer Rate = Zellübertragungsrate
16 Cell Misinsertion Rate = Zelleinfügungsfehlerrate

Was ATM heute auszeichnet, ist, dass mit P-NNI ein intelligentes Routing-Protokoll implementiert ist, welches QoS mit einbezieht, im Gegensatz zur IP-Welt mit beispielsweise OSPF[17]. Dabei wird dieses Routing allerdings sehr komplex, da mehrere Parameter gleichzeitig einbezogen werden müssen. ATM QoS hat sich allerdings in der Praxis für viele Anwender und Netzrealisierungen als komplex und starr herausgestellt. Beispielsweise ist die Parameterbestimmung und das Management sehr aufwendig. Hinzu kommen Skalierungsprobleme, wenn individuelle Reservierungen von Ressourcen für einzelne Sitzungen vorgenommen werden. Hinzu kommt, dass IP hauptsächlich auf ATM-Netzen übertragen wird. RSVP in den Integrated Services kann heute bereits auf ATM-Netze umgesetzt werden (RFC-2379 bis RFC-2382).

Das Problem bleibt, dass die Komplexität in heterogenen Netzen deutlich zunimmt, da mehrere Umsetzungen (Mapping) auf verschiedenen Schichten durchgeführt werden müssen. DiffServ unterscheidet sich außerdem sehr stark von ATM QoS. In der Praxis gibt es zu wenig QoS-Umsetzungen. Bislang sind nur proprietäre Lösungen verschiedener Hersteller wie u.a. Lucent Technologies, Siemens, Cisco Systems und Nortel Networks vorhanden. Somit ist der Ansatz ATM QoS sicherlich nicht der universelle Ansatz wie vor einigen Jahren noch propagiert.

QoS könnte heute im Backbone oder im WAN eingesetzt werden, um auf der einen Seite Applikationen eine bestimmte Dienstgüte garantieren zu können und auf der anderen Seite knappe Ressourcen zu managen. Im LAN ist heute Gigabit-Ethernet (GE) und zu geringen Teilen ATM vorherrschend. Weiterhin wird in solchen Netzen reines Switching eingesetzt. Dies führt häufig zu dem Schluss, dass keine QoS-Mechanismen mehr eingeführt werden müssen, da ausreichende Kapazitäten zur Verfügung stehen. Trotzdem bleiben die Störfaktoren wie Verzögerungen, Jitter und Paketverluste erhalten, die auch bei Überkapazitäten durchaus auftreten können. Um den Best-effort zu vermeiden, können heute unterschiedliche QoS-Ansätze in IP-Umgebung einbezogen werden, die allerdings bislang kaum eingesetzt werden:

- **Reservierung von Bandbreite:** Integrated Services (IntServ) mit RSVP zur Reservierung von aggregierten Datenströmen um Skalierungsprobleme zu vermeiden.
- **Priorisierung des Datenverkehrs:** Differentiated Services (DiffServ) und IEEE802.1D
- **Multiprotocol Label Switching (MPLS):** Priorisierung des Verkehrs unter Berücksichtigung des kürzesten Wegs
- **Subnet Bandwidth Management (SBM):** Mapping der unterschiedlichen Ansätze für ein globales QoS

17 Open Shortest Path First

Folgende Probleme lassen sich bei QoS zusammenfassen, die untersucht werden müssen:

- **Keine End-to-end-Dienstgüte ist vorhanden:** Unterschiedliche Ansätze für unterschiedliche Technologien vorhanden (Mapping notwendig!), proprietäre Verfahren der Hersteller, heterogenes Umfeld im Internet, IPv4 besitzt keine QoS-Mechanismen
- **Applikationen kennen keinen QoS/CoS:** Betriebssysteme unterstützen kaum QoS/CoS, Echtzeitapplikationen nutzen keine Stack-Implementierungen
- **Datennetze kennen keinen QoS/CoS:** Datennetze sind nicht auf Echtzeitanwendungen ausgelegt.
- **Policies sind notwendig:** QoS kann nur über Policies verwaltet werden

1.2.6 Lösungsansätze

Um die Wechselwirkung zwischen unterschiedlichen Verkehrsströmen zu eliminieren, müsste man unterschiedliche Wege beschreiten. Die erste Möglichkeit ist, für bestimmte Dienstarten[18] Ressourcen in jedem Router sowie auf jeder Übertragungsstrecke so zu reservieren, dass bestimmte Dienstgüteparameter eingehalten werden. Dieses kann beispielsweise dadurch realisiert werden, dass für jeden Datenstrom in jedem Router eine einzelne Warteschlange verwaltet wird. Durch Setzen unterschiedlicher Prioritäten für einzelne Queues kann die angebotene Dienstgüte variiert werden. Dieser Lösungsansatz wird als IntServ bezeichnet und wird in den folgenden Kapiteln näher untersucht. Man könnte aber auch den gesamten Datenverkehr anhand seiner Anforderungen in einige wenige Dienstgüte-Klassen unterteilen und danach den Routern unterschiedliche Verarbeitungsmuster für die jeweiligen Klassen mitteilen. Dieser Ansatz wird als DiffServ bezeichnet und oftmals als Nachfolger des IntServ-Ansatzes betrachtet.

Auf der Schicht 2 wird jedes IP-Paket mit einem L2-Header versehen. Dieser Header ist abhängig von der auf Schicht 2 eingesetzten Technik (z.B. ATM oder IEEE 802.3). Bevor ein IP-Paket im Router verarbeitet werden kann, muss das Paket aus dem L2-Header entpackt werden. Das wird bei hohen Datenraten zeitkritisch. Deswegen neigt man immer mehr dazu, in größeren Bereichen des Netzes Datenpakete weiterzuleiten, ohne den L3-Header zu analysieren. Dabei spricht man im Allgemeinen von Layer-2-Switching. Auch hierbei müssen Techniken zur Differenzierung unterschiedlicher Dienstgüte-Klassen eingesetzt werden. Dieses Gebiet ist sehr neu, es gibt hierfür lediglich die ersten Ansätze. Auch sie werden in diesem Buch betrachtet.

18 Im Wesentlichem für multimediale Anwendungen

Da QoS den Dienst Ende-zu-Ende beschreiben muss, sind Mittel zur Anforderung einer Dienstgüte von einer Anwendung aus notwendig. Man spricht dabei von einer Teilnehmersignalisierung. Es gibt allgemein zwei Signalisierungsarten:

- **Innenband-Signalisierung:** Hier wird die Signalisierung unmittelbar mit den Daten mitgesendet. Dafür muss das verwendete Protokoll spezielle Felder vorsehen. Im IP-Protokoll ist das der IP-Header.
- **Außenband-Signalisierung:** Hier wird für die Signalisierung extra Bandbreite benötigt. Es wird zusätzlich, beispielsweise vor dem Verbindungsaufbau eine Anforderung gesendet sowie eine Auflösung der Datenübertragung explizit signalisiert. Hierbei werden Kanäle für Signalisierungsdaten unabhängig von den Nutzdaten-Kanälen verwaltet.

Bei DiffServ hat man den ersten Weg gewählt. Durch bestimmtes Setzen eines Bytes im IP-Header wird die gewünschte Dienstgüte mit jedem gesendeten Datenpaket gemeldet. IntServ hingegen wählt das zweite Verfahren. Deswegen ist für diesen Dienst speziell ein Protokoll entwickelt worden, mit dem ein Teilnehmer einen Dienst anfordern kann. Das ist das so genannte Ressource Reservation Protocol (RSVP). Im Rahmen dieses Buchs werden beide Verfahren untersucht und verglichen sowie weitere Ansätze wie MPLS einbezogen, um ein ganzheitliches Konzept entwickeln zu können.

Das Kapitel 3 untersucht die unterschiedlichen QoS-Ansätze, die sich hier bereits herauskristallisiert haben. Dabei werden im Wesentlichen die Ansätze ATM-QoS und IP-QoS (IntServ, DiffServ, IEEE802.1D) miteinander verglichen, untersucht und gegenübergestellt. Es wird dabei Wert auf die gesamte Übertragungskette gelegt. Das heißt, vom Sender zum Empfänger und wieder zurück muss der gesamte Übertragungspfad eine bestimmten QoS anbieten und garantieren können. Um dies in einer heterogenen Netzumgebung wie dem Internet erreichen zu können, sind unterschiedliche Ansätze möglich, die unterschiedliche Vor- und Nachteile bieten. Die weiteren Kapitel zu diesem Thema ermöglichen auf der einen Seite einen sehr detaillierten Überblick über diese Ansätze. Auf der anderen Seite werden aber auch eigene Lösungsansätze des Autors diskutiert. Um im Kernnetz ebenfalls einen QoS anbieten zu können, müssen Traffic-Engineering-Verfahren zum Einsatz kommen. Dies wird im Kapitel 4 ausführlich untersucht. Multi-Protocol-Label-Switching (MPLS) wird hier als Schwerpunkt mit unterschiedlichen Layer-2-Technologien (ATM und PoS) vorgestellt und evaluiert. Kapitel 6 fasst die Ergebnisse zusammen für die Erstellung einer Echtzeitplattform. Die Messergebnisse unterschiedlicher QoS- und TE-Verfahren in Kapitel 7 runden das Thema ab und bieten eine Sicht in die Zukunft bzw. Funktionsweise solcher Realisierungsmöglichkeiten. Aussichten und abschließende Ergebnisse bietet das Kapitel 8.

1.3 Voice-over-IP (VoIP)

Die Sprachübertragung ist, seitdem die Telefonie eingeführt wurde, immer durch leitungsvermittelte Netze, also verbindungsorientiert, realisiert worden. Für ein Telefongespräch wird deshalb über spezifische Leitungen zwischen zwei Endpunkten eine Verbindung aufgebaut. Diese durchgeschaltete Leitung steht exklusiv für die Teilnehmer während einer Verbindung zur Verfügung und muss sich die Bandbreite nicht mit anderen Benutzern teilen. Wenn das Gespräch zwischen den beiden Parteien beendet wurde, wird die Leitung wieder freigegeben.

Im Gegensatz dazu haben sich die paketvermittelten Netze für die reine Datenübertragung gebildet. Sie sind verbindungslos aufgebaut, verschicken Datenpakete zur Kommunikation und müssen unterschiedliche Datenmengen meistern. Die entstehenden Schwankungen im Verkehrsaufkommen, die durch die unterschiedliche Belastung entstehen, werden mit verschiedenen Signalarten (z.B. Daten und Sprache) und Teilnehmern kombiniert, um die Übertragungsbandbreiten und Vermittlungskapazitäten dynamisch zwischen diesen aufzuteilen. Die Datenpakete benötigen auf dem Weg durch das Netz einen Protokollkopf (Header), der u.a. die Zieladresse enthält. Dabei werden sie von Routern zwischen unterschiedlichen Netzen weitergeleitet. Router setzen zur Erkennung der optimalen Wegwahl Algorithmen ein. Erhöhte Belastungen des Routers werden durch Warteschlangen (Waiting Queue) ausgeglichen, indem die Pakete erst zwischengespeichert und nach der Entlastung des Routers weitergeleitet werden. Diese flexible heterogene Struktur beinhaltet aber höhere Verzögerungszeiten und schwankende Ankunftszeiten der Pakete. Dabei besitzen die Paketlaufzeiten, Paketverluste und Jitter einen erheblichen Einfluss auf die Übertragungsqualität eines Sprachsignals.

Allerdings hat sich innerhalb der letzten Jahre die Qualität der Sprachübertragung stark verbessert. Konnte man vor zwei Jahren bei geringer Belastung des Internets gerade noch dem Gespräch folgen, so kompensieren heute neue Kompressionsalgorithmen und intelligente Empfangspuffer (FIFOs) die meisten unterschiedlichen Laufzeiten der Datenpakete und lassen so die Sprache kaum noch abreißen. Dabei muss man allerdings auf 100 ms Verzögerung durch Netzbelastung im optimalen Fall Rücksicht nehmen. Im schlechtesten Fall kann allerdings die Verzögerung mehrere 100 ms betragen. Das empfindliche menschliche Ohr bemerkt aber bereits ab einer 25-ms-Verzögerung Qualitätsunterschiede. Im Mobilfunkbereich bei GSM-Handys hat sich durch das Problem der Zeit- und Frequenzselektivität eine ähnliche Minderung der Sprachqualität bemerkbar gemacht. Hier werden ebenfalls Paketdaten komprimiert und über die empfindliche Funkschnittstelle gesendet. Die schwankende Qualität hat sich aber nicht negativ auf den Markt ausgewirkt, da der Mehrwert

Mobilität geschaffen wurde. Dafür nimmt der Anwender Qualitätsminderungen in Kauf. Dies kann aber für die Konvergenz von Sprache und Daten im Festnetzbereich nicht gelten.

1.3.1 Komprimierungsverfahren und Qualität

Allerdings ist bei Einsatz des Internets zu berücksichtigen, dass relativ viel Bandbreite für die Übertragung benötigt wird. Das analoge Sprachsignal wird nämlich 8000 Mal in der Sekunde abgetastet, um es zu digitalisieren. Bei einer Umsetzung von 8 Bit pro Abtastung kommt man auf eine typische Datenrate von 64 Kbit/s (PCM nach G.711[19]). Deshalb hat man Algorithmen eingeführt, die diese Bandbreite herabsetzen. Komprimierungsalgorithmen wie G.723.1[20], G.728[21] und G.729[22] komprimieren die Sprache bereits um den Faktor 10. Zusätzlich lässt sich zur Unterdrückung der Sprachpausen Silence Suppression einsetzen. Dadurch kann man bis zu 60% eines Gesprächs unterdrücken, da Gesprächspausen nicht mehr übertragen werden. Zwar wird dabei Bandbreite gespart, allerdings nicht die erhofften 60%, da jeder Gesprächsteilnehmer mehr oder weniger Pausen benötigt.

Tab. 1.3 ITU-T-Standards der Sprachkomprimierung

Standard	Beschreibung	Datenrate in kBit/s	Status
G.711	PCM	48/56/64	H.323 M, H.320 M
G.721	ADPCM	16/24/32/40	Veraltet: durch G.726 ersetzt
G.722	7 kHz	48/56/64	Höhere Qualität
G.723	ADPCM	16/24/32/40	Veraltet: durch G.726 ersetzt
G.723.1	MP-MLQ	5,3/6,3	H.324 M, IMTC IA M
G.726	Adaptive Differential PCM	16/24/32/40	Class 1, FRF 11 O
G.727	Embedded ADPCM	16/24/32/40	Class 1, FRF 11 O
G.728	LD-CELP	16	Optional: Low-Delay
G.729	CS-ASCELP	8	Class 2, FRF 11 M
G.729A	CS-ASCELP	8	Geringe Komplexität

Der ITU-Standard G.723.1 mit dem Verfahren MP-MLQ[23] setzt die effektive Bandbreite von 64 kBit/s bereits auf 6,4 kBit/s herab, wobei noch ein Overhead

19 Pulse Code Modulation (PCM) of voice frequencies
20 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kBit/s
21 Coding of speech at 16 kBit/s using low-delay code excited linear prediction
22 Coding of speech at 8 kBit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)

hinzukommt, sodass man effektiv 15,9 Kbit/s benötigt. Setzt man noch zusätzlich Silence Suppression ein, setzt man die Bandbreite wieder auf 9,5 kBit/s herunter. Dieser Wert schwankt aber und ist abhängig von den Teilnehmern. Der ITU-Standard G.729 mit dem Verfahren CS-ACELP[24] ermöglicht eine Komprimierung auf 8 kBit/s. Der Overhead erhöht die Datenrate anschließend allerdings wieder auf 17,5 kBit/s, während die Unterdrückung der Sprechpausen 10,5 kBit/s umsetzt. G.728 nach LD-CELP[25] besitzt eine effektive Bandbreite von 16 kBit/s. Dies ist mehr als doppelt so viel wie bei G.729 (7 kBit/s) und G.723[26] (6,4 kBit/s) und sollte deshalb auch keine Verwendung mehr finden. Somit lassen sich mehrere Gespräche mit VoIP auf einem 64-kBit/s-Kanal durchführen. Tab. 1.3 zeigt die unterschiedlichen Standards der ITU-T im Bereich der Sprachkomprimierung.

VoIP ist deshalb bereits sinnvoll in Corporate Networks bzw. Intranets, wo die Netzqualität vom Betreiber beherrscht werden kann. Szenarien über ATM-Netze im Backbone zeigten eine deutliche Qualitätsverbesserung bei IP-Applikationen. Das liegt zum einen an der hohen Bandbreite und zum anderen an QoS-Merkmalen. Weiterhin ist zu beachten, dass der Overhead einer Verbindung bei kleinen Paketgrößen die Nutzdatenrate bei weitem überschreitet kann. Hinzu kommen noch die Headers für die jeweiligen Datenapplikationen, da eine Segmentierung im LAN vorgenommen werden muss, um zu stark variierende Verzögerungszeiten aufzufangen. Diese Funktion sollte aber ausschließlich bei Sprachverbindungen eingesetzt werden. Ansonsten würde sie nur unnötigen Verkehr verursachen und das Netz zusätzlich belasten.

1.3.2 Standards

VoIP stellt sehr hohe Anforderung an eine Netzwerkumgebung. Die Laufzeiten sind dabei u.a. für eine störungsfreie Echtzeitübertragung entscheidend. Bisherige Netzprotokolle in der IP-Umgebung sind nicht für eine solche Übertragungsart entwickelt worden. Nimmt man das Protokoll TCP[27], so ist anhand der Quittierungsmechanismen nicht an eine Echtzeitübertragung zu denken, da dies bei reiner Sprachübertragung zu lange dauern würde. Hinzu kommt, dass Pakete nicht wiederholt angefordert werden dürfen, da es sonst zu Störungen im Sprachablauf kommen würde. Das Protokoll UDP[28] bietet hingegen Echtzeitfunktionalität an, da es keine Quittierungsmechanismen kennt. Allerdings sind wiederum nur sehr begrenzte Funktionen implementierbar, da UDP sehr einfach aufgebaut wurde.

23 Multipulse Maximum Likelihood Quantization
24 Conjungate Structur Algebraic Codebook Excited Linear Predictive Coding
25 Low Delay Codebook Excited Linear Predictive Coding
26 in G.726: 40, 32, 24, 16 kBit/s Adaptive Differential Pulse Code Modulation (ADPCM)
27 Transmission Control Protocol
28 User Datagram Protocol

Seit Mitte der neunziger Jahre wurde daher an Echtzeitprotokollen gearbeitet, die die Eigenschaften von diesen Anwendungen besser abbilden können. Das Realtime Transport Protocol (RTP) nach RFC-1889 ist 1996 entwickelt worden, um Paketfolgen auf Seiten des Empfängers besser synchronisieren zu können, damit Jitterbuffer und Sequenzfolgen abgestimmt werden können. Das zusätzliche Überwachungsprotokoll Realtime Transport Control Protocol (RTCP) sorgt für die Kontrolle der Signallaufzeiten und misst die Paketverlustraten. Dadurch kann die Übertragung an die jeweiligen Verbindungseigenschaften angepasst werden.

Trotz der Protokollentwicklung besitzen IP-Protokolle grundsätzlich einen verbindungslosen Charakter. Die Übertragung findet weiterhin hop-by-hop statt. Eine ähnliche Qualität, Verfügbarkeit und Stabilität wie in verbindungsorientierten Netzen ist daher schwer zu realisieren. Auch vorhandene Echtzeitprotokolle wie RTP basieren auf der Netzqualität, die ihnen von den darunter liegenden Schichten bereitgestellt wird. Deshalb spielt QoS ein wichtige Rolle bei der Realisierung von VoIP und anderen Echtzeitapplikationen. Tab. 1.4 zeigt die unterschiedlichen ITU-Standards, die für die Realisierung einer VoIP-Lösung beachtet werden müssen. Hinzu kommen hierbei die Ansätze der IETF.

Tab. 1.4 ITU-Standards im Umfeld VoIP

Netzwerk	**ISDN**	**ATM**	**PSTN**	**LAN**	**POTs**
Standard	H.320	H.321	H.322	H.323 V1/V2	H.324
Jahr	1990	1995	1995	1996/1998	1996
Audio Codec	G.711, G.722, G.728	G.711, G.722, G.728	G.711, G.722, G.728	G.711, G.722, G.723.1, G.728, G.729	G.723.1, G.729
Audio Rates [kBit/s]	64, 48-64	64, 48-64, 16	64, 48-64, 16	64, 48-64, 16, 8, 5.3/6.3	8, 5.3/6.3
Video Codec	H.261	H.261, H.263	H.261, H.263	H.261, H.263	H.261, H.263
Data Sharing	T.120	T.120	T.120	T.120	T.120
Kontrolle	H.230, H.242	H.242	H.230, H.242	H.245	H.245
Multiplexen	H.221	H.221	H.221	H.225.0	H.223
Signalisierung	Q.931	Q.931	Q.931	Q.931	-

1.3.3 Heutige Einsatzmöglichkeiten

Es lassen sich unterschiedliche Szenarien zur Nutzung von VoIP ableiten, die unterschiedliche Qualitäten und Flexibilitäten beinhalten. Die einfachste Lösung ist dabei, eine Software auf dem Client zu installieren (z.B. NetMeeting), um eine direkte Computer-zu-Computer-Kommunikation zu ermöglichen. Natürlich muss der PC mit einer Soundkarte ausgestattet sein und über ein Headset verfügen. Anschließend kann man Punkt-zu-Punkt-Verbindungen über das Internet zu beliebigen anderen Rechnern durch Angabe der IP- oder E-Mail-Adresse aufbauen. Mehrpunkt-Konferenzen lassen sich durch Hinzunehmen von Servern oder durch Multicast-Pakete aufsetzen. Folgende Nachteile ergeben sich allerdings aus der Lösung einer direkten PC-Kommunikation:

- Die Sprachqualität hängt stark von der Belastung des Internets ab, da man über eine Strecke geroutet wird, die nicht vorhersehbar ist.
- Die bestehende TK-Infrastruktur wird nicht mit einbezogen.
- Kommunikation ist nur zwischen zwei PCs möglich, die über die gleiche Software verfügen.

Um die TK-Anlage mit in das Szenario zu integrieren, können IP-Gateways zum Einsatz kommen, die die Umwandlung der TCP/IP-Protokolle auf die TK-Welt vornehmen. Zusätzlich lassen sich auch weiterhin Anrufe in und aus dem öffentlichen Telefonnetz vermitteln. Das Telefon kann dann wahlweise über das Internet das Gespräch aufbauen oder das analoge (PSTN) bzw. das digitale Fernnetz (ISDN) bevorzugen. Diese Möglichkeit beinhaltet aber auch eine größere Komplexität, da die Gateways an der Telefonie-Schnittstelle Signalisierungsverfahren und TCP/IP gleichermaßen unterstützen müssen. Außerdem ergibt dies eigentlich nur Sinn, wenn beide Seiten IP-Gateways zur Verfügung haben.

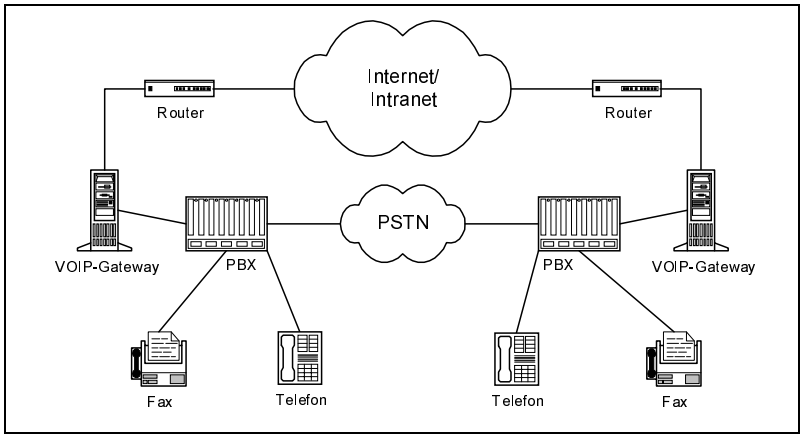


Abb. 1.4
Alle Möglichkeiten eines VoIP-Szenarios

Abb. 1.4 zeigt ein Gesamtszenario eines VoIP-Szenarios, welches sowohl über das Internet als auch über das herkömmliche analoge Telefonnetz (PSTN) miteinander verbunden ist. Somit kann bei schlechterer Qualität der Paketvermittlung sofort wieder das traditionelle Telefonnetz eingesetzt werden. Die Anforderungen an die Netzleistung sind bei der Übertragung von Faxmitteilungen sogar noch höher als bei Sprachübertragung. Hierbei reagiert das Faxprotokoll T.30 äußerst empfindlich auf Laufzeitverzögerungen, da die Synchronisation der Faxverbindung verloren gehen kann, wodurch die Verbindung abgebrochen wird. Paketverluste ab 4% führen so bereits zu einer starken Beeinträchtigung des Durchsatzes. Bei Einsatz eines CNs mit einer ATM-WAN-Verbindung lässt sich allerdings der zusätzliche redundante Pfad auch abschalten und eine höhere Qualität herstellen.

Aus Sicht der Datenkommunikation stellt die Integration von Schnittstellen und Signalisierung einer TK-Anlage ein nicht geringes Problem dar. Hinzu kommt die Planung und Handhabung eines gemischten Daten-/Sprachnetzes, die schwieriger zu beherrschen sind, als das vorher der Fall war. Die verschiedenen Signalisierungsverfahren, die für den Verbindungsaufbau zuständig sind und auch gleichzeitig die Leistungsmerkmale austauschen (z.B. Rufumleitung, Anzeige der Teilnehmernummer und Gebühreninformationen), erhöhen die Integrationsprobleme noch. Die Signalisierung bei digitalen TK-Anlagen basiert im Wesentlichen auf dem ITU-Standard Q.930, während bei der Sprachübertragung über IP der Standard H.323[29] ins Leben gerufen wurde. Hier sind die Funktionen Terminal, Gateway[30] und Gatekeeper[31] festgelegt. Weiterhin ist eine Zugangskontrolle implementiert, die eine Authentifizierung vornimmt, Kontrolle der Bandbreite, Rufsignalisierung, Call-Management, inklusive Status, Fehler und Accounting, und Schnittstellen zu WWW-Servern.

Die Gatekeeper des VoIP-Marktes unterscheiden sich heute vor allem durch ihre Skalierbarkeit, also hinsichtlich der Anzahl der verwalteten Telefone, Zusatzdienste und Redundanz. In redundanten Systemen laufen mehrere Gatekeeper gleichzeitig, sodass beim Ausfall eines Servers die Telefonverbindungen nicht gestört werden bzw. mehr Teilnehmer unterstützt werden können. Die Gateways sind hingegen die Bindeglieder zu anderen Netzen, um beispielsweise mit dem öffentlichen Telefonnetz kommunizieren zu können. Sie konvertieren die verschiedenen Signalisierungsprotokolle und setzen VoIP-Sprachpakete in herkömmliche ISDN- bzw. analoge Sprachsignale um. Für die Konferenzschaltung zwischen mehreren Teilnehmern muss eine Multipoint Controller Unit (MCU) bereitgestellt werden. Sie ist in der Lage, die ankommenden Unicast-Pakete an die unterschiedlichen Empfänger weiterzuleiten, ohne Multicast ein-

29 Packet-based multimedia communications systems
30 Bindeglieder zu anderen Netzen
31 Verwaltung einer Zone, d.h. einer Gruppe von IP-Telefonen

setzen zu müssen. MCUs können eigenständige Geräte sein oder bereits in Gateway oder Gatekeeper integriert werden.

Als Richtlinie für die Kontrolle von störenden Laufzeiteffekten wurde die Empfehlung G.114 von der ITU-T ins Leben gerufen. Für die Sprachübertragung ist man hier von einem Schwellwert von 400 ms für Laufzeiten in eine Richtung ausgegangen. Hierbei ist natürlich eine Echokompensation vorzunehmen, da sonst Verständnisschwierigkeiten der Teilnehmer zu befürchten sind. Echokompensation und eine hohe Verfügbarkeit von Bandbreite ist eine Möglichkeit, um Jitter zu minimieren. Sie kann aber keine Garantien ermöglichen, wenn das Kernnetz keine Dienstgüte vornimmt.

1.3.4 Problemstellungen

Die Laufzeitschwankungen im Internet lassen sich bislang nicht beherrschen. Aufgrund der chaotischen Struktur sind Überlastungen einzelner Netzknoten an der Tagesordnung. In diesem Fall würden die Datenpakete verworfen und später noch einmal angefordert werden. Das ist bei Sprachverkehr nicht tragbar. Höhere Bandbreiten lösen kurzfristig dieses Problem, haben aber keinen direkten Einfluss auf Jitter, Verzögerungen und Komprimierungsverfahren. Hinzu kommt, dass das Internet einem enormen Wachstum unterworfen ist, wodurch neue Standleitungen und Backbones nach relativ kurzer Zeit wieder die volle Auslastungskapazität fahren. Aus diesem Grund müssen andere Wege gefunden werden, um das Netz besser auszunutzen und Ressourcen sowie Laufzeiten garantieren zu können.

Das Protokoll RSVP ist ein erster Ansatz hierzu. RSVP muss allerdings für jede Verbindung vom Netz angefordert werden, wodurch sich dies ungünstig auf die Gesamtleistung des Netzes auswirken kann. Zusätzlich ist es unklar, wie das Netz reagiert, wenn eine große Menge von Teilnehmern RSVP nutzt. Außerdem müssen alle Router auf einem Verbindungspfad RSVP sprechen. Router, die das Protokoll nicht unterstützen, müssen getunnelt werden, was wieder zu neuen Schwachstellen führt. Das heißt, das Netz kann letztendlich die angeforderten Ressourcen verweigern oder diese während einer bestehenden Verbindung zurückfordern. Bei vorhersehbaren Routen durch das Netz lässt sich aber auch mit der jetzigen Form von RSVP und der Realisierung in den Routern (vornehmlich Cisco) die Qualität der Sprachübertragung erhöhen.

VoIP ist aus Gründen der Laufzeiten und Komprimierung sowie der Dienstintegration von mehreren Faktoren stark abhängig, wenn es erfolgreich eingeführt werden soll:

- Bandbreiten im Backbone des Internets
- Quality-of-Service
- Effiziente Komprimierungsalgorithmen
- Verwendung des gleichen und kürzesten Datenpfades beim Routing

- Integration geeigneter Managementtools
- Billing & Accounting
- Integration in herkömmliche TK-Netze und Web-Szenarien

Diese Anforderungen sind heute noch nicht oder nur teilweise erfüllt. Zur Integration von VoIP in bestehende TK-Infrastrukturen und zur Implementierung in Websysteme bzw. in die Internetumgebung müssen diese Probleme allerdings noch gelöst werden.

Hinzu kommt, dass auch bei VoIP unterschiedliche Herstelleransätze vorhanden sind, je nachdem aus welchem Markt der Hersteller ursprünglich stammt, und verschiedene Standards um die Vorherrschaft kämpfen bzw. noch nicht endgültig verabschiedet sind. Die von der ITU[32] entwickelte Protokollfamilie H.323 ist nicht hundertprozentig für VoIP geeignet, da sie nur wenige Komfortfunktionen definiert und mit mobilen Telefonen mit dynamisch zugeordneten IP-Adressen nicht zurechtkommt. Aus diesem Grund hat die IETF[33] das Session Initiation Protocol (SIP) als Alternative zu H.323 entwickelt. Es ist Kern einer Familie von Spezifikationen, die die Kommunikation zwischen VoIP-Endgeräten und den so genannten Gatekeepern festlegt. Darüber hinaus sind weitere Merkmale definiert worden, wie die Interaktion mit unterschiedlichen Gateways und das Media Gateway Control Protocol (MGCP), welches Kontrollfunktionen übernimmt. Welche Alternative sich durchsetzen wird, ist bislang unklar.

Weiterhin bestehen Lücken in den Spezifikationen, die zudem noch nicht alle endgültig verabschiedet sind. Dadurch sind die Hersteller nicht gezwungen, auf die gleichen Standards bzw. Ansätze zu setzen, wodurch immer wieder die Kompatibilität und Interoperabilität unterschiedlicher Systeme leidet. Auch diese Probleme müssen angegangen werden, wenn man nicht in die gleichen Engpässe wie in der herkömmlichen TK-Welt geraten möchte.

Zusammenfassend sind folgende Engpässe zu erwähnen:

- Direkte Kommunikation bei E-Commerce-Lösung ist über das Internet bislang nicht vorhanden, da die Möglichkeiten begrenzt sind und keine ausreichende Dienstgüte angeboten wird.
- VoIP-Implementierungen sind kaum vorhanden und wenn nur in kleineren Test-/Pilotprojekten verfügbar.
- Im Internet und in anderen Datennetzen wird nur Best-effort eingesetzt.
- Bislang ist die Sprach- und Bildqualität im Internet von der Tageszeit abhängig, da unterschiedliche Lasten auftreten. Diese Lasten lassen sich in einem heterogenen Umfeld nicht oder nur schwer beherrschen.

32 International Telecommunication Union
33 Internet Engineering Task Force

- Videokonferenzsysteme als eine Möglichkeit, Echtzeitdaten zu übertragen haben sich nicht durchsetzen können, da die technische Implementierung unzureichend war, die Handhabung sich unfreundlich gestaltete sowie Multicast zur Gruppenkommunikation ein komplexes Themengebiet darstellt.
- Sicherheitsprobleme: Eine Verschlüsselung der Sprachdaten wird bislang weder verwendet noch von den Herstellern angeboten sowie fehlt eine Authentifizierung oder ist unzureichend.

1.3.5 Lösungsansätze

Der Sprachübertragung über Datennetze wird eine große Zukunft bescheinigt. Sie soll die Zusammenführung von Sprach- und Datennetzen vorantreiben und eine Basis für die gemeinsame Konvergenz schaffen. Somit sind nicht mehr zwei getrennte Netze für Sprache und Daten notwendig. Allerdings geht der Ansatz so weit, dass man das Internet sprachtauglich machen möchte, um die globale Verfügbarkeit auszunutzen. Die Internetprotokollfamilie, die auch inzwischen in lokalen Datennetzen vorherrschend ist, wurde aber ursprünglich ausschließlich für Datenübertragung entwickelt und verbessert. Bisherige Anwendungen wie beispielsweise E-Mail oder FTP[34] können mit schwankenden Paketlaufzeiten oder einzelnen Paketverlusten umgehen. Bei reiner Sprachübertragung stören diese Effekte allerdings erheblich.

Die Qualität einer Sprachübertragung über Datennetze hängt deshalb im Wesentlichen von den folgenden Faktoren ab, die beachtet werden müssen:

- Laufzeit der Datenpakete
- Paketverluste
- Paketduplikate
- Paketlänge
- Overhead
- Betriebssysteme
- Hardwarekomponenten (Codec, Soundkarte etc.)
- Verzögerungsschwankungen
- Zwischenpufferung
- Flusssteuerung
- Kanalauslastung
- Netztechnologie

Diese Parameter müssen untersucht und einer Bewertung unterzogen werden. Hinzu kommt, dass man diese Parameter für das Gesamtkonzept berücksichtigen muss, um eine Sprachübertragung bei gleichbleibender Qualität über das Internet (oder andere Datennetze) realisieren zu können. Weiterhin werden die

34 File Transfer Protocol

unterschiedlichen Standards (z.B. H.323 versus SIP[35]) und Ansätze der Standardisierungsgremien und Hersteller aufgezeigt und verglichen. Dabei spielt auch die Sicherheit eine entscheidende Rolle, um den Sprachübertragungspfad zu verschlüsseln, sodass man eine geschlossene Diskussion führen kann. Die Effizienz einer solchen Übertragung sollte ebenfalls in Frage gestellt werden, wodurch Messungen notwendig sind, um die Qualität sowie die Implementierungseffektivität einschätzen und abschließend beurteilen zu können.

Das Kapitel 5 beschäftigt sich mit der Sprachqualität von VoIP-Lösungen. Dabei steht neben der Qualität auch die Komprimierung und Verzögerung im Vordergrund. Hier wird der gesamte Übertragungspfad vom Sender zum Empfänger verfolgt und Probleme angesprochen sowie Lösungsvorschläge unterbreitet. Gerade die Störeffekte wie Bitfehler, Jitter und Echos haben einen großen Einfluss auf heutige Implementierungen. Nachdem die grundlegenden Übertragungseigenschaften anhand der verwendeten Algorithmen erläutert wurden, werden bei der paketorientierten Übertragung die Protokolle und Standards eingehend beleuchtet. Eine große Rolle spielen dabei RTP, RTCP, H.323 und SIP. Hier stehen sich komplexe Telekommunikationsstandards und einfache Internet-Standards gegenüber. Beim Verbindungsaufbau treten die Unterschiede noch deutlicher zutage. Die Sicherheit der Kommunikationsstrecke wird bei der Untersuchung von VoIP ebenfalls angesprochen. Messungen in Kapitel 7 verdeutlichen, welche Verzögerungen im besten Fall auftreten und wie man die Störeffekte kompensieren kann. Die weitere Entwicklung hinsichtlich der Gateway-Protokolle wird ebenfalls in Kapitel 6 angesprochen sowie Möglichkeiten aufgezeigt, VoIP bereits heute effektiv in bestehende Netze zu integrieren. Das abschließende Kapitel 8 zeigt die Aussichten von VoIP bezüglich technischer Umsetzung, Security, H.323/SIP und Gateway-Protokolle auf.

35 Session Initiation Protocol

2

Security

Gerade im Bereich E-Business und E-Commerce ist die Sicherheit eines der wichtigsten Kriterien, damit sich solche Lösungen am Markt etablieren können. Um eine sichere Umgebung zu schaffen, sind bestimmte Anforderungen zu beachten bzw. ist die notwendige Infrastruktur aufzubauen. Hinzu kommt der Einsatz von Echtzeitanwendungen wie Voice-over-IP (VoIP), die weitere Eigenschaften beinhalten, die überdacht werden müssen. Zwar gelten auch hier die allgemeinen Anforderungen an die Informationssicherheit. Darüber hinaus macht allerdings die Einführung verteilter Systeme und der Einsatz von Netzwerken und Kommunikationseinrichtungen zur Übertragung von Daten (inkl. Sprachdaten) zwischen Anwender und Computer sowie zwischen den Rechnern Maßnahmen zum Schutz der Daten während der Übertragung erforderlich (Computer- und Netzwerksicherheit).

Dieses Kapitel wird sich mit den Sicherheitsbedingungen für die Schaffung einer sicheren Kommunikationsplattform für E-Business- und E-Commerce-Anwendungen vertraut machen. Dabei wird auf folgende Hauptanforderungen Rücksicht genommen:

- **Vertraulichkeit:** Der Schutz der übertragenen Daten vor unbefugter Kenntnisnahme Dritter muss gewährleistet werden. Bei der Verbreitung von Nachrichteninhalten lassen sich mehrere Schutzebenen definieren wie Schutz der Anwenderdaten, einzelner Nachrichten oder bestimmter Felder einer Nachricht. Weiterhin muss der Datenstrom vor Analysen abgesichert werden. Ein möglicher Angreifer darf nicht in der Lage sein, die Quelle, den Empfänger oder die Häufigkeit und Dauer von Verbindungen bzw. andere Verkehrsmerkmale zu beobachten, und daraus Rückschlüsse ziehen können.
- **Authentizität:** Eine Mitteilung muss auf Echtheit, Zuverlässigkeit und Glaubwürdigkeit überprüfbar sein. Dies wird durch die Authentizität ermöglicht, die den Ursprung der Nachricht ermittelt. Bei Verbindungen zwischen beispielsweise Client und Server muss also sichergestellt werden können, dass beide Einheiten tatsächlich die sind, die sie vorgeben zu sein, und andererseits muss sichergestellt werden, dass die Verbindung zwischen

den beiden Einheiten nicht manipuliert wird. Dadurch will man verhindern, dass sich ein Angreifer als legitimer Teilnehmer ausgibt oder die Verbindung unberechtigterweise herstellt.

- **Integrität:** Die Unversehrtheit bzw. Unverfälschtheit von Nachrichten wird durch die Integrität beschrieben. Das heißt, es muss gewährleistet werden, dass Nachrichten so empfangen werden, wie sie gesendet wurden. Dies bedeutet, es darf zu keiner Änderung, Einfügung, Verdopplung, Wiederholung, Zerstörung oder Umsortierung kommen. Wie bei der Vertraulichkeit müssen hier auch verschiedene Ebenen unterschieden werden. Dies betrifft die Fragestellungen, ob die Integrität für den gesamten Datenfluss aufrechterhalten werden soll oder nur die einzelnen Nachrichten und Felder. Die verbindungsorientierte Integrität geht noch einen Schritt weiter: sie beinhaltet mit der Änderung des Nachrichtenflusses auch die Verweigerung von Diensten. Verbindungsunabhängige Integritätsdienste bieten hingegen nur Schutz gegen Veränderungen von Nachrichten ohne Rücksicht auf den weiteren Zusammenhang. Zusätzlich wird noch zwischen Integritätsdiensten unterschieden, die Daten wiederherstellen können oder nur eine Meldung generieren.
- **Verbindlichkeit:** Hier wird der Sender oder Empfänger einer Nachricht daran gehindert zu leugnen, dass er die Nachricht gesendet bzw. erhalten hat. Der Empfänger muss beweisen können, dass die Nachricht vom angegebenen Sender stammt. Der Sender wiederum muss belegen können, dass der angegebene Empfänger die Nachricht auch erhalten hat.
- **Verfügbarkeit:** Die Zugriffssteuerung muss für die Beschränkung oder die Freigabe auf den Zugriff auf Systeme und Anwendungen über Kommunikationsverbindungen definiert werden. Eine Einheit, die versucht, Zugriff zu erlangen, muss identifiziert und freigegeben werden, um Zugriffsrechte auf den Einzelnen umsetzen zu können. Eine Vielzahl von Angriffen kann zu Verlust oder Einschränkung der Verfügbarkeit führen (siehe z.B. Denial-of-Service-Attacken).

2.1 Kryptographie

Die sichere Datenübertragung über unsichere Netze kann nur durch die Verschlüsselung der Nutzdaten erreicht werden. Aus diesem Grund ist keine sichere Kommunikation in Netzwerken ohne kryptographische Verfahren möglich. Unter der Kryptographie versteht man die Wissenschaft von den Methoden der Verschlüsselung und Entschlüsselung von Daten. Die Kryptoanalyse hingegen ist die Wissenschaft von den Methoden der unbefugten Entschlüsselung von Daten. Beide Disziplinen werden von der Kryptologie eingeschlossen. Für die Beschreibung bzw. Erläuterung von kryptographischen

Verfahren wird hier ein einfaches Kanalmodell verwendet. Dabei kommunizieren ausschließlich zwei Partner miteinander. Natürlich lässt sich dieses Modell auch auf mehrere Kommunikationspartner ausdehnen, was aber die Komplexität unnötig erhöhen würde.

Von außen wird nun versucht, eine Nachricht abzuhören, zu verfälschen, zu vernichten oder abzufangen. Man kann hierbei folgende Störungen unterscheiden:

- Zufällige Störungen
- Systematische Störungen (physikalisch bedingt)
- Passive Beeinträchtigungen
- Aktive Beeinflussung

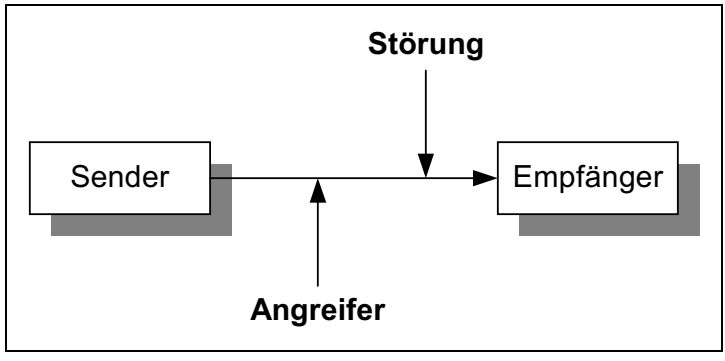


Abb. 2.1 Kryptographie-Kanalmodell

Die beiden erstgenannten Probleme lassen sich mit den Verfahren der Kodierungstheorie und Signalverarbeitung behandeln und werden hier nicht weiter betrachtet. Die beiden letzten Punkte schließen alles vom einfachen Abhören eines passiven Angreifers bis zum aktivem Eingreifen eines aktiven Angreifers mit ein. Unter einem aktiven Angreifer versteht man beispielsweise das Löschen, Verändern und Wiederholen von Daten. Um Angriffe zu verhindern, kommen kryptographische Verfahren zum Einsatz. Dabei sollen sie die Vertraulichkeit, Authentizität, Integrität und Verbindlichkeit gewährleisten.

Hier soll nur die Geheimhaltung betrachtet werden, die für den Aufbau einer sicheren Kommunikationsplattform auf Basis eines Virtual Private Networks (VPN) entscheidend ist, da man das öffentliche Internet verwendet. Zur Zeit erfolgt die Kommunikation im Internet ungeschützt bzw. unverschlüsselt. Vertrauliche oder sensitive Daten werden im Allgemeinen im Klartext gesendet und sind somit für Unbefugte nicht nur lesbar, sondern auch veränderbar! Notwendig ist es deshalb, ein Originaldokument, auch als Klartext bezeichnet, so zu bearbeiten, dass sein Inhalt verborgen bleibt. Das heißt, es liegt in einer nicht lesbaren, nicht verständlichen Form vor. Diesen Vorgang nennt man Verschlüsselung. Der Umkehrvorgang, der aus einem verschlüsselten Dokument, welches

auch als Chiffretext bezeichnet wird, wieder den Klartext generiert, heißt Entschlüsseln. Dabei werden die Begriffe Chiffrieren und Verschlüsseln sowie Dechiffrieren und Entschlüsseln gleichwertig verwendet.

Ob ein kryptographisches System sicher ist, lässt sich nicht einfach beweisen. Es ist leichter zu zeigen, ob es unsicher ist. Aus diesem Grunde gilt ein System nur so lange als sicher, bis das Gegenteil bewiesen werden konnte. Somit ist der Versuch, ein Kryptosystem zu brechen, gleichermaßen für den Anwender und den Gegner interessant. Denn ein erfolgreicher Versuch zeigt, dass das System Schwachstellen beinhaltet. Sichere Kryptosysteme lassen sich in zwei Typen einteilen. Es gibt die absolut sicheren, auch theoretisch sicher genannt, und die praktisch sicheren. Dabei wird ein Verfahren als absolut sicher bezeichnet, wenn es auch unter Einsatz von unbeschränkten Ressourcen (Rechenzeit und Speicherplatz) nicht zu brechen ist. Die Sicherheit muss beweisbar sein. Doch das wird nur von einem Verfahren, One-time Tape genannt, erfüllt. Für fast alle sicheren Verfahren gilt, dass sie unter dem Einsatz von begrenzten Ressourcen sicher sind. Es darf also kein Verfahren bekannt sein, das es ermöglicht, das Kryptosystem unter Einsatz der verfügbaren Ressourcen mit vertretbaren Kosten zu brechen. Solche Kryptosysteme werden auch stark genannt.

Besonders wichtig sind starke Kryptosysteme, für die die folgenden Punkte zutreffend sind:

- Die mathematischen Gleichungen, mit denen sich das System beschreiben lässt, sind so komplex, dass sie mit analytischen Methoden nicht behandelt werden können. Das heißt, das System kann nicht gebrochen werden.
- Bei der Verwendung von einfacheren Methoden wird die Komplexität des Lösungssystems so groß, dass zu viel Rechenzeit und Speicherplatz benötigt werden und damit die Kosten zu hoch werden.

Hingegen unterscheidet man vier verschiedene kryptoanalytische Ansätze, um ein Kryptosystem zu brechen. Im ersten Fall liegt dem Kryptoanalytiker bereits der Chiffretext vor (Ciphertext Only Attack). Eine andere Möglichkeit ist, wenn der Analytiker über zusammenhängende Teile von Klar- und Chiffretext verfügt, um einen Angriff mit bekanntem Klartext durchführen zu können (Know Plaintext Attack). Weiterhin kann der Analytiker einen Klartext seiner Wahl verwenden, um den Algorithmus zu analysieren. Dies wird Angriff mit frei wählbarem Klartext genannt (Chosen Plaintext Attack). Zuletzt kann der Kryptoanalytiker beliebige Chiffretexte dechiffrieren. Die so gewonnenen Klartexte stehen ihm dann bei der Analyse zur Verfügung (Chosen Ciphertext Attack).

Der letztgenannte Ansatz ist bei asymmetrischen Verfahren von Bedeutung. Um die Sicherheit eines Kryptosystems abschätzen zu können, sollte immer von einem Angriff mit frei wählbarem Klartext ausgegangen werden. Dabei gibt es verschiedene Stufen, ein Kryptosystem zu brechen. So ist es möglich, ein Kryptosystem ganz zu brechen, d.h., den Schlüssel zum Dechiffrieren

zu erlangen (Total Break). Eine weitere Möglichkeit ist es, einen alternativen Algorithmus zu finden, mit dem sich der Chiffretext dechiffrieren lässt (Global Deduction). Auch ist es denkbar, nur Teile eines Chiffretextes zu entschlüsseln (Local Deduction) sowie nur Informationen aus einem Chiffretext zu gewinnen (Information Deduction).

Eine wichtige Rolle, besonders für asymmetrische Verfahren, spielen Einwegfunktionen und Trap-Door-Funktionen. Dabei sind Einwegfunktionen Funktionen, deren Umkehrfunktionen sich praktisch nicht berechnen lassen. Trap-Door-Funktionen sind spezielle Einwegfunktionen, bei denen sich die Umkehrfunktion nur mit Hilfe von Zusatzinformationen berechnen lassen. [DEER01]

2.1.1 Unterscheidung von Code- und Kryptosystemen

Die kryptographischen Verfahren zur Geheimhaltung müssen in Code- und Kryptosysteme unterschieden werden. Die Codesysteme arbeiten auf der Basis von semantischen Spracheinheiten. Dabei entsteht ein System zur Ersetzung von Einheiten eines Textes durch neue Zeichenfolgen. Zu den Einheiten des Textes kann man Silben, Wörter etc. zählen. Die Umsetzung erfolgt mit Hilfe einer Umsetztabelle – einem so genannten Codebuch. Jede Zeichenkette, die ersetzt werden soll, muss vorher vereinbart werden. Das beschränkt die Menge der zu ersetzenden Zeichenketten. Codesysteme bieten keine große Sicherheit, da eine bestimmte Klartextzeichenkette immer durch dieselbe Zeichenkette ersetzt wird. Deshalb sollen Codesysteme hier nicht weiter betrachtet werden.

Im Gegensatz dazu arbeiten Kryptosysteme mit Chiffren, auch Kryptofunktionen oder kryptographische Algorithmen genannt. Sie arbeiten unabhängig von semantischen Einheiten und verschlüsseln dabei einzelne Zeichen oder auch Zeichenblöcke. Daher gibt es keine Beschränkung des Klartextes wie bei den Codesystemen. Es kann jeder beliebige Klartext chiffriert werden. Dafür muss es eine Verschlüsselungsfunktion E geben, die den Klartext M in einen Chiffretext C abbildet:

$$E(M) = C$$

Um aus dem Chiffretext wieder den Klartext zu erhalten, muss eine Umkehrfunktion gebildet werden:

$$D(C) = D(E(M)) = M$$

Dabei kann jeder, der die Funktionen D und E kennt, alles ver- und entschlüsseln. Daher gibt es schlüsselabhängige Funktionen D und E, für die ein jeweiliger Schlüssel K und K' existieren muss:

$$E(K, M) = C$$
$$D(K', C) = M$$

Auch hier muss natürlich *C* wieder durch die gesamte Formel ersetzt werden:

$$D(K', E(K, M)) = M$$

Sollten die Schlüssel *K* und *K'* identisch sein, so nennt man das Kryptosystem symmetrisch. *K* und *K'* müssen dabei geheim gehalten werden. Wenn die Schlüssel ungleich sind, spricht man von einem asymmetrischen Verfahren. [JACH97]

2.1.2 Symmetrische Verschlüsselung

Unter den symmetrischen Verfahren, die auch Private-Key-Verfahren genannt werden, sind Verschlüsselungsverfahren zu verstehen, die für die Ver- und Entschlüsselung der Daten den gleichen Schlüssel verwenden. Der Hauptvorteil eines solchen Verfahrens ist die Schnelligkeit. Hardware-Lösungen sind heute in der Lage, mehr als 1 Gbit/s zu verschlüsseln. Software-Verfahren sind zwar nicht so effektiv, erreichen aber ebenfalls bereits mehrere 100 Mbit/s, abhängig vom verwendeten Rechnersystem und Verschlüsselungsverfahren. Um Echtzeitsysteme, wie Voice-over-IP und Videokonferenzsysteme, sicher einsetzen zu können, verwendet man daher die symmetrische Verschlüsslung.

Für die Kennzeichnung der an den Protokollen beteiligten Personen werden in der Literatur oftmals Namen vergeben. Dabei kommt jeder Person eine bestimmte Aufgabe zu. Auch hier werden diese Namen verwendet, um die darzustellen Funktionen und Merkmale exemplarisch erläutern zu können: [SCHN96]

- **Alice:** Erste Person in allen Protokollen
- **Bob:** Zweite beteiligte Person in allen Protokollen
- **Eve:** passiver Gegner (Eavesdropper)
- **Mallory:** aktiver Gegner (Malicious Active Attacker)

Hauptproblem der symmetrischen Verschlüsslung ist das Key Management und die damit verbundene sichere Übertragung der Schlüssel an die Kommunikationspartner. Der Schlüssel muss vor dem Datenaustausch über einen sicheren (abhörsicheren) Kanal ausgetauscht werden. Dies hat zur Folge, dass ein spontaner, gesicherter Datenaustausch nicht möglich ist. Ein weiteres Problem ergibt sich bei der Anzahl der Schlüssel. Unter der Maßgabe, dass man mit *n*-Partnern kommunizieren möchte, aber verhindern will, dass die für einen Partner bestimmte Nachricht von den anderen gelesen werden kann, benötigt man *n* verschiedene Schlüssel. Wenn nun auch die *n*-Kommunikationspartner untereinander exklusiv kommunizieren möchten, sind [n(n-1)/2] Schlüssel notwendig.

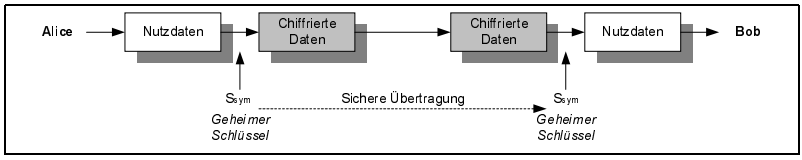


Abb. 2.2
Private-Key-Verfahren

Beim Symmetrieverfahren wird an dieser Stelle einfach vorausgesetzt, dass sich beide Parteien vorab über das Verschlüsselungsverfahren und den Schlüssel geeinigt haben. Alice möchte nun Bob eine Nachricht über einen nicht gesicherten Übertragungskanal zusenden. Also chiffriert Alice das Dokument mit dem Schlüssel S_{sym} und sendet es an Bob:

$$E(S_{sym}, M) = C$$

Bob erhält das chiffrierte Dokument und dechiffriert es mit dem gleichen Schlüssel S_{sym}:

$$D(S_{sym}, C) = M$$

Die Voraussetzung, dass sich beide Parteien vorher über das Verschlüsselungsverfahren und auf einen Schlüssel geeinigt haben, stellt eine große Einschränkung des Verfahrens dar, weil gerade der Vorgang des Einigens eine gute Angriffsmöglichkeit für Gegner bietet. So müssen eigentlich noch zwei Punkte eingefügt werden: Alice und Bob einigen sich auf ein Verfahren und auf einen bestimmten Schlüssel.

Bei einem guten Kryptosystem kann das Einigen auf ein Verschlüsselungsverfahren in der Öffentlichkeit stattfinden, da ein Gegner auch mit diesem Wissen keinen Vorteil für einen Angriff erhalten kann. Der zweite Schritt muss unbedingt geheim ablaufen, da der passive Gegner mit Kenntnis des Schlüssels jede chiffrierte Nachricht entschlüsseln kann. Ein aktiver Gegner hingegen könnte noch größeren Schaden anrichten, da er z.B. die Nachrichten nicht nur lesen, sondern auch verändern kann. Zu den wichtigsten symmetrischen Verfahren zählen bis heute folgende Algorithmen:

- Blowfish
- International Data Encryption Algorithm (IDEA)
- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES)
- Secure and Fast Encryption Routine (SAFER)
- CAST-128
- RC2/5

Hinzu kommen Stromchiffren, die im Gegensatz zu Blockchiffren, welche Klartext und Chiffretext in Blöcken von meist 64 Bit verarbeiten, immer ein Bit oder

Byte (in den wenigsten Fällen 32-Bit-Worte) von Klar- und Chiffretext bearbeiten. Außerdem liefern Blockchiffren bei wiederholter Verschlüsselung desselben Klartextblocks unter Verwendung des gleichen Schlüssels immer den gleichen Chiffretextblock. Bei Stromchiffren ergibt sich hingegen für das gleiche Klartextbit oder -byte bei jeder Verschlüsselung ein anderes Bit oder Byte. Das Funktionsprinzip basiert dabei auf einer einfachen XOR-Verknüpfung der Bitfolge am Ausgang eines Schlüsselstromgenerators mit einem Strom von Klartextbits. Auf der Entschlüsselungsseite wird der Strom der Chiffretextbits mit einem identischen Schlüsselstrom XOR verknüpft. Die Sicherheit von Stromchiffren hängt demnach von dem Schlüsselgenerator ab. Er liefert den Bitstrom, welcher zufällig aussieht, in Wirklichkeit aber deterministisch sein muss, um für die Entschlüsselung wieder reproduziert werden zu können. Für den Einsatz von Blockchiffren gibt es verschiedene Betriebsarten: [GORE99]

- Electronic Code-book (ECB)
- Ciper Block Chaining (CBC)
- Ouput Feedback (OFB)
- Cipher Feedback (CFB)

An dieser Stelle soll nur auf die prinzipielle Funktionsweise der unterschiedlichen Betriebsarten eingegangen werden. ECB ist die einfachste Variante für eine Blockchiffre. Bei ihr wird ein bestimmter Block Klartext immer in den gleichen Block Chiffretext überführt. Die Chiffrierung erfolgt unabhängig von anderen Blöcken immer mit dem gleichen Schlüssel. Eine Nachricht, die sich in mehrere Klartextblöcke M1, M2, ... M_n aufteilt, ergibt bei der Chiffrierung einen Chiffretext aus folgenden unabhängigen Chiffretextblöcken:

$$E(K, M_1), E(K, M_2), \ldots E(K, M_n) \text{ bzw. } C_i = E(K, M_i)$$

Da ein bestimmter Klartextblock immer in den gleichen Chiffretextblock überführt wird, ist der ECB-Modus anfällig für so genannte Codebuch-Analysen. Denn es lässt sich ein Codebuch anlegen, das für jeden Chiffretextblock den entsprechenden Klartextblock enthält. Natürlich ist dieser Ansatz bei einer großen Blocklänge eher theoretischer Natur. Aber bei häufig auftretenden Klartextblöcken bietet sich immer noch eine Möglichkeit und ein Ansatzpunkt für Gegner, um Informationen aus dem Chiffretext zu beziehen.

Eine Variante, die dieses Problem umgeht, ist CBC. In diesem Modus wird das Ergebnis der Verschlüsselung eines Klartextblocks mittels XOR-Funktion mit dem nachfolgenden Klartextblock verknüpft, bevor die nächste Verschlüsselung stattfindet. Der erste Block wird dabei mit einem Initialisierungsvektor (IV) verknüpft. Der IV muss starke Eigenschaften von Zufallszahlen haben, um sicherstellen zu können, dass ein identischer Ursprungstext nicht einen identischen chiffrierten Block ergibt. Damit hängt das Ergebnis einer Verschlüsselung

nicht nur wie bei ECB vom jeweiligen Klartextblock, sondern auch von allen vorangegangenen ab. So werden identische Klartextblöcke im Allgemeinen in unterschiedliche Chiffretextblöcke überführt.

Dadurch wird aus der bisherigen Folge von Klartextblöcken M1, M2, ...M_n der folgende Chiffretext:

$$E(K, M_1 \oplus IV), E(K, M_2 \oplus C_1), E(K, M_3 \oplus C_2)..E(K, M_n \oplus C_{n-1})$$

Die Dechiffrierung verläuft entgegengesetzt zur Verschlüsselung. Jeder Block wird entschlüsselt und mit dem jeweils vor dieser Entschlüsselung dechiffrierten Block durch eine XOR-Funktion verknüpft. Der erste Block wird entschlüsselt und mit dem IV mittels XOR verknüpft:

$$D(K, C_1) \oplus IV, D(K, C_2) \oplus C_1, D(K, C_3) \oplus C_2, ...D(K, C_n) \oplus C_{n-1}$$

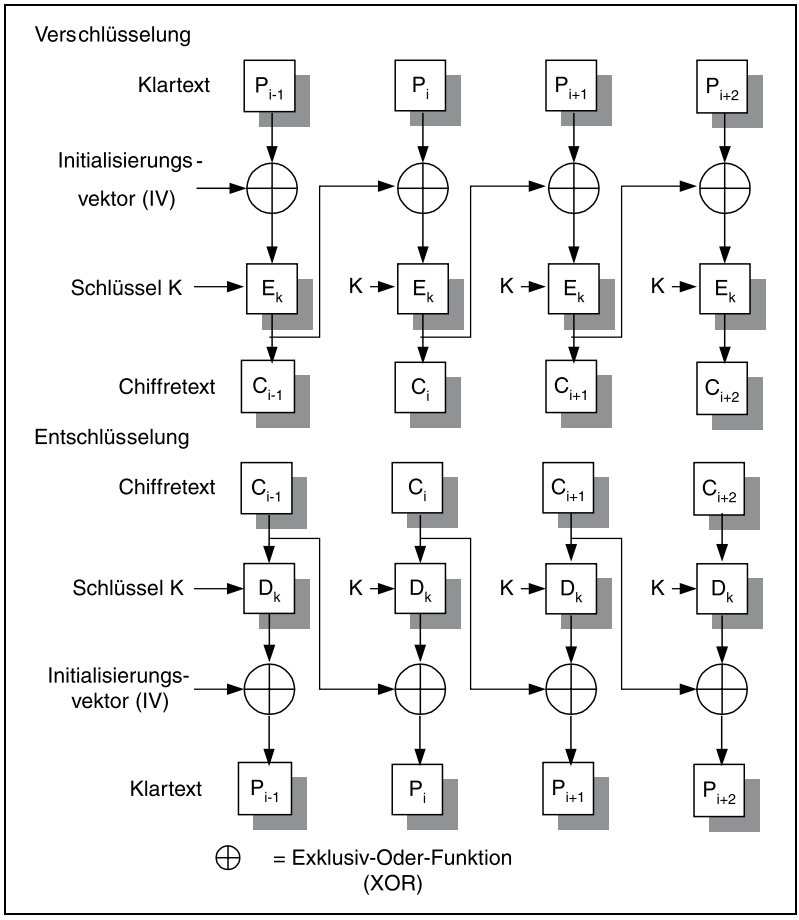
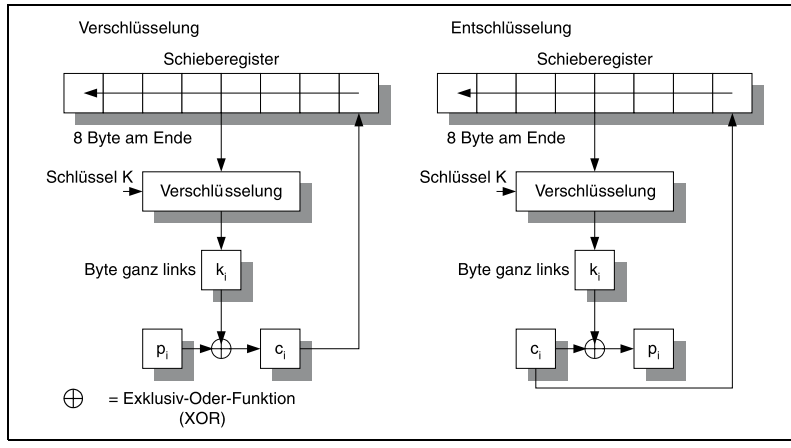


Abb. 2.3
CBC-Betriebsart

Sollte es zu einer fehlerhaften Übertragung kommen, wirkt sich das nur auf den zugehörigen und den nachfolgenden Klartextblock aus. Abb. 2.3 macht die Arbeitsweise des CBC-Verfahrens noch einmal deutlich.

Das CFB ist ebenfalls ein weit verbreitetes Verfahren, bei dem der vorangehende chiffrierte Block verschlüsselt und mit dem aktuellen Textblock mittels XOR verknüpft wird[1]. Bei CFB kann eine selbstsynchronisierende Blockchiffrierung betrieben werden, bei der die Daten in Einheiten verschlüsselt werden, die kleiner als Blockgröße sind. Somit funktioniert CFB bei einer Blockgröße von 8 Bit wie eine Stromchiffrierung auf Byte-Ebene und ist dadurch besonders für Anwendungen geeignet, die einzelne Zeichen[2] separat übertragen. Bei der Funktionsweise wird zuerst ein Schieberegister in der Größe eines Eingangsblocks mit dem IV gefüllt und verschlüsselt. Das ganz links stehende Byte wird mit dem ersten Klartextzeichen XOR verknüpft und anschließend übertragen. Das Schieberegister wird um ein Byte nach links gedrückt und das eben verschlüsselte Zeichen in die frei gewordene rechte Byte-Position geschrieben. Die Entschlüsselung verläuft genau umgekehrt. Für Ver- und Entschlüsselung wird der Blockalgorithmus im Verschlüsselungsmodus betrieben.

Abb. 2.4
CFB-Betriebsart



Ein Fehler im Klartext wirkt sich auf den gesamten folgenden Chiffretext aus, macht sich aber bei der Entschlüsselung wieder rückgängig. Ein fehlerhaftes Byte im verschlüsselten Text bewirkt zunächst einen 1-Byte-Fehler im Klartext, wandert dann aber anschließend durch das Schieberegister und beeinträchtigt bei einer Blockgröße von 64 Bit auch die nächsten 8 Byte. Ein Synchronisierungsfehler wandert durch das Schieberegister und beeinträchtigt 8 Byte. Anschließend erholt sich der CFB-Modus wieder.

1 Der erste Textblock wird nur mit dem IV mittels Exklusiv-Oder verknüpft.
2 Zum Beispiel ASCII-Zeichen einer Tastatureingabe

Beim OFB wird hingegen ein bestimmter Status der Chiffre aufrechterhalten, der wiederholt verschlüsselt und mit Textblöcken mittels XOR verknüpft wird, um einen chiffrierten Text zu erhalten. Ein IV stellt dabei den ursprünglichen Status der Chiffre dar. Dadurch funktioniert er ähnlich wie CFB, wobei allerdings hier nicht das aktuelle Byte C_i des verschlüsselten Textes ins Schieberegister zurückgekoppelt wird, sondern das nach der Verschlüsselung aktuell in der linken Position stehende Byte k_i des Schieberegisters. Auch im OFB-Betriebsmodus wird der Blockalgorithmus für Ver- und Entschlüsselung im Verschlüsselungsmodus betrieben.

Vorteilhaft beim OFB-Modus ist, dass sich Bitfehler während der Übertragung nicht weiter fortpflanzen können. Da weder Klar- noch Chiffretext zurückgekoppelt werden, ist immer nur das aktuelle Zeichen von einem Fehler betroffen. Wenn allerdings die Schieberegister auf der Verschlüsselungs- und Entschlüsselungsseite nicht mehr synchronisiert sind, ist der entstandene Klartext völlig unbrauchbar. Daher müssen Synchronisationsfehler erkannt werden können. Wird dieser Fehler entdeckt, müssen die Schieberegister erneut mit einem IV initialisiert werden. [DOHA00]

Blowfish

Blowfish ist ein Algorithmus, welcher frei verfügbar und nicht patentiert ist. Er wurde von Bruce Schneier entworfen und ist sehr schnell. Blowfish ist für Anwendungen wie die Verschlüsselung von Kommunikationskanälen optimiert worden, bei denen sich der Schlüssel nur selten ändert. Er beinhaltet eine 64-Bit-Blockchiffrierung mit variabler Schlüssellänge von bis zu 448 Bit. Weiterhin besteht er aus den Teilen Schlüsselexpansion und Verschlüsselung. Während der Schlüsselexpansion wird der Benutzerschlüssel in verschiedene Teilschlüssel umgewandelt. Diese haben zusammen 4168 Bit und beinhalten das P-Array, welches aus 18 Teilschlüsseln mit einer Länge von 32 Bit besteht, und vier S-Boxen der Länge 32 Bit sowie 256 Einträgen. Nach der Initialisierung werden 521 Iterationen benötigt, um alle Teilschlüssel zu erzeugen. 16 Runden werden für die Verschlüsselung eines Datenblocks verwendet. [SCHN96]

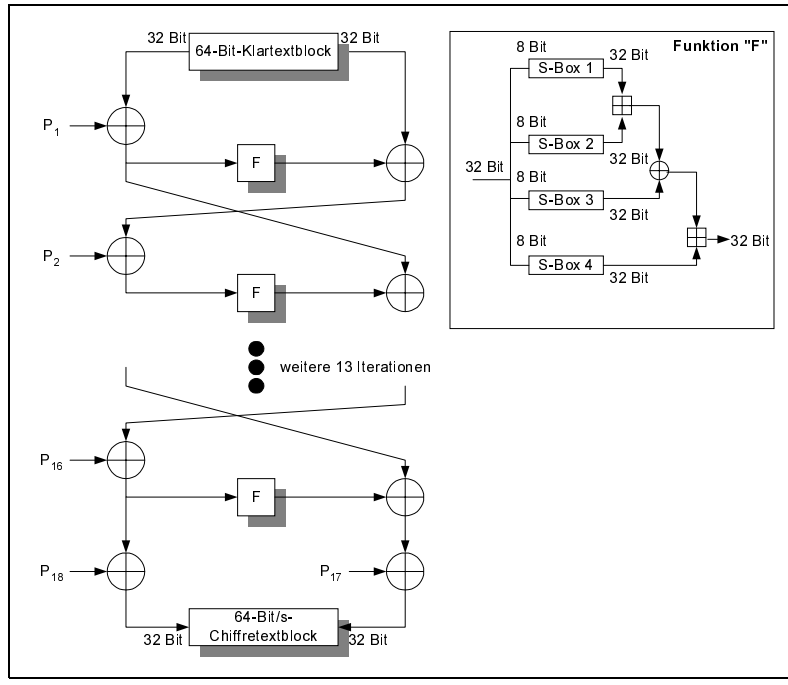
Wie die meisten Blockalgorithmen besteht auch Blowfish aus einem Feistel-Netzwerk. Das heißt, für die Entschlüsselung muss man lediglich die Schlüssel in umgekehrter Reihenfolge anwenden. Bei einem Feistel-Netzwerk wird ein Datenblock X (Klartextblock) der Länge n in die zwei Hälften L und R aufgeteilt (siehe Abb. 2.5). Bei den Iterationen der Blockchiffrierung wird die Ausgabe der i-ten Runde durch die Ausgabe der (i-1)-ten Runde bestimmt:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f\left(R_{i-1}, K_i\right)$$

Ki ist dabei der Teilschlüssel der i-ten Runde, *f* eine beliebige Rundenfunktion. Da *Li-1* mit der Ausgabe der Rundenfunktion XOR verknüpft wird, gilt ebenfalls:

$$L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(R_{i-1}, K_i) = L_{i-1}$$

Abb. 2.5 Funktionalität des Blowfish-Algorithmus



Ein Algorithmus mit diesen Eigenschaften ist umkehrbar, wenn die Eingabewerte der Funktion *f* in jeder Runde rekonstruiert werden können. Dabei spielt die Beschaffenheit von *f* keine Rolle. Dieses Konzept ist ebenfalls in CAST und DES implementiert sowie in anderen Blockchiffrierungen. Blowfish verwendet die 32-Bit-Hälften eines jeden 64-Bit-Blocks. In der Abb. 2.5 (*XL*) werden die 32-Bit-Blöcke in vier 8-Bit-Viertel zerlegt (*a, b, c, d*). Diese werden dann in vier S-Boxen durch 32-Bit-Blöcke substituiert. Die bei Blowfish verwendete Funktion lautet:

$$F(X_L) = \left((S_{1,a} + S_{2,b} \bmod 2^{32}) \oplus S_{3,c}\right) + S_{4,d} \bmod 2^{32}$$

Für die Entschlüsselung werden *P1-P18* in der umgekehrten Reihenfolge verwendet. [GORE99]

Data Encryption Standard (DES)

Auch das Verfahren Data Encryption Standard (DES) ist eine Blockchiffrierung. Hier werden die Daten in Blöcken von 64 Bit in 16 Runden verschlüsselt und als 64-Bit-Chiffretext ausgegeben. Die offizielle Beschreibung des Standards wurde 1977 unter der Bezeichnung Federal Information Processing Standard (FIPS) veröffentlicht. DES verwendet einen 56-Bit-Schlüssel. Wird der Schlüssel als 64-Bit-Zahl bzw. 129-Bit-Zahl bei Triple DES ausgedrückt, wird jedes achte Bit für eine Paritätsprüfung verwendet und sonst ignoriert.

Nach einer Eingangspermutation wird der 64-Bit-Block in zwei 32-Bit-Blöcke zerlegt. Danach folgen 16 Runden identischer Operationen (die Funktion *f*), in der die Daten mit dem Schlüssel kombiniert werden. Den Anschluss bildet die zur Eingangspermutation inverse Schlusspermutation. In jeder Runde werden die Bits des Schlüssels verschoben (Schlüsselpermutation) und 48 aus den 56 Bit des Schlüssels ausgewählt (Kompressionspermutation). Die rechte Hälfte der Daten *R0* nach der Abb. 2.6 wird durch Expansionspermutation auf 48 Bit verbreitert, mit den 48 verschobenen und rotierten Bit des Schlüssels *K1* XOR-verknüpft und durch 8 S-Boxen geschickt, die 32 neue Bit erzeugen. Diese neuen 32 Bit werden wiederum permutiert. Die letzten vier Operationen bilden zusammen die Funktion *f*. Nachdem *R0* die Funktion *f* durchlaufen hat, werden die resultierenden 32 Bit mit der linken Hälfte der Daten *L0* XOR verknüpft. Abschließend werden linke und rechte Hälfte des Datenblocks vertauscht und bilden die Eingangsblöcke für die nächste Runde. Nach der letzten Runde werden rechte und linke Hälfte nicht mehr vertauscht. Dadurch erhält man die in Abb. 2.6 gezeigte symmetrische Struktur, mit der man sowohl verschlüsseln als auch entschlüsseln kann. Damit gilt hier ebenfalls:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f\left(R_{i-1}, K_i\right)$$

Es handelt sich demnach auch hier um ein Feistel-Netzwerk. Für die Entschlüsselung muss man lediglich die Schlüssel in umgekehrter Reihenfolge anwenden.

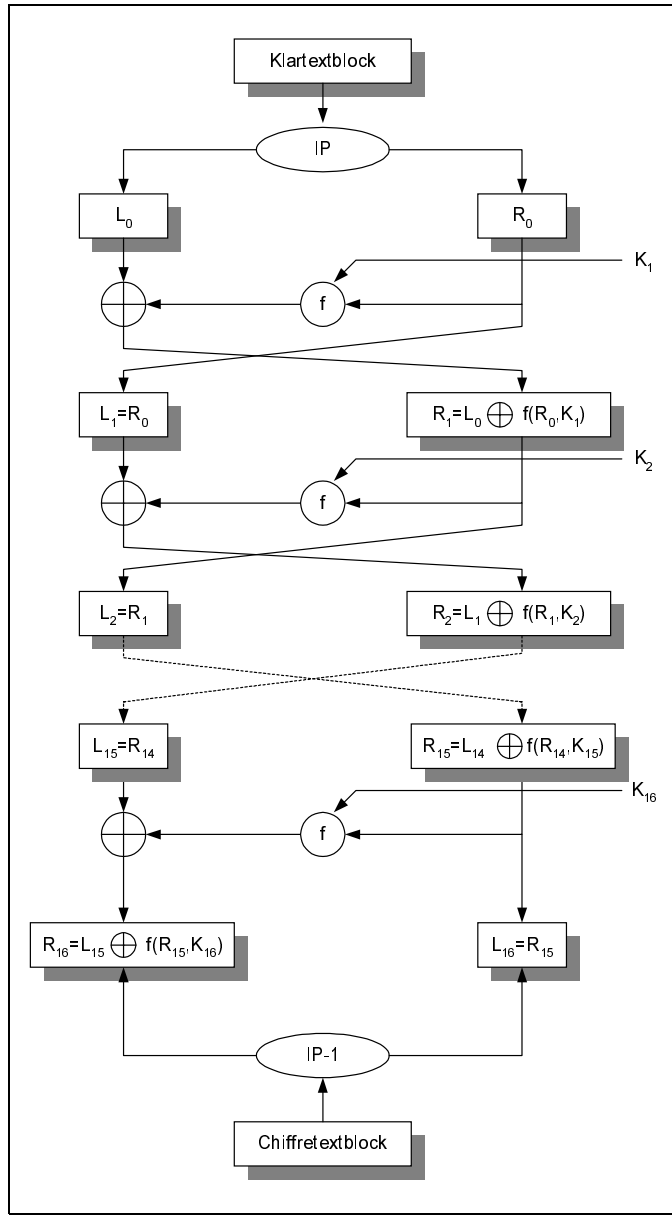
Triple-DES ist hingegen ein Verfahren zur dreifachen Verschlüsselung, welches mit zwei 56-Bit-Schlüsseln arbeitet. Dabei handelt es sich um ein so genanntes Encrypt Decrypt Encrypt (EDE)-Schema, welches in der ersten Stufe den Klartext mit dem ersten Schlüssel chiffriert. In der zweiten Stufe wird DES im Entschlüsselungsmodus mit dem zweiten Schlüssel ausgeführt. Abschließend erfolgt wieder eine Verschlüsselung mit dem ersten Schlüssel. [GORE99]

International Data Encryption Algorithm (IDEA)

Der Algorithmus International Data Encryption Algorithm (IDEA) verwendet gegenüber Blowfish einen 128-Bit-Schlüssel. Daraus werden Daten in Blöcken zu 64 Bit verschlüsselt. IDEA wurde von Xuejia Lai und James Massey am Swiss Federal Institute of Technology entwickelt und ist in Europa sowie den USA

patentrechtlich geschützt. Das heißt, er muss für patentrechtliche Anwendungen lizenziert werden. Patentinhaber ist die Ascom Tech AG in der Schweiz, die über die E-Mail-Adresse idea@ascom.ch kontaktiert werden kann.

Abb. 2.6 Funktionalität des DES-Algorithmus



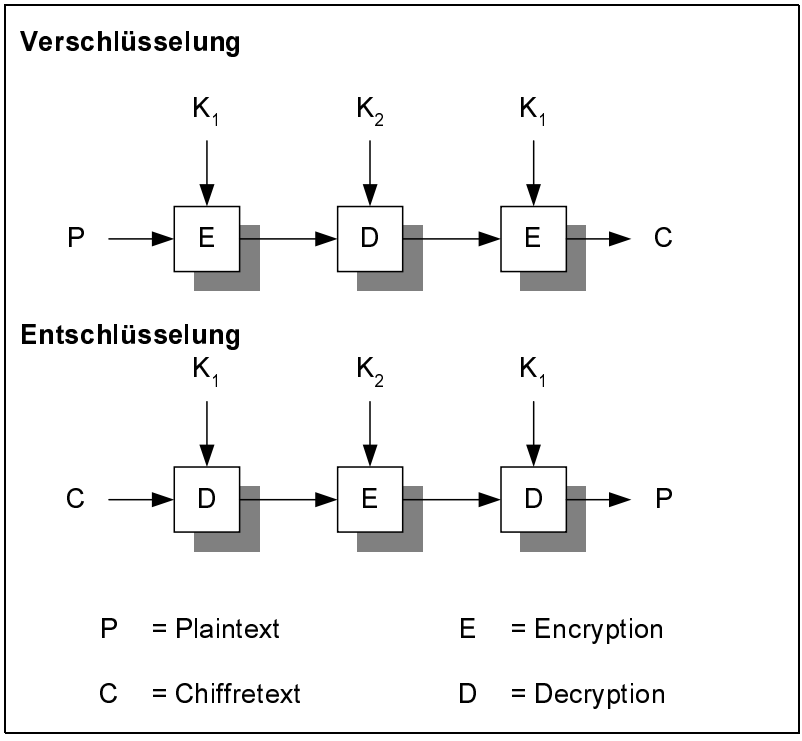


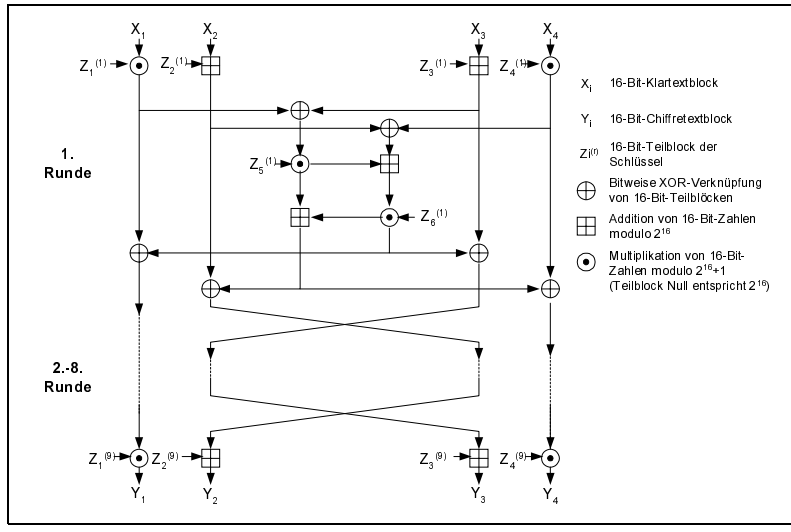
Abb. 2.7 Funktionsweise des Triple-DES-Algorithmus

IDEA selbst besteht aus 8 Iterationen und einer abschließenden Ausgabetransformation. Ein 64-Bit-Eingangsblock wird in vier 16-Bit-Unterblöcke zerlegt. Jede der acht Iterationen berechnet daraus vier 16-Bit-Ausgangsblöcke, die als Eingabe für die nächste Runde dienen. In jeder Iteration werden die vier Teilblöcke untereinander und mit sechs Teilschlüsseln der Länge 16 Bit mit XOR verknüpft, addiert und multipliziert. Zischen den Runden werden der zweite und dritte Teilblock vertauscht. Die abschließende Ausgabetransformation berechnet ebenfalls vier 16-Bit-Blöcke, die zum 64-Bit-Chiffretextblock zusammengefasst werden. Für jede der acht Iterationen werden sechs, für die Ausgangstransformation nochmals vier Teilschlüssel von je 16 Bit Länge benötigt. Diese 52 Teilschlüssel werden direkt aus dem 128-Bit-Benutzerschlüssel abgeleitet. Die ersten acht Teilschlüssel werden dabei direkt aus dem User Key entnommen, während anschließend der User Key um 25 Bit nach links im Kreis verschoben wird und die nächsten acht Unterschlüssel gebildet werden. Dieser Vorgang wird so lange wiederholt, bis alle 52 Teilschlüssel erstellt sind. Der Entschlüsselungsprozess läuft nach dem gleichen Prinzip ab, wobei sich allerdings die Reihenfolge der Teilschlüssel umdreht und diese das additive oder multiplikative Inverse der Chiffreschlüssel ist. [SCHN96]

IDEA stellt eine Alternative zu DES dar und besitzt einen wesentlich größeren Schlüsselraum. Im Gegensatz zu DES verzichtet IDEA auf S-Boxen und Permutationen und arbeitet stattdessen mit drei arithmetischen Elementaroperationen:

- XOR-Funktion,
- Addition modulo 2^{16} und
- Multiplikation modulo $2^{16} + 1$.[3]

Abb. 2.8 Funktionsweise des IDEA-Algorithmus



In der Abb. 2.8 wird der Aufbau des Verschlüsselungsalgorithmus gezeigt. Es ist eine Runde sowie die Output-Transformation dargestellt. Ein 64-Bit-Eingangsklartextblock *X* wird in vier 16-Bit-Teilblöcke X_1, X_2, X_3 und X_4 zerlegt. Danach folgen die acht identischen Runden, die jeweils mit den 16-Bit-Teilschlüsseln $Zi^{(1)}$ bis $Zi^{(6)}$ arbeiten. Dabei gibt der Index *r* die Nummer der Runde an (r = 1, 2...9). Für die Output-Transformation werden die 16-Bit-Teilschlüssel $Z_1^{(9)}$ bis $Z_6^{(9)}$ verwendet. Dadurch werden wie erwähnt 52 Teilschlüssel eingesetzt.

IDEA arbeitet mit den Betriebsarten für Blockchiffren zusammen. Bis heute sind keine Schwächen des Algorithmus bekannt bzw. veröffentlicht worden. Er hat sich als immun gegen die Methode der differentiellen Kryptoanalyse bewiesen. Zwar fand J. Daemen heraus, dass es 2^{51} schwache Schlüssel[4] für IDEA gibt, was jedoch bei einem Schlüsselraum von 2^{218} zu vernachlässigen ist, sodass IDEA als sicher angesehen wird. [JACH97]

3 Wobei der Operand 0 als 2^{16} angesehen wird.

4 Unter einem Weak Key versteht man einen Schlüssel, der bei Verschlüsselung zum einen bestimmte Regelmäßigkeiten erkennen lässt und zum anderen den Grad der Verschlüsselung abschwächt.

CAST-128

Der CAST-Algorithmus stammt aus Kanada und wurde von Carlisle Adams und Stafford Tavares entwickelt. Der Algorithmus verschlüsselt 64-Bit-Klartextblöcke mit 64-Bit-Schlüsseln bzw. 128-Bit-Schlüsseln bei CAST-128. CAST ist patentrechtlich angemeldet und muss für seinen Einsatz beantragt werden. Die Struktur von CAST ähnelt den Strukturen der bisher beschriebenen Algorithmen. Er besteht aus acht Runden. Auch bei CAST wird der Eingangsdatenblock in eine 32 Bit große linke und rechte Hälfte zerlegt. Die rechte Hälfte durchläuft eine Funktion *f*, welche sechs S-Boxen enthält, in denen sie mit zwei Hälften eines Teilschlüssels kombiniert wird. Der 32-Bit-Block wird in vier 8-Bit-Viertel *a, b, c, d* zerlegt, während ein 16-Bit-Teilschlüssel in zwei 8-Bit-Blöcke *e* und *f* aufgeteilt wird. Jede der sechs S-Boxen verarbeitet einen der sechs 8-Bit-Blöcke. Die Ausgaben der sechs S-Boxen werden miteinander zu einem 32-Bit-Ausgabeblock mit XOR verknüpft. Dieser Ausgabeblock wird mit der linken Hälfte der Eingangsdaten mit XOR verknüpft und bildet anschließend die neue rechte Hälfte für die nächste Runde, wie Abb. 2.9 verdeutlicht.

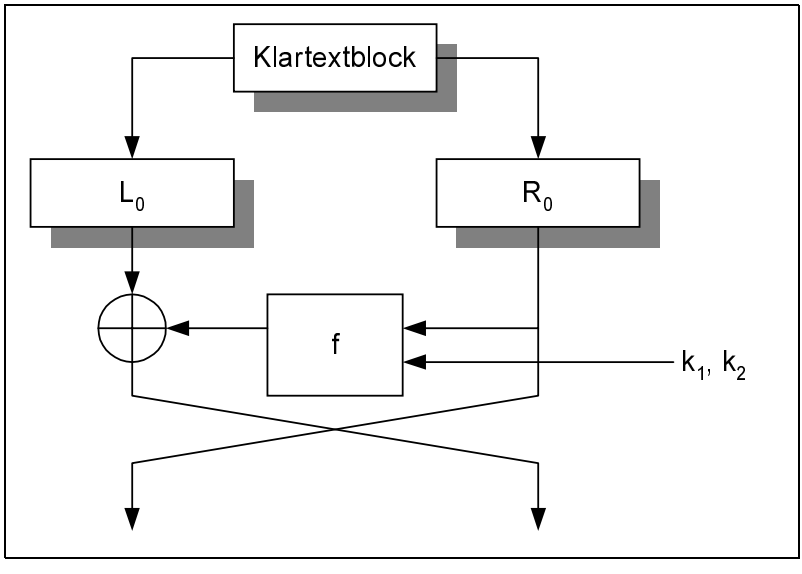


Abb. 2.9
CAST-Iteration

Die Teilschlüssel *k1* und *k2* bestehen einfach aus den ersten beiden Byte des Schlüssels. Nach der achten Runde werden rechte und linke Hälfte nicht mehr vertauscht, sodass man ebenfalls eine symmetrische Struktur erhält. Die Stärke von CAST beruht auf den nach speziellen Kriterien entwickelten S-Boxen. Dabei hängen die S-Boxen nur von der Implementierung ab und nicht vom Schlüssel, wie beispielsweise Blowfish.

RC2, RC4 und RC5 RC2 ist ebenfalls ein 64-Bit-Blockchiffre. Der Unterschied zu den anderen Verfahren liegt in der variablen Schlüssellänge. Der Algorithmus wurde von Ron Rivest für RSA Data Security Inc. (RSA/DSI) entwickelt. RC2 ist allerdings nicht patentiert und wurde auch offiziell nie veröffentlicht. Das Unternehmen stellt aber den Algorithmus zur Verfügung, wenn ein Geheimhaltungsabkommen unterschrieben wurde. Ein 128-Bit-Schlüssel wird mit Hilfe einer Permutationstabelle zunächst auf 1024 Bit expandiert und anschließend auf 128 Bit zurückgefaltet.

RC4 wurde 1987 entwickelt. Sieben Jahre lang wurde er nicht veröffentlicht und war gesetzlich geschützt. 1994 wurde der Quellcode durch ein anonymes Posting in der Cypherpunk-Mailingliste veröffentlicht und so verbreitet. RSA Data Security erhebt immer noch Ansprüche auf die Rechte des Algorithmus. Für RC4 wie auch RC2 gelten die Exportbestimmungen der USA. Diese erlauben bisher nur den Export der Chiffren für Schlüssellängen bis zu 40 Bit.

RC4 zählt zu den Stromchiffren und arbeitet ebenfalls mit Schlüsseln variabler Länge. Die Blocklänge von RC4 beträgt 8 Bit. Es wird die Betriebsart OFB[5] verwendet. Es kommt eine 8x8 S-Box mit S_0, S_1...S_{255} zum Einsatz, deren Einträge Permutationen der Zahlen von 0 bis 255 sind. Die Permutation ist eine Funktion des Schlüssels. Es gibt zwei Zähler *i* und *j*, die mit 0 initialisiert werden. Ein Byte des Schlüsselstroms wird folgendermaßen generiert:

$$i = (i+1) \bmod 256$$
$$j = (j + S_i) \bmod 256$$
$$t = (S_i + S_j) \bmod 256$$
$$K = S_t \qquad \text{S}_\text{i} \text{ wird gegen S}_\text{j} \text{ getauscht}$$

K wird nun mit dem Klartext XOR verknüpft. Für die Initialisierung der S-Box wird der Schlüssel verwendet. Zu Beginn gilt für die Elemente der S-Box $S_i = i$. Damit ist $S_0 = 0$, $S_1 = 1. . . S_{255} = 255$. Dann wird eine 256-Byte-Matrix wiederholt mit dem Schlüssel gefüllt. Somit erhält man dann K_0, K_1... K_{255}. Die 8x8 S-Box wird dann wie folgt berechnet:

$$j = 0$$
$$j = (j + S_i + K_i) \bmod 256 \text{ mit i = 0 bis 255}$$

Es wird ebenfalls wieder *Sj* mit *Si* getauscht. Der Zähler *i* bewirkt, dass jedes Element verändert wird und Zähler *j* steuert das zufällige Verändern eines Elements. RC4 kann 2^{1700} verschiedene Zuständen annehmen.

RC5 wurde ebenfalls von Ron Rivest entwickelt. RSA/DSI haben diesmal aber an ein Patent gedacht. Für die kommerzielle Nutzung muss deshalb bei

5 Output Feedback

RC5 eine Lizenz erworben werden. Blockgröße, Schlüssellänge und Anzahl der Runden sind variabel und können bei der verwendeten Implementierung über entsprechende „#defines“ eingestellt werden. [GORE99]

Secure and Fast Encryption Routine (SAFER)

Die Secure and Fast Encryption Routine (SAFER) ist von James Massey 1994 für die Cylink Corporation entworfen worden. Die Benutzung kann kostenfrei erfolgen, da keine Copyright-Rechte oder Patente vorhanden sind. Der Algorithmus gehört ebenfalls zur Familie der Blockchiffren mit einer Blockgröße von 64 Bit. Die Schlüssellänge hängt von der eingesetzten Variante ab. Weiterhin ist SAFER kein Feistel-Netzwerk, sondern eine iterierte Blockchiffrierung. Das heißt, dieselbe Funktion wird über eine bestimmte Anzahl von Runden immer wieder angewandt.

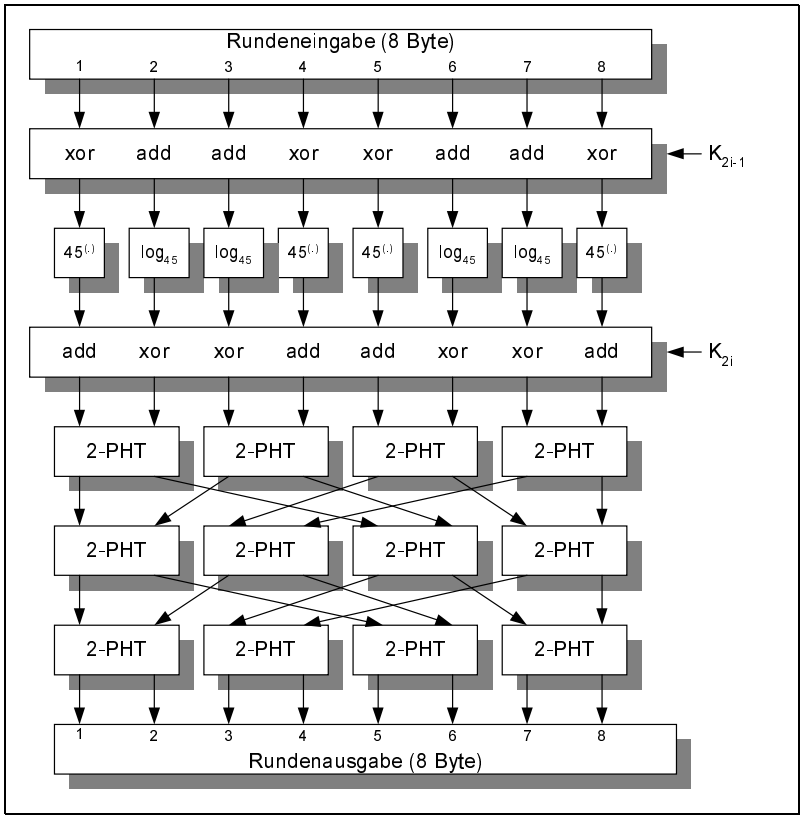


Abb. 2.10 Funktionalität des SAFER-Algorithmus

SAFER zerlegt einen 64-Bit-Eingangsdatenblock in acht 1-Byte-Blöcke. In jeder Runde werden zwei 64-Bit-Teilschlüssel verwendet. Das Besondere von SAFER ist die Verwendung der nichtlinearen Transformation (45^x mod 257 und $\log_{45}$ x) sowie der Verwendung der drei Schichten der Pseudo Hadamard Transformation (PHT). Die Funktionsweise verdeutlicht Abb. 2.10. [SCHN96]

2.1.3 Asymmetrische Verschlüsselung

Das asymmetrische Verfahren, auch Public-Key-Verfahren genannt, ist ein Verschlüsselungsverfahren, das für die Ver- und Entschlüsselung unterschiedliche Schlüssel verwendet. Es besteht aus einem öffentlichen Schlüssel, *Spub,* und einem privaten Schlüssel, *Spriv.* Für ein solches Schlüsselpaar gilt, dass das, was mit dem einen Schlüssel chiffriert wurde, nur mit dem anderen dechiffriert werden kann. Der private Schlüssel bleibt beim Besitzer und wird geheimgehalten, der öffentliche Schlüssel wird verteilt. Die Verteilung des öffentlichen Schlüssels kann im Gegensatz zu den symmetrischen Verfahren auch über unsichere Übertragungskanäle erfolgen. Da der private Schlüssel nur einmal existiert, kann nur sein Besitzer die mit den zugehörigen öffentlichen Schlüsseln chiffrierte Nachrichten dechiffrieren. Damit ist garantiert, dass zwar jeder Nachrichten chiffrieren, aber nur eine Person diese Nachrichten dechiffrieren kann.

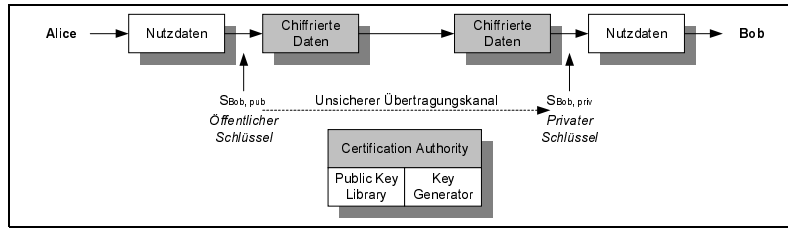
Eine asymmetrische Verschlüsselung verläuft so, wie in Abb. 2.11 gezeigt. Alice verschlüsselt ihr Dokument mit Bobs öffentlichem Schlüssel *SB,pub* und sendet es an Bob:

$$E\left(S_{B,pub}, M\right) = C$$

Bob entschlüsselt anschließend das chiffrierte Dokument mit seinem privaten Schlüssel *SB,priv*:

$$D\left(S_{B,priv}, C\right) = M$$

Abb. 2.11
Public-Key-Verfahren



Jedes Endsystem oder eine Certification Authority[6] (CA) stellt ein Schlüsselpaar für die Ver- und Entschlüsselung zur Verfügung. Solange ein System seinen privaten Schlüssel sicher verwahrt hat, ist die eingehende Kommunikation sicher. Wird der Schlüssel korrumpiert, kann das System jederzeit seinen privaten Schlüssel ändern und den passenden öffentlichen Schlüssel als Ersatz für den alten öffentlichen Schlüssel bekannt geben. Dabei muss natürlich jeder Kommunikationspartner von dem neuen Schlüssel in Kenntnis gesetzt werden.

6 Erstellt Zertifikate für öffentliche Schlüssel und bürgt damit für deren Echtheit.

Mit einem solchen Verfahren kann die Vertraulichkeit einer Nachricht bedingt gewährleistet werden, nicht aber die Authentizität einer Mitteilung, da das Verfahren nicht vor Man-in-the-middle-Attacken schützen kann.

Man-in-the-middle-Attacke

Eine asymmetrische Verschlüsselung verläuft wie in Abb. 2.12 dargestellt. Alice chiffriert ihr Dokument mit Bobs öffentlichem Schlüssel und sendet die Daten an Bob. Bob dechiffriert das chiffrierte Dokument mit seinem privaten Schlüssel. Eine Möglichkeit für einen aktiven Angreifer besteht darin, sich gegenüber Alice als Bob und gegenüber Bob als Alice auszugeben. Dieser Angriff wird als Man-in-the-middle-Attacke bezeichnet. Dabei sendet Alice ihren öffentlichen Schlüssel *SA,pub* an Bob. Die Angreiferin Mallory fängt diesen ab, ersetzt ihn durch ihren eigenen öffentlichen Schlüssel *SM,pub* und sendet ihn weiter an Bob. Bob sendet seinen öffentlichen Schlüssel *SB,pub* an Alice. Mallory fängt diesen erneut ab, ersetzt auch ihn durch ihren eigenen öffentlichen Schlüssel *SM,pub* und sendet ihn weiter an Alice. Wenn Alice Bob eine verschlüsselte Nachricht senden will, chiffriert sie ihre Nachricht mit dem vermeintlich öffentlichen Schlüssel von Bob (*SM,pub*), der in Wirklichkeit Mallory gehört, und sendet die verschlüsselte Nachricht an Bob. Mallory fängt diese ab und dechiffriert sie mit ihrem eigenen privaten Schlüssel *SM,priv*. Danach chiffriert sie die Nachricht mit Bobs öffentlichem Schlüssel *SB,pub* und sendet die verschlüsselte Nachricht weiter an Bob. Wenn Bob Alice eine verschlüsselte Nachricht senden will, chiffriert er seine Nachricht mit dem vermeintlich öffentlichen Schlüssel (*SM,pub*) von Alice, der in Wirklichkeit Mallory gehört, und sendet die verschlüsselte Nachricht an Alice. Mallory fängt diese ebenso ab und dechiffriert sie mit ihrem eigenen privaten Schlüssel *SM,priv*. Anschließend chiffriert sie die Nachricht mit Alices öffentlichem Schlüssel *SA,pub* und sendet die verschlüsselte Nachricht weiter an Alice. [DEER01]

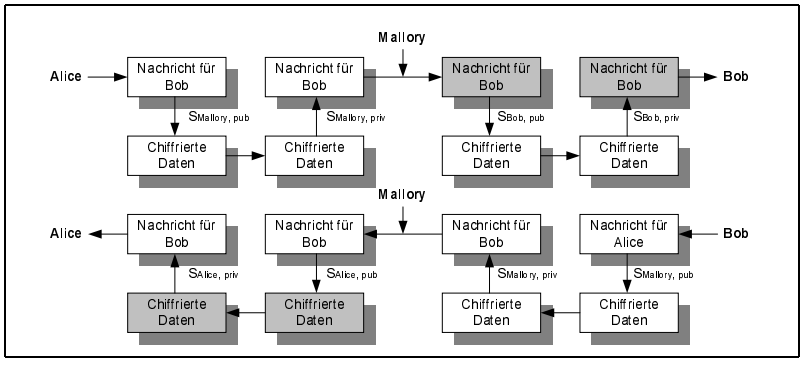


Abb. 2.12 Man-in-the-middle-Attacke

Digital Envelope

Diese Attacke macht deutlich, dass es bei der asymmetrischen Verschlüsselung unabdingbar ist, dass sich die Echtheit (Authentizität) der öffentlichen Schlüssel überprüfen lässt. Denn erst dann lässt sich ein Austausch der Schlüssel über unsichere Übertragungskanäle realisieren, ohne die Sicherheit des Verfahrens zu mindern. Da die asymmetrische Verschlüsselung wesentlich langsamer als die symmetrische ist, werden beide Verfahren in der Praxis kombiniert, um die Vorteile von beiden nutzen zu können. Dabei wird das asymmetrische Verfahren dafür genutzt, einen Schlüssel für das schnellere symmetrische Verfahren gesichert auszutauschen. Dieser Schlüssel wird Sitzungsschlüssel genannt. Der Nachteil des symmetrischen Verfahrens, dass der Schlüssel nämlich vorher erst über einen sicheren Kanal ausgetauscht werden muss, ist damit kompensiert. Das kombinierte Verfahren läuft folgendermaßen: Alice generiert einen zufälligen Sitzungsschlüssel *Ssym*. Sie chiffriert diesen Schlüssel mit Bobs öffentlichem Schlüssel *SB,pub* und sendet ihn an Bob:

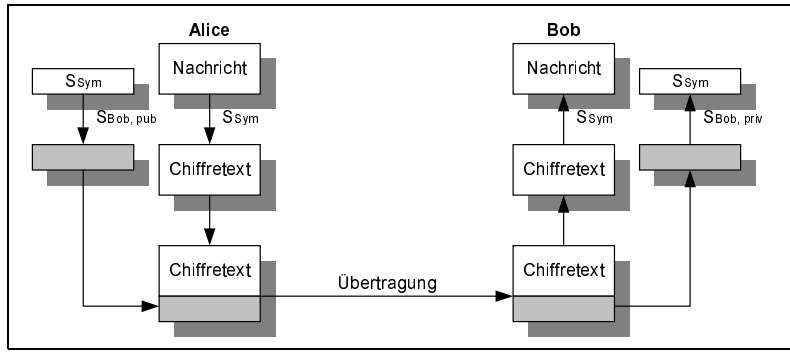
$$E_{S_{B,\,pub}}\left(S_{sym}\right) = C$$

Bob dechiffriert ihn mit seinem privaten Schlüssel *SB,priv*:

$$D_{S_{B,\,priv}}\left(C\right) = S_{sym}$$

Für den weiteren Datenaustausch während dieser Sitzung können nun beide den Sitzungsschlüssel *Ssym* verwenden.

Abb. 2.13 Digital Envelope



Soll kein Datenaustausch in beiden Richtungen erfolgen, kann man nach Abb. 2.13 den Schlüssel *Ssym* dazu verwenden, die Nachricht zu verschlüsseln. Der Schlüssel *Ssym* wird dabei ebenso mit dem öffentlichen Schlüssel des Empfängers (Bob) chiffriert. Beides wird versendet. Mit Hilfe des privaten Schlüssels lässt sich der Schlüssel *Ssym* dechiffrieren, mit dem sich dann die Nachricht entschlüsseln lässt. Dieses Verfahren wird als Digital Envelope bezeichnet. [DEER01]

Authentifizierung

Ein weiterer Vorteil des asymmetrischen Verfahrens besteht darin, dass es Möglichkeiten zur Authentifizierung bietet. Dabei wird ein Dokument mit dem privaten Schlüssel chiffriert statt mit dem öffentlichen Schlüssel und mit dem öffentlichen dechiffriert. Das asymmetrische Verfahren, um das es ging, wird also umgekehrt. Dies dient nicht der Geheimhaltung, denn jeder, der in Besitz des öffentlichen Schlüssels ist, kann das Dokument dechiffrieren und damit auch lesen. Der Vorzug liegt darin, dass derjenige, der das Dokument mit dem öffentlichen Schlüssel dechiffriert, davon ausgehen kann, dass nur der Besitzer des privaten Schlüssels das Dokument chiffriert hat. Dies kann somit zur Authentifizierung verwendet werden. Auf diesem Verfahren aufbauend, lassen sich digitale Signaturen bilden, die nicht nur die Authentizität, sondern wie eine echte Unterschrift auch die Verbindlichkeit eines Dokuments gewährleisten können.

Doch es besteht weiterhin das Problem, dass man die Gewissheit haben muss, dass der öffentliche Schlüssel wirklich zu der Person gehört, die vorgibt, den zugehörigen privaten Schlüssel zu besitzen. Sonst lässt sich zwar prüfen, ob das Dokument mit dem zugehörigen privaten Schlüssel chiffriert wurde, es ist aber fraglich, wer sich hinter diesem Schlüssel verbirgt. Dies ist ein allgemeines Problem bei der asymmetrischen Verschlüsselung. Aus diesem Grund werden Institutionen eingesetzt, die für die Echtheit eines Schlüssels bürgen. Eine solche Institution fungiert als so genannte Trusted Third Party, eine unabhängige dritte Partei, der beide Kommunikationspartner vertrauen. Die Verwundbarkeit dieser Institutionen ist ein Nachteil der asymmetrischen Verfahren. Trust Center oder Certification Authorities (CAs) bieten einen oder mehrere der folgenden Dienste an:

- Schlüsselmanagement: Generierung, Revocation (Rückruf), Personalisierung von Personal Security Environments (PSE)
- Beglaubigung (Registrierung und Zertifizierung, Time Stamping)
- Serverfunktionen (Public-Key-Verzeichnis, Authentisierungsserver etc.)
- Treuhänderfunktion (Key Escrow, Key Recovery)

Aus Effizienzgründen wird nicht das gesamte Dokument mit dem privaten Schlüssel chiffriert. Es wird mit Hilfe einer Einweg-Hash-Funktion, auch Message Digest genannt, eine Kurzkennung des Dokuments generiert. Der so entstandene Message Digest wird dann mit dem privaten Schlüssel chiffriert. Wie schon aus dem Namen hervorgeht, haben die Einweg-Hash-Funktionen die Eigenschaften der Einwegfunktionen. Das heißt, sie lassen sich nicht wieder rückgängig machen.

Digitale Signatur

Das asymmetrische Verfahren kann ebenfalls zur Erstellung einer digitalen Signatur verwendet werden. Das heißt, ein Dokument wird unverschlüsselt, aber signiert übertragen. Wenn dies der Fall ist, generiert Alice mit Hilfe einer Mes-

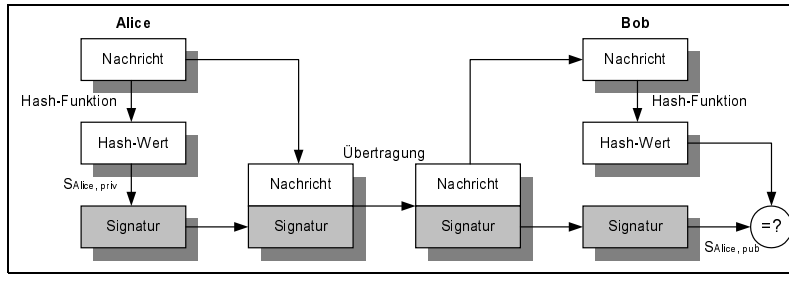
sage-Digest-Funktion einen Message Digest ihres Dokuments. Diesen chiffriert sie mit ihrem privaten Schlüssel *SA,priv*. Anschließend hängt sie den chiffrierten Message Digest an das Dokument und sendet es an Bob. Bob muss dann wiederum den Message Digest vom Dokument trennen und mit dem öffentlichen Schlüssel *SA,pub* von Alice dechiffrieren. Danach generiert Bob ebenfalls mit der gleichen Message-Digest-Funktion einen Message Digest und vergleicht ihn mit dem von Alice. Sind beide gleich, stammt das Dokument von Alice und wurde nicht verändert.

Wenn die Verifikation der digitalen Signatur fehlschlägt, kann man entweder auf ein verändertes Dokument schließen, wodurch die Integrität nicht mehr sichergestellt ist, oder auf einen Übertragungsfehler bzw. die falsche Identität des Senders.

Die Kombination der Public- und Private-Key-Verfahren gewährleistet Authentizität, Integrität und Vertraulichkeit der Nachricht. Für den Austausch von Dokumenten oder E-Mails können asymmetrische Verschlüsselungsverfahren direkt verwendet werden. Durch solche Public-Key-Systeme ist der Einsatz von Key Management und digitaler Signatur möglich. Die Notwendigkeit, über sichere Netze zu kommunizieren, entfällt somit.

Die Public-Key-Verfahren sind aufgrund ihrer Abstammung aus der Komplexitätstheorie sehr rechenintensiv. Das heißt, sie sind relativ langsam und nicht für die Verschlüsselung großer Datenmengen geeignet. So kann man davon ausgehen, dass Hardware-Implementierungen von RSA ca. 1000 Mal langsamer sind als vergleichbare Implementierungen von DES. Aus diesem Grund ist der Einsatz einer Kombination von symmetrischen und asymmetrischen Verfahren sinnvoll.

Abb. 2.14
Digitale Signatur



Die hybriden Verfahren haben genau diese Möglichkeit integriert. Bei einem hybriden Verfahren wird über ein Public-Key-Verfahren ein Sitzungsschlüssel für die Nutzdatenverschlüsselung sicher übermittelt. Dazu wird der Sitzungsschlüssel über eine Zufallszahl generiert und vom Absender mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. Der Empfänger entschlüsselt den Session Key mit seinem privaten Schlüssel. Die Nutzdatenverschlüsselung wird dabei über ein symmetrisches Verfahren realisiert.

RSA

RSA ist ein asymmetrisches Verfahren zur Verschlüsselung und Authentifizierung. Es wurde 1977 von Ron Rivest, Adi Shamir und Leonard Adleman erfunden und nach seinen Entwicklern benannt. Die Sicherheit des Algorithmus basiert auf der Schwierigkeit, größere Zahlen zu faktorisieren. So ist es einfach, zwei große Zahlen zu multiplizieren, aber die Umkehrung, um wieder die Faktoren zu erhalten, gestaltet sich ab einer gewissen Größe als nicht trivial. Dieser Vorgang lässt sich somit als Einwegfunktion ansehen. Bei RSA wird aus der Multiplikation zweier großer Primzahlen p und q der Modulus berechnet:

$$n = q \cdot p$$

Dieses n ist nun der Modulus[7]. Es wird weiterhin ein öffentlicher Exponent e gewählt, der teilerfremd mit $(p\text{-}1)(q\text{-}1)$ ist. Diese Bedingung muss eingehalten werden, da sonst kein modulares Inverses existiert. Mit Hilfe des erweiterten euklidischen Algorithmus wird nun ein d bestimmt:

$$e \cdot d \equiv 1 (\mathrm{mod}((p-1)(q-1)))$$

Die Bedingung, dass der Modulus und m teilerfremd sein müssen, ist für den RSA-Algorithmus nicht notwendig. Das Zahlenpaar (n, e) ist der öffentliche und d der private Schlüssel. Die Primzahlen p und q müssen unter Verschluss gehalten oder zerstört werden, da sich aus ihnen der private Schlüssel d berechnen lässt. Eine Nachricht m wird folgendermaßen verschlüsselt:

$$c = m^e \bmod n$$

Das Ergebnis dieser Verschlüsselung c wird folgendermaßen berechnet und somit wieder verschlüsselt:

$$m = c^d \bmod n$$

Nur die Person, die d kennt, kann Nachrichten dechiffrieren. Um die Effizienz des Verfahrens zu steigern, wird der öffentliche Exponent klein gewählt, ohne jedoch die Sicherheit des Algorithmus zu beeinträchtigen. Dies hat zur Folge, dass die Operationen mit dem öffentlichen Schlüssel schneller als die mit dem privaten ablaufen. Somit ist das Verschlüsseln schneller als das Entschlüsseln und das Verifizieren schneller als das Signieren. Natürlich darf nur der öffentliche Exponent klein gewählt werden und nicht der private. Denn sonst ließe sich die Suche nach dem privaten Schlüssel stark einschränken, was die Sicherheit mindern würde. Interessant ist, dass der Algorithmus nicht dadurch unsicherer

7 Der Operand n bei der Modulo-Operation $a \bmod n = b$ wird als Modulus bezeichnet.

wird, wenn man den öffentlichen Exponenten *e* immer gleich wählt. Zwei beliebte Werte für *e* sind 3 und 65537.

Es gibt verschiedene Ansätze, um RSA zu brechen. Beispielsweise kann man versuchen, sich Zugang zum privaten Schlüssel zu verschaffen oder einen direkten Angriff auf die RSA-Implementierung vorzunehmen. Ebenfalls wäre es denkbar, dass man mathematisch über die n-te Wurzel und modulo-Verfahren die Nachricht zu dechiffrieren versucht. Der erste Punkt ist klar verständlich, da es als Voraussetzung gilt, den geheimen Schlüssel sicher aufzubewahren. Die beiden anderen Punkte sind jedoch interessanter. Gelingt es, ein Verfahren zu finden, mit dem sich die n-te Wurzel über Modulo-Verfahren ziehen ließe, so könnte man chiffrierte Nachrichten dechiffrieren. Doch bis jetzt sind keine allgemeinen Verfahren bekannt. Der zweite Punkt zeigt die Notwendigkeit auf, dass ein wirklich sicheres RSA auch einer sicheren Implementierung bedarf.

Die Größe für den Modulus hängt von den Sicherheitsbedürfnissen der Anwender ab. Je länger der Modulus ist, desto größer wird die Sicherheit, aber desto langsamer werden auch die Operationen. Momentan wird angenommen, dass 512-Bit-Schlüssel mit dem Erscheinen eines neuen Faktorisierungsalgorithmus und dem Einsatz verteilter Systeme keine ausreichende Sicherheit mehr bieten. Nach 1998 sollten diese Schlüssel bereits nicht mehr verwendet werden. Die Vorschläge der RSA Laboratories für die Länge der Schlüssel sind folgendermaßen:

- **Personal Use:** 768 Bit
- **Corporate Use:** 1024 Bit
- **Extremley Valuable Keys:** 2048 Bit

Unter der letztgenannten Schlüssellänge versteht man beispielsweise Schlüssel einer Institution, die Zertifikate für Schlüssel vergibt. Mit diesen Zertifikaten soll die Identität eines Besitzer eines öffentlichen Schlüssels überprüft werden können. Es wird angenommen, dass die 768-Bit-Schlüssel bis zum Jahr 2004 sicher sind. Im Allgemeinen hat jeder Schlüssel nur eine bestimmte Gültigkeitsdauer. Diese Einschränkung der Lebenszeit eines Schlüssels soll dazu dienen, den Zeitraum für eine Kryptoanalyse möglichst gering zu halten. Nach Ablauf dieser Zeit sollte dann ein neuer Schlüssel generiert werden, der um eine angemessene Anzahl von Bits länger ist, um die Entwicklung der Hardware und der Faktorisierungsalgorithmen in dieser Zeit zu berücksichtigen. [DEER01]

Challenge Response Protocol

Das Challenge Response Protocol ist ein Authentifizierungsprotokoll, das heißt, es hat ausschließlich die Aufgabe zu prüfen, ob die Kommunikationspartner diejenigen sind, die sie zu sein vorgeben. Ein weiterer Zweck besteht darin, die Kommunikationspartner in die Lage zu versetzen, Sitzungsschlüssel (Session Keys) einzurichten, die anschließend für die Verschlüsselung von Nachrichten verwendet werden können. Für jede neue Verbindung kann dann ein neuer sowie zufällig gewählter Sitzungsschlüssel verwendet werden. Es gibt heute eine

große Anzahl von solchen Protokollen, die zum Teil sehr komplex sind. Einige arbeiten mit Key Distribution Center (KDC), wie beispielsweise die Protokolle Wide Mouth Frog, Needham Schroeder oder Otway Rees.[8] Die anderen Arten funktionieren mit einem Authentifizierungsserver und einem Ticket-Ausgabe-Center wie Kerberos.

Das Challenge Response Protocol funktioniert hingegen folgendermaßen. Im ersten Schritt sendet der Teilnehmer *A* mit der Setup-Nachricht seine Kennung bzw. Identität an den Teilnehmer *B*. Dieser Teilnehmer wählt anschließend eine große Zufallszahl R_B (z.B. 128 Bit) und sendet sie als Nachricht 2 an den ersten Teilnehmer zurück (Challenge). Teilnehmer *A* verschlüsselt die Zufallszahl mit dem Shared Secret Key K_{AB} und sendet den Chiffretext $K_{AB}(R_B)$ an den Teilnehmer *B* zurück (Response). Damit hat sich Teilnehmer *A* authentifiziert, was einer einseitigen Maßnahme entspricht. Um eine gegenseitige Authentifizierung zu erhalten, wählt Teilnehmer *A* ebenfalls eine 128-Bit-Zufallszahl R_A aus und sendet diese an Teilnehmer *B*. Dieser verschlüsselt auch diese Zufallszahl mit demselben Shared Secret Key K_{AB} und sendet den Chiffretext $K_{AB}(R_A)$ an Teilnehmer *A* zurück. Nun haben sich beide Teilnehmer authentifiziert. Soll anschließend ein Sitzungsschlüssel für den folgenden Datenaustausch eingerichtet werden, kann man diesen mit dem Shared Secret Key K_{AB} verschlüsselt weiterleiten. [GORE99]

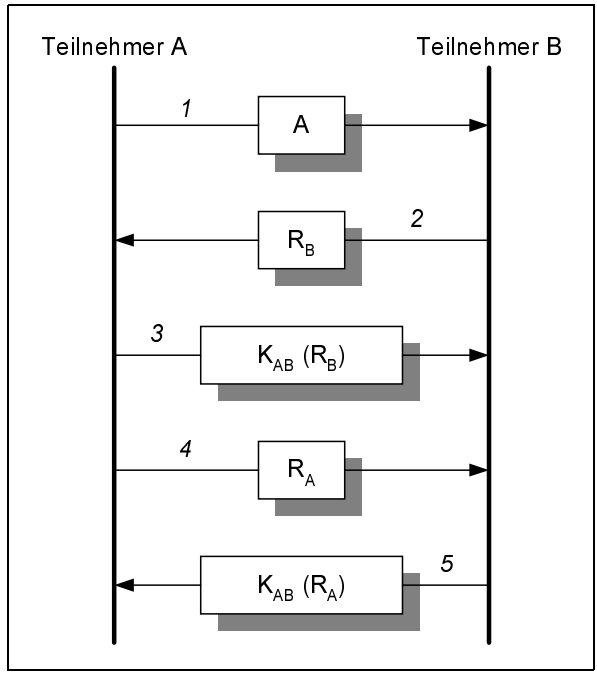


Abb. 2.15 Funktionsweise des Challenge Response Protocol

8 Zu mehr Details siehe [CCITT87]

Diffie-Hellman Der Algorithmus Diffie-Hellman ist ebenfalls ein ausschließliches Authentifizierungsprotokoll, welches bereits 1976 veröffentlicht wurde. Er basiert auf der Komplexität der Berechnung diskreter Logarithmen. Diffie-Hellman ermöglicht die Definition eines gemeinsamen geheimen Schlüssels, den nur die beiden Teilnehmer kennen, auch wenn sie über einen unsicheren Kanal miteinander kommunizieren. Dieser geheime Schlüssel wird dann zur Verschlüsselung der Daten verwendet und beinhaltet einen Verschlüsselungsalgorithmus, über den sich beide Kommunikationspartner einigen müssen.

Zunächst einigen sich die Kommunikationspartner auf eine große Primzahl *p* und eine Ganzzahl *g*, die eine einfache Wurzel von *p* ist. Teilnehmer *A* wählt eine große Ganzzahl *XA* (z.B. 512 Bit) und überträgt Folgendes an den Teilnehmer *B*:

$$Y_A = g^{X_A} \bmod p$$

Der Teilnehmer *B* wählt entsprechend eine große Ganzzahl X_B und sendet den folgenden Wert an den Teilnehmer *A*:

$$Y_B = g^{X_B} \bmod p$$

Beide Teilnehmer sind nun in der Lage, einen Schlüssel *K* zu berechnen, da nach dem Gesetz der modularen Arithmetik Folgendes gilt:

$$(Y_B)^{X_A} \bmod p = (Y_A)^{X_B} \bmod p = K$$

Dadurch ist ein geheimer Schlüssel geschaffen worden, der bedenkenlos über ein ungesichertes Medium wie das Internet übertragen werden kann. Ein möglicher Angreifer kann nur die Werte *p*, *g*, Y_A und Y_B auslesen. Um daraus den Schlüssel zu berechnen, müsste er den diskreten Logarithmus berechnen und X_A oder X_B ermitteln. Die Sicherheit liegt nun darin begründet, dass im Gegensatz zur relativ einfachen Ermittlung exponentieller Modulo einer Primzahl kein praktischer Algorithmus für die Berechnung diskreter Logarithmen von Primzahlen bekannt ist, die mehrere hundert Bits lang sein können. Die Sicherheit hängt allerdings ebenfalls von der Wahl von *p* und *g* ab. Die Zahl *p* sollte möglichst groß gewählt werden, wobei *(p-1)/2* ebenfalls eine Primzahl ergeben sollte.

Ein Nachteil von Diffie-Hellman ist allerdings seine Anfälligkeit für Man-in-the-middle-Attacken. Solche Angriffe können allerdings unterbunden werden, wenn beide Teilnehmer ihre öffentlichen Werte mit einer digitalen Signatur versehen haben. Das heißt, Diffie-Hellman lässt sich geschickt mit Hash-Funktionen oder digitalen Signaturen koppeln. Weiterhin ist Diffie-Hellman nicht auf zwei Kommunikationspartner begrenzt. Es lassen also auch Konferenzen mit drei oder mehr Parteien einsetzen.

Das Zusammenspiel mit einer Public Key Infrastructure (PKI) bietet noch mehr Flexibilität bezüglich der sicheren Kommunikation. Wenn öffentliche Schlüssel Y_i in einer zentralen Datenbank so gewählt wurden, dass für den öffentlichen Schlüssel des Anwenders *A* gilt:

$$Y_A = g^{X_A} \bmod p$$

Dann kann sich ein anderer Teilnehmer, der mit dem Anwender *A* kommunizieren will, diesen Schlüssel aus der Datenbank besorgen, den Session Key *K* berechnen und Anwender *A* eine verschlüsselte Nachricht übermitteln. Anwender *A* besorgt sich anschließend den Public Key des anderer Teilnehmers ebenfalls aus der Datenbank und kann nun seinerseits den Session Key *K* berechnen. Dadurch ist er in der Lage, die Nachricht zu entschlüsseln. Beide Teilnehmer verfügen nun über einen gemeinsamen Session Key, ohne vorher direkt in Kontakt getreten zu sein. Bei der PKI-Datenbank müssen allerdings beide Schlüssel zertifiziert sein. [GORE99]

2.1.4 Hash-Funktion

Es wurde bereits die Möglichkeit beschrieben, wie durch zweifache Anwendung eines Public-Key-Verfahrens die Integrität einer Nachricht überprüft werden kann. Um dies durchzuführen, verwendet man eine Einweg-Hash-Funktion. Diese Funktion wird auch als Fingerabdruck oder kryptographische Prüfsumme bezeichnet. Einweg-Hash-Funktionen haben die Aufgabe, einen Eingabe-String variabler Länge in einen wesentlich kürzeren Ausgabe-String fester Länge umzuwandeln. Dies ist der eigentliche Hash-Wert. Es gibt Algorithmen, für die kein Schlüssel erforderlich ist. Einweg-Hash-Funktionen werden auch für die digitale Signatur verwendet. Hier wird die Hash-Funktion des Dokuments mit dem privaten Schlüssel des Absenders verschlüsselt.

Das Besondere an einer Einweg-Hash-Funktion ist, dass sie nur in einer Richtung funktioniert. Dabei lassen sich folgende Eigenschaften beschreiben:

- Es ist nicht möglich, zu einem bestimmten Hash-Wert ein passendes Original zu finden.
- Zwei verschiedene Originale dürfen nicht den gleichen Hash-Wert ergeben, das heißt, es muss Kollisionsfreiheit gegeben sein.
- Die Ausgabe darf nicht nachvollziehbar von der Eingabe abhängen.
- Die Änderung eines einzigen Bits der Eingabe muss einen völlig anderen Hash-Wert ergeben.

Einweg-Hash-Funktionen sind beispielsweise Secure Hash Algorithm (SHA), Message Digest 5 (MD5) oder RIPE-MD 128/160. Codes zur Authentifizierung von Nachrichten sind so genannte Message Authentication Codes (MAC) – also Hash-Funktionen – mit einem zusätzlichen geheimen Schlüssel. Der Hash-Wert hängt in diesem Fall dann sowohl von der Eingabe als auch vom Schlüssel ab.

Es ist ebenfalls möglich, symmetrische Blockchiffren als Einweg-Hash-Funktion zu verwenden. Dabei wird die Nachricht in den Modi Cipher Block Chaining (CBC) oder Cipher Feedback (CFB) mit einem festen Schlüssel und einem Initialisierungsvektor verschlüsselt. Der letzte Chiffretextblock liefert dabei den Hash-Wert, welcher noch einmal im CBC- oder CFB-Modus verschlüsselt wird. [DEER01]

Funktionsmerkmale

Um den Begriff der Hash-Funktion deutlicher zu fassen, lässt sich folgende Feststellung treffen. Eine Hash-Funktion H erzeugt für eine beliebig lange Nachricht M einen festen Hash-Wert:

$$h = H(M)$$

Dabei werden folgende Anforderungen gestellt:

1. Der Hash-Wert h ist leicht zu berechnen.
2. Es ist praktisch unmöglich, für einen gegebenen Hash-Wert X eine Nachricht M zu finden, die den gleichen Hash-Wert liefert: $H(M) = X$
   Eine Hash-Funktion, die diese Anforderung erfüllt, wird schwache Hash-Funktion genannt. Ein Paar verschiedener Eingabewerte, die den gleichen Hash-Wert liefern, bezeichnet man als Kollision. Die zweite Anforderung beinhaltet, dass man zu einer gegebenen Nachricht M keine zweite Nachricht M' findet, die den gleichen Hash-Wert hat:

   $$H(M) = H(M')$$

   Dies schließt jedoch nicht aus, dass es irgendeine Kollision gibt. Für starke Hash-Funktionen müssen die Anforderungen enger gefasst werden:
3. Es ist quasi unmöglich, ein Paar verschiedener Eingabewerte zu finden, die eine Kollision ergeben.

Hash-Funktionen, die die erste und dritte Anforderung erfüllen, werden kollisionsfrei genannt. In der Regel werden Hash-Funktionen durch eine Folge gleichartiger Kompressionsfunktionen realisiert. Eine Kompressionsfunktion liefert für eine Eingabe fester Länge eine kürzere Ausgabe ebenfalls fester Länge. So wird, wie in Abb. 2.16 gezeigt, blockweise aus der Nachricht ein Hash-Wert erzeugt. Die Integrität einer Nachricht lässt sich mit einer Hash-Funktion dadurch schützen, dass ein Hash-Wert der Nachricht erzeugt wird. Wird bei einem späteren Test ein anderer Hash-Wert erzeugt, so wurde die Integrität der Nachricht verletzt. Wird der Hash-Wert noch mit einem privaten Schlüssel chiffriert, lässt sich ebenfalls die Authentizität der Nachricht überprüfen.

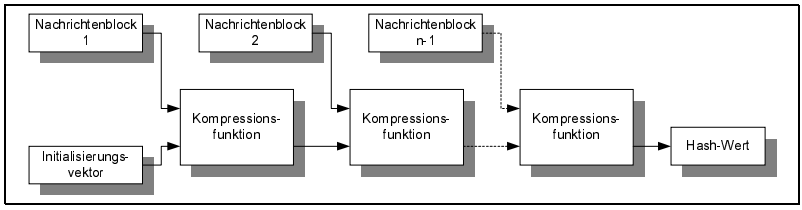


Abb. 2.16 Aufbau einer Hash-Funktion

Message Digest 2 (MD2)

MD2 ist eine von Ron Rivest entworfene Hash-Funktion, die einen 128-Bit-Hash-Wert erzeugt. Dieser Hash-Wert wird auch Message Digest genannt. Der Algorithmus wurde speziell für Anwendungen der digitalen Signatur vorgesehen. Das Erzeugen des Message Digests erfolgt in fünf Schritten. Eine Nachricht, die als Eingabe für die MD2-Hash-Funktion dienen soll, wird als Reihenfolge von Byteblöcken angesehen. Im ersten Schritt wird die Nachricht so aufgefüllt, dass ihre Länge in Bytes ein ganzzahliges Vielfaches von 16 ist. Das Auffüllen, das so genannte Padding, erfolgt ebenfalls, wenn die Länge vorher schon ein Vielfaches von 16 ist. So werden minimal 1 Byte und maximal 16 Byte an die Nachricht angehängt. Im zweiten Schritt wird der Nachricht eine 16 Byte lange Prüfsumme angehängt. Zur Erzeugung der Prüfsumme wird eine 256-Byte-Permutation verwendet, die auf den Ziffern einer Zahl basiert, die aus einer Substitutionstabelle gewonnen wird. Im dritten Schritt wird die Initialisierung eines 48-Byte-Blocks, dem so genannten MD-Puffer, vorgenommen. Abschließend wird im vierten Schritt die Nachricht in 16-Byte-Blöcken verarbeitet. Dazu verwendet sie auch die Permutation der Substitutionstabelle. Der letzte Schritt liefert die Ausgabe. Der Message Digest befindet sich in den ersten 16 Blöcken eines MD-Puffers. Bis jetzt haben sich noch keine Schwächen der MD2-Hash-Funktion gezeigt. Sie ist jedoch langsamer als andere Hash-Funktionen.

Message Digest 5 (MD5)

MD5 ist ebenfalls ein von Ron Rivest entwickelter Message-Digest-Algorithmus, der einen 128-Bit-Hash-Wert erzeugt. MD5 stellt eine modifizierte Version des Message Digest Algorithmus MD4 dar, von dessen Einsatz abgeraten werden muss. Bei der Entwicklung des MD5-Algorithmus wurde darauf geachtet, dass er effizient auf Rechnern mit 32-Bit-Architektur läuft. Der Aufbau von MD5 weicht von dem von MD2 ab. Die Erzeugung eines Message Digest erfolgt aber ebenfalls in fünf Schritten. Eine Nachricht wird dabei als beliebig lange Bitreihenfolge m_0, $m_1 ... m_{b-1}$ angesehen[9]. Dies ist ein Unterschied zum MD2-Algorithmus, bei dem die Nachricht als Reihenfolge von Byteblöcken angesehen wird. Ein 32-Bit-Datenwort wird beim MD5-Algorithmus so interpretiert, dass das Low-order-, Least-significant-Byte zuerst kommt.

9 *b* entspricht der Länge der Nachrichten in Bit.

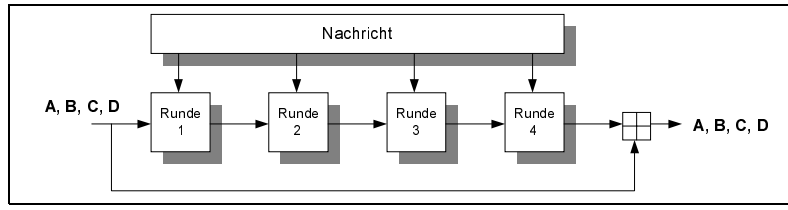


Abb. 2.17 MD5-Hauptschleife

Im ersten Schritt werden Padding-Bits an die Nachricht angefügt, dabei wird die Nachricht so erweitert, dass für die neue Länge *b'* gilt:

$$b' \bmod 512 = 64 \quad \text{dabei ist} \quad b' = b + z$$

Der Anzahl der Padding-Bits entspricht *z*. Die Nachricht wird immer erweitert, auch wenn für *b* schon *b mod 512 = 64* gelten sollte. So werden minimal 1 Bit und maximal 512 Bit an die Nachricht angehängt. Das erste Bit der Erweiterung hat den Wert 1, die folgenden den Wert 0. Im nächsten Schritt wird die 64-Bit-Darstellung der Länge *b* an die Nachricht angehängt. Sollte die Nachricht länger als 2^{64} Bit lang sein, werden die Low-order-64-Bit verwendet. Die Nachrichtlänge hat dann ein Vielfaches von 512 Bit und somit natürlich auch ein Vielfaches von 16 Datenwörtern der Länge 32 Bit. Die Nachricht lässt sich durch M_0, $M_1 ... M_{N-1}$ darstellen, wobei *N* ein Vielfaches von 16 ist. M_i hat eine Länge von 32 Bit. Der dritte Schritt dient der Initialisierung eines MD-Puffers, der aus vier Puffern besteht. Jeder Puffer wird mit einem 32-Bit-String initialisiert:

$$A = 01234567_{16}$$
$$B = 89abcdef_{16}$$
$$C = fedcba98_{16}$$
$$D = 76543210_{16}$$

Die Nachricht wird im vierten Schritt in 16-Datenwörter-Einheiten verarbeitet. Dies findet in der Hauptschleife des Algorithmus statt, siehe Abb. 2.17. Die Hauptschleife besteht aus vier Runden, in denen jeweils eine unterschiedliche nichtlineare Funktion 16 Mal angewendet wird[10]:

$$F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$$

10 $\wedge$ steht für die AND-Verknüpfung, $\vee$ für die OR-Verknüpfung und $\neg$ für den NOT-Operator

In jeder Runde wird eine unterschiedliche Operation angewendet, die jeweils eine der nicht linearen Funktionen enthält, siehe Abb. 2.18. Unter *s* soll ein zyklischer Links-Shift um s-Bit verstanden werden. T_i stellt hingegen den Wert der folgenden Funktion dar:

$$2^{32} \cdot abs(\sin(i))$$

Die i-Schleife stellt die Hauptschleife dar und stellt sicher, dass jeder 512-Bit-Block bzw. jede Einheit von 16 Datenwörtern berücksichtigt wird. Die innere j-Schleife arbeitet dann jedes Datenwort einer Einheit in aufsteigender Folge ab. Für *i* und *j* gelten dabei:

$$i = 0 \ldots \frac{N}{16} - 1$$
$$j = 0 \ldots 15$$
$$X_j = M_{i \cdot 16 + j}$$

Der letzte Schritt dient der Ausgabe. Der Message Digest setzt sich aus den Inhalten der Puffer *A*, *B*, *C* und *D* zusammen. Dabei wird die Ausgabe entsprechend der Vorgabe für den MD5-Algorithmus so interpretiert, dass mit dem Low-order-Byte von Puffer *A* begonnen und beim High-order-Byte von Puffer *D* geendet wird.

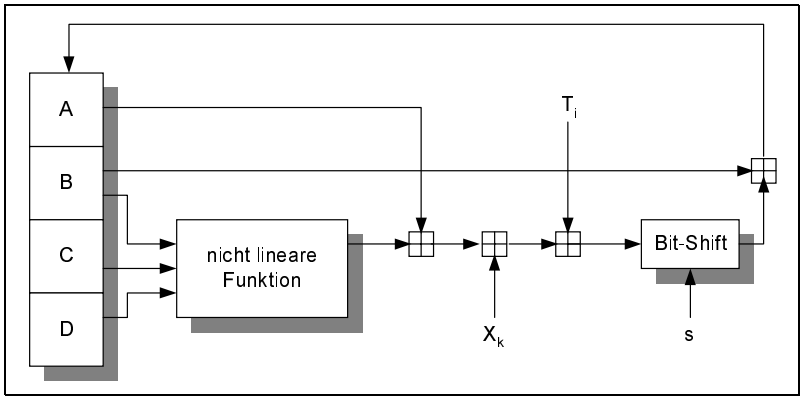


Abb. 2.18 MD5-Operation

Beim MD5-Algorithmus wurden Pseudokollisionen gefunden. Unter diesen Pseudokollisionen versteht man Kollisionen, die bei den Kompressionsfunktionen auftreten. Dabei werden mit unterschiedlichen Initialisierungen gleiche Hash-Werte erzeugt. Pseudokollisionen könnten hilfreich sein, echte Kollisionen zu finden. Im Allgemeinen gelten sie nicht als gravierende Schwäche. Unter dieser Voraussetzung wird MD5 noch als sicher angesehen. Es ist jedoch eine gewisse Vorsicht geboten. [DEER01]

Secure Hash Algorithm (SHA)

SHA ist in der Spezifikation Secure Hash Standard (SHS) festgeschrieben. Dieser Standard wurde für den Einsatz im Digital Signature Standard (DSS) entwickelt. Eine überarbeitete Version von SHA heißt SHA-1. SHA ist der MD4-Familie ähnlich. SHA arbeitet mit Eingaben der Länge bis zu 2^{64} Bit und erzeugt einen 160-Bit-Hash-Wert. Die Limitierung der Eingabelänge stellt keine große Einschränkung dar, weil die Größe der Eingabe immer noch weit jenseits der TByte liegt: 2^{64} Bit = 2^{21} TB. Das Padding erfolgt wie in MD5. Die Nachricht wird so aufgefüllt, dass ihre Länge modulo 512 genau 64 ergibt. Dabei hat das erste Padding-Bit den Wert 1, die folgenden den Wert 0. Auch das Anhängen einer Prüfsumme verläuft genau wie in MD5. So wird die 64-Bit-Darstellung der Länge der ursprünglichen Nachricht angehängt. Im nächsten Schritt werden fünf 32-Bit-Variablen initialisiert: [DEER01]

$$A = 67452301_{16}$$
$$B = \text{efcdab89}_{16}$$
$$C = \text{98badcfe}_{16}$$
$$D = 10325476_{16}$$
$$E = \text{c3d2e1f0}_{16}$$

Danach folgt die Hauptschleife des Algorithmus. Hier wird die Nachricht in 512-Bit-Einheiten verarbeitet. Die Hauptschleife besteht aus 4 Runden mit jeweils 20 Operationen. Zuerst werden die Variablen *A, B, C, D, E* in den Variablen *a, b, c, d, e* gespeichert. Jede Operation enthält dabei eine nicht lineare Funktion, die auf den Variablen *a, b, c, d, e* arbeitet und wie folgt definiert ist:

$$f_T(X,Y,Z) = \begin{cases} (X \wedge Y) \vee ((\neg X) \wedge Z) & t = 0...19 \\ X \oplus Y \oplus Z & t = 20...39 \\ (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z) & t = 40..59 \\ X \oplus Y \oplus Z & t = 60...79 \end{cases} \quad \textit{für}$$

Weiterhin wird eine Konstante K_t definiert:

$$K_t = \begin{cases} \text{5a827999}_{16} & t = 0...19 \\ \text{6ed9eba1}_{16} & t = 20...39 \\ \text{8f1bbcdc}_{16} & t = 40...59 \\ \text{ca62c1d6}_{16} & t = 60...79 \end{cases} \quad \textit{für}$$

Dann werden für jeden 512-Bit-Block die 16 32-Bit-Datenwörter mittels einer Expansionsabbildung zu 80 32-Bit-Datenwörtern transformiert. Dies wird durch den folgenden Algorithmus erreicht[11]:

$$W_t = M_t \qquad t = 0...15$$
$$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-16}) \lll 1 \quad \textit{für} \quad t = 16...79$$

Abb. 2.19 stellt einen der 80 Verarbeitungsschritte der Hauptschleife dar, die nach folgender Vorschrift abgearbeitet werden:

$$TEMP = (a \lll 5) + f_t(b,c,d) + e + W_t + K_t$$
$$e = d$$
$$d = c$$
$$c = b \lll 30 \qquad \textit{für} \quad t = 0...79$$
$$b = a$$
$$a = TEMP$$

Im Anschluss daran werden die Variablen *a* zu *A*, *b* zu *B*, *c* zu *C*, *d* zu *D* und *e* zu *E* addiert. Die Hauptschleife des Algorithmus wird entsprechend der Anzahl der 512-Bit-Einheiten wiederholt. Der 160-Bit-Hash-Wert von SHA befindet sich am Ende des Algorithmus in den fünf 32-Bit-Variablen *A*, *B*, *C*, *D* und *E*. [JACH97]

Aufgrund des längeren Hash-Wertes ist SHA unempfindlicher gegen Brute-Force-Attacken als Hash-Funktionen mit 128-Bit-Hash-Werten. SHA stellt eine modifizierte Version von MD5 dar. Bis jetzt sind keine erfolgreichen kryptographischen Angriffe gegen SHA bekannt. Man kann ihn daher bedenkenlos einsetzen.

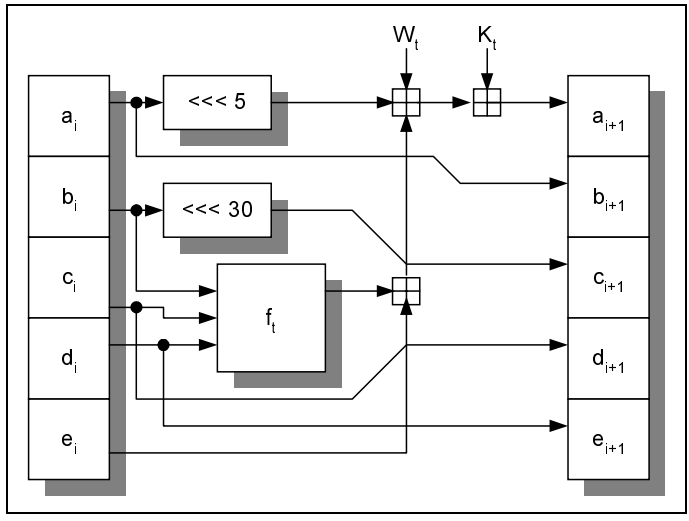


Abb. 2.19 SHA-Operation

11 Der zyklische Links-Shift um ein Bit (<<< 1) ist nicht eigentlicher Bestandteil von SHA; er wurde zur Erhöhung der Sicherheit eingeführt

2.2 Key Management

Nachdem die verschiedenen Verschlüsselungsverfahren erläutert worden sind, bleibt die grundsätzliche Frage offen, wie die extrahierten Schlüssel innerhalb einer Benutzergruppe bzw. zwischen den Komponenten verteilt werden. Das Key Management beantwortet diese Frage und spielt für die Sicherheit der asymmetrischen Verschlüsselungsverfahren eine entscheidende Rolle, da sich Angriffe auf Public-Key-Systeme meistens nicht gegen das zugrunde liegende kryptographische Verfahren, sondern gegen die Ebene des Key Managements richten. Folgende Basisanforderungen werden daher an das Key Management gestellt:

- Einem Benutzer muss es möglich sein, auf sichere Art und Weise ein Schlüsselpaar zu erhalten, das den Ansprüchen an Effizienz und Sicherheit genügt.
- Es muss eine Möglichkeit geben, die öffentlichen Schlüssel anderer Benutzer zu erhalten und seine eigenen zu veröffentlichen.
- Es muss gewährleistet sein, dass der Benutzer Vertrauen in den öffentlichen Schlüssel seines Kommunikationspartners haben kann.
- Es muss die Möglichkeiten bestehen, öffentlich bekannt zu geben, dass ein Schlüssel verloren oder kompromittiert[12] wurde.
- Ein Benutzer muss in der Lage sein, seinen privaten Schlüssel sicher aufzubewahren.
- Schlüssel sollten nur bis zu einem bestimmten Verfallsdatum (Expiration Date) gültig sein.

Neben der Schlüsselverwaltung müssen Zertifikate vorhanden sein, um das Vertrauen in den öffentlichen Schlüssel des Kommunikationspartners gewährleisten zu können. Die Basisanforderungen lauten hier:

- Zertifikate müssen fälschungssicher und auf sichere Art und Weise zu erhalten sein.
- Sie müssen so gestaltet sein, dass sie von keinem Angreifer missbraucht werden können.
- Die Verteilung der Zertifikate muss sicher gegen Angreifer sein.

Aufgrund möglicher kryptoanalytischer Angriffe muss die Gültigkeitsdauer von Schlüsseln begrenzt werden. Dadurch wird verhindert, dass einem Angreifer durch die ständig wachsende Zahl von verschlüsselten Texten, die mit dem gleichen Schlüssel chiffriert wurden, Rückschlüsse auf den Schlüssel gestattet werden. Dies kann ab einer bestimmten Menge durchaus passieren. Hinzu

12 Kompromittierte Schlüssel sind Schlüssel, auf die ein Angreifer auf irgendeine Weise Zugriff erlangt hatte, sodass er in der Lage ist, die verschlüsselten Nachrichten zu lesen oder selber zu verschlüsseln.

kommt, dass natürlich die Schlüssellänge an den aktuellen Stand der Kryptoanalyse und Computertechnik angepasst werden muss. Ein anderer Grund besteht darin, den Schaden, den ein kompromittierter Schlüssel verursachen kann, zu minimieren. Denn in der Regel erfährt der Benutzer nichts davon, dass sein Schlüssel kompromittiert wurde.

Folgende Phasen werden während einer Gültigkeitsdauer eines Schlüssels im allgemeinen abgearbeitet:

- Schlüsselgenerierung und mögliche Registrierung
- Verteilung des öffentlichen Schlüssels
- Ersetzen oder Update des Schlüssels
- Rückruf des Schlüssel
- Ablauf der Gültigkeit des Schlüssels

Für die Schlüsselgenerierung gibt es zwei verschiedene Ansätze. Bei dem ersten Ansatz generiert der Benutzer sein Schlüsselpaar lokal selbst, während dies beim zweiten Ansatz von einer zentralen Instanz für ihn übernommen wird. Bei der ersten Möglichkeit wird vorausgesetzt, dass das Schlüsselpaar von einer vertrauenswürdigen und sicheren Software innerhalb einer gesicherten Umgebung generiert wird. Entsprechend muss es sich bei der zweiten Möglichkeit um eine vertrauenswürdige und sichere Instanz handeln.

Die Registrierung des Schlüssel wird von einer Certification Authority (CA) bzw. einem Trust Center übernommen. Ein Zertifikat bestätigt dabei die Echtheit des öffentlichen Schlüssels. Für die Verteilung der Schlüssel gibt es ebenfalls wieder mehrere Möglichkeiten. Die Verteilung kann persönlich erfolgen, das heißt, man teilt jeder Person, mit der man kommunizieren möchte, seinen öffentlichen Schlüssel mit. Dies ist allerdings eine sehr umständlichste Variante, da man jede Person explizit kontaktieren muss, um seinen Schlüssel auszuhändigen oder ihren Schlüssel zu erhalten. Eine weitere Möglichkeit ist es, öffentliche Schlüssel zentral zu speichern, sodass sie jeder abrufen kann. So könnte beispielsweise ein CA einen Server zur Verfügung stellen, von dem aus die Schlüssel abgerufen werden können. An diese Server werden natürlich hohe Sicherheitsansprüche gestellt, da die Schlüsselverzeichnisse und deren Einträge sicher gegen Verfälschungen sein müssen. In Kombination mit Zertifikaten können die Sicherheitsvorkehrungen gesenkt werden, da die Echtheit des öffentlichen Schlüssels durch das beiliegende Zertifikat gewährleistet wird.

Sollte der Verdacht auftreten, dass der private Schlüssel kompromittiert wurde, ist es erforderlich, sofort ein neues Schlüsselpaar zu generieren. Wenn der öffentliche Schlüssel mit einem Zertifikat versehen wurde, muss des Weiteren auch die CA davon in Kenntnis gesetzt werden, damit der alte Schlüssel auf eine so genannte Certificate Revocation List (CRL) gesetzt wird. Diese Listen enthalten Zertifikate, die widerrufen wurden. Diese Zertifikate werden vor dem Ablauf ihrer Gültigkeit für ungültig erklärt. Vor dem Überprüfen einer Signatur

kann es somit sinnvoll sein, sich zu vergewissern, ob das zugehörige Zertifikat nicht widerrufen wurde. Da abgelaufene Zertifikate auf keinen Fall akzeptiert werden sollten, enthalten CRL nur Zertifikate, die zwar widerrufen wurden, aber noch nicht abgelaufen sind. Wenn ein widerrufenes Zertifikat abläuft, wird es von der CRL entfernt.

Um einer Signatur eines Dokuments auch nach dem Ablauf des öffentlichen Schlüssels Gültigkeit zu verleihen, können Signaturen mit Zeitstempeln, so genannten Timestamp versehen werden, die Aussagen über den Erstellungszeitpunkt der Signatur geben. Liegt dieser vor dem Ablauf der Gültigkeit des öffentlichen Schlüssels, so hat die Signatur noch Gültigkeit. Im Allgemeinen sind die Gültigkeitsdauern für öffentliche Schlüssel und die der zugehörigen Zertifikate identisch.

2.2.1 Funktionsweise

Bevor Verschlüsselung eingesetzt werden kann, muss also ein angepasstes Schlüsselmanagement initiiert werden. Berücksichtigt werden muss dabei Folgendes:

1. **Schlüsselgenerierung:** Die Auswahl der Schlüssel muss sich am eingesetzten Verfahren orientieren. Schlüssel dürfen nicht leicht zu erraten oder rekonstruierbar sein (analog zum Passwort). Für eine effektive Schlüsselwahl eignen sich insbesondere Zufallszahlengeneratoren. Auch muss sichergestellt werden, dass bei der Installation des Verschlüsselungsverfahrens voreingestellte Schlüssel geändert werden.
2. **Aufbewahrung und Hinterlegung:** Der Vertraulichkeitsschutz durch Verschlüsselung kann nur dann erreicht werden, wenn die verwendeten kryptographischen Schlüssel geheim gehalten werden können. Bieten die IT-Systeme, auf denen das Verschlüsselungsverfahren eingesetzt ist, keinen ausreichenden Zugriffsschutz für die Schlüssel, sollten diese nicht auf diesem IT-System gespeichert werden. Besser ist eine bedarfsorientierte manuelle Eingabe. Um jedoch zu vermeiden, dass die Schlüssel zu diesem Zweck aufgeschrieben werden, bieten sich entweder so genannte Tamperproof-Boxen an – das sind Geräte, in denen die Schlüssel sicher gespeichert und verarbeitet werden können – oder Chipkarten, die den Schlüssel speichern. Letztere Möglichkeit benötigt zwar auch zusätzliche Hardware, erlaubt aber die personenbezogene Benutzung von kryptographischen Schlüsseln. Werden Schlüssel nicht mehr benötigt oder verwendet, sind sie physikalisch zu löschen oder zu vernichten. Bei Bedarf ist aus Gründen der Notfallvorsorge das Hinterlegen der verwendeten Schlüssel in gesicherten Bereichen (Tresoren) vorzusehen.
3. **Übermittlung:** Die Schlüssel sollten von den verschlüsselten Daten (zeitlich und räumlich) getrennt zum Empfänger übertragen werden. Hierfür ist ggf.

ein Bote oder der Versand mittels PIN-Brief (geschwärzter Umschlag wie bei Gehaltsmitteilungen) vorzusehen. Eine Übermittlung per Telefon ist aber in vielen Fällen ausreichend.

4. **Schlüsselwechsel:** Die verwendeten Schlüssel sind abhängig von der Häufigkeit ihres Einsatzes, von dem Bedrohungspotenzial und der Sicherheit ihrer lokalen Aufbewahrung hinreichend oft zu wechseln. Besteht der Verdacht, dass ein verwendeter Schlüssel offen gelegt wurde, so ist dieser Schlüssel nicht mehr zu verwenden und alle Beteiligten sind zu informieren. Bereits mit diesem Schlüssel verschlüsselte Informationen sind zu entschlüsseln und mit einem anderen Schlüssel zu verschlüsseln.

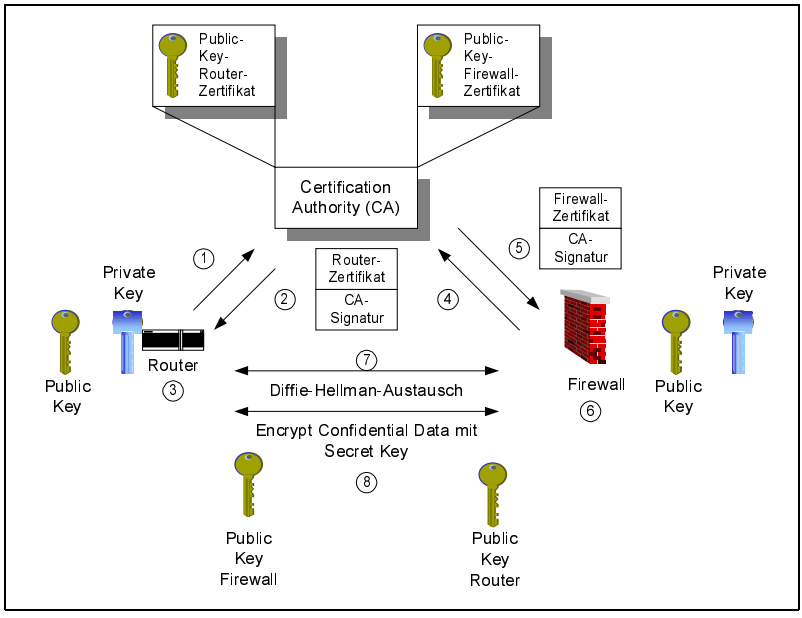


Abb. 2.20 Sichere Kommunikation über Certification Authority

Abb. 2.20 zeigt die sichere Kommunikation über eine Certification Authority (CA). Der Router und die Firewall besitzen hier jeweils ein Paar von öffentlichen/privaten Schlüsseln. Dabei wird davon ausgegangen, dass es der CA gelungen ist, ein X.509-Zertifikat auf sichere Weise der Firewall/Router zu hinterlegen. Firewall und Router haben somit auf sichere Weise eine Kopie des öffentlichen Schlüssels der CA erhalten. Der Router will nun Datenverkehr zur Firewall mit Authentifizierung etablieren, um eine vertrauliche Zustellung der Daten gewährleisten zu können. Der Router sendet der CA eine Anforderung des öffentlichen Schlüssels der Firewall. Anschließend sendet die CA das Zertifikat der Firewall verschlüsselt mit ihrem eigenen privaten Schlüssel. Der Router entschlüsselt das Zertifikat mit dem öffentlichen Schlüssel des CA, um den öffentlichen Schlüssel der Firewall zu erhalten. Die Firewall sendet der CA eine

Anforderung des öffentlichen Schlüssels des Routers. Dann sendet die CA das Zertifikat des Routers, verschlüsselt mit seinem eigenen privaten Schlüssel. Die Firewall entschlüsselt wiederum das Zertifikat mit dem öffentlichen Schlüssel der CA, um den öffentlichen Schlüssel des Routers zu erhalten. Router und Firewall durchlaufen den Diffie-Hellman-Austausch mit Verschlüsselung durch öffentliche Schlüssel, um den Austausch zu authentifizieren. Mit Hilfe des vom Austausch abgeleiteten geheimen Schlüssels tauschen Router und Firewall vertrauliche Daten aus.

Ein weiterer wichtiger Punkt bei kommunizierenden Gruppen ist das Schlüsselmanagement bei Multicast, da hier wesentlich mehr Datenverkehr als bei Unicast erzeugt wird. Dies liegt daran, dass der gemeinsame Schlüssel an jedes Gruppenmitglied und an alle Sender verteilt werden muss. Soll auch die Authentizität gesichert werden, muss jeder Sender seinen Authentifizierungsschlüssel an alle Mitglieder verteilen. Man sieht bereits, dass sich dabei asymmetrische Verfahren noch weniger eignen, da jeder Sender den öffentlichen Schlüssel jedes Empfängers haben müsste. Außerdem haben asymmetrische Verfahren eine zu schlechte Performance. Bei einem Netzdurchsatz von 100 Mbit/s kann selbst eine leistungsfähige Workstation nicht jedes Paket mit einer RSA-Ver- bzw. Entschlüsselung versehen. Selbst Hardwareimplementierungen kommen in den meisten Fällen nicht über einen oder mehrere Mbit/s hinaus. Die einfachste Lösung ist ein symmetrisches Verfahren in Verbindung mit dem Einsatz eines Gruppenverantwortlichen. Mitglieder, die ein- oder austreten wollen, senden ihre Anforderung an den Gruppenverantwortlichen und der generiert und verteilt alle benötigten Schlüssel.

Ideale Verschlüsselungssysteme für Multicast-Gruppen sollten für jeden Teilnehmer einen Schlüssel haben. Dies wäre ein *n*-Wege-Verschlüsselungssystem. Ein Mitglied, das einen Schlüssel kennt, muss in der Lage sein, Nachrichten, die mit einem der anderen Schlüssel verschlüsselt worden sind, zu entschlüsseln und umgekehrt. Es ist möglich, das normale Deffie-Hellmann-Verfahren zum Schlüsselaustausch zu erweitern.

Die Problematik von Multicast-Gruppen zeigt sich in vollem Umfang bei großen und sehr dynamischen Gruppen, das heißt, wenn viele Teilnehmer zu- und austreten. Denn es muss sichergestellt werden, dass neue Mitglieder nicht in der Lage sind, die Kommunikation vor dem Eintritt zu verstehen, und Mitglieder, die austreten, nicht der weiteren Kommunikation folgen können. Zudem muss gerade bei großen Gruppen das Key Management eine geringe Komplexität haben, um garantieren zu können, dass die aktuellen Mitglieder immer im Besitz des richtigen Schlüssels sind. Die naheliegende Lösung, bei jedem Ein- oder Austritt einen neuen Schlüssel zu generieren und an alle aktuellen Mitglieder zu verteilen, z.B. über multiplen Unicast mit Deffie-Hellmann, ist für große Gruppen nicht zu bewältigen. Die Lösung ist eine Art Versionsver-

waltung der Schlüssel: wenn ein neuer Teilnehmer hinzukommt, bekommt er den Schlüssel mit einer neuen Versionsnummer. Alle alten Teilnehmer erfahren diesen Versionswechsel durch ein normales Datenpaket, in dem diese Nummer immer mitgesendet wird und müssen nun ihren alten Schlüssel mittels einer Einweg-Hash-Funktion auf die aktuelle Version bringen. Allerdings müssen trotzdem bei Austritt eines Mitglieds alle anderen Mitglieder neue Schlüssel bekommen, um zu verhindern, dass der Austretende die weitere Kommunikation verfolgen kann. [DEER01]

2.2.2 Zertifikate

Verteilte asymmetrische Kryptosysteme benötigen als Grundlage Zertifikate, um einen öffentlichen Schlüssel *x* in Verbindung mit einer Person *y* zu bringen. Somit kann man unter einem Zertifikat ein signiertes Dokument verstehen, das den Namen des Schlüsselbesitzers und den dazugehörigen öffentlichen Schlüssel enthält. Eine zertifizierende Instanz erstellt mit ihrem privaten Schlüssel eine Signatur für das Dokument und bürgt damit für dessen Echtheit.

Durch Zertifikate ist es möglich, dass ein Benutzer Gewissheit über die Echtheit des öffentlichen Schlüssels eines unbekannten Kommunikationspartners erlangt. Das setzt allerdings voraus, dass der Benutzer Vertrauen zur zertifizierenden Instanz hat. Der Benutzer kann die Echtheit des Zertifikats über die Signatur und den öffentlichen Schlüssel der zertifizierenden Instanz prüfen. Stellt sich das Zertifikat als echt heraus, weiß der Benutzer, dass der vorliegende öffentliche Schlüssel zu der im Zertifikat angegebenen Person gehört. Nur diese Person besitzt den korrespondierenden privaten Schlüssel. An dieser Stelle kann man erkennen, dass das Ausstellen von Zertifikaten nur von vertrauenswürdigen Institutionen vorgenommen werden darf, da sonst Sicherheitslücken entstehen würden. Weiterhin sollte die zertifizierende Instanz bekannt geben, wie sie die Identität des Schlüsselbesitzers geprüft hat, das heißt, ob beispielsweise ein einfaches Schreiben genügt oder die Vorlage eines amtlichen Lichtbildausweises etc. notwendig ist. Dem Benutzer wird somit die Prüfung der Identität des Kommunikationspartners von der Instanz abgenommen. Er muss nur noch die Prüfung des öffentlichen Schlüssels der zertifizierenden Instanz übernehmen.

Am Beispiel von X.509 können Aufbau und Inhalt eines Zertifikats gezeigt werden. Der Standard X.500[13] wurde bei der ITU entwickelt und enthält Konzepte zu Verzeichnissen und Directory Services. Der Unterstandard X.509[14] enthält dabei die Spezifikationen zur Struktur von Public-Key-Zertifikaten und zur Unterstützung von Authentifizierungsdiensten zwischen zwei Clients bzw. Benutzern. Es wird dabei sowohl eine Authentifizierung über Passwort[15] als

13 The Directory: Overview of Concepts, Models and Services
14 The Directory: Authentication Framework

auch der Einsatz kryptographischer Verfahren[16] beschrieben. Das kryptographische Verfahren basiert auf Public-Key-Kryptographie. Auf X.509 bauen sowohl PKCS#6 als auch Privacy Enhanced Mail (PEM) nach RFC-1422 auf. Im Allgemeinen enthält ein Zertifikat nach X.509 folgende Informationen: [X.509(00)]

- **Version:** gibt Versionsnummer des X.509 Zertifikates an.
- **Serial Number:** ist eine einmalige Seriennummer des Zertifikats. Jedes Zertifikat wird mit einer eigenen Seriennummer versehen, die keiner der anderen gleichen darf.
- **Signature:** ist nicht die eigentliche Signatur, sondern nur das Verfahren, mit dem die zertifizierende Instanz die Signatur erstellt hat.
- **Issuer Name:** ist der Name der zertifizierenden Instanz. Über den Namen der Instanz lässt sich der richtige öffentliche Schlüssel zur Validierung des Zertifikats bestimmen.
- **Validity Period:** legt den Anfangs- und den Endpunkt der Gültigkeit des Zertifikates fest.
- **Subject Name:** ist der Name der Person bzw. Organisation o.Ä., für die das Zertifikat ausgestellt wurde. Damit wird der Name des Schlüsselbesitzers eindeutig in Verbindung mit seinem öffentlichen Schlüssel gebracht
- **Subject Public Key:** enthält zum einen den öffentlichen Schlüssel als auch den verwendeten Public-Key-Algorithmus sowie zugehörige Parameter.

Ein X.509-Zertifikat wird durch die folgende ASN.1-Darstellung spezifiziert: [KENT93]

```
Certificate ::= SEQUENCE {
CertificateInfo  CertificateInfo,
signatureAlgorithm AlgorithmIdentifier,
signature        BIT STRING
}
```

Die `CertificateInfo` enthält die eigentlichen Informationen. Diese werden dann von der zertifizierenden Instanz mit dem in `signatureAlgorithm` angegeben Algorithmus signiert und die daraus resultierende Signatur wird unter Punkt `signature` dem Zertifikat beigegeben. Die ASN.1-Spezifikation von `CertificateInfo` sieht dabei folgendermaßen aus: [KENT93]

```
CertificateInfo ::= SEQUENCE {
Version [0]              Version DEFAULT v1988,
serialNumber             CertificateSerialNumber,
signature        AlgorithmIdentifier,
issuer           Name,
validity         Validity,
subject          Name,
```

15 Simple Authentication
16 Strong Authentication

```
subjectPublicKeyInfo     SubjectPublicKeyInfo
issuerUniqueID [1] IMPLICIT UniqueIdentifier
OPTIONAL
subjectUniqueID [2]      IMPLICIT UniqueIdentifier
OPTIONAL
extensions [3]   Extensions
OPTIONAL
}
```

Über den `AlgorithmIdentifier` wird hier der verwendete Signaturalgorithmus spezifiziert. So steht für die Signatur beispielsweise MD2 mit RSA-Verschlüsselung oder SHA-1 mit RSA-Verschlüsselung zu Verfügung. Aktuell ist die Version 3 von X.509, die folgende Erweiterungen enthält: [KENT93]

```
CertificateInfo ::= SEQUENCE {
version [0]                      Version DEFAULT v1,
...
subjectPublicKeyInfo     SubjectPublicKeyInfo
issuerUniqueID [1]       IMPLICIT UniqueIdentifier
OPTIONAL
subjectUniqueID [2]              IMPLICIT UniqueIdentifier
OPTIONAL
extensions [3]           Extensions
OPTIONAL
}
Version ::= INTEGER { v1(0),v2(1),v3(2)}
Extension ::= SEQUENCE {
extnID           OBJEXT IDENTIFIER,
critical                 BOOLEAN DEFAULT FALSE,
extnValue                        OCTET STRING
}
```

Die `UniqueIdentifier` werden dazu benutzt, die Wiedervergabe von `subject`- und `issuer name` zu ermöglichen. Die `Extensions` bieten die Möglichkeit, weitere Attribute in das Zertifikat aufzunehmen. So gibt es eine Reihe von Standard-Extensions, auf die zugegriffen werden kann, die hier aber nicht weiter erläutert werden. [DEER01]

2.2.3 PKCS

Der Public Key Cryptography Standard (PKCS), ist eine Sammlung von Spezifikationen für Public-Key-Verfahren. Sie wurde von den RSA Laboratories erstellt und dient dazu, verschiedene RSA-Implementierungen verträglich bezüglich der Zusammenarbeit zu gestalten. PKCS ist kompatibel zu Privacy Enhanced Mail (PEM) und X.509. Die zur Zeit gültigen Standards sind die folgenden[17]: [JACH97]

17 PKCS#2 und #4 sind nicht mehr gültig und in #1 eingearbeitet worden

- **RSA Encryption Standard (PKCS#1):** definiert den Mechanismus der Verschlüsselung und des Signierens mit dem RSA Public-Key-Verfahren. So wird unter anderem die Syntax der öffentlichen und privaten Schlüssel sowie die der Signaturen festgelegt.
- **Diffie-Hellman Key Agreement Standard (PKCS#3):** beschreibt die Methoden für den Diffie-Hellman-Schlüsselaustausch.
- **Password-based Encryption (PKCS#5):** enthält hingegen die Methoden für die Verschlüsselung eines Oktet-Strings mit einem geheimen Schlüssel, der aus einem Passwort abgeleitet wird. Die vorrangige Aufgabe besteht darin, private Schlüssel zu chiffrieren.
- **Extended Certificate Syntax Standard (PKCS#6):** gibt die Definition für die Syntax erweiterter Zertifikate. Diese bestehen aus einem X.509-Zertifikat und einigen Zusätzen.
- **Cryptographic Message Snytax Standard (PKCS#7):** erläutert die allgemeine Syntax für Dokumente, die kryptographische Erweiterungen wie digitale Signaturen oder Verschlüsselungen enthalten bzw. auf die kryptographische Verfahren angewendet wurden.
- **Private Key Information Standard (PKCS#8):** definiert ein Format für Private-Key-Informationen. Dieses umfasst einen privaten Schlüssel für ein Public-Key-Verfahren und optionale Zusätze.
- **Selected Attribute Type Standard (PKCS#9):** Hier werden die Zusätze, die in anderen PKCS-Standards verwendet werden, beschrieben.
- **Certification Request Syntax Standard (PKCS#10):** befasst sich mit der Syntax der Requests für Zertifikate.
- **Cryptographic Token Interface Standard (PKCS#11):** beschreibt eine technologie-unabhängige Programmierschnittstelle für kryptographische Dienste und Anwendungen, wie z.B. Smart-Cards. Enthalten sind Schnittstellen zu Kryptobibliotheken mit den ausführenden Algorithmen und Schlüsselmanagementfunktionen, aber auch von Anwendungen zu einem Cryptography Service Provider [GORE99].

Da die Sammlung der Standards sehr ausführlich ist, soll hier speziell nur gezeigt werden, wie die Darstellung der Schlüssel und der Mechanismus der Verschlüsselung bzw. Entschlüsselung definiert sind.

Der Standard PKCS#1 beschreibt die Syntax der Schlüssel mit Hilfe von Abstract Syntax Notation One (ASN.1). Die Transfersyntax von ASN.1 definiert, wie Werte von ASN.1-Typen eindeutig[18] in eine Bytefolge zur Übertragung konvertiert werden. Die angewandte Transfersyntax ist in Basic Encoding Rules (BER) definiert. Die Codierregeln basieren auf dem Prinzip, dass jeder übertragene Wert aus vier Feldern besteht: Bezeichner (Typ oder Tag), Länge

18 zur eindeutigen DeKodierung

des Datenfeldes in Byte, Datenfeld und Inhaltsende-Flag, falls die Datenlänge unbekannt ist. Für die Schlüssel gilt damit:

```
RSAPublicKey              ::= SEQUENCE {modulus INTEGER, -- n
publicExponent INTEGER -- e}
RSAPrivateKey ::= SEQUENCE { version Version,
modulus INTEGER, -- n
publicExponent INTEGER, -- e
privateExponent INTEGER, -- d
prime1 INTEGER, -- p
prime2 INTEGER, -- q
exponent1 INTEGER, -- d mod (p-1)
exponent2 INTEGER, -- d mod (q-1)
coefficient INTEGER -- (inverse of q) mod p}
```

Die Versionsnummer ist aufgrund der Kompatibilität zu zukünftigen Überarbeitungen des Standards gedacht. Zurzeit ist die Versionsnummer gleich Null. Die Werte *p*, *q*, *d mod (p-1)*, *d mod (q-1)* und *(inverse of q) mod p* sind aus Effizienzgründen mit im privaten Schlüssel enthalten. Die Kodierung der Schlüssel erfolgt nach den Regeln der Basic Encoding Rules (BER), X.209. Die allgemeinen Regel der BER schreiben vor, dass die Kodierung vier Komponenten in der folgenden Reihenfolge umfassen soll: [X.209(88)]

1. Identifier Octets
2. Length Octets
3. Contents Octets
4. End-of-Contents Octets

Der Aufbau wird in Abb. 2.21 gezeigt. Jedoch sind die End-of-Content Octets nur erforderlich, wenn diese explizit in den Length Octets gefordert werden. Ansonsten ist auch ein Aufbau ohne dieses Feld möglich, was an dieser Stelle auch nur betrachtet werden soll.

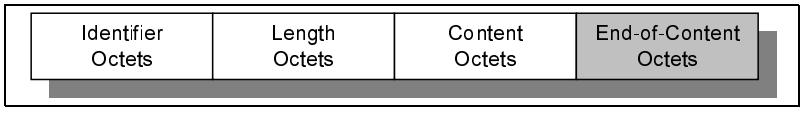


Abb. 2.21 Kodierungsaufbau

Die Identifier Octets kodieren das ASN.1-Tag der Daten. Die Length Octets sind in zwei Formen spezifiziert, der definierten und der undefinierten Form. Die undefinierte Form arbeitet mit den End-of-Content Octets und besteht nur aus einem Byte. Die hier betrachtete definierte Form gibt es in einer kurzen und einer langen Ausführung. Im Allgemeinen umfassen die Length Octets in der definierten Form ein oder mehrere Bytes. In der kurzen Ausführung besteht es genau aus einem Byte, bei dem das achte Bit Null ist und die Bits 1 bis 7 die Anzahl der Content Octets als ein Unsigned Binary Integer kodieren. Die

Anzahl der Bytes ist dabei auf 127 beschränkt. Umfassen die Content Octets beispielsweise 42 Byte, so gilt dann für das Length Octet:

L = 001010102 = 2A16 = 4210

Die lange Ausführung kann eine Anzahl von mehr als 127 Byte codieren. Dabei bestehen die Length Octets aus einem Einführungsbyte und einem oder mehreren folgenden Bytes. Das Einführungsbyte setzt sich folgendermaßen zusammen.

1. Das achte Bit ist 1.
2. Die Bits 7 bis 1 kodieren die Anzahl der folgenden, zu den Length Octets gehörenden Bytes.
3. Das Einführungsbyte soll nicht den Wert FF haben.

Die Bits der folgenden Bytes sollen ebenfalls als Unsigned Binary Integer die Anzahl der Content Octets codieren. Umfassen dann diese beispielsweise 222 Byte, so wird dies folgendermaßen kodiert:

L = 100000012 110111102 = 8116 DE16

Die Content Octets bestehen aus keinem, einem oder mehreren Bytes. Die Daten werden je nach Typ kodiert.

Die Verschlüsselung erfolgt in vier Schritten:

1. Erstellen eines Blockformats für die Verschlüsselung, Verschlüsselungsblock
2. Konvertierung des Oktet-Strings in eine ganze positive Zahl
3. RSA-Verschlüsselung
4. Rückkonvertierung der ganzen positiven Zahl in einen Oktet-String

Für den Verschlüsselungsprozess werden ein Oktet-String, der die zu verschlüsselnden Daten umfasst, der Modulus und ein Schlüssel benötigt. [JACH97]

2.2.4 Certification Authority (CA)

Instanzen, welche Zertifikate ausstellen, spielen eine entscheidende Rolle in Kryptosystemen, deren Key Management auf Zertifikaten basiert. Denn sie überprüfen die Identität des Schlüsselbesitzers und bürgen dann mit einem von ihnen signierten Zertifikat für die Echtheit des jeweiligen öffentlichen Schlüssels. Damit wird dem Benutzer die Überprüfung seines Kommunikationspartners abgenommen und das Authentifizierungsproblem gelöst, da der Benutzer nun einfach prüfen kann, ob der jeweilige öffentliche Schlüssel wirklich zu seinem Kommunikationspartner gehört.

Die Instanz unterschreibt die Zertifikate mit ihrem privaten Schlüssel, weshalb sie ihren öffentlichen Schlüssel bekannt geben muss, damit die Unterschriften der Zertifikate für gültig erklärt werden können. Dies trägt natürlich das angesprochene Authentifizierungsproblem auf eine neue Stufe. Denn nun

muss ebenfalls gewährleistet werden, dass der öffentliche Schlüssel der zertifizierenden Instanz echt (authentisch) ist. Hinzu kommt, dass die Sicherheit des zertifikatbasierten Key Managements eng mit der Vertrauenswürdigkeit und Ehrlichkeit der CA verknüpft ist. Denn auch wenn das verwendete asymmetrische Kryptoverfahren sicher ist und die Zertifikate fälschungssicher sind, kann das System durch eine unehrliche Instanz korrumpiert werden.

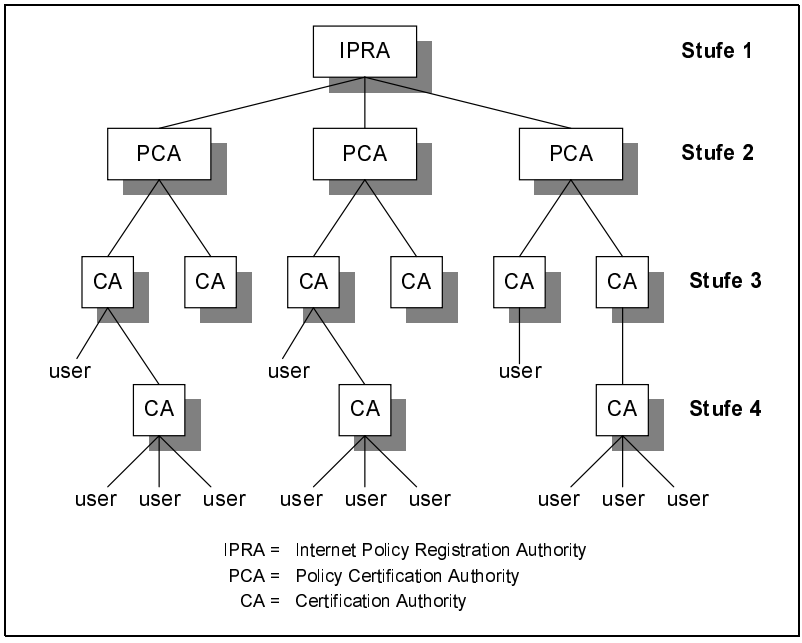


Abb. 2.22 Hierarchische CA-Struktur

Damit ist es von essentieller Bedeutung, dass es sich bei der CA um eine vertrauenswürdige und ehrliche Instanz handelt. Das noch vorhandene Authentisierungsproblem lässt sich dadurch lösen, dass eine weitere Instanz den öffentlichen Schlüssel einer anderen Instanz zertifiziert. Dadurch entsteht eine Kette von Zertifikaten, die so lange verfolgt werden muss, bis man auf eine Zertifizierungsinstanz trifft, der man vertraut. Diese Kette wird auch Certification Path (CP) genannt. Es gibt verschiedene Möglichkeiten, die CAs anzuordnen, damit ein Anwender einen CP durchlaufen kann. So bietet der RFC-1422 eine strikt hierarchische Struktur von Zertifizierungsinstanzen als Lösung an. Diese Hierarchie ist unabhängig vom jeweils verwendeten asymmetrischen Signaturalgorithmus.

Der RFC-1422 schreibt drei verschiedene CAs vor. An der Spitze dieser Hierarchie auf der Stufe 1 steht die Internet Policy Registration Authority (IPRA), die unter der Schirmherrschaft der Internet Society arbeitet. Die IPRA zertifiziert nur die auf der Stufe 2 stehenden, so genannten Policy Certification Authorities (PCA) und legt innerhalb des Internets die notwendigen Richtli-

nien fest, an die die PCAs gebunden sind. Jede PCA muss vor der Registrierung ihre eigenen Richtlinien für die Zertifizierung bei der IPRA festlegen. Diese werden dann als RFC-Dokumente in der Kategorie „Informational" veröffentlicht. Eine Kopie davon wird von der IPRA signiert und ist frei erhältlich. Damit ist gewährleistet, dass die Richtlinien nicht nachträglich geändert werden können. Die Autorisierung der PCAs in der Hierarchie erfolgt zum einen durch die Veröffentlichung des Richtliniendokuments und zum anderen durch ein von der IPRA signiertes Zertifikat. Die Richtlinien der einzelnen PCAs für die Zertifizierung können auf die unterschiedlichen Anforderungen für gesicherte Kommunikation ausgerichtet sein. So gibt es PCAs mit unterschiedlich Sicherheitsstandards. Auf der Stufe 3 und darunter befinden sich die CAs, die wiederum von den PCAs zertifiziert werden. Die CAs können beispielsweise zu bestimmten Organisationen, Firmen, Behörden oder Einrichtungen wie Universitäten gehören. Sie können auch auf bestimmte lokale Gebiete beschränkt sein. Der einzelne Anwender befindet sich in dieser Hierarchie erst auf der Stufe 4 oder darunter. [KENT93]

Dieses statische Modell beinhaltet einige Nachteile. So führt zu jedem Anwender nur ein einziger CP, wodurch beispielsweise CAs nicht von mehreren PCAs zertifiziert werden können. Damit sind Cross Certificates (CC), bei denen sich zwei CAs gegenseitig zertifizieren, nicht möglich. CCs haben aber den Vorteil, dass sich der CP verkürzt. Weiterhin wird erwartet, dass die Anwender mehr Vertrauen in die CAs in ihrer eigenen Domain als in den gesamten CP haben. Die Möglichkeit der unterschiedlichen Richtlinien stellt ein Hindernis für eine weitgehende automatisierte Prüfung der Zertifikate dar.

Aufgrund dieser Nachteile gibt es inzwischen Spezifikationen für eine flexiblere Struktur der CAs. Die PKIX Working Group arbeitet an diesen Spezifikationen für eine Public Key Infrastructure (PKI) im Internet und hat bereits einige Standards definiert. Durch die neue Version der Zertifikate X.509v3 unterliegt die Struktur der CA nicht mehr den strikten Einschränkungen, wie dies bei X.509v1 der Fall war. Diese Überlegungen über eine flexiblere Struktur sind Bestandteil der von der PKIX Working Group erarbeiteten Spezifikationen. [HFPS99] [ADFA99]

2.2.5 Public Key Infrastructure (PKI)

Sichere Kommunikation und automatisierter Schlüsselaustausch mit globaler Reichweite bedingt die Errichtung und Nutzung von öffentlichen Infrastrukturen, so genannte Public Key Infrastructures (PKIs), die auf dem Public-Key-Verfahren basiert. Dies ist leicht verständlich, da Private-Key-Verfahren aufgrund der hohen Zahl für die Kommunikation notwendigen Schlüssel [$nx(n-1)$] bei n Kommunikationsteilnehmern ungeeignet sind. Die PKI besteht gemäß dem Signaturgesetz aus den folgenden fünf Instanzen:

- Regulierungsbehörde
- Zertifizierungsstellen
- Prüfstellen für Hard- und Software
- Anbieter spezieller Soft- und Hardware
- Anwender

Das Zusammenwirken von Certification Authority (CA) und Root Authority sowie gegebenenfalls Prüfstellen und Benutzern bilden zusammen die PKI. Kernaufgaben des Trust Centers bzw. der Certification Authority sind Schlüssel- und Zertifikatsmanagement, Revokation des öffentlichen Schlüssels, Schlüssel-Backup und Wiederherstellung, Identitätsüberprüfung, Schlüsselsperrung und Kreuzzertifizierung (mit anderen CAs). Die Qualität und Vertrauenswürdigkeit der Zuordnung zwischen Zertifikat und Inhaber ist entscheidend für die Verwendbarkeit der digitalen Zertifikate.

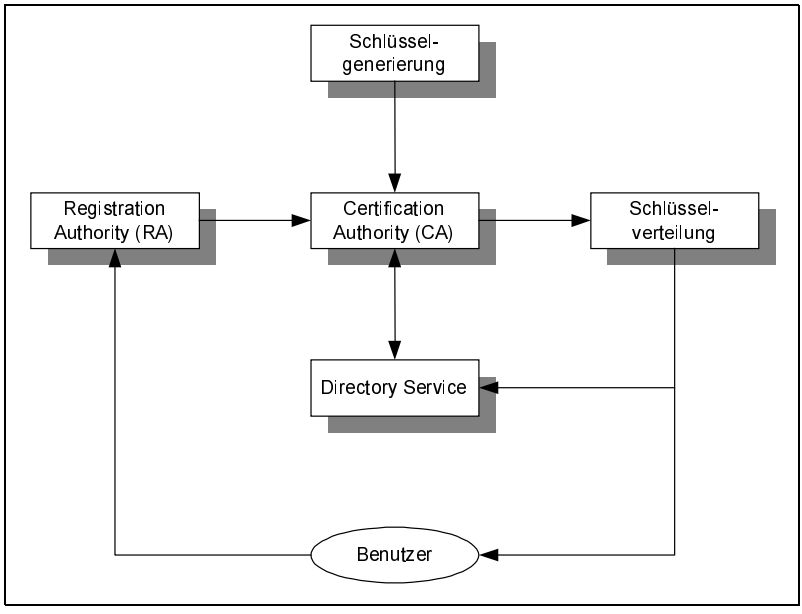


Abb. 2.23 Aufgaben einer Certification Authority (CA)

Trust Center sind die vertrauenswürdigen Instanzen, die über digitale Zertifikate die Übereinstimmung der Identität eines Benutzers mit seinem öffentlichen Schlüssel bestätigen und für diese bürgen. Die Regulierungsbehörde für Telekommunikation und Post (RegTP) in Mainz ist seit dem 25. Januar 1999 die oberste Betriebs-, Genehmigungs- und Kontrollinstanz. Sie fungiert als Wurzelinstanz (Root Authority) bei den Signaturschlüsselzertifikaten und genehmigt Trust Center mit der Bestätigung ihrer öffentlichen TC-Schlüssel. TeleSec (TC der DTAG) hat im Januar 1999 als erstes signaturgesetzkonformes Trust Center in Deutschland den Betrieb aufgenommen. PKIs sind damit die

Basis für jegliche Art von Authentifizierungs-, Verschlüsselungs- und Signierdiensten und somit der Ausgangspunkt für sämtliche Maßnahmen zur Absicherung von elektronischer Information und Kommunikation.

Die Aufgaben einer Certification Authority beschränken sich nicht nur auf das Ausstellen der Zertifikate, sondern beinhaltet beispielsweise auch einen Directory Service für die Veröffentlichung der zertifizierten Schlüssel zuzüglich Zertifikat, wie Abb. 2.23 zu verdeutlichen versucht. Der Begriff CA wird aber nicht immer einheitlich benutzt. Für einige ist mit einer CA nur die zertifizierende Instanz gemeint, andere fassen den Begriff weiter und schließen in diesem Begriff alle für das Key Management wichtigen Aufgaben ein. Im deutschsprachigen Raum gibt es auch den Begriff des Trust Centers. Dieses Trust Center schließt alle für das Key Management wichtigen Aufgaben ein. Dabei werden jedoch nur die Aufgaben betrachtet, die die asymmetrischen Kryptoverfahren betreffen. So ist eine Hauptaufgabe des Trust Centers die Zertifizierung eines öffentlichen Schlüssels inklusive der Generierung des Schlüsselpaars. Nach der Identitätsprüfung wird dem Anwender ein eindeutiger Name zugewiesen. Diese Registrierung wird von der so genannten Registration Authority (RA) übernommen. Im Trust Center fällt die Zertifizierung des Schlüssels in den Aufgabenbereich der Certification Authority. Weiterhin erfolgt von der CA ein Eintrag des Zertifikats in ein öffentliches Schlüsselverzeichnis[19]. Alle gültigen Zertifikate können aus diesem Verzeichnis abgefragt werden. Der Unterhalt dieses Schlüsselverzeichnisses und der Directory Service ist Aufgabe des Directory Managements. Für die Schlüsselverteilung werden vom Produktzentrum TeleSec der Deutschen Telekom AG (DTAG), die solche Trust Center unterhält, Chipkarten verwendet. Es existiert ebenfalls das Projekt Policy Certification Authority (PCA) im DFN, welches den Aufbau einer PKI für das Deutsche Forschungsnetz vorgenommen hat. Im Rahmen dieses Projekts ist eine Public-Key-Infrastruktur für DFN-Mitglieder und weitere Einrichtungen aus Wissenschaft und Forschung aufgebaut worden. Es soll eine internationale Einbindung auf Ebene anderer PKIs erfolgen. [DEER01]

2.3 Sicherer Übertragungskanal

Nachdem die Grundlagen der Kryptographie und das Key Management erläutert wurden, werden nun die möglichen Sicherheitsmechanismen für den Übertragungskanal behandelt. Diese beinhalten Layer-2- und Layer-3-Tunneling-Verfahren, die unterschiedliche Möglichkeiten und Verschlüsselungen bieten, um zwei Kommunikationspartner gegenseitig abzusichern. Abschließend findet eine Evaluierung dieser Verfahren statt.

19 Public Key Directory

Beim Tunneln sind drei Protokolltypen involviert:

1. Das **Passagierprotokoll** (eingekapselte bzw. getunnelte Protokoll):
   - Beim Layer-2-Tunneling (z.B. L2TP) ist es PPP.
   - Beim Layer-3-Tunneling (z.B. IP, IPX oder AppleTalk) wird direkt übertragen.
2. Das **Kapsel- bzw. Tunnelprotokoll**, welches für den Auf- und Abbau des Tunnels verantwortlich ist:
   - VTP (3Com)
   - L2F (Cisco)
   - PPTP (Microsoft)
3. Das **Transport- oder Carrier-Protokoll**, das für den Transport des Kapselprotokolls verantwortlich ist:
   - IP, da es robuste Routing-Fähigkeiten besitzt und als De-facto-Internet-Standard auf heterogenen Systemen läuft.
   - ATM, da es Routing-Fähigkeiten besitzt und unabhängig vom Layer-3-Protokoll arbeitet.

2.3.1 Layer-2-Tunneling

Um die Kommunikation nach außen abzusichern, beispielweise über ein Virtual Private Network (VPN), sind verschiedene Verschlüsselungen bereits in den Routern vorhanden, die für das Tunnelprotokoll eingesetzt werden können. Dabei kann man als Hauptmechanismen nennen:

- Point-to-Point Tunneling Protocol (PPTP)
- Layer-2-Forwarding (L2F)
- Layer-2-Tunneling Protocol (L2TP)
- Microsoft Point-to-Point Encryption (MPPE)

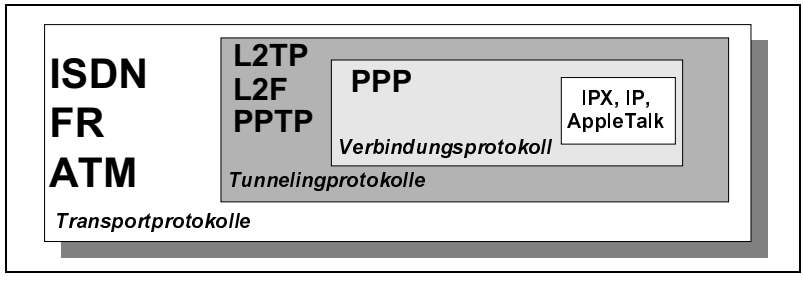
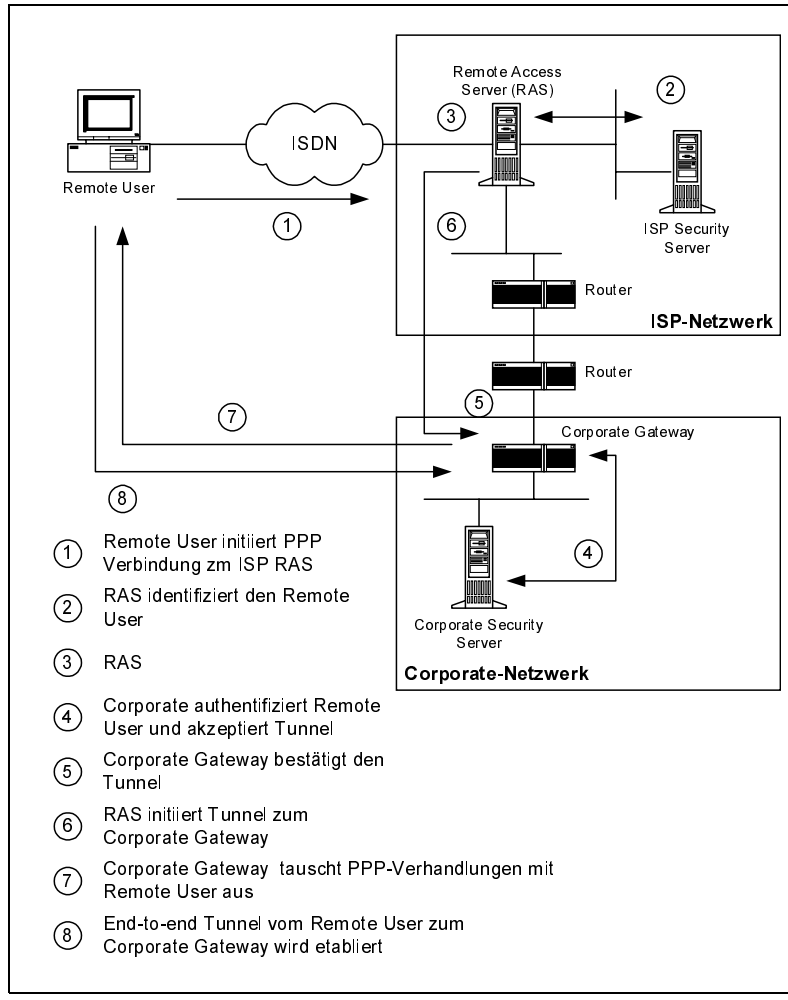


Abb. 2.24 Transport-, Tunneling- und Verbindungsprotokolle

Das Point-to-Point Protocol (PPP), auf dem diese Protokolle basieren, unterstützt bereits in der Sicherungsschicht eine Fehlererkennung, mehrere Protokolle, die Vergabe von IP-Adressen zum Zeitpunkt des Verbindungsaufbaus und Authentifizierung. Es handelt sich hierbei um ein verbindungsorientiertes Netzwerk zwischen zwei Rechnern.

Ein PPP-Datenrahmen besteht aus einem Header (Protokollkopf) und der Payload (Nutzdaten). Die Gesamtlänge eines Rahmens ist im Allgemeinen auf 1500 Byte festgelegt. In den ersten 32 Bit des Rahmens ist der Header enthalten. Die restlichen Bytes enthalten die Nutzdaten und eine Prüfsumme zum Abgleich von Übertragungsfehlern. Der Header enthält Informationen über das transportierte Protokoll selber (z.B. IP, IPX oder LCP) und acht Flags zur Steuerung. In den Nutzdaten ist dann das nächste Paket enthalten. Es könnte sich zum Beispiel um ein TCP/IP-Paket für eine Internet-Verbindung handeln.

Abb. 2.25 Virtueller Einwählvorgang



Generelle Nachteile der Layer-2-Tunneling-Verfahren sind:

- Der Schicht-2-Rahmen braucht einen zusätzlichen Header, der die benötigten Ressourcen erhöht und zusätzliche Übertragungsgebühren verursacht.
- Die Skalierung ist eingeschränkt, da viele parallele PPP-Verbindungen die Ressourcen angreifen. Dadurch ist die Lösung abhängig von der Client-Anzahl.
- Der Rechenaufwand an den Tunnelenden ist enorm hoch, wodurch Verbindungsabbrüche geschehen können. Spezielle Anwendungen (z.B. Fax) reagieren sehr empfindlich auf kleine Time-outs.

Abb. 2.25 zeigt einen virtuellen Einwählvorgang, der über einen RAS-Server auf einen ISP erfolgt und anschließend auf das sichere Corporate Network (CN) weitergeleitet wird. Dadurch wird erst im achten Schritt eine direkte Verbindung zum CN aufgebaut und ein End-to-end-Tunnel realisiert.

Point-to-Point Tunneling Protocol (PPTP)

Das Point-to-Point Tunneling Protocol (PPTP) nach RFC-2637 von Microsoft, Ascend, 3Com und U.S. Robotics ist ein Layer-2-Tunneling-Protokoll und setzt einen Client voraus, der PPP-Pakete beherrscht. Diese Pakete werden in eine modifizierter Form des Generic Routing Encapsulation Protocols der Version 2 (GRE V2) eingepackt und über das Netz zum Remote Access Server (RAS) des ISP transportiert. GRE ist ein fluss- und sättigungsgesteuerter gekapselter Datagrammdienst für den Transport von PPP-Paketen. Bei RAS findet keine Authentifizierung statt, wodurch eine statische Zuordnung erfolgen muss. Der Teilnehmer ist dadurch nicht in der Lage, den Endpunkt der Tunnel zu beeinflussen. Zusätzlich lassen sich für den Betreiber des Zugangssystems keine Accounting-Daten erfassen. Auch auf die übertragene Menge von Datenpaketen sowie auf die Verbindungsdauer muss verzichtet werden.

PPTP definiert eine Client-/Server-Architektur, um Funktionen zu entkoppeln, die in heutigen RAS existieren und Virtual Private Networks (VPNs) unterstützen. Der PPTP Network Server (PNS) soll laut Spezifikation auf einem beliebigen Betriebssystem laufen, während der Client als PPTP Access Concentrator (PAC) auf einer Einwählplattform arbeitet. PPTP spezifiziert ebenfalls ein Anrufssteuerungs- und Managementprotokoll, welches dem Server die Zugangskontrolle für eingewählte Anrufe aus einem Telefonnetz sowie die Initiierung von ausgehenden Verbindungen ermöglicht. PPTP verlässt sich hinsichtlich der Sicherheit auf IPsec. [KAEO98]

Über die Steuerverbindung zwischen PAC und PNS regeln die beiden Seiten Aufbau, Verwaltung und Beendigung einer Tunnelverbindung. Dabei wird die Beziehung zwischen PAC und PNS und nicht die Kommunikation der im Tunnel vorhandenen Einzelverbindungen betrachtet. Eine bestehende Steuerverbindung bedeutet demnach nicht, dass ebenfalls ein Tunnel hergestellt wird. Die Steuerverbindung kann dabei von beiden Seiten aus etabliert werden. Ein

Echo-Dialog dient zur Überwachung der Verbindung. Ein so genannter Echo-Request wird von beiden Seiten gesendet, um die Verfügbarkeit der Gegenstelle über die bestehende Verbindung zu erfragen, wenn innerhalb von 60 Sekunden keine Daten empfangen wurden. Neben dem Controll Connection Management spezifiziert PPTP auch ein Call Management, Error Reporting und PPP-Session Control.

Das Call Management steuert den Aufbau der Tunnelverbindung. Über den Tunnel können anschließend Nutzdaten in modifizierten GRE-Rahmen transportiert werden. Um den Transfer eines Clients vom PAC zum PNS zu starten, fordert der PAC mit einem Incoming Call Request beim PNS eine Verbindung an. Dieser fordert den PAC mit einem Outgoing Call Request auf, eine externe Verbindung für die Gegenrichtung herzustellen, damit Daten zum PAC übertragen werden können. In der Meldung Incoming Call Connected werden bestimmte Parameter wie Übertragungsrate oder Fenstergröße an den PNS übergeben. Innerhalb der Steuerverbindung werden auch Informationen zur Rufnummer übermittelt. Die detaillierte Beschreibung aller Abläufe ist in der IETF-Spezifikation RFC-2637 definiert.

Wenn ein Verbindungswunsch zum Remote Network über den PNS vorgenommen werden soll, sendet der PAC die Nachricht Incoming Call Request, die aus einem Standard-TCP-Header und den folgenden Feldern der Abb. 2.26 bestehen:

- **Length:** Die Länge definiert die Anzahl der Bytes der gesamten PPTP-Nachricht.
- **Message Type:** gibt den Nachrichtentyp an und beinhaltet bei einer Control Message den Wert 1.
- **Magic-Cookie:** wird immer auf den angegebenen Wert gesetzt, um die Synchronisation gewährleisten zu können. Wird auf der Empfangsseite ein anderer Wert erkannt, kommt es zum Verlust der Synchronisation und die TCP-Verbindung wird abgebrochen.
- **Control Message Type:** beschreibt den Zweck der Steuernachricht. Bei einem Incoming Call Request wird der Wert 9 verwendet. Die Auswertung des Feldes ist sehr wichtig, da die Struktur der Steuernachricht nur in den ersten Bytes mit denen eines modifizierten TCP-Headers identisch ist. Anschließend wird für jede Nachricht eine spezielle Struktur definiert.
- **Call ID:** kennzeichnet den Tunnel. Alle gemultiplexten Verbindungen, die durch diesen Tunnel geführt werden, besitzen dabei die gleiche Call ID.
- **Call Serial Number:** Die Unterscheidung der Verbindungen erfolgt über Seriennummern.

Die anschließenden Parameter beschreiben das Einwahlmedium und die Rufnummerndaten. Der GRE-Header wird bei PPTP abgeändert, sodass Prüfsumme und ein optionales Offset-Feld entfallen. Zusätzlich wird das Element

Key in zwei Bereiche gegliedert, die Payload-Länge und Call ID für die Assoziation zur jeweiligen Steuerverbindung enthalten.

Ein Nachteil dieses Verfahrens ist der notwendige Einsatz von Windows NT/2000 als Home Gateway. Zusätzlich muss eine feste IP-Adresse vergeben werden, da diese für eine statische Vergabe der Route ausschlaggebend ist. Viele Unternehmen besitzen aber private IP-Adressen nach der Spezifikation RFC-1918, wodurch Probleme bei der Zuordnung entstehen. Hinzu kommt, dass keine Authentifizierung der Tunnelenden vorgesehen ist. Sie erfolgt über den durch den Tunnel geführten PPP-Dialog. Die Tunnelverbindung muss demnach vor der Authentifizierung des Clients beim Server mit PAP oder CHAP etabliert werden. Dadurch ist das Angriffspotenzial innerhalb des Übertragungspfades relativ hoch. Vorteile von PPTP sind die Möglichkeit der Übertragung von IP- und IPX/SPX-Paketen (Multiprotocol Tunneling) sowie der Dial-out. Dial-out ermöglicht die Anwahl einer Rufnummer von der Zentrale aus, um eine direkte Verbindung mit dem einzubindenden Arbeitsplatz herzustellen. [HPVT+99]

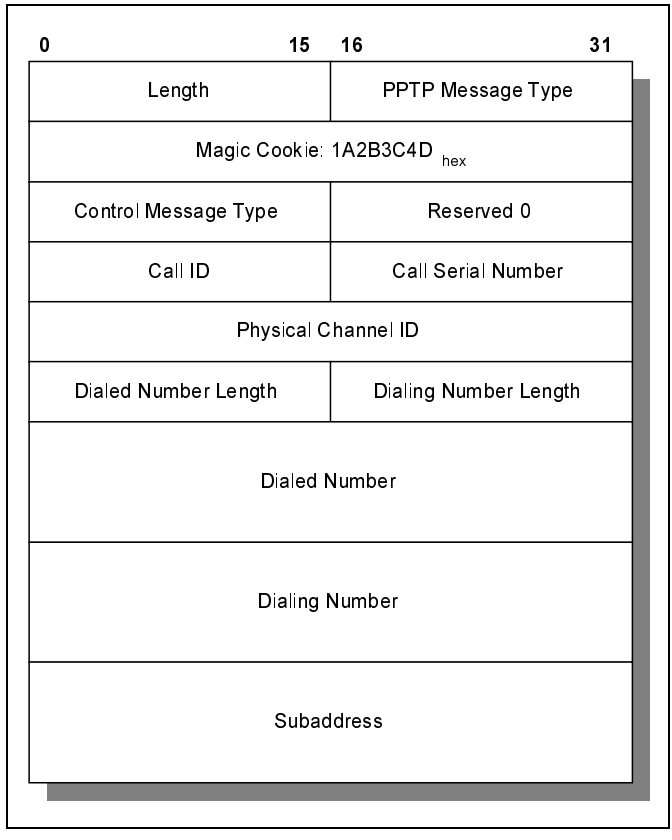
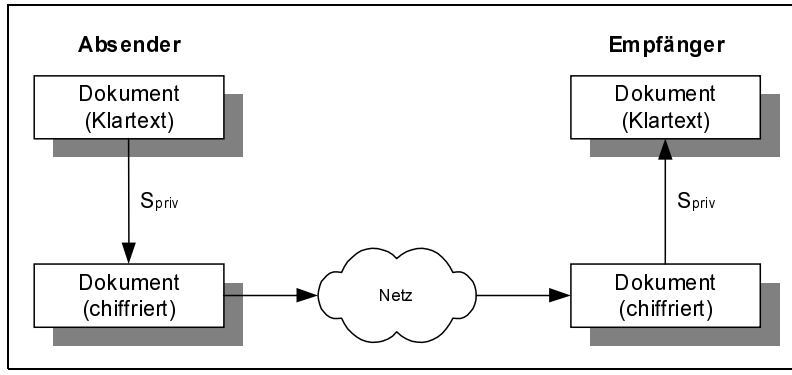


Abb. 2.26
PPTP-Nachricht
Incoming Call Request

Microsoft Point-to-Point Encryption (MPPE)

MPPE erlaubt eine Ende-zu-Ende-Verschlüsselung für Standard-Microsoft-Wählverbindungen über PPTP. MPPE verschlüsselt auf der Client-Seite PPP-Pakete vor dem PPTP-Tunneling. Nach der Verhandlung des PPP zwischen Client und Tunnel-Server wird die Verschlüsselungs-Session initiiert. Dabei wird MS-CHAP (Microsoft Challenge Handshake Protocol) für die Authentifizierung eingesetzt. Die zweite Schicht des OSI-Referenzmodells, die Sicherungsschicht, wird dafür verwendet, dass die zu verschickenden Pakete alle wichtigen Informationen enthalten und auch in der richtigen Reihenfolge versendet werden. Hier findet die Verschlüsselung über MPPE mit dem RSA-RC4-Algorithmus statt.

Abb. 2.27 MPPE-Verschlüsselung



Hierbei verfasst der Absender ein Dokument. Dieses Dokument wird dann, bevor es als ein Paket die Schicht 2 erreicht, chiffriert und erst anschließend als ein kodiertes Paket an die zweite Schicht weitergegeben. Danach wird das Paket über die erste Ebene (Hardware) des Schichtmodells zum Empfänger geschickt. Wenn das Paket beim Empfänger angekommen ist, wird dort das chiffrierte Dokument dechiffriert und kann zum Schluss vom Empfänger geöffnet werden.

Bei der Verschlüsselung unterscheidet man unterschiedliche Schlüssellängen, die sich u.a. durch die Exportbeschränkung der USA ergeben haben. Während man in der Vergangenheit 40- oder 56-Bit-Verschlüsselung im Allgemeinen international eingesetzt hat, wird heute die 128-Bit-Verschlüsselung offiziell verwendet, da die Exportbeschränkungen aufgeweicht wurden.

Die MPPE-Verschlüsselung setzt sich direkt in das Betriebssystem (Netzwerksystem) von Windows, es ist keine Benutzerinteraktionen nötigt. Die Verschlüsselung wird also unabhängig von dem Benutzer und den verwendeten Applikationen durchgeführt. Die verwendeten Schlüssel werden dabei laufend gewechselt, die genaue Zeitdauer kann entsprechend den Sicherheitsbedürfnissen konfiguriert werden. Das heißt im extremsten Fall, dass jedes Paket einen anderen Schlüssel enthält. Die Vielzahl der Schlüssel müssen natürlich auch verwaltet werden, sodass eine höhere Rechenkapazität erforderlich ist.

In jedem Paketkopf gibt es noch zusätzlich zu der Sicherungsschicht einen 12-Bit-Zähler zur Synchronisation der Verschlüsselung (RSA RC4). Dieser Zähler wird lediglich von 0 bis 4095 heraufgezählt und fängt dann wieder bei null an. Falls die Pakete noch zusätzlich komprimiert worden sind, muss der Empfänger zunächst dekomprimieren. Es werden nur Pakete mit der Nummerierung zwischen 0x0021 und 0x00FA für MPPE verwendet.

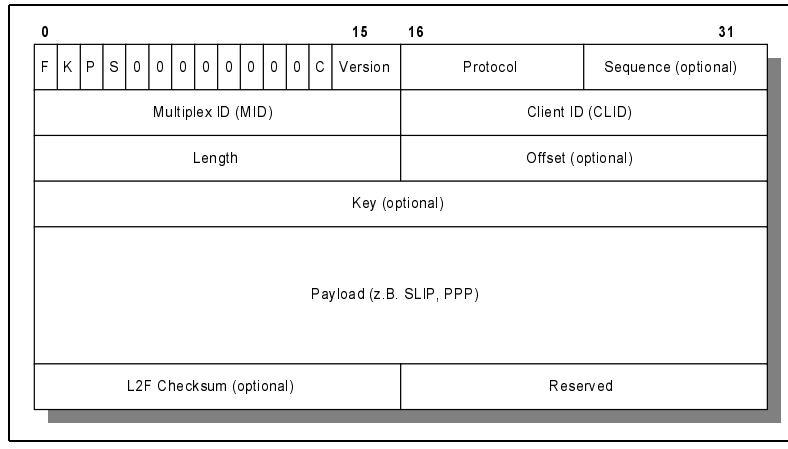
Layer 2 Forwarding (L2F)

Die zweite Möglichkeit ist das Protokoll Layer 2 Forwarding (L2F) von Cisco nach RFC-2341. Als Home Gateway kann ein Router eingesetzt werden, während als RAS Access Server zum Einsatz kommen. Die Client-Software ist beim L2F wie beim PPTP beliebig. Allerdings unterstützt L2F neben PPP auch das Serial Line Protocol (SLIP). Es wird somit das Tunneling der Linkebene von Protokollen höherer Ebene ermöglicht. Auch ist die Zuordnung von RAS und Home Gateway, im Gegensatz zu PPTP, dynamisch, wodurch Multi-Providing ermöglicht wird. Die Kennzeichnung der Datenpakete erfolgt durch eine Multiplex ID (MID) und ermöglicht dadurch den gleichzeitigen Betrieb mehrerer Tunnel. Die Ermittlung des Teilnehmers und des Passworts ist ebenfalls gegeben und wird im Zugangssystem ermittelt. Dadurch kann das Zugangssystem den Zielort des Tunnels durch den Namen oder die Wahlziffer bestimmen. Dieser Tunnel ermöglicht außerdem Anwendungen, die Unterstützung benötigen, für privat adressierte IP-, IPX- und AppleTalk-Einwahl mittels SLIP/PPP über die vorhandene Internet-Infrastruktur. Durch dieses Multi-Providing kann man direkt auf das Internet oder das VPN zugreifen. Das L2F-Paket besteht aus drei Elementen:

- L2F-Header
- Payload Packet
- L2F-Prüfsumme (optional)

Der L2F-Header enthält zwei ID-Parameter: Multiplex ID (MID) und Client ID (CLID). Mit deren Hilfe können gleichzeitig verschiedene Verbindungsziele und damit mehrere Tunnel sowie innerhalb dieser Tunnel jeweils mehrere logische Verbindungen unterstützt werden. L2F definiert für jeden Tunnel eine individuelle Punkt-zu-Punkt-Verbindung, kann allerdings auch Punkt-zu-Mehrpunkt-Verbindungen etablieren. Weiterhin können die drei Bit des Versionsfeldes in kodierter Form das Protokoll der Payload beschreiben. So steht 02_h für eine PPP-Payload und 03_h für eine SLIP-Payload. Weiterhin sind optionale Parameter enthalten, deren Nutzung verschiedene Flags in den ersten beiden Byte anzeigen. Sequence und Checksum leisten einen Beitrag zur fehlerfreien Übertragung im L2F-Tunnel. Die optionale Checksumme ergibt sich aus einem Cyclic Redundancy Check (CRC) und erlaubt es, Übertragungsfehler zu erkennen. Durch die Sequence Number kann der Empfänger hingegen verlorene oder doppelt ankommende Pakete erkennen und entsprechend reagieren.

Abb. 2.28
L2F-Paketstruktur



Der Tunnelaufbau wird durch eine L2F-Request-Configuration-Nachricht gestartet. Dabei wird der Name des RAS und eine Zufallszahl zur Authentifizierung des RAS beim Home Gateway übergeben. Im zweiten Schritt antwortet das Home Gateway dem rufenden RAS und enthält den Namen des Home Gateway, eigene Vorgaben für die Client-ID und eine Zufallszahl zur Authentifizierung. Beide Seiten identifizieren sich über ein Challenge-Handshake-Verfahren. Anhand der Authentifizierungsdaten öffnet nun jede Seite, beginnend mit dem RAS, den Tunnel mit einer Accept-Configuration-Nachricht. Die Sequence Number wird jeweils um den Wert Eins erhöht. Als CLID werden die von der Gegenseite vorgegebenen Werte verwendet. Die Antwort besteht aus einem Hash-Wert aus der zuvor empfangenen Zufallszahl und dem intern gespeicherten Zugangskennwort. Der Hash-Wert wird anhand des MD5-Algorithmus gebildet und im Key-Feld sowie als Response-Parameter übertragen. Nach erfolgreicher Authentifizierung besteht eine Tunnelverbindung zwischen dem RAS und dem Home Gateway. Abschließend können sich Clients über diese Tunnelverbindung beim Home Gateway anmelden.

Der Datenaustausch zwischen dem Client und dem Zielnetzwerk erfolgt durch Encapsulation der Payload, inklusive des verwendeten Transportprotokolls. Sowohl der Client als auch der Tunnel selbst sind durch die Daten des Header (MID[20], CLID[21] und Key) eindeutig identifizierbar. Auch das Transportprotokoll der Payload wird im Header angezeigt. Die Verfügbarkeit kann mit einer Echoabfrage (ähnlich einem Ping-Befehl) getestet werden.Bei L2F_ECHO wird eine Nachricht mit einer Payload-Länge von maximal 64 Byte und der MID=0 verschickt. Besteht der Tunnel, antwortet die Gegenseite mit L2F_ECHO_RESP.

20 Multiplex ID
21 Client ID

Weiterhin kann L2F über eine Vielzahl von paketorientierten Netzen (u.a. ATM und Frame-Relay) transportiert werden, wodurch dieses Verfahren unabhängig von der verwendeten Technologie ist. Im Gegensatz zur Übertragung von IP-Paketen, die verbindungslos arbeiten, werden L2F-Tunnel bei Bedarf auf- und wieder abgebaut. Da als Home Gateway meistens ein Router eingesetzt wird, kann auch eine Adressumsetzung über NAT erfolgen. Ein Dial-out, das heißt die Übertragung von Wählverbindungen (Rufnummern), wird somit von L2F nicht unterstützt. Die Unterstützung virtueller Multiprotokoll-Einwählanwendungen ist für den Endbenutzer sowie den ISP ein Vorteil, da die gemeinsame Nutzung die Investitionen im Zugangs- und Kerninfrastrukturbereich relativ gering hält und für den Kunden Ortsgespräche ermöglicht werden. Zusätzlich werden auch Anwendungen über das Internet unterstützt, die nicht IP sprechen. Das Tunnelverfahren setzt allerdings auch keine Verschlüsselung automatisch ein, was durch andere Verfahren wie beispielsweise IPsec sichergestellt werden muss.[VLK98]

Layer 2 Tunneling Protocol (L2TP)

Wegen der Vor- und Nachteile beider Verfahren ist ein drittes Protokoll spezifiziert worden. Cisco und Microsoft haben sich dabei auf die Zusammenarbeit an der Entwicklung eines einzigen Standards innerhalb der IETF nach RFC-2661 geeinigt, welches als Layer 2 Tunneling Protocol (L2TP) bezeichnet wird. L2TP ist ein hybrides Verfahren aus den beiden zuerst genannten. Es ist jedoch stärker mit dem L2F-Protokoll verwandt und erweitert dieses um den fehlenden Dial-out. Das heißt, wenn keine physikalische Verbindung vorhanden ist, wird ein ausgehender Call vom Zugangssystem zur Gegenstelle aufgebaut. Die Abwärtskompatibilität zu L2F ist ebenfalls gesichert. Weitere Vorteile sind die Unterstützung mehrerer Tunnelverbindungen und die Authentifizierung der Gegenstellen während des Tunnelaufbaus. Allerdings verzichtet L2TP auf SLIP als Transportprotokoll.

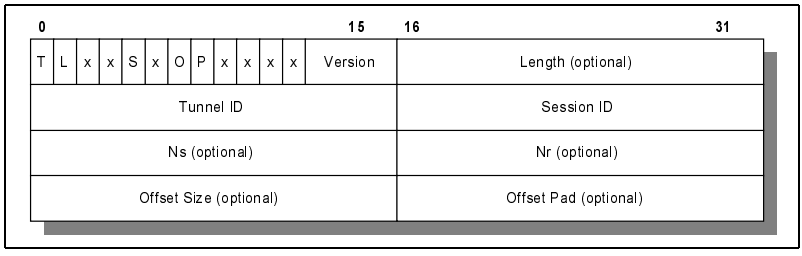


Abb. 2.29 L2TP-Paketstruktur

L2TP definiert zwei Nachrichtentypen, die L2TP Data Message und die L2TP Control Message. Die Steuerverbindung sorgt für die Etablierung, Verwaltung und Beendigung des Tunnels. Für den eigentlichen Datenaustausch werden spezielle Datenformate verwendet. Beide Nachrichtentypen besitzen eine

einheitliche Header-Struktur, wobei die Längenangabe und die Folgenummern für Steuernachrichten zwingend sind. Bei Datenpaketen sind sie hingegen optional.

Der L2TP-Header beginnt mit einer Reihe von Flags, die Informationen über die Verwendung optionaler Elemente beinhalten. Das erste Bit unterscheidet zwischen Daten- (T=0) und Steuerrahmen (T=1). Das L-Bit zeigt hingegen, ob das Längenfeld vorhanden ist (L=1), das die Länge des gesamten Pakets in Byte anzeigt. Das S-Bit steht wieder für Sequence. Enthält der Header ein Offset-Padding-Feld, dessen Länge das Feld Offset Size beschreibt, zeigt das O-Bit das Vorhandensein dieser Felder an. Die Tunnel ID und die Session ID haben ähnliche Funktionen wie MID und CLID bei L2F. Hinzu kommt allerdings, dass sich damit auch die Referenz eines Datenpakets zu einer Steuerverbindung, die den Tunnel und dessen Status beschreibt, herstellen lässt. Dadurch wird die Bildung von vielen Tunnel ermöglicht. Dies geschieht, indem die empfangene Sendefolgenummer um jeweils eins erhöht und als Empfangsfolgenummer mit der nächsten Nachricht an den Absender zurück übertragen wird.

Die Steuernachrichten enthalten nach dem Header einen oder mehrere Parameter, die man als Attribute Value Pairs (AVP) bezeichnet. Dies ist eine weitere Rahmenstruktur, die neben dem Attributtyp und einem Parameter mit variabler Länge auch die Gesamtlängenangabe des AVP, eine Herstellerkennung nach RFC-1700 und zwei Flags umfasst. Eine Steuernachricht beginnt dabei immer mit einem Message Type AVP. Dieser beschreibt die Nachricht und dessen Bedeutung. Anschließend folgen Parameter in jeweils eigenen AVPs. Diese Parameter können u.a. Bitrate und Rufnummerninformationen enthalten. Die Steuernachrichten gliedern sich in Nachrichten zur Verwaltung einer Tunnelbeziehung zwischen dem L2TP Access Controller (LAC) und dem L2TP Network Server (LNS) bzw. einer einzelnen Session innerhalb eines Tunnels sowie in Nachrichten zur Meldung von Fehlern und der Koordination der PPP-Session.

Nimmt der LAC einen Anruf über das Wählnetz entgegen, der über eine Tunnelverbindung weitergeleitet werden muss, spricht man von einem Incoming Call. Dieser Anruf löst entweder den Aufbau des Tunnels durch den LAC aus oder führt, wenn der Tunnel bereits besteht, zum direkten Aufbau eines Session innerhalb dieses Tunnels. Dazu sendet der LAC mit einer Incoming Call Request Message die wesentlichen Daten des Anrufers an den LNS. Der LNS antwortet mit einem Incoming Call Reply und die Verbindung wird aufgebaut. Sowohl eine Session als auch eine Tunnelverbindung werden mit einer einfachen Nachricht beendet. Ein Grund für eine Beendigung einer Verbindung kann die Annahme sein, dass die Gegenseite nicht mehr verfügbar ist. Um dies festzustellen, definiert L2TP eine Keep-Alive-Prozedur mit einer Hello-Nach-

richt. Der Empfänger muss diese Nachricht quittieren, da sonst davon ausgegangen wird, dass er nicht mehr erreichbar ist. Für die Übertragung der Daten werden die CRC-Prüfsumme und Zero Bit Stuffing entfernt, um die Performance möglichst hoch zu halten. Im Tunnel werden die PPP-Rahmen mit einem L2TP-Header versehen, über dessen Parameter (Tunnel und Session ID) eine Zuordnung zu den jeweiligen Wegen vorgenommen wird. Für beide Kommunikationspartner wirkt der Übertragungspfad wie eine transparente Strecke, auf der das individuelle LAN-Protokoll gefahren wird.

Ein wesentlicher Vorteil von L2TP gegenüber PPTP ist, dass die Authentifizierung beider Tunnelenden über CHAP vorgenommen wird. Der allgemeine Ablauf der Kommunikation bleibt dabei gleich, jedoch werden in den Nachrichten des Dialogs die entsprechenden AVPs integriert. Ein Vorteil gegenüber L2F ist, dass L2TP beiden Seiten ermöglicht, eine Verbindung anzufordern. Dazu müssen die Steuernachrichten in ihren AVPs alle Informationen zur Übertragungsrate, benötigte Rufnummern etc. enthalten. [TVRP+99]

2.3.2 Layer-3-Tunneling

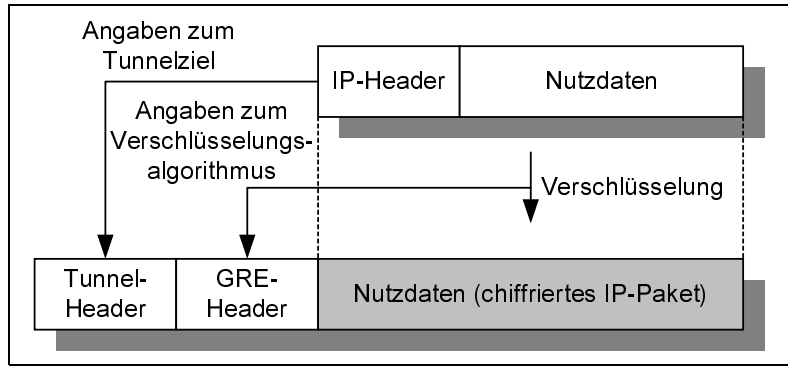
Aufgrund der Nachteile (immerhin sind alle Layer-2-Tunneling-Verfahren keine abgeschlossen Standards, sondern Internet Drafts) und unterschiedlicher Ansätze versucht man mit dem Layer-3-Tunneling Netzwerkprotokolle direkt in das Tunnelprotokoll einzubinden. Die Implementierung auf Schicht 3 besitzt zusätzlich den Vorteil, dass man den Datenverkehr logisch unabhängig übermitteln kann und demnach eine höhere Vertraulichkeit vorhanden ist. Folgende Verfahren lassen sich u.a. als Layer-3-Tunneling beschreiben:

- Generic Routing Encapsulation (GRE)
- Internet Protocol Security (IPsec)

Generic Routing Encapsulation (GRE)

Ein erster Standard für das Tunneling war das Generic Routing Encapsulation (GRE) nach den Spezifikationen RFC-1701 und RFC-1702. Dabei handelt es sich um eine Richtlinie darüber, wie die Tunnelpakete aufgebaut sein sollen. In einem GRE-Paket werden dabei drei Abschnitte unterschieden: ein Tunnelkopf, der das Tunnelziel enthält, ein GRE-Kopf, der Informationen über das verwendete Tunnelprotokoll und die Verschlüsselungsalgorithmen enthält, und die Nutzlast (Payload), also das zu transportierende LAN-Paket. Abb. 2.30 zeigt den Aufbau eines solchen GRE-Pakets. Als Beispiel ist hier die Kapselung eines IP-Pakets gewählt. Mit GRE-Paketen lassen sich aber auch beliebige andere Netzwerkprotokolle tunneln, weshalb es u.a. bei PPTP zum Einsatz kommt. [HLFT94a] [HLFT94b]

Abb. 2.30
GRE-Paket



IP Security (IPsec) Die Arbeitsgruppe IPsec der IETF hat 1998 einen Entwurf für einen Standard vorgelegt, um zukünftig eine sichere IP-Architektur bereitzustellen. Grundlage dieser Spezifikation bildet ein vor ein paar Jahren erarbeiteter Standard (RFC-1825, jetzt RFC-2401). Der Vorschlag legt fest, auf welche Weise Authentifizierung und Verschlüsselung auf der IP-Schicht einzurichten sind. Dabei hat sich die IP Security Protocol Working Group der IETF zur Aufgabe gemacht, Richtlinien für mögliche Implementierungen von IPsec für Hard- und Softwarehersteller zu erstellen und auch zu pflegen. Firewalls, die sich an diese Spezifikation halten, können untereinander chiffrierte Daten austauschen, auch wenn sie von unterschiedlichen Herstellern stammen und verschiedene Verschlüsselungsverfahren verwenden. Dies war bislang nicht möglich, da es sich immer um proprietäre Lösungen handelte.

Mit IPsec ist es möglich, folgende Eigenschaften einer Netzwerkübertragung zu gewährleisten:

- **Integrität:** Die Daten können vor Verfälschung während des Transports geschützt werden, indem eine verschlüsselte Prüfsumme in das Datenpaket eingefügt wird, die der Empfänger wiederum verifizieren kann.
- **Authentizität:** Die Authentizität des Datagramms wird dadurch garantiert, dass die Integritätsbedingung auch auf die Absenderadresse zutrifft.
- **Vertraulichkeit:** Die zu transportierenden Daten können mit Hilfe eines kryptographischen Verfahrens geschützt werden.

Dabei ist IPsec nur in der Lage, reine Datenübermittlung zu schützen, die Daten auf dem Quell- und Zielrechner bleiben unverschlüsselt. Damit arbeitet IPsec unabhängig von der jeweiligen Anwendung und deren Verschlüsselungsmethoden. Für den weiteren Schutz der Daten sind dann wiederum Anwendungsprogramme zuständig.

IPsec ist ein Layer-3-Protokoll, d.h. basierend auf dem OSI-Referenzmodell ist IPsec als Erweiterung von IP in der dritten Schicht, also der Vermittlungsschicht anzusiedeln. Die Abb. 2.31 zeigt dabei die Zuordnungen zwischen OSI-Modell und TCP/IP-Referenzmodell sowie die grobe Einordnung des IPsec Protokolls.

Zu der Spezifikation RFC-2401 kommen noch RFC-2402 und RFC-2406 hinzu, die alle aufeinander aufbauen. Zusätzlich sind eine Menge Spezifikationen definiert worden, die unterschiedliche Themen behandeln. Ebenfalls sind spezielle Spezifikationen für die Nutzung konkreter Verschlüsselungs- und Authentifizierungsalgorithmen entstanden. Key-Management-Protokolle sind ebenfalls angegeben, sodass man zusammenfassend folgende IPsec-Funktionen herausstellen kann:

- Authentifizierung und Integrität durch AH (Authentication Header)
- Kombinierte Authentifizierung, Vertraulichkeit und Integrität durch ESP (Encapsulating Security Payload)
- Schlüsselaustauschmechanismen (Key Management)

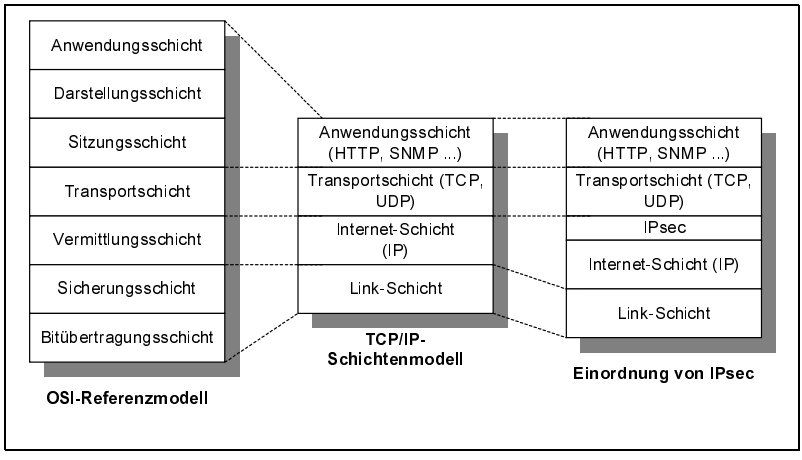


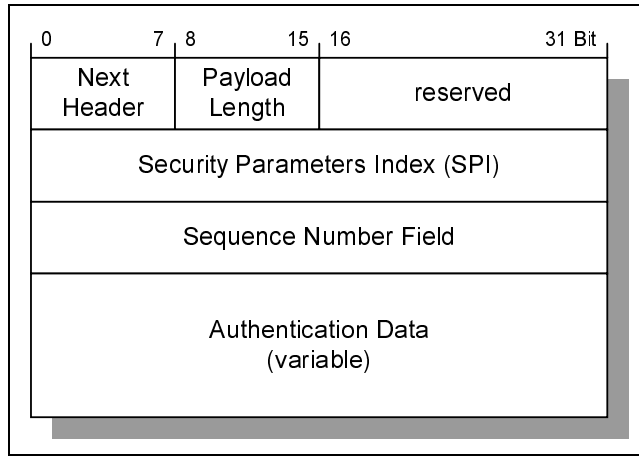
Abb. 2.31
Einordnung von IPsec ins OSI-Modell

Mit Hilfe des IP Authentication Header (AH) wird die Integrität der übertragenen Daten geschützt und die Authentizität gewährleistet. Auch ohne die Möglichkeit, die Daten verschlüsseln zu können (Nutzdaten werden unverschlüsselt übertragen), findet hier eine Sicherung der Datenübertragung statt, indem eine Prüfsumme der Daten erstellt wird (z.B. mit Hilfe von MD5 oder SHA-1) und verschlüsselt in den Authentication Header vor dem TCP-Header eingefügt wird. Zum Schutz gegen das Wiedereinspielen (Anti-Replay) von Paketen können Sequenznummern verwendet werden. Zur Erzeugung des AH verwendet der Sender einen zu Beginn des Kommunikationsflusses vereinbarten gemeinsamen Schlüssel (gemäß dem Key-Management-Protokoll), den der Empfänger zur Überprüfung des Pakets einsetzt. AH kann alleine oder in Kombination mit ESP eingesetzt werden. Viele Kombinationen, die zwischen beiden Verfahren möglich sind, kann man dabei im praktischen Einsatz vernachlässigen. AH sollte in jedem Fall eingesetzt werden, wenn der benutzte ESP-Algorithmus keine Integrität der verschlüsselten Daten garantiert.

Abb. 2.32 zeigt den Aufbau eines Authentication Headers nach RFC-2402, der aus folgenden Teilen besteht: [KEAT98c]

- **Next Header:** identifiziert den Typ des nächsten Nutzdatenpakets nach dem Authentication Header
- **Payload Length:** enthält die Längenangabe der Nutzlast dieses Pakets in 32-Bit-Werten
- **Reserved:** reserviert für zukünftige Nutzung
- **Security Parameters Index (SPI):** ein 32-Bit-Wert, der die zur Zieladresse des Datenpakets gehörige Security Association (SA) identifiziert
- **Sequence Number:** enthält eine Sequenznummer, falls Anti-Replay genutzt werden soll. Dabei ist das Vorhandensein dieser Nummer auf Senderseite erforderlich, auf der Empfängerseite wird erst entschieden, ob die Sequenznummern ausgewertet werden. In diesem Fall darf sich keine Nummer wiederholen, d.h. nach 2^{32} Paketen muss auf beiden Seiten ein Reset durchgeführt werden.
- **Authentication Data:** der Integrity Check Value (ICV) für dieses Paket. Der ICV ist ein in Abhängigkeit des eingesetzten Prüfsummenverfahrens entstandener Wert, der durch die während des Transports nicht veränderbaren Felder des Datagramms bestimmt wird.

Abb. 2.32 Authentification Header



Der IP Encapsulating Security Payload (ESP) Header kapselt die zu schützenden Daten ein und gewährleistet bei Bedarf deren Vertraulichkeit durch Verschlüsselung. Außerdem sind Möglichkeiten zum Schutz der Integrität und zur Authentizität der Datenpakete vorgesehen. Wie beim AH können auch hier Sequenznummern zum Schutz gegen das Wiedereinspielen von Paketen genutzt werden.

Die Vertraulichkeit kann durch ein geeignetes Verschlüsselungsverfahren (z.B. DES[22] oder Triple-DES) gewährleistet werden, der Schutz der Integrität der Daten wird durch eine kryptographische Prüfsumme (z.B. mit Hilfe von MD5[23] oder SHA-1[24]) garantiert. Nicht durch das ESP eingekapselte Felder eines Pakets können nicht geschützt werden.

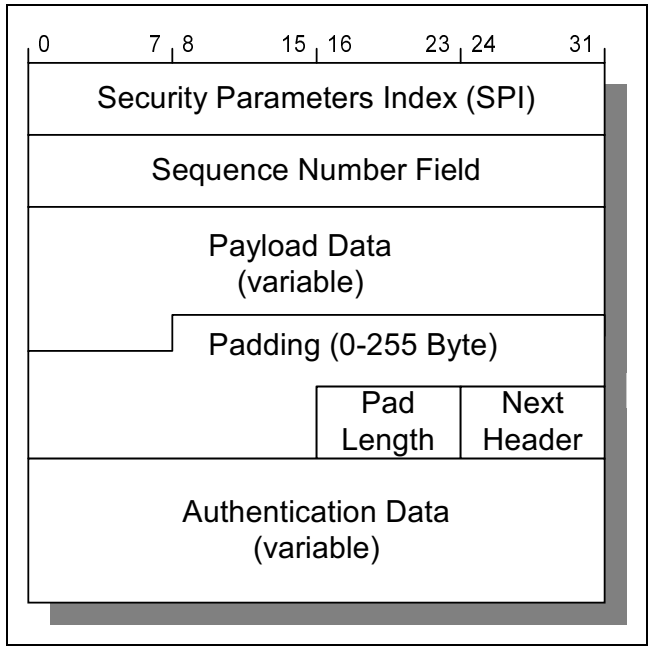


Abb. 2.33 Encapsulating Security Payload

Abb. 2.33 zeigt den Aufbau des Encapsulating Security Payload Headers nach RFC-2406, welcher aus folgenden Feldern besteht: [KEAT98b]

- **Security Parameters Index (SPI):** ein 32-Bit-Wert, der die zur Zieladresse des Datenpakets gehörige Security Association (SA) identifiziert
- **Sequence Number:** enthält eine Sequenznummer, falls Anti-Replay genutzt werden soll. Dabei ist das Vorhandensein dieser Nummer auf Senderseite erforderlich, auf der Empfängerseite wird erst entschieden, ob die Sequenznummern ausgewertet werden. In diesem Fall darf sich keine Nummer wiederholen, d.h., nach 2^{32} Paketen muss auf beiden Seiten ein Reset durchgeführt werden.

22 Data Encryption Standard
23 Message Digest Version 5
24 Secure Hash Algorithm

- **Payload Data:** In diesem Feld sind die verschlüsselten Daten untergebracht. Je nachdem, welches Verschlüsselungsverfahren eingesetzt wird, kann es nötig sein, den Daten einen Initialization Vector (IV) voranzustellen.
- **Padding:** enthält Fülldaten, die beispielsweise beim Einsatz eines Block-Verschlüsselungsalgorithmus benötigt werden, um die Daten bis zu einem vollen Block aufzufüllen. Außerdem muss sichergestellt werden, dass die chiffrierten Daten bis zur nächsten 4-Byte-Grenze reichen, was ebenfalls durch Auffüllen erreicht wird.
- **Pad Length:** ist die Anzahl von Bytes, der Padding-Daten aus dem vorherigen Feld und darf zwischen 0 und 255 groß sein, wobei 0 bedeutet, dass kein Padding in diesem Header genutzt wurde.
- **Next Header:** identifiziert den Typ des nächsten Nutzdatenpakets nach dem Authentication Header.
- **Authentication Data:** ist der Integrity Check Value (ICV) für dieses Paket.

Bei den Implementierungen kann man drei grundsätzliche Möglichkeiten unterscheiden, um IPsec in die IP-Protokollschicht zu integrieren:

- **Oberhalb der IP-Schicht:** Bei dieser Methode muss der Source-Code der logischen Schnittstelle zwischen TCP und IP verfügbar sein. Außerdem ist nur ESP möglich, da die IPsec-Pakete logisch durch die IP-Schicht getunnelt werden müssen, was bei AH nicht möglich ist.
- **Innerhalb der IP-Schicht:** Dies ist besonders bezüglich der Performance die beste Lösung, da die IPsec-Funktionalität direkt in den IP-Schicht-Treiber integriert wird. Allerdings ist hierbei selbstverständlich der Source-Code der IP-Schicht nötig.
- **Unterhalb der IP-Schicht:** Mit dieser Methode macht man die IPsec-Implementierung unabhängig vom TCP/IP-Protokollstack, allerdings ist es hierbei nötig, Fragmentierung und IP-Checksummenbildung neu zu integrieren.

IPsec schützt IP-Pakete vor Modifikation und Abhören und beeinflusst keine anderen Protokolle oder Anwendungen. Ausnahmen bestätigen allerdings auch hier die Regel: SNA[25] über IP. Hier wird Einfluss auf die Übertragung genommen, da einige für SNA notwendige Informationen durch IPsec verschlüsselt werden, sodass eine einfache Migration nicht gewährleistet ist. Normalerweise findet die Datenübertragung über IPsec aber transparent statt. IPsec-Pakete werden so über Router oder Switches weitergeleitet, ohne dass eine Softwareangleichung erfolgen muss. Das liegt daran, dass IPsec unterhalb der Transportschicht des OSI-Referenzmodells arbeitet. Somit ist es unabhängig bezüglich der vorhandenen Client/Server-Software. Es lässt sich also eine End-to-end-Kommunikation über beliebige Netze herstellen, ohne auf die bestehende

25 Systems Network Architecture

Hard- und Software Rücksicht nehmen zu müssen. Demnach ist IPsec einfach in bestehende IPv4-Umgebungen integrierbar.

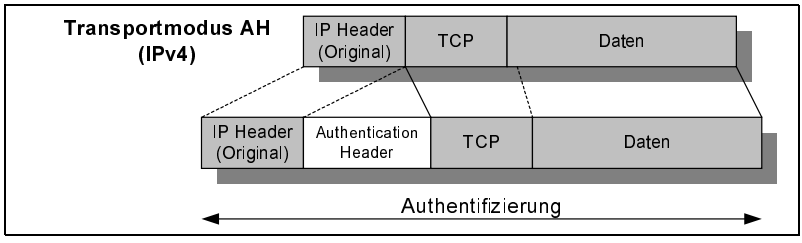


Abb. 2.34
AH im Transportmodus

Zum Transport der Pakete erhält man zwei verschiedene Modi, den Transportmodus und den Tunnelmodus, die sich im Wesentlichen durch den Aufbau der Pakete und die Einsatzmöglichkeiten unterscheiden. Im Transportmodus werden nur die Daten der Transportschicht geschützt, nicht aber der IP-Header. Im Tunnelmodus wird das gesamte Datenpaket mitsamt seines IP-Headers verschlüsselt und verpackt und mit einem neuen Header versehen. Es ist also möglich, die eigentlich zu schützenden Daten in einem weiteren IP-Paket zu kapseln und damit die so zusammengebauten Pakete über Gateways zu versenden, die selbst kein IPsec unterstützen. Abb. 2.34 zeigt, dass das Originalpaket aufgespaltet und der Authentication Header nach dem IP-Header und vor den Nutzdaten eingefügt wird.

Bei der Benutzung des ESP ist es nötig, die eigentlichen Nutzdaten in das ESP zu integrieren, da diese hier verschlüsselt werden. In der Abb. 2.35 ist das angedeutet, indem das ESP in Header und Trailer aufgeteilt ist, die die verschlüsselten Nutzdaten einschließen, und außerdem kann wie oben beschrieben ein Authentizitätsfeld angehängt werden.

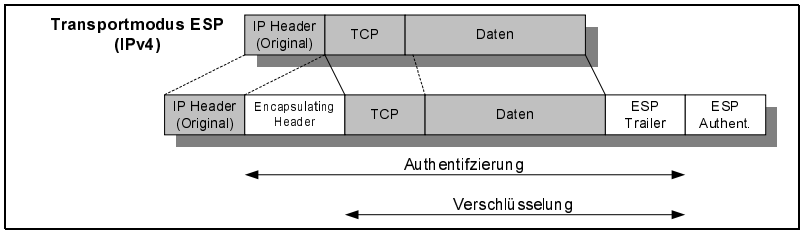


Abb. 2.35
ESP im Transportmodus

Der Tunnelmodus hingegen erfordert, dass das ursprüngliche Pakete vollständig neu verpackt wird und daher der Authentication Header und ein neuer IP-Header vorangestellt werden müssen, wie in Abb. 2.36 gezeigt.

ESP schließt im Tunnelmodus das ursprüngliche Paket ein, um es insgesamt zu verschlüsseln. Dem so neu entstandenen Paket wird ein neuer IP-Header vorangestellt. Abb. 2.37 zeigt den ESP im Tunnelmodus.

Abb. 2.36
AH im Tunnelmodus

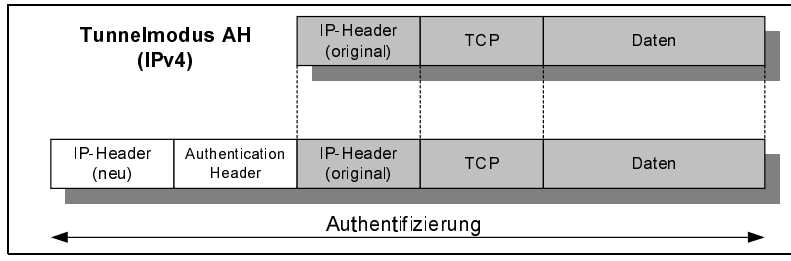
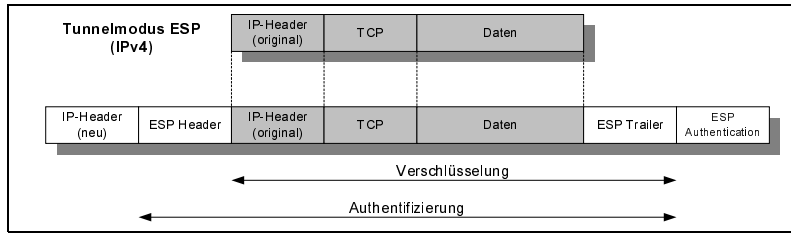


Abb. 2.37
ESP im Tunnelmodus



Die verzögerte Einführung eines standardisierten Key Management liegt an dem zeitlichen Ablauf der Diskussion der Arbeitsgruppe IPsec der IETF. Diese hatte zunächst die Sicherheitsfunktionen für IP spezifiziert und erst anschließend ein entsprechendes Protokoll für Key Management. Dabei stammt je eine Implementierung von dem National Institute of Standards and Technology (NIST) und Cisco Systems. Die erste Möglichkeit sieht dabei vor, dass die Schlüsselinformationen mit dem IP-Paket durch zusätzliche Header mitgeschickt werden. Das heißt, das Key Management verbleibt auf der gleichen Schicht. Der zweite Ansatz ist ein separates, universelles Key-Management-Protokoll, welches auf der Anwendungsebene arbeitet. Suns SKIP (Simple Key Management over IP) ist ein Beispiel für den ersten Ansatz, während ISAKMP die zweite Möglichkeit darstellt.

Um aber der Sicherheit der Kommunikation und sich schnell ändernden Konfigurationen Rechnung zu tragen, muss auf jeden Fall ein Key Management eingeführt werden. Dieses sollte automatisch den Schlüsselaustausch zwischen den Teilnehmern vornehmen. Durch die Aufnahme neuer oder den Wegfall bestehender Kommunikationspartner ist daher von einer manuellen Konfiguration abzusehen. Große Installationen sind auf ein automatisches Key Management angewiesen, da eine manuelle Konfiguration der Security Association (SA) zwischen den Gateways (Firewall-Firewall) und Hosts (Client/Server) zu aufwendig ist. Genau dazu ist das Protokoll ISAKMP beziehungsweise dessen Realisierung in IKE entwickelt worden. Zusätzlich müssen die unterschiedlichen Sicherheitsrichtlinien der Teilnehmer aufeinander angeglichen werden. IKE arbeitet in zwei Phasen, der Authentifizierungs- und Schlüsselgenerierungsphase. Vorteile des Verfahren sind, dass ohne großen Aufwand neue

Schlüssel generiert werden können und dass das Verschlüsselungsverfahren ausgetauscht werden kann. Diese abstrakten Definitionen werden in der RFC-2409 genau beschrieben. Allerdings sind nicht alle bisherigen Entwicklungen außerhalb der USA frei verfügbar.

SA ist eine Art Vertrag über die einzuhaltenden Sicherheitsparameter einer bestimmten Kommunikationsbeziehung, die eindeutig festgelegt sind durch die Zieladresse und einen Security Parameter Index (SPI). Sie bilden einen Satz von Sicherheitsparametern, wie Authentifizierung und/oder Verschlüsselung, Schlüssel, Initialvektor für Verschlüsselung, Gültigkeitszeitraum der Schlüssel und der SA sowie die Adresse des Senders. Der Empfänger definiert den SPI und legt den Inhalt der SA fest.

Sowohl im AH als auch im ESP wird durch den Security Parameter Index (SPI) und die Zieladresse eine so genannte Security Association (SA) identifiziert und damit festgelegt, welche Parameter für die Authentisierung und Verschlüsselung eingesetzt werden sollen. Diese Parameter werden in RFC-2401 genauer spezifiziert: [KEAT98a]

- Authentisierungsalgorithmus und Schlüssel für die Authentisierung im AH
- Verschlüsselungsalgorithmus und Schlüssel für die Verschlüsselung im ESP
- Vorhandensein/Nichtvorhandensein und Spezifikationen von bestimmten kryptographischen Methoden (ESP)
- Algorithmus für die Authentisierung und Schlüssel, wenn gewünscht, für ESP
- Gültigkeitszeit der Schlüssel (AH und ESP)
- Gültigkeitszeit der Security Association
- Quelladresse der Security Association
- Sensibilitätsgrad der zu schützenden Daten, falls mehrstufige Sicherheit unterstützt werden soll

ISAKMP ist nicht auf bestimmte Sicherheitsmechanismen, kryptographische Verfahren oder Schlüsselgenerierungsverfahren beschränkt, sondern kann auch in anderen Bereichen als nur in den Anwendung IPsec verwendet werden. Als besondere Stärken von ISAKMP werden Schutz vor Denial-of-Service-Angriffen, vor Replay- oder Reflection-Angriffen, vor Man-in-the-middle-Angriffen und vor ungewollten Verbindungsumleitungen angesehen. ISAKMP definiert keinen Schlüsselaustausch. Dies bleibt anderen Protokollen wie IKE bei IPsec überlassen. IKE benutzt die zwei Phasen von ISAKMP: die erste Phase etabliert die IKE-SA, während die zweite Phase diese SA nutzt, um die IPsec-SA festzulegen. [MSST98]

Internet Key Exchange (IKE) ist ein in RFC-2409 beschriebenes konkretes Schlüsselaustauschverfahren, das speziell auf die Benutzung mit ISAKMP abgestimmt ist. Die Beschreibung von IKE umfasst genaue Definitionen für das gesamte Verfahren des Schlüsselaustauschs. Dabei werden von einer IKE-Implementierung folgende Eigenschaften erwartet: [HACA98]

- Unterstützung des Data Encryption Standard (DES) im CBC[26]-Modus
- Unterstützung der Hash-Funktionen MD5 und SHA
- Authentifizierung durch zuvor ausgehandelte Schlüssel

IKE kann als universelles Austauschprotokoll verwendet werden. Es wird bei IPsec für den Austausch von Strategien und auch zur Authentifizierung von verschlüsselten Materialien unterschiedlichster Art benutzt werden, wie beispielsweise SNMPv3[27] oder OSPFv2[28]. Die Spezifikation wofür IKE verwendet wird, kann in der Domain-of-Interpretation (DOI) definiert werden. Es existiert eine DOI für IPsec, wo nach RFC-2407 festgelegt wird, wie Sicherheitsassoziationen (SA) ausgehandelt werden. Wenn IKE für andere Protokolle benutzt wird, müssen hier eigene DOI definiert werden.

Die SA von IKE unterscheidet sich von der IPsec-SA. Die IKE-SA definiert einen Weg, auf dem zwei Partner miteinander kommunizieren; beispielsweise welcher gemeinsamer Algorithmus verwendet werden soll. Die IKE-SA wird anschließend dazu benutzt, eine beliebige Menge von IPsec-SA zwischen diesen beiden Teilnehmern zu erzeugen. Daher ist die Mitteilung der entsprechenden Sicherheitsanforderung mittels IKE die Aktion, die durchgeführt werden muss, wenn ein Eintrag in der Datenbank für Sicherheitsstrategien auf einen Null-Eintrag verweist, um anschließend festzulegen, wie IPsec die SA erzeugen soll.

Die durch IKE erzeugten IPsec-SA können optional Perfect Forwarding Secrety (PFS)[29] der Schlüssel bieten, wodurch nach Wunsch auch die Identität der Teilnehmer geheim gehalten werden. Mehrere Paare von IPsec-SA können auf einmal durch einen einzelnen IKE-Austausch erzeugt werden. Eine beliebige Anzahl solcher Austauschvorgänge kann durch eine einzelne IKE-SA ausgeführt werden. Diese Vielfalt an Optionen bewirkt, dass IKE stark erweiterungsfähig und komplex ist.

Während die von IPsec beschriebenen Möglichkeiten in IPv4 noch optional implementiert werden können, sind die Sicherheitsmechanismen in IPv6 fest integriert. Damit besteht also trotz der Verzögerung der Einführung von IPv6 schon jetzt die Möglichkeit, Datenverkehr auf IP-Ebene hinsichtlich der Eigenschaften Authentizität, Integrität und Vertraulichkeit zu verarbeiten. Auf IPsec wird bei der Etablierung einer Sicherheitsplattform zu einem späteren Zeitpunkt noch genauer eingegangen, da es heute die größte Gewichtung im Internet besitzt und sich alle später definierten Protokolle nach dieser Spezifikation richten müssen. [DEER01]

---

26 Ciper Block Chaining
27 Simple Network Management Protocol, Version 3
28 Open Shortest Path First, Version 2
29 vorwärts gerichtete Sicherheit bei IPsec, bei der alle Daten geschützt übertragen werden, unter anderem auch die Identität der Teilnehmer

3

Quality-of-Service

In diesem Kapitel sollen die unterschiedlichen Ansätze zur Erreichung der Dienstgüte auf der Layer-2-Ebene mittels ATM und Layer-3-Ebene mittels IP detailliert betrachtet werden, um daraus eine spätere Lösung abzuleiten, wie man Qualitäten im Netzwerk vergeben kann. Dabei liegt der Schwerpunkt auf IP-QoS, weil das Internet dominiert und man auf der Layer-3-Ebene heute Prioritäten einstellen kann. Unterteilt wird dieser Abschnitt deshalb in die folgenden zwei Hauptbereiche:

- ATM-QoS: CBR, VBR und ABR
- IP-QoS: IntServ, RSVP, DiffServ

3.1 ATM-QoS

Um eine Dienstgüte (Quality-of-Service) einer virtuellen ATM-Verbindung anbieten zu können, sind bestimmte Dienstklassen definiert worden. Die dadurch mögliche Echtzeitfähigkeit ist für ATM von entscheidender Bedeutung. Keine andere Technologie ist in der Lage, isochrone Datenströme so zu handhaben wie der Asynchrone Transfer Modus (ATM). Reine Datenübertragung kann auftretende Zellenverluste immer durch Wiederholung der Daten ausgleichen. Dies ist bei der Echtzeitkommunikation via Audio und Video nicht möglich. Laufzeitschwankungen, so genannte Jitter, stören hier empfindlich die Übertragungsqualität. Gerade bei der Sprachübertragung fallen kleinere Verzögerungen und Zellenverluste sehr unangenehm auf.

Die Hauptmotivation, um ATM einzuführen und weiterzuentwickeln, ist die Flexibilität von ATM sowie die Unterstützung existierender und neuer Dienste. Untersuchungen in europäischen Pilot- und Forschungsprojekten[1] haben jedoch bewiesen, dass die Verkehrskontrollfunktionen auf den Bedarf der speziellen Anwendungskategorien und ihren unterschiedlichen QoS-Anforderungen angepasst werden müssen, um einen effizienten Transport der Daten

1 Unter anderem Experimental Platform for Engineering Research and Trials – EXPERT (AC094), European Information Exchange Services between harbour sites – EIES (AC075)

gewährleisten zu können. Weiterhin existieren bedingt durch Verwendung des statischen Multiplexing keine inhärenten Bandbreitenbegrenzungen eines Kanals. Das bedeutet, bei dem ausschließlichen Angebot einer Dienstklasse wie Constant Bit Rate (CBR) könnte unter schlechten Bedingungen das ATM-Netz bereits bei 40% voll belastet sein[2], da nicht alle Teilnehmer diese Bitrate vollständig ausnutzen. Damit die vorhandene Bandbreite intelligent ausgenutzt werden kann, hat das ATM-Forum und die ITU-T ATM-Dienstkategorien entwickelt, die verschiedene Verkehrsarten beinhalten und unterschiedliche Anforderungen an das Netz stellen. Diese ermöglichen eine Auslastung eines ATM-Netzes von bis zu 90 %. Somit kommen die folgenden Dienstklassen zum Einsatz, die durch verschiedene Verkehrskontrollfunktionen unterstützt werden:

- Constant Bit Rate (CBR)
- Real-time – Variable Bit Rate (rt-VBR)
- Non-real-time – Variable Bit Rate (nrt-VBR)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)
- ATM Block Transfer (ABT)
- Deterministic Bit Rate (DBR)
- Statistical Bit Rate (SBR)

Die letzten drei Dienstklassen beziehen sich dabei auf die Spezifikationen der ITU-T und haben einen wesentlich geringeren Bekanntheitsgrad als die des ATM-Forums. Deshalb werden bei heutigen Produkten auch nur CBR, UBR und VBR angeboten. ABR gibt es als ITU-T- und ATM-Forum-Spezifikation. Sie stellt aber eine kompliziertere Realisierung dar, die später umgesetzt wurde, weshalb sie nicht in allen verfügbaren Switches und Adapterkarten integriert ist.

Für Echtzeitübertragungen sind die Dienstklassen CBR sowie rt-VBR bestimmt. Die CBR-Dienstklasse benötigt eine ganz bestimmte Bandbreite und Dienstgüte, die weder signifikante Verzögerungen noch Jitter oder Zellenverluste erträgt. Typische Anwendungen sind Telefonie, Video-on-Demand (VoD) und Videokonferenzen. Rt-VBR verhält sich ähnlich zu CBR, wobei geringfügige Veränderungen der Bandbreite (z.B. Video) sowie kleinere Zellenverluste toleriert werden, da keine Sicherung der Übertragung gewährleistet wird. Die Sustainable Cell Rate (SCR) ist als zusätzlicher Parameter bei rt-VBR gegenüber CBR eingefügt worden. Die Verkehrsfestlegungen beider Klassen sind definiert und die benötigten Verkehrskontrollfunktionen leicht verständlich. Die Dienstklasse nrt-VBR bietet hingegen die gesicherte Datenübertragung an. Das heißt, diese Verkehrsart unterstützt Resending-Eigenschaften, welche sich bei Echt-

2 Je nachdem, wie viel Bandbreite von den Teilnehmern reserviert wird (z.B. 20 Mbit/s pro Teilnehmer führen zu einer Begrenzung von 7 Teilnehmern).

zeitübertragungen negativ auswirken würden. Die Bandbreite wird statisch zugewiesen und ermöglicht ausschließlich Datenverkehr, beispielsweise die gesicherte Kommunikation zwischen zwei LANs. Während der Übertragung können Perioden ohne Datenverkehr auftreten. Somit ist die Verwaltung des VBR-Dienstes schwieriger, als das bei CBR der Fall ist.

Unter der Dienstklasse UBR versteht man Datenverkehr, der keine Echtzeitfähigkeit benötigt. Anwendungen sind beispielsweise E-Mail- und Telnet-Applikationen, die Burst-Charakteristik aufweisen. Aus diesem Grund werden keine Spezifikationen, wie der QoS, verwendet. Damit bei UBR aber Netzwerkfunktionen sicher unterstützt werden, sind trotzdem einige Funktionen spezifiziert worden. UBR überträgt weiterhin die Daten mit der maximal zur Verfügung stehenden Geschwindigkeit. Bei dem Scheitern einer Datenübertragung durch zu hohen Zellenverlust wird der Transport zu einem späteren Zeitpunkt durch ein höheres Schichtenprotokoll (z.B. TCP) wieder aufgenommen. UBR ist damit mit dem Best-effort im Internet zu vergleichen. Das heißt, die Bandbreite wird nur nach der Verfügbarkeit zugewiesen.

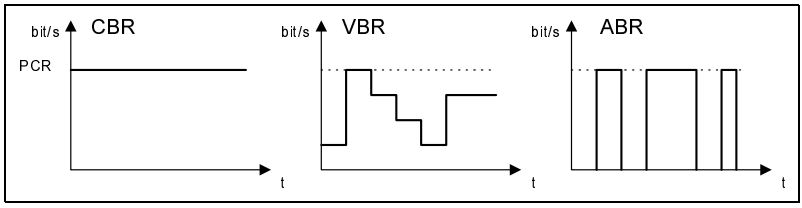


Abb. 3.1 Eigenschaften von ATM-Dienstklassen

Als letzte wichtige Serviceklasse bleibt ABR zu nennen. Diese Klasse ist für zeit- und fehlerunkritischen LAN-Datenverkehr geeignet. Das heißt, ABR besitzt Burst-artige Eigenschaften wie sie im LAN (Ethernet, Token Ring, FDDI) entstehen. Während der kurzen Übertragungsphasen werden alle verfügbaren Zellen mit Nutzinformationen besetzt. Dabei wird die Bandbreite vom Netz elastisch zugewiesen. Im Unterschied zu UBR findet bei der ABR-Dienstklasse eine kontrollierte Einspeisung in das Netzwerk statt, wobei nicht nur der Burst, sondern auch Paketgrößen eine wichtige Rolle spielen. Die Paketgrößen bestehen aus der ATM Adaptation Layer Protocol Data Unit (AAL-PDU) und liegen meistens weit über der ATM-Zellengröße von 53 Byte. Die Übertragung in AAL-PDU-Paketen hat zur Folge, dass bei Datenverlust nur einer Zelle das gesamte Paket unbrauchbar wird. Für den ABR-Dienst ist der Verkehrskontrakt und das Rahmenwerk für die Rückmeldungskontrollen ebenfalls spezifiziert. Im Gegensatz zu VBR und CBR beschränkt das Netz automatisch die Anforderung der Endgeräte, um das Netz nicht zu überlasten. ABR orientiert sich immer an der Netzauslastung und stellt dem Benutzer nur die freien Ressourcen zur Verfügung. Dabei ist der Benutzer auch in der Lage, die gewünschte Datenmenge skalierbar einzustellen.

Verschiedene Anwendungen besitzen unterschiedliche Anforderungen, wie Bitraten, Zellenverlustrate und Zeitverzögerung, um bestimmte Leistungskriterien erfüllen zu können. Deshalb hängt die Verkehrscharakteristik sehr davon ab, für welche Aufgaben die Applikationen entworfen und entwickelt wurden. Das Einrichten der Kontrollfunktionen zusammen mit der Pufferarchitektur und den Dienstprioritäten wird für die effektive Unterstützung der verschiedenen QoS-Anforderungen benötigt. Zusammengefasst stellt dies eine integrierte Verkehrskontrollarchitektur dar. Dabei muss bei der Definition einer solchen Architektur sichergestellt werden, dass alle verschiedenen Dienstklassen mit ihren speziellen Kontrollfunktionen auf effektive Art und Weise miteinander kooperieren. Als Ergebnis werden definierte Parameter für die aktuelle Verbindung über einen Traffic Contract eingehalten. Dieser enthält Parameter, die zwischen dem Endgerät und dem ATM-Switch ausgehandelt werden und Verkehrsprofil, maximale Zeitverzögerung, zulässige Zellenverlustrate und Dienstgüte enthalten. Die Usage Parameter Control (UPC) überwacht dabei die Einhaltung des jeweils verhandelten Verkehrsprofils sowie die Richtigkeit der Pfad- bzw. Zellenidentifikation (VPI[3], VCI[4]). Zusätzlich ermöglicht Policing das Verwerfen von Zellen, die den Traffic Contract nicht einhalten (z.B. Übertretung der Spitzenlast). Durch das Header-Bit Cell Loss Priority (CLP) können die Zellen einer niedrigen Priorität von vornherein zugeordnet werden, indem man CLP=1 setzt. Diese Zellen können dann bei starker Belastung des Netzes verworfen werden, um sensitive Daten zu schützen.

Vor einem Verbindungsaufbau werden Verhandlungen zwischen den ATM-Endgeräten und den ATM-Switches geführt, um die Übertragungsqualität über unterschiedliche Parameter festzulegen. Der Traffic Contract garantiert die Einhaltung dieser Parameter. Die Messtechnik muss für die Überprüfung eine system- und netzunabhängige Vertragsprüfung durchführen und die Regelverstöße dokumentieren. Um den QoS garantieren zu können, sind folgende Verkehrs- und Performance-Parameter vereinbart worden, die eine Änderung des Verkehrsvertrags sofort anzeigen:

- **Verkehrslast:** Die absolute Verkehrslast auf einer Verbindung oder an einem Port wird definiert durch die Zellentransportrate. Die Einheit lässt sich angeben durch [Zellen/sec].
- **Relative Verkehrslast:** Die relative Verkehrslast ist das Verhältnis der Anzahl der genutzten Zellenslots dividiert durch die Anzahl der vorhandenen Zellenslots. Die Einheit wird in [%] angegeben. In der ATM-Schicht werden die nicht verwendeten Zellenslots auf einer physikalischen Verbindung immer mit Leerzellen belegt.

3 Virtual Path Identifier
4 Virtual Channel Identifier

- **Peak Cell Rate (PCR):** Zulässige Spitzenzellenrate, die nicht überschritten werden darf, es sei denn, es wird einer Überbuchung und damit Übertretung des Traffic Contract stattgegeben.
- **Sustainable Cell Rate (SCR):** Die zulässige Zellenrate, die für die Verbindung dauerhaft zur Verfügung gestellt werden kann. Dabei müssen alle definierten Verkehrsparameter eingehalten werden.
- **Mean Cell Transfer Delay (MCTD):** Dieser Wert wird durch die Aufenthaltsdauer der Zelle im ATM-Netz bestimmt und ist somit eine statistisch schwankende Größe. Durch diese Zufallsvariable sind N unabhängige Messungen erforderlich, um den Mittelwert bestimmen zu können. Dadurch können Änderungen, bedingt durch Warteschlangen, Vermittlung usw., berücksichtigt werden.
- **Cell Delay Variation (CDV):** Die Laufzeitschwankungen sind für Echtzeitanwendungen möglichst gering zu halten. Dabei können sowohl Zellenverluste als auch Änderungen der Laufzeit, bedingt durch Zwischenspeicherung und Vermittlung, zu Störungen führen.
- **Cell Loss Ratio (CLR):** Der Zellenverlust ist das Verhältnis der Anzahl der verlorenen Zellen zu der Gesamtzahl der gesendeten Zellen innerhalb eines bestimmten virtuellen Kanals. Eine genauere Erfassung des CLR-Wertes kann durch die Auskopplung von langen Fehler-Bursts (SECBR[5]) erfolgen.
- **Cell Error Ratio (CER):** Der CER-Wert gibt den Anteil aller Zellen an, die fehlerhaft übertragen wurden. Bei reiner Datenübertragung bedeutet der Verlust eines Pakets die Wiederholung der Übertragung und damit eine Verringerung des Durchsatzes.
- **Severely Errored Cell Block Ratio (SECBR):** Der SECBR-Wert gibt den Anteil aller Zellenblöcke an, die fehlerhaft übertragen wurden. Das heißt, ein kompletter Zellenblock ist ungültig und wird verworfen.
- **Cell Transfer Rate (CTR):** Die Zellenmenge, die während eines Datenaustauschs transportiert wird, wird durch diesen Parameter angegeben. Bei isochronen Datenströmen muss dieser Wert sehr klein sein, um eine Verständlichkeit zweier Kommunikationspartner gewährleisten zu können.
- **Cell Misinsertion Rate (CMR):** Die Zellen, welche fehlerhaft in den aktuellen Zellenstrom eingefügt werden, kann man durch diesen Parameter feststellen. Das heißt, die Zelle ist über den falschen virtuellen Kanal und/oder Pfad empfangen worden, wodurch Zellen verworfen werden mussten.

5 Severely Errored Cell Block Ratio

Tab. 3.1 Gegenüberstellung der Dienstklassen und der Verkehrsparameter

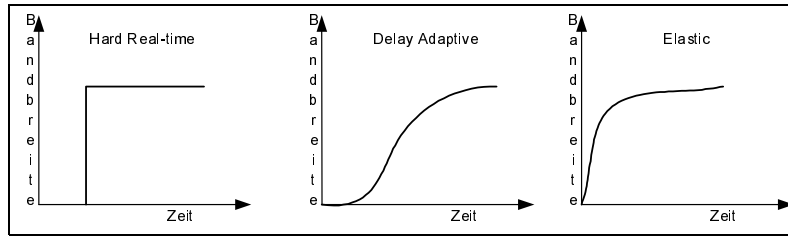
Dienst-klasse	Verkehrsparameter	Bandbreite	Zellen-verlustrate	Transitzeit
CBR	PCR, CDV	Ja	Ja	Ja
Rt-VBR	PCR, SCR, CDV	Ja	Ja	Ja
Nrt-VBR	PCR, SCR, MBS	Ja	Ja	Nein
ABR	PCR, MCR	Ja	Ja	Nein
UBR	(PCR)	Nein	Nein	Nein

Nachteilige Beeinflussungen der ausgehandelten Verkehrsparameter ergeben sich durch Übertragungsfehler, Pufferkapazität, Netzlast, virtuelle Kanal-/Pfadkapazität und die Durchschaltverzögerung der ATM-Switches. Diese Beeinflussungen treten allerdings in jedem Netz auf. Ziel der ATM-QoS ist es, die Netzparameter zu berücksichtigen, ohne eine Veränderung der Dienstqualität zu bekommen. Anhand der vorhandenen Verkehrsparameter lassen sich wiederum die bekannten Dienstklassen bilden. Von welchen Parametern die jeweiligen Dienstklassen abhängen, klärt Tab. 3.1. [DETK00a]

3.2 IP-QoS

Echtzeitdaten lassen sich nicht ohne weiteres über IP-Netze transportieren. Die Voraussetzung dafür ist, dass Ressourcen wie Bandbreite, Bitrate oder Verarbeitungsleistung in einer garantierten Qualität zur Verfügung stehen. Das Internet arbeitet jedoch nach dem Best-effort, das keine garantierte Dienstgüte unterstützt. Um dieses Problem in den Griff zu bekommen, wurden QoS-Mechanismen für Multimedia-Anwendungen entwickelt. Sie beruhen auf abgestuften QoS-Niveaus. Solche Ansätze sind die Integrated Services (IntServ) und Differentiated Services (DiffServ).

Abb. 3.2 Unterschiedliche Eigenschaften der Anwendungen



Die Dienstgüteanforderung ist je nach Anwendung unterschiedlich. Dabei steht die Verzögerung einzelner Pakete im Vordergrund, die durch die maximale und minimale Verzögerung begrenzt wird. Echtzeitapplikationen benötigen dabei eine garantierte Verzögerung, während andere Anwendungen weniger anfällig

sind. Die Integrated Service Group unterteilt diese Anwendungen in Hard Realtime, Delay Adaptive und Elastic Applications. Die erste Gruppe ist sehr empfindlich gegenüber Störungen, während die zweite Gruppe leichte Verzögerungen toleriert. Die dritte Gruppe bilden reine Datenanwendungen, die große Zeitverzögerungen zulassen und das Zwischenspeichern von Daten zulassen.

Basierend auf den Eigenschaften wurden von der IETF[6] fünf unterschiedliche Serviceklassen definiert. Eine Serviceklasse beinhaltet dabei jeweils eine Untergruppe möglicher Dienstgüte-Parameter. Die Eigenschaften dieser Klassen decken sich direkt mit denen der Anwendungen, die sie angefordert haben. Die Gesamtheit aller Serviceklassen spiegelt nun die Leistungsfähigkeit des Übertragungssystems wider. Folgende Serviceklassen wurden definiert, wobei nur die ersten beiden heute noch Bedeutung besitzen:

- Controlled Load Service
- Guaranteed Service
- Controlled Delay Service
- Predictive Delay Service
- Committed Rate Service

Aus diesem Serviceklassen heraus hat man ebenfalls Parameter definiert, die den Datenverkehr beschreiben sollen und sich ähnlich wie die ATM-Verkehrsparameter verhalten:

- **Bucket Rate (r):** die Erzeugungsrate der Token entspricht der mittleren Datenrate des Verkehrsflusses in Bytes/s.
- **Bucket Depth (b):** die maximal zulässige Burst-Grösse für den Datenfluss in Bytes.
- **Peak Rate (P):** die höchstzulässige Datenrate für den betrachteten Datenfluss. Mathematisch ist die Spitzenrate als der kürzeste Zeitabstand zwischen zwei ankommenden Bytes zu sehen. Die Einheit ist Bytes/s.
- **Maximum Packet Size (M):** die im ausgehandelten Profil maximal zugelassene Paketgröße in Bytes. Diese muss kleiner oder gleich der kleinsten MTU-Size[7] entlang des Datenpfades sein. Somit wird sichergestellt, dass unterwegs keine Fragmentierung der IP-Pakete stattfindet.
- **Minimum Policed Unit (m):** eine Größe in Bytes. Ist ein Paket kleiner als *m*, so wird es beim Accounting mit der Größe *m* berechnet. Dieser Wert muss kleiner als M sein. Als Richtwert sollte man m auf die Summe aller Header (IP-, TCP/UDP-, RTP-Header) im IP-Paket setzen.
- **Burst-Time (tb):** eine Zeitspanne, die notwendig ist, um den größtzulässigen Burst mit der maximal möglichen Datenrate zu übertragen.

6 Internet Engineering Task Force

7 Maximum Transmission Unit = maximale übertragbare Paketgröße

Der Guaranteed Service nach RFC-2212 berücksichtigt alle Serviceparameter und Kombinationen aus ihnen. Das heißt, hier wird nicht nur der Durchsatz als Maß der Dienstgüte herangezogen, sondern ebenfalls die Verzögerungszeiten. Es fehlen dennoch notwendige Parameter wie Jitter und Latenzzeiten. Ein Datenfluss wird durch die Beschreibung eines Token Bucket definiert. Durch diese Verkehrsbeschreibung ist das Netzwerk in der Lage, verschiedene Parameter zu berechnen, die beschreiben, wie der Datenfluss gehandhabt wird. Dadurch wird die maximale Verzögerung ermittelt, die sich durch eine feste sowie eine Pufferverzögerungszeit definiert. Die feste Verzögerung beinhaltet die Auswahl des Weges durch das Netz, bestehende Übertragungsverzögerungen usw. Die Pufferverzögerungszeit, bestimmt durch den Guaranteed Service, wird durch zwei primäre Funktionen von zwei Parametern beschrieben:

- Token Bucket
- Angeforderte Datenrate der Applikation

Der Token Bucket ist eine Verkehrsbeschreibung, die zur Kontrolle der Verkehrsparameter in QoS-fähigen Netzen eingesetzt wird. Die Funktionsweise ist vergleichbar mit einem Behälter mit dem Volumen *b*. Von einem Token-Generator werden Token mit der konstanten Rate *r* erzeugt, wie dies in Abb. 3.3 gezeigt wird. Damit wird der Behälter (Bucket) kontinuierlich gefüllt. Kommt ein Datenpaket an, so muss die Anzahl der Token im Bucket mindestens so groß wie das Paket sein[8]. Ist dies der Fall, so wird das Paket als filterkonform weiterverarbeitet, der Inhalt des Buckets wird um die Größe des Datenpakets verringert. Sonst wird das Paket als nicht filterkonform behandelt.[9] Kommen über einen längeren Zeitabschnitt keine Datenpakete an, so füllt sich der Bucket. Ist der Bucket voll, so werden neu erzeugte Token verworfen. Dieser Mechanismus hat große Ähnlichkeit mit dem Leaky-Bucket-Verfahren, welches in ATM spezifiziert wurde. Mit den beiden Parametern *r* und *b* des Bucket-Filters ist es bei diesem Verfahren gewährleistet, dass für beliebige Zeitspannen T maximal n Bytes als filterkonform behandelt werden, wobei n folgende Ungleichung erfüllt:

$$n \leq r \cdot T + b$$

Angewandt auf die Datenübertragung bedeutet jetzt *r* die mittlere Übertragungsrate der Daten und *b* die maximale Burst-Größe. Solche Filter lassen sich zur Kontrolle des Datenverkehrs folgendermaßen einsetzen: Liegt *b* in der Größenordnung von *r*, so kann damit der Datenverkehr anhand seiner Paketverkehrsaufkommen gefiltert werden. Setzt man dagegen $b << r$, so wird anhand

8 Unterschied zum ITU-Modell des Leaky-Bucket-Mechanismus, da statt Zellen (bei ATM) Pakete unterschiedlicher Größe (bei IP) verwendet werden.

9 Die Behandlung dieser Daten ist je nach Anwendungsfall sehr unterschiedlich. Es können beispielsweise die Daten verworfen oder als Best-effort weitertransportiert werden.

der Datenrate gefiltert. Man kann durch Reihenschaltung mehrerer Token-Bucket-Filter komplexere Filter konstruieren. Außerdem sollte eine Dienstspezifikation den Datenfluss mit Hilfe von Token-Bucket-Parametern beschreiben.

Um diesen Mechanismus bei IntServ nutzen zu können, muss in jedem Endgerät und Router eine Admission-Control-Einheit vorhanden sein, was den Aufwand natürlich erheblich erhöht. Wenn nun eine Reservierungsfrage auf einer bestimmten Schnittstelle ankommt, prüft diese Einheit, ob ausreichende Ressourcen verfügbar sind und ob der angefragte Dienst mit dem auf der Schnittstelle vorhandenen Paket-Schedule-Verfahren garantiert werden kann. Dabei hängt die Arbeitsweise der Admission-Control-Einheit von der Art des Systems und dem eingesetzten Paket-Schedule-Verfahren ab, wodurch wieder Herstellerunterschiede auftreten. Zusätzlich zur Reservierungsanfrage wird die Berechtigung des Teilnehmers geprüft.

Im Gegensatz zur Admission-Control-Einheit prüft die Policy-Control-Einheit die Berechtigungen der Teilnehmer zur QoS-Anforderungen. Außerdem ist sie zuständig für die Authentisierung der Teilnehmer im Netz. Allerdings gibt es auch im Bereich der Policy Control offene Punkte bezüglich der einzusetzenden Verfahren. Dabei ist ohne Implementierung der Policy-Control-Einheiten kein ernsthafter Einsatz der IntServ-Dienste möglich. Die Policy-Entscheidungen können zurzeit anhand der Sender- bzw. Empfänger-IP-Adresse und/oder der jeweiligen Port-Nummer getroffen werden. Was fehlt ist die personenbezogen Policy-Entscheidung.

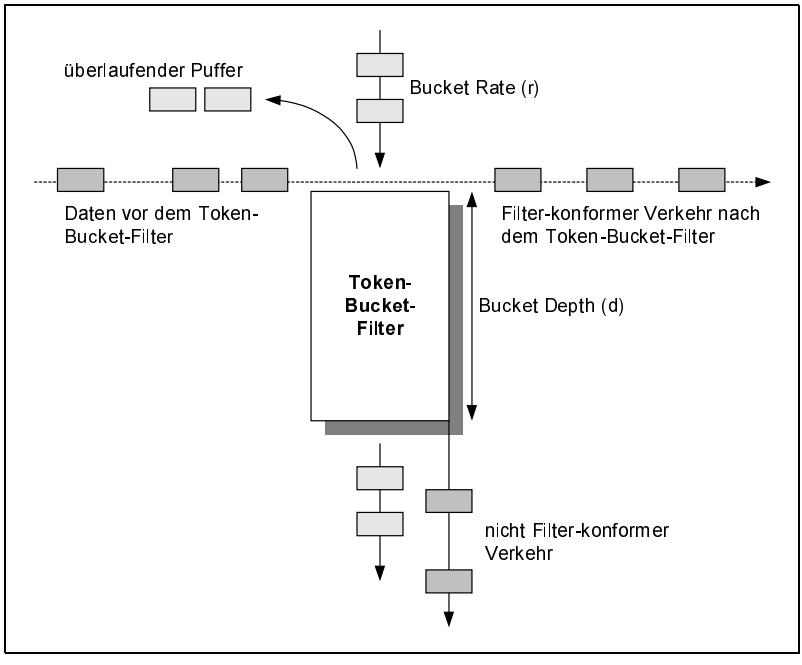


Abb. 3.3 Funktionsweise des Token-Bucket-Filter

3.2.1 Integrated Services (IntServ)

Diese ersten Ansätze der IETF, um QoS auch in IP-Umgebung umsetzen und nutzen zu können, wurde durch die Arbeitsgruppe Integrated Services (IntServ) ins Leben gerufen. Ziel war es, besonders Echtzeitapplikationen effizienter und mit der maximal möglichen Performance zu unterstützen. Der reine Best-effort sollte durch ein komplexeres Modell abgelöst werden, um die Anforderungen neuer Applikationen erfüllen zu können. Um dies zu ermöglichen, sollten Prioritätsklassen eingeführt werden, die man den unterschiedlichen Anforderungen zuordnen kann. Folgende Annahmen hat man dabei für das IntServ-Modell getroffen:

- Ressourcen müssen explizit verwaltet werden, um die Anforderungen der Anwendungen erfüllen zu können.
- Die Dienstgarantien für Echtzeitapplikationen können nicht ohne Reservierung von Ressourcen erfolgen.
- Die End-to-end-Verzögerungszeiten müssen begrenzt werden, um die dynamische Anpassung an sich ändernde Netzbedingungen gewährleisten zu können.
- Statistisches Aufteilen zwischen Echtzeit- und Datenapplikationen ist vorteilhaft, wenn man über eine gemeinsame Infrastruktur beide Anwendungen nutzen will

Dabei kann die Übertragung über Unicast-Pakete[10] oder Multicast-Pakete[11] stattfinden. Um dies umzusetzen, wird ein virtueller Kanal zwischen den Kommunikationsteilnehmern geschaltet. Ausgehend von den Anforderungen des Datenverkehrs wird ein QoS-Profil mit fest vorgegebenen Verkehrsparametern vereinbart. Dementsprechend werden dann entlang der Route Ressourcen in jeder Übertragungseinrichtung reserviert. Aus den IntServ-Spezifikationen nach RFC-1633 entstanden die folgenden Dienstklassen, die 1997 zum Proposed Standard erhoben worden:

1. **Controlled Load Network Element Service (CL-Service):** hat zum Ziel, dem Teilnehmer ein unbelastetes Netz in Zeiten der Überlast vorzutäuschen. Daher wird dieser Dienst auch als geringfügig besser eingestuft als bei Best-effort. Die Spezifikation RFC-2211 beschreibt diesen Service. Einsatzgebiete sind: Audio-/Videostreaming und web-basierte Transaktionen. [WROC97a]
2. **Guaranteed Quality of Service (Guaranteed Service, GS):** Dieser Dienst legt die Einhaltung fester QoS-Parameter im Netz fest. Diese Parameter werden vor dem Beginn der Übertragung oder parallel dazu ausgehandelt. Die Spezifikation RFC-2212 legt diese Parameter fest. Einsatzgebiete sind interak-

10 Unicast = Punkt-zu-Punkt-Kommunikation zwischen zwei Teilnehmern

11 Multicast = Punkt-zu-Mehrpunkt-Kommunikation zwischen einer bestimmten Gruppe

tive Dienste, die eine harte Anforderung an QoS haben, wie IP-Telefonie, VoIP und Videokonferenzen. [SPG97]

Folgende Begriffe werden im Zusammenhang mit IntServ zusätzlich verwendet, die am Anfang definiert werden sollten:

- **Policing:** ist ein Prozess für die Überwachung des Datenstroms auf die Einhaltung zum vereinbarten Verkehrsprofil. Der Prozess selbst hat nichts mit der Prüfung der Policies zu tun!
- **Traffic Shaping:** Der Datenverkehr im Internet besitzt keine gleichmäßige Charakteristik, sondern hat größere oder geringere Paketaufkommen. Bei auftretendem Burst werden Warteschlangen gefüllt, um das Datenaufkommen zwischenzuspeichern[12]. Eine Vergrößerung der Warteschlange verursacht größere Wartezeiten. Um diese Warteschlangen abzubauen, muss der entsprechende Datenfluss eine Zeitlang schneller verarbeitet werden. So kommt es leicht dazu, dass der Verkehr, der beim Sender noch den Vorgabeparametern entsprach, es im Netz nicht mehr tut (z.B. wegen zu hoher Peak-Rate). Um das zu vermeiden, können zur Realisierung bestimmter Dienste Traffic Shaper eingesetzt werden. Somit werden die einkommenden Pakete verzögert, bis der ausgehende Verkehr den vorgegebenen Parametern entspricht. Fehlt für ein Packet das Token (z.B. weil die Burst Size überschritten wird), so wird das angenommene Paket aussortiert und je nach Spezifikation gesondert behandelt. Die Realisierung besteht dabei aus drei Komponenten:
  - Token Bucket Filter
  - Buffer, der die Bursts aufnimmt und kompensiert.
  - Datenratenregler, der sicherstellt, dass die gewünschte Datenrate nicht überschritten wird.
- **Reshaping:** Anwendung des Traffic-Shaping-Verfahrens außerhalb der Datenquelle. Der Begriff Reshaping deutet implizit darauf hin, dass der Verkehr ursprünglich der *TSPEC* entsprach, inzwischen allerdings durch mehrfache Verzögerung in unterschiedlichen Warteschlangen verformt wurde. Wird in den Warteschlangen statistisches Multiplexen angewendet, so wächst beim Passieren der Warteschlange das Paketaufkommen innerhalb eines Datenstroms. [DETK00a]

3.2.2 Controlled Load Network Element Service

Die dedizierte Zuweisung von Bandbreite ermöglicht der Controlled Load Network Element (CLNE) Service nach RFC-2211. Dadurch wird die Kontrolle der von den Applikationen angeforderten Übertragungsraten ermöglicht. Alle anderen QoS-Parameter bleiben jedoch unberücksichtigt. Das heißt, Netzpara-

12 Store-and-Forward-Prinzip

meter können nicht mit einbezogen werden. Die Implementierung ist auch oberhalb der Sicherungsschicht möglich, da die maximale Transferrate eines Übertragungsmediums eine unveränderliche Größe darstellt. Wenn man also den CLNE-Service in das Internet einführen würde, was beispielsweise über das Protokoll RSVP[13] möglich ist, könnten die unteren Schichten ohne Änderungen übernommen werden. Man müsste somit keine Hardware austauschen, da allein der Durchsatz berücksichtigt wird.

Der CLNE-Service ist aufgrund seiner Zielsetzung einer einfachen Integration in das bestehende Netz auch nur auf die einfachsten Funktionen beschränkt geblieben. Es werden bei der Anforderung des CLNE-Dienstes keine festen Werte für die einzelnen QoS-Parameter wie Paketverzögerung oder Paketfehlerrate gesetzt. Dabei wird vom Netz folgendes Verhalten erwartet:

- unterdurchschnittliche Warteschlangenverzögerung für Zeitspannen *T*, die wesentlich größer als die Burst-Time *tb* ist ($T >> t_b$)
- wenige oder keine Paketverluste bezüglich Überlastungen im Netz, betrachtet über eine Zeitspanne ($T >> t_b$)

Der CLNE-Service wird durch die Traffic Specification (*TSPEC*) beschrieben. Die Beschreibung des Token Bucket enthält eine Bucket-Rate (*r*) und eine Bucket Depth (*b*). Über alle Zeitperioden *T* darf die Länge eines Daten-Bursts nicht mehr als *rT+b* betragen. Wenn IP-Pakete dies nicht berücksichtigen oder die Maximum Transmission Unit (MTU) verletzt wird, können sie nicht für diesen Service eingesetzt werden. Weiterhin ist es möglich, dass zu große Daten-Bursts kontinuierlich empfangen werden, sodass immer eine Zwischenspeicherung erfolgen muss. Der Puffer wird somit nie gelöscht und die Latenzzeiten steigen kontinuierlich an. Um diesen Verkehr trotzdem zu handhaben, bietet der Controlled Load Service verschiedene Möglichkeiten an:

- Das Leihen von der angeforderten Bandbreite durch einen Traffic Scheduler, um die Daten-Bursts des Netzwerks bearbeiten zu können.
- Einbeziehen von statistischem Multiplexing unter Berücksichtigung einer gemessenen Admission Control

Um den CLNE-Dienst bereitzustellen, muss jedes Netzelement bei einer Dienstanforderung bestimmte Fähigkeiten besitzen. Erstens muss die Einheit Admission Control Kenntnisse über die Belastung und noch verfügbare Bandbreite im Netz besitzen. An dieser Stelle können auch prädiktive Schätzungsverfahren zur Vorhersage der Verbindungsnutzung eingesetzt werden. Die Wahl eines passenden Verfahrens wird dem Hersteller überlassen. Weiterhin müssen Bandbreitenreserven vorhanden sein oder ein Mechanismus zum Teilen der Bandbreite zwischen unterschiedlichen Datenflüssen, welches die Anwendung des statischen Multiplexens beinhaltet. Dies ist wichtig, um die Queues auf den

13 Resource Reservation Protocol

Schnittstellen nach Auftreten von Datenbursts auch wieder abbauen zu können. Hinzu kommt die Tatsache, dass der Verkehr eine Burstiness[14] besitzt, das heißt, unterschiedliche Datenaufkommen beinhaltet. Dies macht ein Puffermanagement erforderlich, welches in den Netzwerkkomponenten enthalten sein muss, wodurch man in der Lage ist, Puffer für unterschiedliche Datenströme gemeinsam nutzen zu können. Der CLNE-Dienst ist nicht in der Lage Traffic Shaping durchzuführen, weshalb das Datenaufkommen entlang der Übertragungsstrecke an den Knoten immer weiter ansteigen kann. Dabei müssen Paketverluste in Kauf genommen werden, wenn keine gemeinsamen Pufferbereiche verwendet werden.

Ein Netzelement ist somit mittels des CLNE-Dienstes in der Lage, erste Maßnahmen gegen Netzlastspitzen zu ergreifen. Beispielsweise kann die Datenrate für eine bestimmte Anforderung begrenzt werden, sodass Pakete, die von der vereinbarten Verkehrssituation abweichen, je nach Netzauslastung als Best-effort-Verkehr weitergeleitet oder verworfen werden. Somit kann die Situation im Netz unter Kontrolle gehalten werden. Zusätzlich muss die Fähigkeit der Ressourcenüberprüfung implementiert sein. Änderungen der *TSPEC* erfordern diese erneute Überprüfung durch die Admission Control, die wie bei einer neuen Ressourcenanforderung vorgenommen werden muss, nach der dann eine neue Entscheidung gefällt wird. Wenn die *TSPEC* verringert wird, so muss weiterhin sichergestellt werden können, dass die Anforderung nicht abgewiesen wird, und eine Prüfung kann entfallen, während bei einer Erhöhung diese Überprüfung auf jeden Fall vorgenommen werden muss.

Anforderung eines CLNE-Dienstes

Die Anforderung eines CLNE-Dienstes wird vorgenommen, indem eine T_{SPEC} an das entsprechende Netzelement gesendet wird. Die einzelnen Parameter der T_{SPEC} sind in Tab. 3.2 aufgeführt.

Tab. 3.2 T_{SPEC}-Datenstruktur

Parameter	Bezeichnung	Typ	Einheit
R	Rate des Token-Bucket-Filters	Floating Point	[byte/s]
B	Bucket Depth (maximale Bucket Size) des Token-Bucket-Filters	Floating Point	[byte]
P	Peak Rate	Floating Point	[byte/s]
m	Minimum Policed Unit	Integer	[byte]
M	Maximum Paket Size	Integer	[byte]

14 Unterschiedliche Datenaufkommen, die stochastisch – also nicht voraussagbar – auftreten: das Verhältnis der maximalen zur mittleren Datenrate.

Der CLNE-Dienst nimmt keine Reservierung mit festen Parametern vor, da die Peak-Rate nur zur Kompatibilität zu anderen IntServ-Diensten eingeführt worden ist. Dieser Parameter kann vom Packet Scheduler zur Berechnung der Größe des zu allozierenden Puffers verwendet oder durch den Knoten innerhalb des CLNE-Dienstes auch ignoriert werden. Falls die Peak-Rate ignoriert wird, d.h. von dem Managementprozess auf unendlich gesetzt wird, so wird stattdessen die maximale Datenrate auf der eingehenden Schnittstelle in die Berechnung der Traffic Control einbezogen. Optional kann ein CLNE-Block der *ADSPEC*[15] der PATH-MESSAGE[16] beigefügt werden (siehe RSVP[17]). In diesem Block können die allgemeinen Parameter der *ADSPEC* überschrieben werden, z.B. falls man die für den CLNE-Dienst die verfügbare Bandbreite beschränken will.

Policing Das Policing behandelt Regeln, anhand deren der *TSPEC*-Verkehr auf Konformität bzw. Nichtkonformität eingestuft wird. Zusätzlich werden hier die Regeln zur Behandlung des überschüssigen Verkehrs festgelegt. Dies beinhaltet, dass die definierten Bedingungen für beliebige Zeitspannen T erfüllt sein müssen. Hierbei sind R und b die Token-Bucket-Parameter der *TSPEC*. Kommt ein Datenpaket an, für das diese Bedingung nicht erfüllt wird, so wird es als nichtkonform behandelt. Alle Pakete, die kleiner als m sind, fließen mit der Größe m in die Berechnung ein, während Pakete, deren Größe die MTU auf der Ausgangsverbindung überschreiten, als nichtkonform gelten.

Weiterhin wird als Verhalten bezüglich der nichtkonformen Daten vom Netzwerk verlangt, dass eine Überschreitung der *TSPEC* nicht als Fehler betrachtet wird. Konforme und nichtkonforme *TSPEC*-Daten werden aber auf jeden Fall unterschiedlich behandelt. Liegt beispielsweise eine Multicast-Übertragung vor und werden in den Heterogeneous Branch Points auf den Verbindungen mit unterschiedlichen Charakteristiken verzweigt, so muss damit gerechnet werden, dass auf einer Verbindung mit geringer Bandbreite die angemeldete *TSPEC* permanent überschritten wird! Dies ist deshalb der Fall, weil die *TSPEC* vom Sender vorgegeben wird und sie nach der Weiterleitung durch die Router nicht mehr angepasst werden kann. Zusätzlich darf ein Netzelement nicht versuchen, Daten, die die aktuelle *TSPEC* überschreiten, als CLNE-Service-Daten weiterzuleiten. Die Menge der nichtkonformen Daten darf das Weiterreichen der konformen Daten nicht negativ beeinflussen!

Das Netzelement kann aber versuchen, die nichtkonformen *TSPEC*-Pakete als Best-effort-Daten weiterzuleiten. Die CLNE-Service-Datenströme dürfen dabei aber den Best-effort-Verkehr nicht vollständig verdrängen. Diese Bedingung hat erstens die Konsequenz, dass bei der Dimensionierung des Interface

15 Advertising Specification
16 Mit der PATH-Nachricht meldet ein Sender die Datenübertragung an.
17 Resource Reservation Protocol

eine maximale Grenze der Bandbreite für QoS-Verkehr festgelegt werden soll, zweitens kann unter Umständen verlangt werden, dass Pakete, die die aktuelle *TSPEC* überschreiten, nicht weitergeleitet, sondern verworfen werden. Die Spezifikation des CLNE-Dienstes überlässt den Herstellern die Entscheidung, wie die nichtkonformen Daten behandelt werden. Hierbei sind drei Szenarien möglich:

1. Nur die überschüssigen Daten werden als nichtkonforme T_{SPEC}-Daten behandelt; die restlichen Daten werden weiterhin als CLNE-Dienst weiterverarbeitet. In diesem Fall werden diese überschüssigen Pakete als Best-effort-Daten behandelt, da sie mit großer Wahrscheinlichkeit eine höhere Laufzeit haben werden als die CLNE-Datenpakete. Die Pakete kommen somit beim Empfänger in falscher Reihenfolge an. Das kann beispielsweise bei einer TCP-Übertragung zur Paketwiederholung führen, was wiederum die Anzahl der Datenpakete erhöht.
2. Der gesamte Datenfluss wird zum Best-effort-Datenstrom degradiert. Der Nachteil ist hier, dass weiterhin alle Daten dieses Datenflusses als Best-effort-Verkehr deklariert werden, wenn die *TSPEC* einmal überschritten ist.
3. Die nichtkonformen Datenpakete werden verworfen.

Dem Anwender bleibt bei der Nutzung des CLNE-Dienstes allerdings verborgen, welches Verfahren angewendet wird, um die Daten nach den IntServ-Definitionen zu behandeln.

Regeln der T_{SPEC}

Die *TSPEC* ist eine mehrdimensionale Größe, weshalb bestimmte Regeln zur Addition, zum Größenvergleich oder zur Zusammenführung definiert werden mussten. Ein Größenvergleich wird beispielsweise durchgeführt, wenn definierte *TSPEC*-Parameter überschritten werden. Bei der Zusammenfassung bzw. Verteilung unterschiedlicher Reservierungen müssen zusätzliche Regeln definiert werden. Jede IntServ-Implementierung muss also eigene Regeln zum Vergleich bzw. für die Manipulationen von Datenstrukturen einführen. Die folgenden Regeln stellen die Basis für den CLNE-Dienst dar. Gegeben sind dabei zwei *TSPEC*-Datenstrukturen *A* und *B*, wobei *A* größer oder gleich *B*, wenn alle fünf Bedingungen erfüllt sind:

$$r(A) \geq r(B)$$
$$b(A) \geq b(B)$$
$$P(A) \geq P(B)$$
$$m(A) \leq m(B)$$
$$M(A) \geq M(B)$$

Dabei ist zu beachten, dass hieraus die vierte Regel nicht als eine Negierung des obigen Ergebnisses gebildet werden kann. Vielmehr gilt, falls bei zwei ungleichen *TSPEC*-Parametern nur ein Teil der fünf oben genannten Bedingungen erfüllt ist, ist die Ordnung zueinander undefiniert.

Hieraus folgt die Regel zur Minimumbildung zweier *TSPEC*-Parameter. Ist die obige Regel auf die *TSPEC*-Parameter A und B anwendbar und ist $A > B$, so ist B das Minimum von den beiden. Ist die Ordnung der beiden zueinander undefiniert, so wird das Minimum komponentenweise gebildet[18]:

$$r(C) = \min\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \min\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \min\{M(A), M(B)\}$$

Beim Zusammenfassen (Mergen) unterschiedlicher Reservierungsanforderungen auf einer Teilstrecke wird aus den einzelnen *TSPEC*-Strukturen eine neue gemeinsame *TSPEC* nach folgenden Regeln gebildet[19]:

$$r(C) = \max\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \max\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \min\{M(A), M(B)\}$$

In einigen Fällen möchte man eine Menge Datenflüsse (Flows) mit der geringsten gemeinsamen *TSPEC* beschreiben. Dafür ist die Regel zur Bildung der geringsten gemeinsamen *TSPEC* wie folgt definiert[20] worden:

$$r(C) = \max\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \max\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \max\{M(A), M(B)\}$$

18 C ist hier die resultierende *TSPEC*

19 Auch hier sind A und B die zu manipulierenden *TSPEC*-Parameter und C der daraus resultierende Parameter.

20 In derselben Notation wie vorher

Die Regel zur Summierung von *n TSPEC*-Parametern lautet hingegen[21]:

$$r(C) = \sum_{i=1}^{n} r(A_i)$$

$$b(C) = \sum_{i=1}^{n} b(A_i)$$

$$P(C) = \sum_{i=1}^{n} P(A_i)$$

$$m(C) = \min\{m(A_1), \ldots, m(A_n)\}$$

$$M(C) = \max\{M(A_1), \ldots, M(A_n)\}$$

Implementierung

Bei der Implementierung eines CLNE-Dienstes muss man unterschiedliche Eigenschaften beachten. Zum einen ist der Service nicht in der Lage, feste QoS-Parameter zu garantieren. Dadurch kann es sinnvoll sein, dass die Summe der *r*-Parameter für einzelne Anforderungen die verfügbare Bandbreite auf der Schnittstelle überschreitet. Dieser Fall tritt auf, wenn mehrere Datenströme im Mittel eine geringere Datenrate als die angemeldete Rate *r* aufweisen. Solche Überbelegung sollte man aber nur in Verbindung mit dem Einsatz prädiktiver Schätzungsverfahren der Auslastung auf dem Interface anwenden.

Weiterhin ist zu beachten, dass das Verwerfen der Pakete durch Policing bei einigen Protokollen zum Wiederholen der Pakete[22] führen kann, was wiederum zur Erhöhung der Datenrate des Verkehrsstroms führen kann. Bei der Evaluierung eines CLNE-Dienstes sollte man deshalb das Netzverhalten in folgenden Situationen betrachten:

- Vergleich des Verhaltens beim Best-effort-Verkehr mit dem CLNE-Service bei geringer Last.
- Vergleich auf einer stark belasteten Strecke. Dabei ist auf die Änderung der Gesamtverzögerung des CLNE-Verkehrs sowie die Änderung der Paketverlustrate zu achten.
- Erzeugen eines Datenstroms, der die angemeldete *TSPEC* nicht einhält: Überschreitung der mittleren Datenrate sowie der Burstiness.
- Erzeugen einer Überlast mittels reservierter Datenströme ohne Best-effort-Verkehr.

Im ersten Fall dürften sich keinerlei Verluste messen lassen. Dies müsste sich im zweiten Fall ändern, da hier die Verlustrate im Falle einer Reservierung auf der stark belasteten Strecke steigt, allerdings noch vernachlässigbar. Im dritten Fall sollte es zu Verzögerungen und möglichen Paketverlusten kommen. Hiermit wird die Funktionalität der Einheiten Admission Control geprüft. Ändert sich das Übertragungsverhalten bei einer Überschreitung der *TSPEC* nicht, so wird die Effizienz vom Einsatz des CLNE-Service gering oder gar verschwindend

21 Mit den *TSPEC*-Parametern *A1... An* und *C* als Ergebnis

22 Z.B. bei dem Einsatz von TCP-Fenstermechanismus auf der Anwendungsebene

bleiben. Bei der letzten Möglichkeit sollten bei korrekt funktionierenden Admission-Control-Einheiten zusätzliche Reservierungsanforderungen abgewiesen werden, bevor die Paketverlustrate und die Verzögerung signifikant steigen. Diese Punkte müssen auf jeden Fall vor einer abschließenden Implementierung untersucht werden. [WROC97a]

3.2.3 Guaranteed Quality-of-Service (GQOS)

Der Guaranteed Service nach RFC-2212 kontrolliert im Gegensatz zum CLNE-Dienst nicht die Minimum- oder Durchschnittsverzögerung der IP-Pakete, sondern die maximale Pufferverzögerung. Dieser Dienst garantiert somit, dass Pakete das Ziel mit der angeforderten Verzögerung erreichen. Jitter werden dabei nicht berücksichtigt. Dieser Dienst wird ebenfalls durch die Parameter der Traffic Specification (*TSPEC*) und Request Specification (*RSPEC*) beschrieben. *TSPEC* wird durch die Anwendungen beschrieben und enthält den Token Bucket (*r, b*), die Peak-Rate *P*, eine Minimum Policed Unit (MPU) und eine Maximum Datagram Size (MDS). Die Spitzenrate *P* ist die maximale Datenrate, die der Sender produzieren kann. MPU beinhaltet dementsprechend, dass jedes Paket mit einer Größe kleiner als MPU zur Minimum Policed Unit hinzugezählt wird. *RSPEC* wird durch den Datenfluss, der durch eine bestimmte Dienstgüte beschrieben wird, definiert. Er beinhaltet die Parameter Reservation-Rate *R* und Slack Term *S* in Mikrosekunden. *S* bezeichnet den Unterschied zwischen dem gewünschten und dem realen Delay, den man durch eine Reservation-Rate *R* erhält. Durch die Benutzung von *S* kann man eine spezifische Verzögerungsgarantie bekommen.

Zur Kontrolle des Guaranteed Services werden zwei Policy-Arten vorgeschlagen, die mit Simple Policing und Reshaping bezeichnet werden. *TSPEC* ist die Simple Policing, während Reshaping sich auf die Rekonstruktion einer Verkehrsform bezieht, um *TSPEC* anzupassen. Ein Datenfluss verletzt *TSPEC*, wenn das Reshaping fehlschlägt oder der Puffer überläuft. Policing wird immer am Rande eines Netzes angewendet, während das Reshaping an allen heterogenen Zweigpunkten und Brennpunkten eingesetzt wird. Bislang ist der Guaranteed Service allerdings nicht für das Internet umgesetzt worden.

Ein GQOS soll zusammenfassend einen sehr hochwertigen Dienst bereitstellen. Dabei wird das folgende Verhalten von diesem Dienst erwartet:

- Es wird eine vereinbarte Bandbreite zugesichert
- Überschreitet der Datenfluss die in der *TSPEC* vereinbarten Grenzwerte nicht, so verhält sich das Netz folgendermaßen:
  - Es treten keine Warteschlangenverluste in den einzelnen Netzelementen auf. Verluste infolge von transienten Fehlern in Netzelementen (Änderung der Route, Ausfall eines Routers etc.) werden im Rahmen von IntServ nicht berücksichtigt. Es wird auch nicht versucht, Verluste

infolge solcher Fehler zu beheben, da man hofft, dass der Anteil solcher Verluste nicht ins Gewicht fällt.
- Die verfügbare Bandbreite und die maximale Verzögerung sind stabil. Das heißt, sie ändern sich nicht, solange der Datenpfad erhalten bleibt.
- Die Abweichung des Datenflusses vom GQOS-Flussmodell überschreitet die ausgehandelten Grenzwerte nicht.

Der GQOS berechnet oder berücksichtigt bei der Vergabe der Dienstanforderungen keine Jitter. Es wird lediglich die maximale Verzögerung berechnet, ohne die mittlere oder minimale Verzögerung einzubeziehen. Da aber die maximale Verzögerung immer für die ungünstigsten Verhältnisse mit Reserve berechnet wird, ist damit zu rechnen, dass die meisten Pakete schneller am Ziel ankommen. Es ist wichtig, diesen Faktor bei der Programmierung zeitkritischer Anwendungen, die auf dem Guaranteed Service aufsetzen, zu berücksichtigen. Die früher ankommenden Daten müssen entsprechend verarbeitet bzw. zwischengespeichert werden. Genau wie beim CLNE-Dienst wird der Datenfluss mit Hilfe des Token-Bucket-Filters beschrieben. Jedes Netzelement entlang des Datenpfades innerhalb des Services berechnet unterschiedliche Parameter auf dem entsprechenden Teilabschnitt. Aus diesen Parametern für die Teilabschnitte werden die Ende-zu-Ende-QoS-Parameter bestimmt. Dies ist in erster Linie die maximale Ende-zu-Ende-Verzögerung. Die Bestimmung und Steuerung der Gesamtverzögerung beinhaltet ein großer Teil der GQOS-Spezifikation nach RFC-2212. Dabei besitzt der GQOS-Dienst drei wichtige Eigenschaften, die man berücksichtigen sollte:

1. Obwohl die genauen Anforderungen an das begleitende Management-Protokoll in der Spezifikation beschrieben sind, ist der Dienst unabhängig vom Management-Protokoll. Eine GQOS-Reservierung kann sowohl mit Hilfe des RSVP, als auch mit Hilfe eines anderen Protokolls (z.B. COPS[23]) oder manuell in den Knoten entlang des Datenpfades eingetragen werden.
2. Um die Verzögerung der Daten zwischen dem Sender und Empfänger zu minimieren, müssen alle Knoten entlang des Pfades GQOS unterstützen! Das bedeutet im Umkehrschluss allerdings keineswegs, dass eine GQOS-Reservierung ohne GQOS-fähige Teilstrecken keinerlei Nutzen bringt. Zusätzlich kann man den Dienst innerhalb von Teilnetzen (z.B. innerhalb eines Campusnetzes) oder ausschließlich im Kernnetz zwischen unterschiedlichen Routern sinnvoll einsetzen.
3. Die innerhalb des Dienstes berechnete Ende-zu-Ende-Verzögerung spiegelt zwar eine Fähigkeit des betrachteten Teilnetzes wider, worauf die Endanwendung keinen direkten Einfluss hat, doch kann (und sollte) die Anwendung durch Wahl geeigneter Reservierungsparameter die Gesamtverzögerung indirekt beeinflussen.

23 Common Open Policy Server

Steuerung der Gesamtverzögerung

Bei der Betrachtung bzw. Beschreibung der Dienstgüte des GQOS-Dienstes wird das so genannte Flussmodell (Fluid Modell) herangezogen. Dabei entspricht das Flussmodell einem Service mit der Rate *r* und einer dedizierten Leitung mit der Bandbreite *R* zwischen Quelle und Senke. Der Datenstrom auf der dedizierten Leitung ist vollständig von allen anderen Datenströmen entkoppelt! Die Datenflusseigenschaften werden anschließend mit dem Flussmodell verglichen. Die Flusseigenschaften in jedem Netzelement werden durch die Bandbreite[24] *R* und Puffergröße[25] *TSPEC* beschrieben. Das Netzelement muss sicherstellen, dass sein bereitgestellter Dienst dem Flussmodel entspricht und kein Pufferüberlauf stattfindet. Wird der gesendete Datenstrom mit Hilfe des Token-Bucket-Filters mit Bucket-Filterrate *r* und Bucket Depth *b* beschrieben, so hat die maximale Verarbeitungsverzögerung im Flussmodell den folgenden maximal Wert:

$$\frac{b}{R} \text{ unter der Bedingung R < r}$$

Die Fortpflanzungsverzögerung wird hierbei nicht berücksichtigt. Da die Warteschlangen- und Verarbeitungsverzögerung um gewisse Größenordnungen höher als die Fortpflanzungsverzögerung ist, muss man diese auch nicht einbeziehen. Für genaue Berechnungen kann die Fortpflanzungsverzögerung aus dem dienstunabhängigen Teil der *ADSPEC* -Struktur in die Gesamtverzögerung einbezogen werden.

Somit muss jedes Netzelement gewährleisten, dass die Warteschlangenverzögerung den folgenden Grenzwert nicht überschreitet:

$$\frac{b}{R} + \frac{C}{R} + D$$

Dabei beschreiben *C* und *D* die maximale Abweichung vom Flussmodell. Die grundlegenden Anforderungen an die einzelnen Netzelemente sind ähnlich den Anforderungen für den CLNE-Dienst. Zusätzliche Anforderungen sind vorhanden, die noch erläutert werden.

Dienstanforderungen

Beim GQOS-Dienst wird bei einer Dienstanforderung außer der schon aus dem CLNE-Dienst bekannten *TSPEC*-Parameter, die den aktuellen Datenfluss beschreibt, auch die gewünschte Dienstgüte in Form einer *RSPEC* angegeben. Entsprechend wird zusätzlich zu jeder Änderung der *TSPEC* auch jede *RSPEC*-Änderung von der Einheit Admission Control als eine neue Dienstanforderung betrachtet. Das heißt, es wird vom Netz gefordert, dass bei Minderung[26] sowohl

24 Maximale Bandbreite, die auf der physikalischen Verbindung genutzt werden darf.
25 Puffergröße, die der Datenfluss maximal belegen darf.
26 Siehe *RSPEC* -Regeln.

der *TSPEC* als auch der *RSPEC*, bei jeweils gleichbleibender zweiten Komponente, die Anforderung nicht abgelehnt werden darf. Die Struktur der T_{SPEC} entspricht wie mehrfach erwähnt der im CLNE-Dienst. Es ändert sich lediglich die Semantik der Peak-Rate. Es muss für die Menge der ankommenden Daten *n* für beliebige Zeitspannen *T* die folgende Bedingung eingehalten werden:

$$n \leq M + pT$$

Wird der ankommende Datenstrom diese Bedingung verletzen, so wird er als nicht konform bezüglich der T_{SPEC}-Regeln behandelt. Ist die Peak-Rate unbekannt oder undefiniert, so wird sie als unendlich angenommen.

Tab. 3.3 R_{SPEC}-Datenstruktur

Parameter	Bezeichnung	Typ	Einheit
R	Reservation Level (Reservierungsgrad)	Floating Point	[byte/s]
S	Slack Term (Schlupfterm)	Integer	[µs]

Tabelle 3.2 zeigt die Struktur von *RSPEC*. Dabei muss *R* mindestens so groß wie *r* sein. Das Größenverhältnis $R > r$ ist zugelassen und sinnvoll, denn eine Bandbreitenreserve bewirkt kürzere Warteschlangen in den Knoten. *S* gibt die zulässige Abweichung der Verzögerung an, wobei größere Werte eine geringere Dienstgüte bedeuten. Es muss dabei gelten:

$$S \geq 0$$

Falls *S* nicht explizit angegeben wird, wird der Wert auf Null gesetzt. Die Puffergröße *B* ist in der *RSPEC* nicht enthalten. *B* hat lokale Gültigkeit und muss in jedem Knoten getrennt berechnet werden, damit die Bedingung der Verlustfreiheit in den Warteschlangen gewährleistet wird. Zur Berechnung der notwendigen Puffergröße kann der jeweilige Knoten auf folgende Strukturen und deren Komponenten zugreifen:

- **TSPEC:** r, b, p
- **RSPEC:** R, S
- **ADSPEC:** C_{tot}, D_{tot}, C_{sum}, D_{sum}

Verzögerungsparameter

Die Verzögerungsparameter *C* und *D* stehen stellvertretend für zwei Arten von Verzögerungen, die in den Netzelementen auftreten:

- Der Parameter *C* fasst alle datenratenabhängigen Verzögerungen im Netzwerkelement zusammen. Dabei fließt die Datenrate umgekehrt proportional in die Verzögerungsberechnung mit ein. Als Beispiel für eine Verzögerung dieser Art kann man das Mappen eines Datagramms in ATM-Zellen

betrachten: ist die Übertragungsrate der ATM-Strecke konstant, so ist die Mapping-Verzögerung proportional zu *l/r*. Da beim Flussmodell ein Durchsatz *R* gemessen wird, ist der Term *C/R* in die Berechnung einzubeziehen. *C* hat dabei die Einheit [Byte].

- Der Parameter *D* fasst alle datenratenunabhängigen Verzögerungen im Netzelement zusammen. Ein Beispiel dafür ist die Verzögerung, die bei der Verwendung von TDMA[27]-Verfahren entsteht. Nutzt man ein TDMA-Verfahren mit *M*-Slots mit der Zeitdauer *tslot* für einen Zeitslot, so muss auf dem Interface im ungünstigsten Fall eine Zeit (M-1) *tslot* gewartet werden. *D* hat dabei die Einheit [µs].

Abb. 3.4 Verzögerung bei TDMA-Verfahren

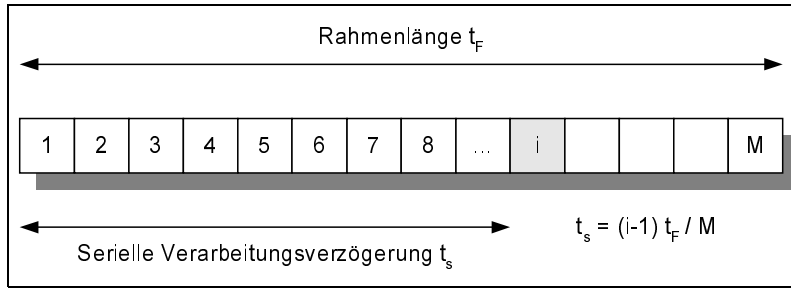
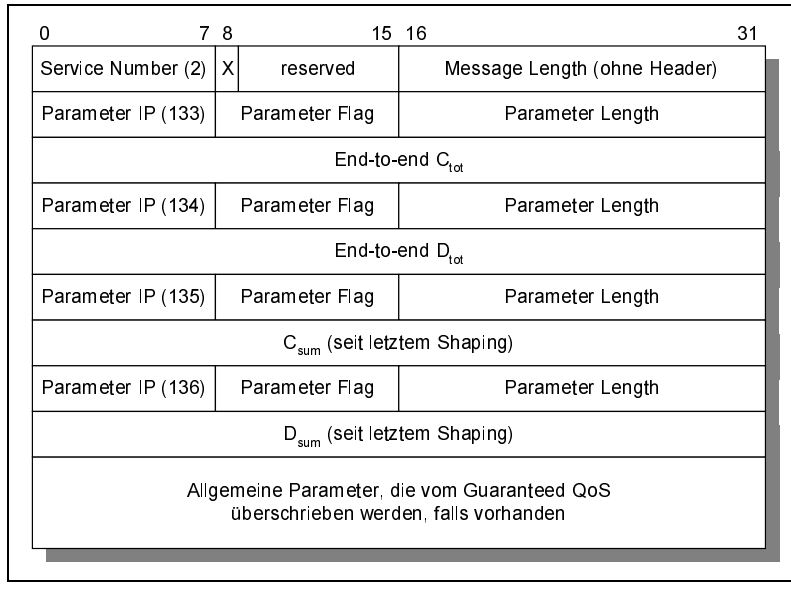


Abb. 3.5 GQOS-Dienstfragment der AD_{SPEC}



27 Time Division Multiple Access

Jedes Netzelement muss seine Größen für *C* und *D* kennen. Es ist dabei freigestellt, ob diese Werte von der Einheit Traffic Control berechnet, gemessen oder vom Netzadministrator manuell eingetragen werden. Außerdem muss jedes Netzelement Methoden besitzen, um die Größen *Ctot*, *Dtot*, *Csum* und *Dsum* zu empfangen, eigene *C*- bzw. *D*-Werte zu addieren und die modifizierten Werte in Richtung des Empfängers weiterzureichen. Dabei stehen die Werte *Ctot* und *Dtot* für die Summe der Einzelwerte *C* bzw. *D* vom Sender bis zum aktuellen Punkt. *Csum* und *Dsum* repräsentieren dagegen die Summe der entsprechenden einzelnen Werte vom vorherigen Reshaping-Punkt bis zum aktuellen Punkt. Wird zur Realisierung des GQOS-Dienstes das RSVP-Protokoll benutzt, so werden die zusammengesetzten Terme im GQOS-spezifischen *ADSPEC*-Block übertragen (siehe Abb. 3.5).

Policing

Bei dem GQOS-Dienst unterscheidet man zwei unterschiedliche Policing-Verfahren. Das erste Verfahren entspricht dem Policing-Verfahren im CLNE-Dienst und ist einfacher gehalten. Hier wird nur der Datenverkehr durch die *TSPEC*-Parameter kontrolliert und entsprechend eingeordnet. Das zweiten Policing-Verfahren versucht durch Verkehrsformung (Traffic Shaping) die ursprüngliche Verkehrscharakteristik wiederherzustellen. Auch hier wird die Verkehrsverletzung der *TSPEC*, beispielsweise durch Pufferüberlauf des Traffic Shaper, erkannt. Einfaches Policing muss immer beim Eintreten und wenn möglich auch beim Verlassen in ein GQOS-fähiges Netz stattfinden. An folgenden Stellen muss das Reshaping erfolgen:

- In den Heterogeneous Source Branch Points (HSBP), wenn die ausgehende T_{SPEC} geringer ist als die *TSPEC* unmittelbar davor.
- In den Source Merge Points (SMP)
- Weitere Punkte können definiert werden, in denen Traffic Shaping durchgeführt werden soll.

Sind die Parameter *C* und *D* zwischen zwei Punkten, in denen Reshaping vorgenommen wird, bekannt[28], so kann auch die maximale Puffergröße *Bmax* in den Traffic Shaper für einen Datenfluss angegeben werden:

$$B_{\max} = b + C_{sum} + \left(D_{sum} \cdot r\right)$$

Dabei sind *Csum* und *Dsum* die entsprechenden kumulierten Verzögerungsterme von dem letzten Reshaping-Punkt, die hier also zwischen zwei Shaping-Punkten sitzen. Zu beachten ist dabei die Verzögerungsänderung der Datenpakete im Traffic Shaper in Abhängigkeit von der Puffergröße, da die Verzögerung nur um die Verarbeitungszeit (also unwesentlich) steigt, solange der ankommende Verkehr eine geringe Burstiness aufweist. Der Datenpuffer bleibt dann

28 Dieses wird vom GQOS-Dienst gefordert.

demnach leer. Kommt ein Datenburst an, füllt sich der Puffer und die Verzögerung der Pakete steigt damit an. Diesen Effekt will man durchaus erreichen, da er dem Sender ermöglicht, sich streng an die Token-Bucket-Rate zu halten. [SPG97]

RSPEC-Regeln Bereits beim CLNE-Dienst mussten Regeln zum Vergleich und zum Operieren mit *TSPEC* eingeführt werden. Bei dem GQOS-Dienst kommt die *RSPEC*-Struktur noch hinzu. Gegeben sind hier zwei *TSPEC*-Datenstrukturen *A* und *B*. *A* ist größer oder gleich *B*[29], wenn alle fünf folgenden Bedingungen erfüllt sind:

$$r(A) \geq r(B)$$
$$b(A) \geq b(B)$$
$$P(A) \geq P(B)$$
$$m(A) \leq m(B)$$
$$M(A) \geq M(B)$$

Gegeben sind weiterhin die Datenstrukturen aus der vorherigen Definition. Dann ist *A* kleiner oder gleich *B*, wenn folgende Bedingungen erfüllt sind:

$$r(A) \leq r(B)$$
$$b(A) \leq b(B)$$
$$P(A) \leq P(B)$$
$$m(A) \geq m(B)$$
$$M(A) \leq M(B)$$

Nach den folgenden Regeln wird das Minimum von zwei *TSPEC* gebildet. Ist *A* kleiner oder gleich *B*, so ist *A* das Minimum von den beiden. Sonst wird *C* als Minimum der beiden *TSPEC* nach folgender Regel komponentenweise gebildet:

$$r(C) = \min\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \min\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \min(M(A), M(B))$$

Beim Zusammenschmelzen (Mergen) unterschiedlicher Reservierungsanforderungen auf eine Teilstrecke wird aus den einzelnen *TSPEC*-Strukturen eine neue gemeinsame *TSPEC* nach folgenden Regeln gebildet[30]:

29 Wird oft auch als genauso gut oder besser bezeichnet.

30 Auch hier sind *A* und *B* die zu manipulierende *TSPEC* und *C* die daraus resultierende *TSPEC*.

$$r(C) = \max\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \max\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \min\{M(A), M(B)\}$$

Die *RSPEC* wird beim Mergen folgendermaßen gebildet[31]:

$$R(C) = \max\{R(A), R(B)\}$$
$$S(C) = \min\{S(A), S(B)\}$$

Die geringste gemeinsame *TSPEC* wird verwendet, um mehrere *TSPEC*-Parameter auf einer Strecke durch eine gemeinsame zu beschreiben. Sie wird folgendermaßen gebildet[32]:

$$r(C) = \max\{r(A), r(B)\}$$
$$b(C) = \max\{b(A), b(B)\}$$
$$P(C) = \max\{P(A), P(B)\}$$
$$m(C) = \min\{m(A), m(B)\}$$
$$M(C) = \max\{M(A), M(B)\}$$

Die Regel zur Summierung von *n TSPEC*-Parametern lautet dabei wie folgt[33]:

$$r(C) = \sum_{i=1}^{n} r(A_i)$$
$$b(C) = \sum_{i=1}^{n} b(A_i)$$
$$P(C) = \sum_{i=1}^{n} P(A_i)$$
$$m(C) = \min\{m(A_1), \ldots, m(A_n)\}$$
$$M(C)) \max\{M(A_1), \ldots, M(A_n)\}$$

Die *RSPEC* dient der Anforderung der Dienstgüte auf dem Datenpfad. Empfängt ein GQOS-fähiger Knoten eine R_{SPEC} in der Form *Rin* und *Sin*, so wird sie entweder verarbeitet und eine neu berechnete R_{SPEC} in der Form *Rout* und *Sout* in Richtung des Senders weitergereicht, oder die Reservierungsanforderung wird abgewiesen. Die neu berechnete *RSPEC* muss die folgenden Bedingungen

31 *A* und *B* seien die zusammenlaufenden *RSPEC*, *C* die zusammengesetzte *RSPEC*.
32 In derselben Notation wie vorher.
33 Mit den *TSPEC*-Parametern *A1*... *An* und *C* als Ergebnis.

erfüllen, wobei *Ctoti* der kumulierte C-Term vom Empfänger inklusive des aktuellen Knoten *i* ist:

$$S_{out} + \frac{b}{R_{out}} + \frac{C_{toti}}{R_{out}} \leq S_{\sin} + \frac{b}{R_{in}} + \frac{C_{toti}}{R_{in}}$$

$$r \leq R_{out} \leq R_{in}$$

Zusammenfassend lassen sich die folgenden Schlüsse aus den Berechnungsmöglichkeiten ziehen. Die Reservierungsanforderung *R* kann von Knoten zu Knoten in Richtung des Senders nicht vergrößert werden. Zusätzlich wird der Schlupfterm *S* verbraucht, wenn ein Knoten den Reservierungsgrad *R* verringert. Damit wird die gesamte Dienstgüte nicht verringert, das heißt, die Abweichung vom Flussmodell wird nicht größer. Gibt der Empfänger bei einer Reservierungsanforderung einen kleinen *S*-Wert vor, so steigt die Wahrscheinlichkeit, dass die Anforderung von einer Admission-Control-Einheit im Netz abgewiesen wird. Setzt der Empfänger bei einer Reservierungsanforderung den Schlupfterm *S* auf 0, so kann *R* entlang des Datenpfades nicht verringert werden. Die letzten Bedingungen sind nicht erfüllbar, wenn der Datenfluss in einem Heterogeneous Branch Point auf mehrere Links verteilt wird, wobei die verfügbare Datenrate für den Service auf einem davon kleiner als *R* sein muss. Es ist für solche Punkte keine Anpassung der *TSPEC* und der *RSPEC* vorgesehen. Dieses Problem ist bislang ungelöst geblieben. [SPG97]

Implementierung Bei der Realisierung von IntServ-Diensten bestehen mehrere Freiheitsgrade. So hat man die Wahl bei den Parametern *S*, *R* der *RSPEC* sowie bei der Peak-Rate der *TSPEC*. Durch diese Wahl kann die *FLOWSPEC* implizit die QoS-Parameter beeinflussen. Dies kann negative Auswirkungen auf die QoS-Parameter haben.

Bei der Annahme, dass eine Anwendung eine maximale Verzögerung von *Dreq* zulässt, muss der Schlupfterm *S* gewählt werden. Die Verzögerung *DGQOS* bei Nutzung des Guaranteed Service wird dann wie folgt bestimmt, aus der heraus dann der Schlupfterm ermittelt werden kann:

$$D_{GQOS} = \frac{b}{r} + \frac{C_{tot}}{r} + D_{tot}$$
$$S = D_{req} - D_{GQOS}$$

Ein Knoten kann damit bei Bedarf den Reservierungsgrad *R* heruntersetzen und entsprechend der letzten Bedingung der *RSPEC* den Schlupfterm verringern[34]. Somit wird ein Dienst mit der benötigten Dienstgüte dort zugelassen, wo bei *S=0* die Anforderung abgewiesen wäre. Es ist dabei zu beachten, dass

man diesen Spielraum nur dann hat, wenn $R > r$ gewählt wird. In den bisherigen Abschätzungen der notwendigen Puffergrößen bzw. bei der Berechnung maximaler Ende-zu-Ende-Verzögerung war man bislang immer davon ausgegangen, dass die Bucket Size zu jeder Zeit plötzlich ausgeschöpft werden kann. Man hat also damit die Peak-Rate *P* implizit auf unendlich gesetzt. Aber genau diese Größe lässt den Bedarf an Puffer sowie die Gesamtverzögerung reduzieren. Wird *P* auf einen endlichen Wert gesetzt, so bedeutet dies, dass der Puffer sich nur mit einer maximalen Geschwindigkeit *P-r* füllen lässt.

Wählt man *R* im Bereich $r < R < p$, so ist die maximale Gesamtverzögerung *DGQOSmax* vom Sender bis zum Empfänger sowie die entsprechende Puffergröße *Bmax* wie folgt:

$$D_{GQOS\max} = \left[\frac{(b-M)}{R} \cdot \frac{(P-R)}{(P-r)}\right] + \frac{M + C_{tot}}{R} + D_{tot}$$

$$B_{\max} = M + \left[(b-M) \cdot \frac{(P-R)}{(P-r)}\right] + C_{tot} + (D_{tot} \cdot R)$$

Man sieht daran, dass durch die Vergrößerung von *R* und die Verringerung von *P* der Bedarf an Pufferspeicher sowie die Gesamtverzögerung reduziert werden kann. [SHEN97]

3.2.4 Resource Reservation Protocol (RSVP)

Um Bandbreite zukünftig im Internet reservieren zu können, ist von der gleichlautenden Arbeitsgruppe der IETF das Resource Reservation Protocol (RSVP) vorgeschlagen worden. Dadurch kann man erstmals Echtzeitdienste in einer verbindungslosen Umgebung mit höherer Qualität nutzen. Das Protokoll RSVP untersucht dafür die Router-Eigenschaften auf einem Übertragungspfad. RSVP ist seit September 1997 als Standard Track nach RFC-2205 von der IETF verfügbar. Als eine Erweiterung ist die bislang fehlende Policy Control in der Spezifikation RFC-2750 hinzugekommen. Das RSVP-Protokoll wird aber auch künftig weiterentwickelt, da es auch für andere Ansätze wie MPLS verwendet werden kann.

RSVP kommt wie das Internet Protocol (IP) ebenfalls aus dem militärischen Bereich. Das Information Science Institute (ISI), ein Ableger der US Defense Research Projects Agency (DARPA), hat bereits 1993 begonnen, diese Protokollstrukturen zu entwickeln und sie bei der IETF vorgeschlagen (http://www.isi.edu). Die Hauptmerkmale von RSVP lassen sich dabei folgendermaßen zusammenfassen:

34 Wenn z.B. falls eine Reservierung mit *Rin* wegen belegter Ressourcen nicht möglich ist.

- **Merging:** Der abschnittsweise festgelegte QoS kann in bestimmten Fällen Verschmelzungen von Datenströmen mehrerer Sender ermöglichen.
- **Unidirektionale Ressourcenetablierung:** Ressourcen werden unidirektional angefordert; Up- und Downstream sind bezüglich ihrer Dienstgüte voneinander entkoppelt.
- **Regelmäßige Bestätigung der Ressourcen:** Soft-State muss regelmäßig bestätigt werden, sonst wird die reservierte Bandbreite wieder freigegeben.
- **Empfänger initiiert die Ressourcen:** die Reservierung der Ressourcen wird durch den Empfänger einer Verbindung eingeleitet.
- **Transparenz:** Strecken, die nicht RSVP-fähig sind, können aufgrund der Transparenz des Protokolls ebenfalls einbezogen werden.
- **Ressourcen:** können nur abschnittsweise zur Verfügung gestellt werden (hop-by-hop).
- **Ausnutzung vorhandener Routing-Mechanismen:** RSVP nutzt bereits Unicast oder Multicast-Routing-Mechanismen.

Das Merging stellt eine Hauptfunktionalität von RSVP dar. Datenflüsse, die aufgrund ihrer Asynchronität die sequentielle Nutzung derselben Ressourcen durch mehrere Sender und Empfänger ermöglichen, können hierdurch effizient durch das Netz transportiert werden. Dadurch lassen sich Engpässe vermeiden. Man kann während einer Verbindung zusätzliche Ressourcen anfordern bzw. nicht benötigte freigeben sowie andere QoS-Parameter ändern. Als Zieladresse kann sowohl eine Unicast- als auch Multicast-IP-Adresse angegeben werden (siehe Abb. 3.6). Die Sender-IP-Adresse einer RSVP-Session ist allerdings immer eine Unicast-Adresse. Dadurch kennt der RSVP-Empfänger die genaue Anzahl der Sender innerhalb der Session und kann die Ressourcenreservierung optimal verwalten. RSVP ist als ein Management-Protokoll zu sehen, welches keine eigenen Routing-Mechanismen besitzt, sondern seine Daten entlang der schon existierenden Routen sendet. Somit ist für den Einsatz von RSVP keine grundsätzliche Änderung der Netzstruktur oder der IP-Router notwendig.

Da die RSVP-Objekte in IP-Pakete eingebettet werden und alle Knoten mit ihrer IP-Adresse angesprochen werden, ist es möglich, Nachrichten zwischen zwei RSVP-fähigen Endstationen über nicht RSVP-fähige Teilnetze transparent zu transportieren. Dabei kann in diesen Teilbereichen der Datenverkehr nur nach dem Prinzip Best-effort behandelt wird. Diese Möglichkeit ist sinnvoll, da man nicht alle Internet-Router sofort umrüsten kann und wenn ausreichend Ressourcen (Bandbreite) zur Verfügung steht. RSVP besitzt ähnlich zum TCP bzw. UDP eine eigene IP-Protokoll-ID (Integerzahl 47). Jedes RSVP-Paket wird mit dieser ID im IP-Header gekennzeichnet. Empfängt ein RSVP-fähiger Knoten ein IP-Paket mit der ID 47, so wird es, bevor es weitergeleitet wird, im Netzknoten ausgewertet und gegebenenfalls entsprechend der dienstspezifischen Regeln verändert.

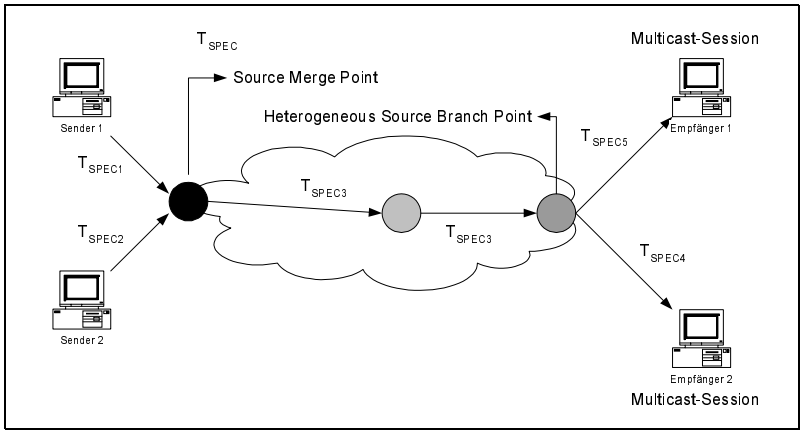


Abb. 3.6
RSVP-Modell mit unterschiedlichen T_{SPEC}

Die Ressourcen-Reservierung in IP-Hosts und Routern wird bei RSVP durch den Soft-State vorgenommen. Das bedeutet, dass die Reservierungen nicht permanent vorhanden sind, sondern in bestimmten Zeitintervallen über PATH- und RESV-Nachrichten nachgefragt werden. Dies wird so umgesetzt, da es sich bei IP um ein unsicheres Protokoll handelt, das nicht garantieren kann, dass alle Nachrichten den Empfänger erreichen. Folglich würde eine solche Verbindung im Netz bleiben, bis sich irgendwann der Speicher in einem beteiligten Netzknoten gefüllt hat. Wenn dabei ein definiertes Zeitlimit überschritten wird, kommt es zum Verbindungsabbruch. Darüber hinaus müssen die reservierten Ressourcen nachträglich gelöscht werden. Ist das Ausbleiben der Nachricht auf einen Paketverlust zurückzuführen, so wird mit der nächsten RESV-Nachricht erneut eine Reservierung initiiert.

Weiterhin erlaubt die Soft-State-Methode beim RSVP die Reservierung von QoS-Parametern für einen Datenfluss. Diese Parameter sind dynamisch und können jederzeit während der Verbindungsdauer verändert werden. Bei ATM ist das nicht möglich, da hier ein statischer QoS verwendet wird, der sich im Zustand Hard-State befindet. Zusätzlich müssen alle Empfänger, die an einer RSVP-Sitzung beteiligt sind, in der Lage sein, dem Netz sinnvolle Reservierungen abzuverlangen. Dies ist nur dann möglich, wenn vor dem Initiieren der eigentlichen Reservierung Informationen über die gegenwärtige Leistungsfähigkeit der Übertragungsstrecke und des Senders zur Verfügung gestellt werden. Abb. 3.7 verdeutlicht diese Problematik. Nicht RSVP-fähige Router können demnach auch keine Reservierung der Bandbreite vornehmen, wodurch kein QoS am Verbindungsende vorhanden ist.

Eine Reservierung wird immer von einem Empfänger eingeleitet. Somit können neue Sender während einer Sitzung zugeschaltet bzw. die Sender, die die Sitzung verlassen, wieder abgeschaltet werden. Die Reservierungsparameter

können für jeden Empfänger einzeln oder auch von mehreren Empfängern gemeinsam ausgehandelt werden.

Beim RSVP wird immer von einer Simplex-Verbindung ausgegangen. Obwohl eine Anwendung als Sender und Empfänger gleichzeitig agieren kann, werden die Richtungen logisch als zwei getrennte RSVP-Sitzungen mit jeweils voneinander unabhängigem Zustand behandelt. Somit ist die Angabe einer Downstream- bzw. Upstream-Richtung und die eines Previous-und Next-Hop in einer RSVP-Sitzung eindeutig. [BRAD97]

Aufbau einer RSVP-Verbindung

Ein RSVP-Sender sendet eine PATH-Message Downstream entlang des Unicast- bzw. Multicast-Pfades mit der IP-Adresse des Empfängers. Somit kann diese Nachricht problemlos nicht RSVP-fähige Netze passieren. Ist dieses der erste Sender für einen Empfänger, so wird ein Path-State-Block (PSB) in jedem RSVP-Knoten entlang des Datenpfades erzeugt. Er enthält folgende Daten:

- Die Empfängeradresse sowie die Port-Nummer und das verwendete IP-Protokoll, die die RSVP-Session kennzeichnen.
- $SENDER_{TEMPLATE}$[35]
- Unicast-IP-Adresse des PHOP[36]
- Sender-T_{SPEC}[37]

Die Sender-T_{SPEC} wird im PSB des jeweiligen Knotens gespeichert und unverändert downstream weitergereicht. Enthält die empfangene PATH-Nachricht eine AD_{SPEC}[38]-Struktur, so fragt der RSVP-Prozess beim System die notwendigen Parameter ab, modifiziert damit die AD_{SPEC}-Struktur und leitet sie modifiziert mit der PATH-Message in Richtung des Empfängers weiter. Als PHOP-Adresse der PATH-Nachricht wird beim Weiterleiten der Nachricht die IP-Adresse der Ausgangsschnittstelle angegeben. Existiert im Knoten schon ein PSB für diese Session, so werden die Empfängerdaten diesem hinzugefügt.

Die PATH-Message sammelt Informationen über die Qualität der Verbindungsstrecke und die möglichen Empfänger-Clients (B, C und D). Die RSVP-Router müssen nach Erhalt der PATH-Meldung die vom Sender unterstützten

35 Hat das Format einer $FILTER_{SPEC}$. Diese Nachricht wird der PATH-Message beigefügt. Anhand dieser werden unterschiedliche Sender innerhalb einer RSVP-Sitzung identifiziert.

36 IP-Adresse des nächsten Upstream-RSVP-Knotens. Jeder RSVP-Knoten trägt hier beim Weiterleiten der PATH-Message seine Adresse ein. Beim Empfang der PATH-Message wird diese Adresse im PSB gespeichert.

37 Eine Datenstruktur, die den Datenverkehr eines RSVP-Senders beschreibt. Sie beinhaltet Parameter wie mittlere Datenrate, maximale Datenrate, maximale Burst-Größe etc. Diese Datenstruktur wird vom RSVP-Prozess an den Routing-Prozess bzw. an den Anwendungsprozess unverarbeitet weitergereicht.

38 Advertising Specification: diese Struktur wird in der PATH-Message von RSVP hop-by-hop durchgereicht und dient zur Bestimmung der gesamten Dienstgüte vom Sender bis zum Empfänger.

Dienstklassen einer kritischen, lokalen Prüfung unterziehen. Sind einzelne Parameter nicht anwählbar, protokolliert ein Flag dieses. Dabei kann ein identischer QoS nicht bereitgestellt werden, da Empfängerunterschiede hinsichtlich der Dienstparameter wie Jitter-Verhalten, Verzögerungen und Bandbreite bestehen. RSVP bedient sich dabei des Prinzips des Least Upper Bound (LUB), wodurch nur die Maximalwerte der Dienstparameter Berücksichtigung finden. Diese Datenflussbeschreibung wird anschließend in einer RESV-Nachricht an den nächsten RSVP-Router weitergeleitet. Auch nicht RSVP-fähige RSVP-Router können über Flags in den RESV-Meldungen entdeckt werden. Ist eine End-to-end-Reservierung nicht möglich, reduziert sich die Übertragungsqualität sofort auf den Best-effort.

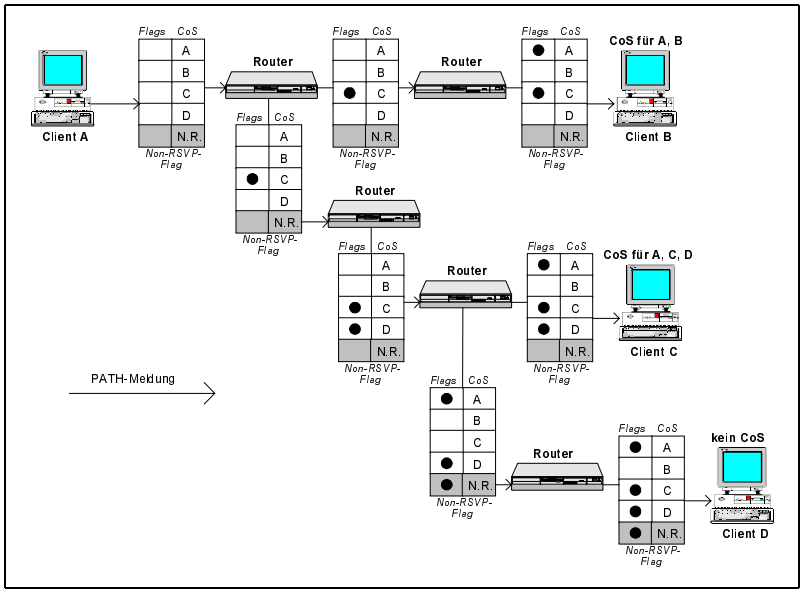


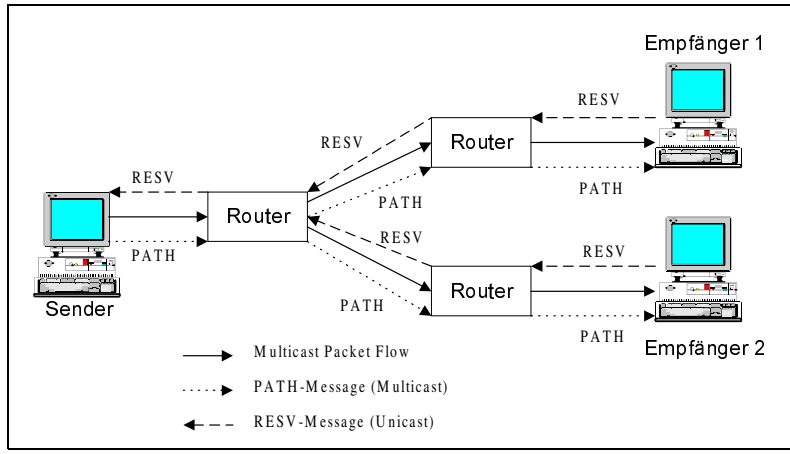
Abb. 3.7 Übertragungsstrecke einer RSVP-Verbindung

Nach Empfang der PATH-Message wird im Empfänger die *FLOWSPEC*[39]-Datenstruktur erzeugt. Sie beschreibt die angeforderte Reservierung. Meistens wird die $FLOW_{SPEC}$ aus den einzelnen Parametern der Sender-T_{SPEC} und der *ADSPEC* abgeleitet. Das steht aber im Ermessen des Anwendungsprogrammierers. Er kann die einzelnen Parameter auch mit eigenen Werten überschreiben.

39 Die $FLOW_{SPEC}$-Datenstruktur beschreibt die gewünschten QoS-Parameter einer Reservierungsanforderung. Diese Struktur besteht wiederum aus einer T_{SPEC}, die die Parameter des Datenflusses beschreibt, und einer R_{SPEC}, die zusätzliche Reservierungsparameter beschreibt. Das Format und der Inhalt der $FLOW_{SPEC}$ wird in den jeweiligen Dienstspezifikation des IntServ festgelegt und ist für das RSVP unsichtbar.

Ist die *FLOWSPEC* gebildet, so wird eine RESV-Nachricht gesendet. Sie wird immer mit der IP-Adresse des PHOP adressiert. Dadurch wird sichergestellt, dass die RESV-Nachricht exakt den Weg der PATH-Nachricht in umgekehrter Reihenfolge durchläuft.

Abb. 3.8 PATH/RESV-Meldungen bei RSVP



Empfängt ein RSVP-Knoten eine RESV-Nachricht, so wird als Erstes die Policy-Control-Einheit nach den Berechtigungen abgefragt. Ist das Ergebnis positiv, so kommt eine Anfrage an die Admission-Control-Einheit. Ist auch hier das Ergebnis positiv, so wird der Reservierungsstatus im RSB gespeichert. Jetzt muss man zwischen verschiedenen Fällen unterscheiden:

- Ist der aktuelle Knoten schon der RSVP-Sender und es existiert noch kein RSB-Eintrag, so wird ein PSB-Eintrag für den Empfänger erzeugt. Ist das RESV-CONFIRM-Bit in der PATH-Message gesetzt, so wird eine RESV-CONFIRM-Nachricht an den reservierenden Empfänger gesendet.
- Wenn allerdings für diese Session noch kein RSB existiert und der aktuelle Knoten nicht der Zielknoten ist, so wird ein RSB-Eintrag mit der Unicast-Adressse des RSVP-Empfänger erzeugt und die RESV-Nachricht weiter an den PHOP gesendet.
- Existiert allerdings für diese Session schon ein RSB mit einer größeren $FLOW_{SPEC}$ und mit einer anderen Unicast-Adrese des RSVP-Empfängers, so wird der aktuelle Empfänger im PSB eingetragen und es wird, falls das RESV-CONFIRM-Bit gesetzt ist, eine RESV-CONFIRM-Nachricht an den reservierenden Empfänger gesendet.[40]

40 Nur in Kombination mit einem Multicast RSVP-Empfänger möglich.

- Existiert im Knoten schon ein RSB für diese Session mit einer anderen Adresse des RSVP-Empfängers, dessen *FLOWSPEC* aber nicht größer als die empfangene *FLOWSPEC* ist, so wird von einer dienstspezifischen Routine[41] eine neue gemeinsame *FLOWSPEC* gebildet und eine RESV-Nachricht mit der neuen *FLOWSPEC* an den PHOP gesendet. [42]
- Existiert im Knoten schon ein RSB mit derselben Adresse des RSVP-Empfängers, so wird lediglich die *FLOWSPEC* im RSB modifiziert und die neuen Parameter dem Packet Scheduler mitgeteilt. Hat sich auch die *FLOWSPEC* nicht verändert, so handelt es sich lediglich um eine REFRESH-Nachricht. Diese Nachricht wird nicht weitergeleitet.

Da die RESV-Nachricht immer an die IP-Adresse des vorherigen Hops gesendet wird, lassen sich nicht RSVP-fähige Teilnetze überbrücken. Dies wird in der Abb. 3.9 dargestellt.

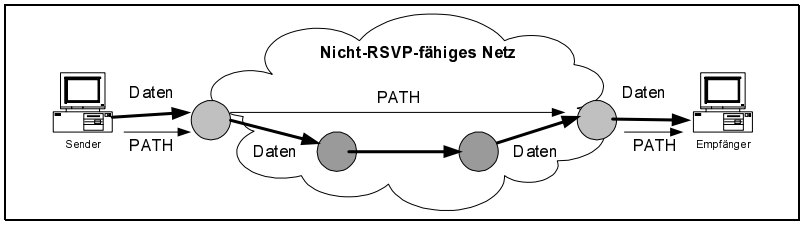


Abb. 3.9 RSVP-Reservierung über nicht RSVP-fähige Hops

Bei RSVP werden zur Aufrechterhaltung der Zustände wie erwähnt Soft States verwendet. Dabei werden Refresh- bzw. Cleanup-Timer gesetzt. Nach Ablauf des jeweiligen Refresh-Timers wiederholt der Netzknoten die PATH- bzw. RESV-Nachrichten. Der Cleanup-Timer wird auf das Vielfache vom Refresh-Timer gesetzt. Jeder RSVP-Router im Netz prüft, ob für jeden PATH bzw. RESV-Zustand innerhalb des Cleanup-Intervalls eine entsprechende Nachricht eingegangen ist. Ist das nicht der Fall, sendet der Router eine Teardown-Nachricht (PATH_TEAR in der Upstream-Richtung bzw. RESV_TEAR in der Downstream-Richtung). War die Verbindung fehlerhaft gelöscht, weil mehrere aufeinander folgende PATH- bzw. RESV-Nachrichten verloren gegangen sind, so wird mit der nächsten korrekt empfangenen Nachricht der richtige Zustand wiederhergestellt. Da das Refresh-Intervall im Bereich mehrerer Sekunden bis zu einigen Minuten liegt, ist die Wahrscheinlichkeit, dass mehrere aufeinander folgende Nachrichten ausfallen, sehr gering. Es ist hierbei zu beachten, dass der Refresh-Timer für den jeweiligen RSVP-Knoten nur als Richtwert gilt. Daraus wird mit Hilfe eines Zufallsgenerators der genaue Wert für den Refresh-Inter-

41 Der RSVP-Knoten muss für jeden unterstützten Dienst solche Routinen bereitstellen, der RSVP-Prozess muss lediglich diese Routine aufrufen können und von ihr das Ergebnis empfangen.

42 Nur in Kombination mit einem Multicast RSVP-Empfänger möglich.

vall berechnet. Diese Maßnahme dient der Vorbeugung der Timersynchronisierung im Netz, damit es nicht zur Signalisierungsflut kommt. Ist die von dem Sender bzw. Empfänger empfangene RESV- bzw. PATH-Nachricht nur als Refresh-Nachricht gesendet, so wird die nicht weitergeleitet. Somit erspart man sich Rechenzeit für die Interprozess-Kommunikation.

Um die Ressourcen-Reservierung zu ermöglichen, muss das IP-typische dynamische Routing autonomer Pakete unterbleiben. Vielmehr ist sicherzustellen, dass RSVP-Steuerungspakete (PATH- und RESV-Meldungen) sowie alle Nutzdaten den Pfad nehmen, der durch die PATH-Meldung gewählt wurde. Die hierzu notwendigen Informationen speichert jeder Router innerhalb verschiedener Datenstrukturen, den so genannten State Blocks, ab:

- **Path State Block (PSB):** PATH-Meldungen werden hier abgespeichert. Zusätzlich wird der nächste Router im Upstream erkannt. PSB kann einer dritten Sitzung, einem Sender und einer Schnittstelle zugeordnet werden.
- **Reservation State Block (RSB):** Sitzungsabhängige Informationen bezüglich der RESV-Meldung werden festgehalten. RSB kann einer dritten Sitzung, einem Next-Hop und Filtermechanismen zugeteilt werden.
- **Traffic Control State Block (TCSB):** Im Gegensatz zum RSB werden hier Ressourcen-Informationen bezüglich einer Schnittstelle festgehalten.
- **Blockade State Block (BSB):** Blockierung von Ressourcen zum Merging ist möglich.

Beide Meldungen (PATH und RESV) können auf ihrem Weg vom Sender zum Empfänger eklatanten Änderungen unterworfen werden bzw. den Empfänger/Sender niemals erreichen. Das kann durch das Merging passieren, das die effiziente Nutzung von Netzressourcen zur Aufgabe hat. Wie und ob das Merging von Ressourcen vorgenommen wird, entscheiden zwei Reservierungsarten: Distinct Reservation und Shared Reservation. Die Distinct Reservation ermöglicht dem Sender, Ressourcen für die eigene Nutzung zu reservieren. Shared Reservation teilt sich hingegen die verfügbaren Ressourcen mit allen beteiligten Sendern. Dabei muss eine Filterung der IP-Datenströme erfolgen. Folgende Verfahren sind möglich:

- **Wildcard-Filter (WF):** Die vom Empfänger angeforderten Ressourcen lassen sich durch alle beteiligten Sender nutzen.
- **Fixed-Filter (FF):** Jeder Empfänger stellt explizite Anforderungen an die einzelnen Sender. Anforderungen an unterschiedliche Sender können nicht berücksichtigt werden. Anforderungen unterschiedlicher Empfänger sind aber möglich.
- **Shared-Explicit-Filter (SE):** Ähnlich dem Wildcard-Filter werden hier die Möglichkeiten zum Merging der Anforderungen an verschiedene Sender gegeben. Allerdings findet eine explizite Auflistung der Sender statt.

TCP/IP-Stack-Implementierung

Die Realisierung von RSVP mit seinem komplexen Reservierungsmodell muss eine klare Trennung in verschiedene Module beinhalten. Um über RSVP Nachrichten auszutauschen, läuft in jedem RSVP-Knoten ein RSVP-Prozess, der die notwendigen Nachrichten erzeugt bzw. empfangene RSVP-Nachrichten verarbeitet und weiterleitet. Zur Kommunikation zwischen der Anwendung und dem RSVP-Prozess muss eine Schnittstelle definiert werden. Eine Möglichkeit hierfür ist das Resource Reservation Protocol Application Programming Interface (RAPI), welches auch in Abb. 3.10 zu erkennen ist. [BRAD98] Damit diese unabhängig von der jeweiligen RSVP-Implementierung bleibt, sind die Anforderungen an diese Schnittstelle im selben RFC-2205 festgehalten. Die exakte Beschreibung der auszutauschenden Datenstrukturen muss außerhalb des RSVP definiert werden.

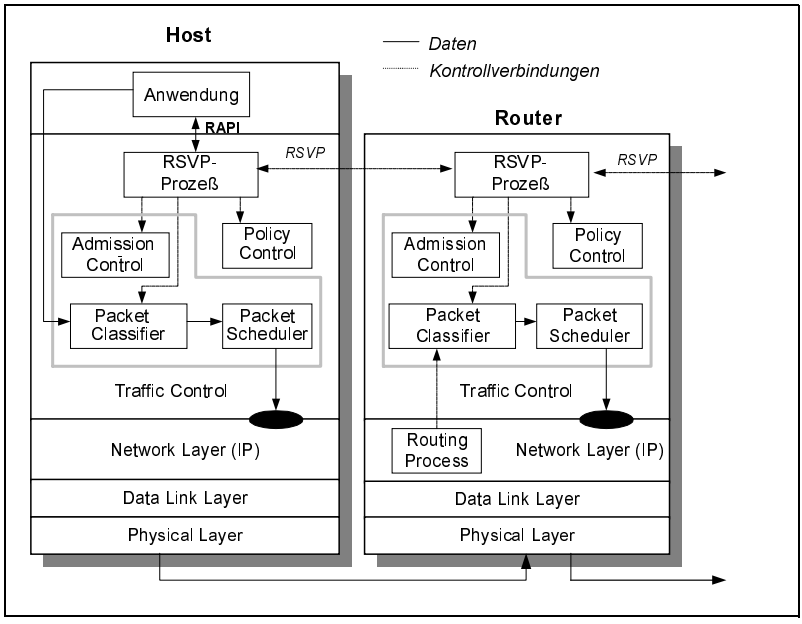


Abb. 3.10 Einordnung von RSVP in den TCP/IP-Protokollstapel

Die Implementierung eines RSVP-Prozesses muss zwischen Router und Host unterschieden werden. Abb. 3.10 zeigt eine solche Unterteilung zwischen Host und Router. Hier wird die Einordnung des RSVP-Protokolls in den TCP/IP-Stack verdeutlicht. RSVP ist auf der OSI-Schicht-4 aufgrund der Nutzung bereits vorhandener Routing-Mechanismen und der abschnittsweisen Verwendung von QoS-Parametern angesiedelt. Die beteiligten Hosts entlang eines Übertragungspfads werden trotzdem nicht als Gateways, sondern als RSVP-Router angesehen. Somit ist RSVP im Grunde kein Transportprotokoll, sondern ist ausschließlich für die Transportüberwachung zuständig. In den Routern kommuni-

ziert der RSVP-Prozess entsprechend mit einem Routing-Prozess. Die Schnittstelle zwischen RSVP und dem Routing-Prozess ist herstellerspezifisch. Außerdem wird ein Policy-Prozess auf der Anwendungsebene benötigt, der die Berechtigungen zur Kanalreservierung anhand definierter Policies prüft. Auf den unteren Schichten kommen Paket-Klassifizierer (Paketeinteilung), Paket-Scheduler (Paketplaner) bzw. eine Traffic-Control (Verkehrskontrolle) und eine Admission-Control (Zulassungskontrolle) zum Einsatz.

Die Einheit Admission Control prüft die Verfügbarkeit physikalischer Ressourcen zur Realisierung gewünschter Dienstgüte auf dem Interface. Somit kann eine Kanalreservierung nur nach erfolgreicher Abfrage der Policy Control und der Admission Control durchgeführt werden. Der Paketklassifizierer entscheidet anhand des Paketinhalts (z.B. anhand des IP-Headers und der TCP[43]/UDP[44]-Port-Nummer) über die Zugehörigkeit der verarbeitenden Pakete zu einer bestimmten QoS-Klasse. Die Entscheidungsregeln für die Policy Control, Admission Control und Paketklassifizierer werden in den Spezifikationen der entsprechenden Dienste (z.B. eines IntServ- oder DiffServ-Dienstes) festgelegt und sind nicht Bestandteil des RSVP.

Die Traffic Control wird unterteilt in Zulassungskontrolle (Admission Control), Paketklassifizierer (Packet Classifier) und Paketplaner (Packet Scheduler). Die Zulassungskontrolle entscheidet aufgrund von Informationen über die Netzauslastung, ob Reservierungen zurückgewiesen oder gestattet werden. Die Dienste der Zulassungskontrolle sind somit nur beim Aufbau bzw. bei Änderungen der Sitzungsparameter notwendig. Der Paket-Scheduler ist systemabhängig. Er wird beispielsweise als Kernel-Modul realisiert und sorgt für die Einhaltung der zugesicherten QoS-Parameter auf dem entsprechenden Netzinterface. Das Protokoll definiert eine RSVP-Sitzung, die durch die IP-Adresse des Empfängers, das verwendete IP-Protokoll (TCP oder UDP) und die Empfänger-Portnummer eindeutig bestimmt wird. Dabei kann es sich bei der Empfängeradresse sowohl um eine Unicast- als auch um eine Multicast-Adresse handeln. Damit wird gewährleistet, dass während einer Sitzung neue Sender problemlos hinzukommen bzw. andere die Sitzung verlassen können. In den Knoten entlang des Datenpfades werden für jede Sitzung der Path-State-Block (PSB), Reservation-State-Block (RSB) und Blockade-State-Block (BSB) gespeichert und verwaltet. Dabei beinhaltet der PSB die relevanten Daten aller bei der Session angemeldeten Sender, der RSB die Daten aller zustande gekommener bzw. fehlgeschlagener, aber noch nicht aufgelöster Reservierungen, während der BSB die höchsten zulässigen Grenzwerte für neue Reservierungsanforderungen im Blockadezustand beinhaltet.

43 Transmission Control Protocol
44 User Datagramm Protocol

Zusammenfassend ist die Verkehrskontrolle für die Entkopplung der reservierten Sender-Ressourcen zuständig. Falls unterschiedliche Sender über denselben Pfad erreichbar sind, ist der Empfänger in der Lage, verschiedene Sender anzusprechen und diese ohne Änderung der Reservierung zu wechseln. Die Paketeinteilung muss nur entsprechend angepasst werden. Zusätzlich kann man die Dienstelemente zwischen den physikalischen Schnittstellen einzeln zuordnen. Eine Unterscheidung findet lediglich in Incoming- und Outgoing-Interfaces statt. Eine Hardware-Karte kann beide Arten annehmen, aber nie gleichzeitig. [BRAD97]

Merging

Eine RSVP-Session beinhaltet im Allgemeinen einen Empfänger und mehrere Sender. Dabei kann als Empfängeradresse auch eine Multicast-Adresse angegeben sein. Somit können tatsächlich m x n Sitzungen realisiert werden.

Eine Hauptfunktionalität von RSVP ist die Möglichkeit der Zusammenfassung mehrerer einzelner Reservierungen zu einer gemeinsamen. Eine solche Zusammenfassung ist vorteilhaft, wenn im Voraus bekannt ist, dass mehrere Sender die Ressourcen nicht gleichzeitig belegen werden. Das ist beispielsweise bei einer Audiokonferenz der Fall. Bei einer Audiokonferenz sprechen in der Regel nicht mehrere Teilnehmer gleichzeitig, somit macht es keinen Sinn, mehr als einen Sprachkanal zu reservieren. Solche Zusammenfassungen mehrerer Reservierungsanforderungen zu einer einzigen wird als Merging bezeichnet. Diese Zusammenfassung findet in der Regel direkt beim Empfänger statt.

Ein anderes Merging findet bei Multicast-Paketen statt. Hier sendet jeder einzelne Empfänger eine RESV-Meldung mit einer eigenen $FLOW_{SPEC}$. Kommen diese Anforderungen im HSBP[45] zusammen, so muss aus den einzelnen $FLOW_{SPEC}$ eine neue gemeinsame *FLOWSPEC* gebildet werden. Die Regeln für solche Zusammenfassung einzelner Strukturen werden in den jeweiligen Dienstspezifikationen festgelegt. Ein anderes Thema ist die Zusammenfassung der Reservierungen für mehrere RSVP-Sender. Dafür wird für jede Anforderung ein Reservierungsstil angegeben. Folgende Stile sind definiert worden:

Tab. 3.4 Reservierungsstile für RSVP

Senderangabe	**Reservierung**	
	Distinct Style	**Shared Style**
Explicit	Fixed Filter (FF)	Shared Explicit (SE)
Wildcard	-	Wildcard Filter (WF)

45 Heterogeneous Source Branch Point

Bei einem FF-Reservierungsstil wird für jeden Sender eine eigene *FLOWSPEC* gebildet und in jedem RSVP-fähigen Knoten entlang des Pfades eingetragen. Man kann zwar bei Verwendung des FF-Styles die Reservierung auf jeden Sender einzeln abstimmen, dabei werden aber auch für jeden Sender explizit Ressourcen reserviert, was im bereits erwähnten Szenario eine Ressourcenverschwendung bedeuten würde. Bei dem Shared-Explicit-Style (SE-Stil) werden für alle Sender nur einmal Ressourcen reserviert. Dabei werden die Sender explizit in einer *FILTERSPEC* aufgelistet. Bei den Wildcard-Filter-Style (WF-Stil) wird genauso je Empfänger eine gemeinsame *FLOWSPEC* gebildet, wofür aber die Empfängeradresse nicht explizit angegeben wird, sondern man setzt diese als Wildcard. [BRAD97]

Der WF-Stil hat gegenüber den SE-Stil nur den Vorteil, dass man bei sehr vielen Empfängern Overhead einspart. Weist das Netz eine vermaschte Struktur auf, so besteht bei der Weiterleitung der RESV-Nachricht an die (Wildcard-)Empfänger die Gefahr der Schleifenbildung. Um dies zu vermeiden, wird in die RESV-Nachricht bei einem WF-Stil beim Weiterreichen zu einem PHOP[46] auf jedem Interface des aktuellen Knotens ein Scope-Object eingefügt, in dem die Sender, die über das aktuelle Interface erreicht werden sollen, einzeln aufgelistet werden. Somit verringert sich auch der Vorteil der Verwendung des WF-Stils.

Abb. 3.11 zeigt unterschiedliche Reservierungsbeispiele in vereinfachter Form. Dabei gehören die Empfänger *R1* bis *R3* zu einer Multicast-Session und entsprechend zu einer RSVP-Session. Dies ist der allgemeinste Fall einer 3x3-Session. Der Router ist bei diesem Beispiel der HSBP[47] der Multicast-Session. Somit werden die $FLOW_{SPEC}$ einzelner Empfänger unabhängig vom Reservierungsstil zusammengefasst. Die Reservierungsstile sind alle von der *FLOWSPEC* abhängig.

Der FF-Stil stellt die Fortpflanzung der Anforderung dar. Anforderungen verschiedener Empfänger werden somit zu einer Anforderung zusammengefasst. Auf dem Link zwischen dem Router und dem Sender wird dagegen für jeden Sender einzeln reserviert. Beim SE-Stil wird erst der kleinste gemeinsame *FLOWSPEC* im jeweiligen Empfänger gebildet (hier durch Maximumbildung aller Sender-*TSPEC*) und danach wird die Verbindung zwischen Sender und Router (wegen einer Multicast-Session) noch einmal zusammengefasst. Beim WF-Stil gelten dieselben Regeln für das Zusammenschmelzen wie beim SE-Stil. Dabei werden aber definitionsgemäß Anforderungen für alle Sender (und nicht nur für eine Teilmenge der Sender) zusammengefasst.

46 Previous Hop
47 Heterogeneous Source Branch Point

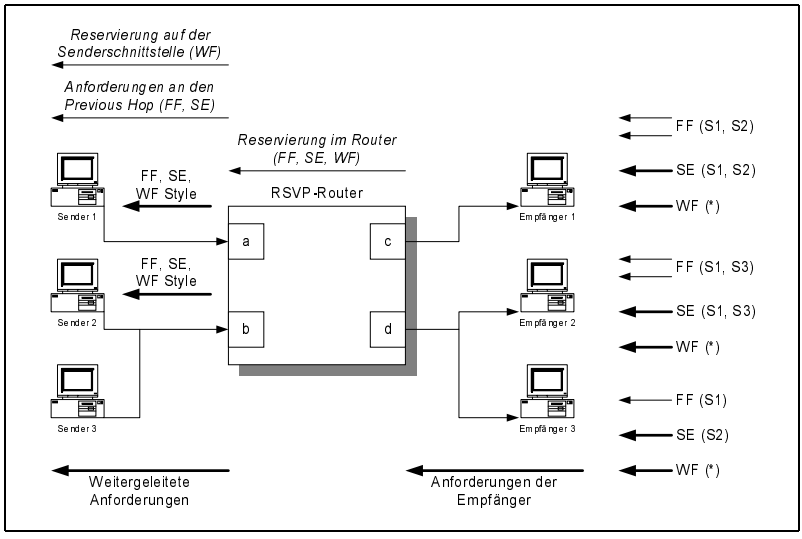


Abb. 3.11 Multicast RSVP-Reservierungsbeispiele

Die Regeln der Zusammenfassung der *FLOWSPEC* sind dienstabhängig und somit auch außerhalb der RSVP-Spezifikation definiert. Allerdings können folgende Probleme im Multicast-Einsatz auftreten:

1. **Reservierungsablehnung:** Bei einer Reservierung kann durch Setzen eines Flags in der PATH-Message eine explizite Bestätigung der Reservierung verlangt werden. Die Reservierung wird hop-by-hop an das jeweilige PHOP[48] gesendet. Kommt es in einem Knoten zum Mergen der Anfrage mit einer größeren *FLOWSPEC*, so wird die RESV-Nachricht nicht mehr weitergereicht, sondern lediglich der Eintrag RESV-State im Knoten modifiziert. Dieser Netzknoten erzeugt dann eine RESV_CONFIRM-Message. Danach wird eine Reservierung von der Admission-Control-Einheit in einem Knoten näher zum Sender abgewiesen. Es kommt demnach zu keiner Ende-zu-Ende-Reservierung, obwohl der Empfänger eine Bestätigung empfangen hat.
2. **Killer-Reservierung (1):** Wenn eine Reservierungsanfrage zu einer existierenden Reservierung hinzukommt, werden diese beiden Anforderungen im Merging Point zusammengefasst. Damit ist die Anfrage ab dem HSBP Upstream nicht mehr sichtbar, sondern lediglich die Reservierung. Beim Weiterreichen der Anfrage an den Previous Hop wird dann die Anfrage durch eine Admission-Control-Einheit abgewiesen. Da aber die Anfrage in der Reservierung verdeckt ist, würde auch die Reservierung wegen der Abweisung durch die Admission-Control-Einheit aufgelöst.

48 Previous Hop

3. **Killer-Reservierung (2):** Ein ähnliches Problem tritt bei einer existierenden Reservierung auf, wenn ein Empfänger permanent eine Anfrage sendet, die in einem Knoten immer wieder durch die Admission-Control-Einheit abgewiesen wird. Kommt es zu einer weiteren Anforderung, so würde diese mit der permanenten Anforderung abgewiesen werden.

Die Lösung des ersten Problems wäre, wenn ein RSVP-Knoten eine RESV_ERROR-Nachricht erzeugt, wodurch diese nicht nur an den die Reservierung initiierenden Empfänger weitergeleitet wird, sondern auch an alle anderen Empfänger in dieser Session. Somit bekommt der Empfänger einer RESV_CONFIRM- und später auch eine RESV_ERROR-Nachricht.

Um das zweite Problem zu vermeiden, wird festgelegt, dass in einem solchen Fall der vorherige Zustand entlang des ganzen Pfades wiederhergestellt werden soll. Dieses Problemszenario könnte man natürlich auf mehr als zwei Anfragen ausdehnen. Es ist hierbei aber nicht festgelegt worden, wie viele Reservierungsanforderungen ein RSVP-fähiger Knoten in der Lage sein soll, rückgängig zu machen. Dies wird natürlich komplexer, je mehr Anfragen zu bewältigen sind.

Um das dritte Problem zu vermeiden, muss ein Schutzmechanismus eingeführt werden: Weist die Admission Control eine Anforderung ab, so wird eine RESV_ERROR-Nachricht erzeugt, die als Hop-by-hop-Downstream zu jedem Empfänger in der Session durchgereicht wird. Jeder RSVP-Knoten, der eine solche Nachricht empfängt, merkt sich dabei die QoS-Parameter der fehlgeschlagenen Anforderung in einem BSB[49]. Dabei wird ein Blockade State Timer gesetzt. Ab jetzt werden alle Anforderungen in jedem Knoten bis zum Ablauf des Timers ignoriert. Kommt eine neue Anforderung vor Ablauf des Timers, so wird der Timer neu gestartet. Anfragen, die sich innerhalb der definierten Rahmenbedingungen befinden werden dabei ungestört verarbeitet und weitergereicht. [SIEM99]

Bestimmung der Dienstgüte

Bislang wurde ausschließlich der RSVP-Sender und -Empfänger betrachtet. Das heißt, der T_{SPEC} meldet dem Sender seine Daten, während der *RSPEC* die konkretem Anforderungen an das Netz charakterisiert. Dabei wird allerdings nicht berücksichtigt, ob das zwischen dem Sender und Empfänger liegende Netz überhaupt RSVP-fähig ist oder die Daten über nicht RSVP-fähige Abschnitte geschickt werden. Weiterhin bleibt die Frage unbeantwortet, welche Dienste die Strecke zwischen Sender und Empfänger unterstützt, ob man damit rechnen kann, dass die Anforderungen auch erfüllt werden und mit welcher Latenzzeit zu rechnen ist. Es fehlen somit noch Kenntnisse vom dazwischenliegenden Netz. Mit Hilfe der *ADSPEC* [50]-Struktur wird dies geändert.

49 Blockade-State-Block

50 Advertising Specification: diese Struktur wird in der PATH-Message von RSVP hop-by-hop durchgereicht und dient zur Bestimmung der gesamten Dienstgüte vom Sender bis zum Empfänger.

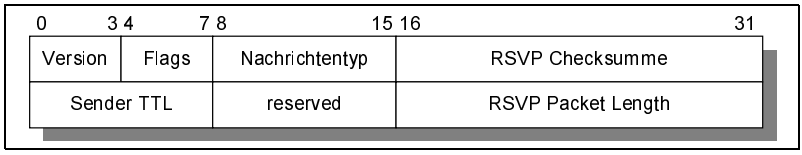


Abb. 3.12
RSVP-Header

Abb. 3.12 zeigt den RSVP-Header, der einer RSVP-Nachricht vorangestellt wird. Der Header ist 8 Byte (64 Bit) lang und enthält wie der IP-Header ein Feld Time-to-Live (TTL), welches ermöglicht, nicht RSVP-fähige Abschnitte zu erkennen. Beim Sender wird der TTL-Wert im RSVP-Header auf den entsprechenden Wert des IP-Headers gesetzt. Passiert das Paket einen RSVP-Knoten, so wird mit dem TTL-Wert des IP-Headers auch der TTL-Wert des RSVP-Headers heruntergesetzt. Ist der aktuelle Knoten nicht RSVP-fähig, so wird der RSVP-Header als normaler Dateninhalt behandelt, der TTL-Wert im RSVP-Header bleibt unverändert, wird aber im IP-Header heruntergesetzt. Kommt das Paket wieder in einem RSVP-Knoten an, so merkt dessen Traffic-Control-Einheit anhand der unterschiedlichen TTL-Werte, dass eine nicht RSVP-fähige Strecke dazwischen liegt. Das entsprechende Bit der *ADSPEC* -Struktur wird gesetzt. Durch Auswertung dieses so genannten Break-Bits kann der Empfänger über die Präsenz nicht RSVP-fähiger Abschnitte im Datenpfad erfahren. Diese Vorgehensweise ist aber nicht sicher genug; sie versagt vor allem bei Verwendung von Tunnelmechanismen. Somit wird sie von der IETF als Notlösung betrachtet. Es ist allgemein ein Bit in der *ADSPEC* zur Signalisierung der Überquerung von Nicht-RSVP-Strecken vorgesehen.

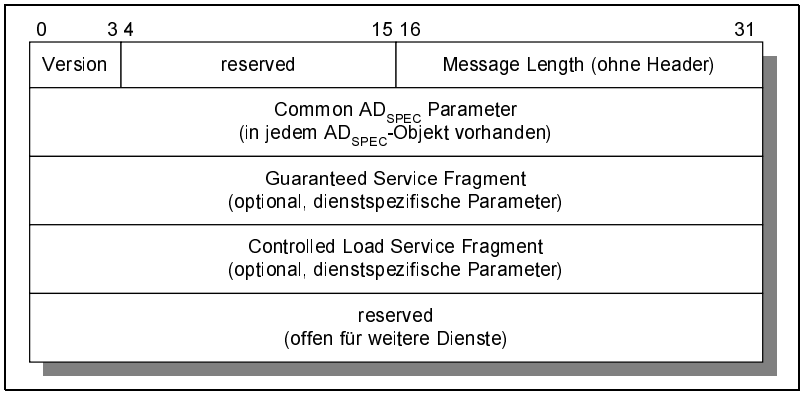


Abb. 3.13
AD_{SPEC}-Struktur

Die *ADSPEC*-Struktur dient der Bestimmung der QoS-Eigenschaften einer RSVP-Strecke. Sie ist modular aufgebaut. Jede AD_{SPEC}-Struktur beinhaltet einen Nachrichtenkopf und einen allgemeinen Block. Darauf können weitere dienstspezifische Blöcke folgen. Jedes Element im Block besteht aus einer eindeutigen Nachrichten-ID, einem Flag-Byte, einem Nachrichtenlängen-Feld

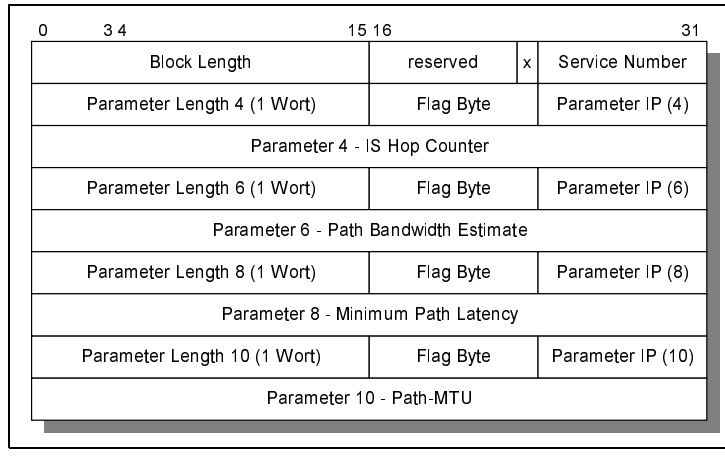
und dem Nachrichteninhalt. Damit kann die Struktur allgemein gehalten werden. Der RSVP-Prozess muss also nicht die einzelnen Objekte kennen.

Bei den *ADSPEC*-Parametern muss zwischen lokalen und zusammengesetzten Parametern unterschieden werden. Die lokalen Parameter charakterisieren eine einzelne Verbindung, die sich aus Parametern der gesamte Strecke vom Sender bis zum erreichten Knoten zusammengesetzten. Die *ADSPEC* dient nur der Übertragung bzw. der Kumulierung, nicht aber der Beschaffung von Informationen. Die Informationen werden von der Traffic Control oder auch von einer anderen, möglicherweise auch dienstspezifischen Einheit bereitgestellt. Manuelle Einträge in den Routern werden zugelassen.

Abb. 3.14 zeigt die *ADSPEC*-Struktur mit den dienstübergreifenden Parametern. Das Global-Break-Bit[51] wird auf 1 gesetzt, falls eine Nicht-RSVP-Strecke entdeckt wurde. Die Bestimmungsregel für die Berechnung der zusammengesetzten Parameter aus den lokalen Parametern ist eine ODER-Verknüpfung. Die PATH-MTU wird auf den MTU-Wert der zutreffenden Verbindung gesetzt. Der zusammengesetzte Parameter wird durch eine Minimumbildung berechnet.

Der Parameter Path Bandwidth Estimate liefert die auf der Verbindung verfügbare Bandbreite in [Byte/s]. Auch hier wird entlang des Pfades der Gesamtwert durch eine Minimumbildung berechnet. Die Minimum PATH Latency gibt die minimale Verzögerung auf der Strecke an. Dabei wird nur die Fortpflanzung des Signals, nicht aber die Verzögerung in den Warteschlangen berücksichtigt. Somit stellt dieser Wert immer eine Unterschätzung dar. Die Zusammensetzungsregel ist hierbei eine Summierung. Der IS Hop Counter, der anschließend kommt, zählt die überquerten RSVP-Knoten. Auch hier werden die einzelnen Werte entlang des Pfades aufsummiert.

Abb. 3.14
AD_{SPEC}-Parameter



51 Wird in der Abb. 3.14 als „x“ bezeichnet.

Die *ADSPEC* wird von dem Sender zum nächstliegenden RSVP-Knoten erzeugt. Somit besitzt der Sender keine Kenntnisse davon, welchen IntServ-Dienst der Empfänger wählt bzw. welchen Dienst das dazwischenliegende Netz unterstützt. Deshalb fügt der Sender in aller Regel je einen dienstspezifischen Block für jeden von ihm unterstützten Dienst der *ADSPEC*-Struktur hinzu. Außerdem gilt in der ersten Version des RSVP die Beschränkung, das alle Reservierungen innerhalb einer RSVP-Sitzung denselben IntServ-Dienst wählen müssen. Damit sind die Dienste CLNE[52] oder GQOS[53] gemeint, wobei aber auch weitere IntServ-Dienste spezifiziert werden können, ohne dafür RSVP anpassen zu müssen.

Die jeweiligen Spezifikationen der IntServ-Dienste beschreiben die jeweiligen Inhalte der dienstspezifischen *ADSPEC*-Blöcke. Das erste 4-Byte-Wort ist dabei in allen Blöcken gleich. Es beinhaltet die Elemente Identifikationsnummer, Break-Bit und Blocklänge. Die Identifikationsnummer des Dienstes innerhalb des RSVP kennzeichnet folgende Inhalte:

- 1 für Common-Block
- 2 für Guaranteed Service
- 5 für Controlled Load Service

Das Break-Bit ist in jedem Dienst im 23. Bit. Die Blocklänge beträgt 32 Bit, wobei man das erste Wort nicht mitzählt. Bei der Verarbeitung der *ADSPEC* müssen bestimmte Regeln von den RSVP-Knoten eingehalten werden. Wird beispielsweise ein nicht RSVP-fähiger Abschnitt erkannt, so wird das 23. Bit des allgemeinen Blocks auf 1 gesetzt. Das heißt, ab jetzt können alle Werte, die in der *ADSPEC* eingetragen sind, nicht korrekt sein. Erkennt der Knoten einen gültigen allgemeinen oder auch dienstspezifischen Block, so werden entsprechende Routinen der Einheit Traffic Control aufgerufen, um die entsprechenden Werte abzufragen und zu setzen. Da die einzelnen Parameter der *ADSPEC* die Eigenschaften des Datenpfades beschreiben, werden sie als allgemeine Charakterisierungsparameter[54] bezeichnet.

Ist die Dienstkennung in einem dienstspezifischen Header unbekannt, so wird das Bit 23 (Break-Bit) in diesem Header auf eins gesetzt und die Länge des Blocks auf Null. Das Setzen der Datenlänge auf null wird nicht zum Fehler führen, da die Gesamtlänge der *ADSPEC*-Struktur außerhalb des einzelnen Blocks definiert wird und somit auch unverändert bleibt. Bekommt ein Empfänger eine *ADSPEC*, in der das Break-Bit für einen Dienst gesetzt ist, so ist bekannt, dass ein Knoten auf der Strecke diesen Dienst nicht unterstützt. In diesem Fall darf der entsprechende Dienst vom RSVP-Empfänger nicht angefordert werden.

52 Controlled Load Network Element
53 Guaranteed Quality-of-Service
54 Generalized Characterisation Parameter (GCP)

Wird in einem dienstspezifischen Block derselbe Parameter wie im allgemeinen Header gesetzt, ist das nicht als Fehler zu interpretieren. Damit wird für den speziellen Dienst der allgemeine Parameter überschrieben. Dies wird allgemein als Overriding bezeichnet. So etwas kann auftreten, wenn man für einen Dienst mit härteren QoS-Anforderungen die allgemeinen Parameter einschränken will. Es kann beispielsweise im Guaranteed-Service-Block der *ADSPEC* die verfügbare Bandbreite auf einen kleineren Wert als im allgemeinen Block gesetzt werden. Der Empfänger weiß in diesem Fall, dass er für den entsprechenden Dienst nicht die allgemeinen, sondern die dienstspezifischen Parameter anwenden soll. [BRAD97]

Ausblick RSVP ermöglicht es dem Anwender, von einer Anwendung aus Netzressourcen zur exklusiven Verfügung zu reservieren. Das birgt aber auch einige Gefahren, da man dem Missbrauch von Reservierungen vorbeugen muss, und gleichzeitig möchte man die Bereitstellung einer relativ garantierten Dienstgüte extra tarifieren bzw. abrechnen. Das erste Problem wird durch das Konzept der Policy Control im RSVP gelöst. Zu dessen Realisierung kann man jeder RSVP-Nachricht ein so genanntes Integrity Object anfügen, welches die Authentisierungsdaten enthält. Zur Authentisierung sollte der MD5-Digest-Agorithmus eingesetzt werden. Dafür muss jeder RSVP-Knoten die Nutzer seiner direkten Nachbarn und deren Zugriffsrechte kennen. Außer dem Integrity Object kann jede Nachricht auch ein Policy Object beinhalten, das die Zugehörigkeit der Nutzer zu einer bestimmten Nutzergruppe, die Account-Nummer usw. enthalten kann. Die Authentisierungsalgorithmen werden aber immer noch nicht einheitlich unterstützt. Obwohl Policy-Regeln entscheidend für den Einsatz von RSVP in den Netzen ist, herrscht in diesem Bereich noch einige Unklarheit bezüglich der einzusetzenden Verfahren. Für das Accounting fehlen zurzeit sowohl Lösungen als auch Ansätze.

Um die RSVP-Nachrichten einer Sitzung bzw. einem Sender zuordnen zu können, müssen die RSVP-Knoten nicht nur auf den IP-Header, sondern auch auf Nutzdaten zugreifen, und zwar um die Sender- bzw. Empfänger-Port-Adresse zu bestimmen. Um das zu gewährleisten, muss entweder die Fragmentierung der IP-Pakete unterbunden werden oder sie müssen hop-by-hop defragmentiert werden. Beim RSVP wird der erste Weg eingeschlagen. Dasselbe Problem taucht bei der Verschlüsselung auf den unteren Schichten des IP-Stacks auf, z.B. bei IP-Security (IPsec). Als Alternative zur Port-Adresse wird in diesem Fall der Security Association Identifier (IPsec SPI) statt der TCP/UDP-Port-Nummer verwendet. [BEOM97]

3.2.5 Differentiated Services (DiffServ)

Neben dem Ansatz der Integrated Services sind die Differentiated Services (DiffServ) der IETF entstanden, um die Probleme der IntServ zu lösen bzw.

über einen verbindungslosen Ansatz eine bessere Beherrschbarkeit bzw. Skalierung zu ermöglichen. Der Hauptunterschied zu IntServ besteht somit darin, dass man keine Signalisierung Ende-zu-Ende durchführt. Erste RFCs (RFC-2474, RFC-2475 und RFC-2598) für die Differentiated Services liegen bereits vor. Man möchte hier die bereits gewonnenen Erkenntnisse und Erfahrungen aus ATM einzusetzen, um IP-Traffic-Management-Funktionalität und -Dienstgüte zu implementieren. Im ersten Schritt wird dies durch eine neue Interpretation der TOS-Bits im IP-Header umgesetzt.

Das Konzept von DiffServ ist dabei, dass die Dienste in einige wenige QoS-Klassen unterteilt werden. Für jede so entstandene Dienstklasse wird ein Satz von Behandlungsregeln[55] definiert. Datenpakete, die in das Netz eintreten, werden im DS-Feld des Paketkopfes entsprechend ihrer Dienstklasse markiert und unter Berücksichtigung der dafür definierten PHB weiterverarbeitet. Mehrere unterschiedliche Verkehrsströme mit ähnlichen QoS-Anforderungen werden somit zu einem größeren Verkehrsbündel zusammengefasst (Aggregation), das im Netz auf gleiche Weise behandelt wird. Man spart sich somit die vielen Zustände und deren Verwaltung im Netz. Stattdessen wird die Vorverarbeitung des Verkehrs bzw. die Markierung der QoS-Klasse, das Policy Control und Traffic Shaping nur einmal, nämlich am Eingang in das DiffServ-Netz, vorgenommen. Der Ansatz DiffServ bringt somit eine völlig neue Sicht auf die QoS-Architektur und das Zusammenspiel einzelner Bereiche mit sich.

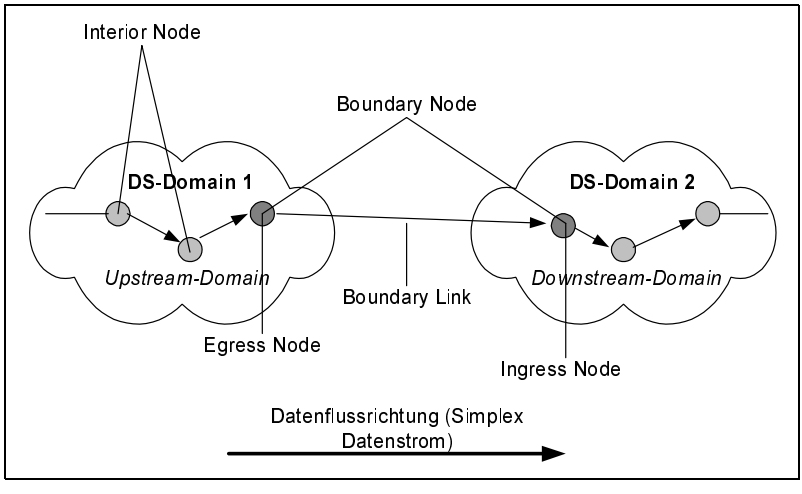


Abb. 3.15 DiffServ-Architektur

Im Mittelpunkt aller Betrachtungen steht eine administrative Einheit, die oft als DiffServ Cloud bzw. DS-Domain bezeichnet wird. Die administrativen Vereinbarungen und Abkommen müssen bei DiffServ eine übergeordnete Rolle spie-

55 Per-hop-Behavior (PHB)

len. Man will mit der Entwicklung dieses Modells keine einheitliche Behandlung des DiffServ-Verkehrs weltweit im Internet erzielen, sondern bildet vielmehr mehrere solcher DiffServ-Wolken, innerhalb deren feste Verarbeitungsregeln definiert sind. Darüber hinaus wird die Behandlung der Daten beim Eintritt bzw. beim Verlassen solcher Bereiche in bestimmten Service Level Agreements (SLAs) festgehalten und entsprechend auf die Datenströme angewendet. Diese Sichtweise liegt zwar der gegenwärtigen Internetverwaltung näher, ist im QoS-Bereich aber völlig neu. Alle Betrachtungen beziehen sich dabei wie auch bei den Integrated Services auf einen Simplex-Datenstrom. [SEUM01]

DS-Feld Als DS-Feld bezeichnet man in DiffServ das TOS-Feld des IPv4-Headers [POST81] bzw. das Traffic Class Oktett des IPv6-Headers [DEHI98], siehe Abb. 3.16. Dabei wird vorausgesetzt, dass dieses Feld nach den DiffServ-Regeln gesetzt wird. Die ersten 6 Bit des Bytes, die zur Differenzierung unterschiedlicher Flussaggregate eingesetzt werden,werden als DS-Codepoint bzw. abgekürzt DSCP bezeichnet. Die letzten zwei Bit bleiben zur Zeit unbenutzt. Anhand des DS-Codepoints, den ein Paket besitzt, entscheidet ein DS-Knoten, welche PHB darauf angewandt wird. Jedem DS-Codepoint entspricht genau ein PHB. Es können außerdem mehrere DSCP auf ein PHB abgebildet werden. Somit führt die Abbildung DSCP zu PHB auf eine Tabellenauswertung, bei der die DSCPs den Tabellenindex bilden. PHB kann dabei folgende Merkmale beinhalten:

- Drop Threshold
- Buffer Allocation
- Service Priority
- Service-Rate

Die Per-hop-Behavior (PHB) beschreibt somit das nach außen sichtbare Verhalten eines Netzknotens, das auf ein Aggregate angewandt wird. Dabei können als einzelne Komponenten des PHB QoS-Parameter wie Jitter, Paketverlustwahrscheinlichkeit oder Verzögerung angegeben werden. Die Komponenten des PHB können als relative oder absolute Größe angegeben werden[56]. Weiterhin werden relative Angaben zu einem anderen PHB gemacht[57]. Eine relative Angabe wird meistens verwendet, wenn eine PHB-Gruppe, die gemeinsam einen bestimmten Dienst bildet, definiert wird. Dabei können auch Regeln, anhand deren der Puffer geteilt wird, als PHB-Element angegeben werden. Es sollte beachtet werden, dass, obwohl die PHB mit Hilfe von Puffer-Management- und Paket-Scheduling-Verfahren realisiert wird, die PHBs nie als Parameter dieser Verfahren angegeben werden, sondern vielmehr als nach außen

56 Z.B. n-Prozent der Bandbreite der Schnittstelle oder eine Angabe in Byte/sec
57 Z.B. relative Prioritäten zueinander

sichtbare Parameter. Damit wird die Implementierungsfreiheit gewährleistet. Man legt also Grenzwerte fest und nicht Methoden, mit denen diese Werte eingehalten werden.

Bei der Definition des DS-Codepoints hat man versucht, eine möglichst weite Abwärtskompatibilität zum definierten TOS-Feld nach RFC-791 zu behalten. Im RFC-791 werden die drei Bit 0-2 als Prioritätsbits verwendet. Dabei entspricht das Setzen eines dieser Bits einer der drei definierten Prioritäten. Eine dieser Prioritäten hat per Definition nur netzweite Gültigkeit. Somit standen nur zwei Prioritäten netzweit zur Verfügung: Routing Traffic und Routine Traffic. Die tatsächliche Auswertung dieser Bits wurde allerdings im Internet nie implementiert und eingesetzt. Um eine Abwärtskompatibilität zu schaffen, sind einige Codepoints vordefiniert worden. Für diese so genannten Class Selector Codepoints (CSC) sind Mindestanforderungen für die entsprechende PHBs definiert worden. Die CSC haben den folgenden Wertebereich:

$$xxx000 \quad mit \quad x := \{0|1\}$$

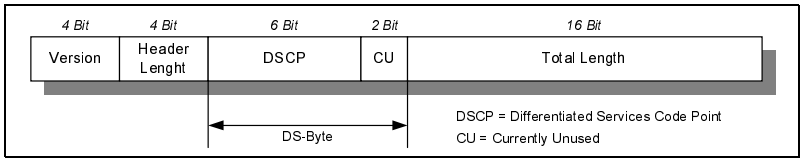


Abb. 3.16 Differentiated Service Byte Field

Neben der einstellbaren Priorität sollen dann auch Durchsatz, Verzögerung, Jitter und Paketverluste mit einbezogen werden. Zusätzlich müssen die Netzknoten bestimmte Funktionalitäten wie Paketklassifizierung, Forwarding und Verkehrszustände (Metering, Marking, Shaping, Policing) abdecken. DiffServ soll die Probleme bezüglich der Skalierbarkeit von den Ansätzen der Integrated Services kompensieren. Dies soll durch die Einführung einer komplexen Klassifizierung und Zustandsfunktionen an den Randknoten sowie durch PHB zum Zusammenfassen des Verkehrs bei IPv4 und IPv6 über das DS-Feld erfolgen. PHB wurden spezifiziert, um eine feine Unterteilung der zugewiesenen Puffer- und Bandbreiten-Ressourcen an jedem Netzknoten vornehmen zu können. Die Bereitstellung des Dienstes und die Verkehrszustände sind dabei völlig unabhängig von den Forwarding-Funktionen.

Eine Unterteilung ist gewährleistet durch: [NBBB98]

- Zusammenfassen des Verkehrs durch den Service
- Gesetzte Funktionen und PHB für die Umsetzung der Services
- Wert des DS-Feldes (DSCP) für die Markierung der IP-Pakete, um PHB zu selektieren.
- Eigener Knotenmechanismus, welcher das PHB realisiert.

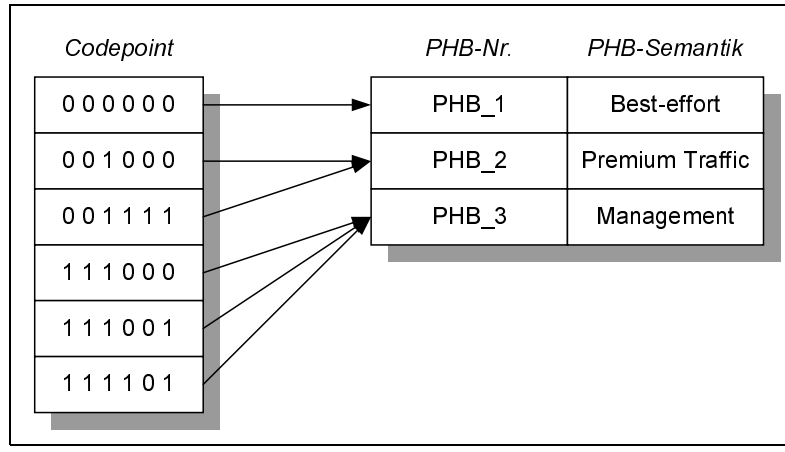


Abb. 3.17 Abbildung der Codepoints auf PHBs

Der Code Point 000000 ist ein Standard Code Point, der von allen DiffServ-Knoten erkannt werden muss. Das zugehörige PHB entspricht dem Verhalten Best-effort. Die ersten 3 Bit des DSCP werden vor den restlichen hervorgehoben. Sie charakterisieren eine DS-Klasse. Es können DiffServ-Domänen existieren, in denen nur die ersten 3 Bit des DSCP ausgewertet werden. Anschließend werden alle DS-Code-Points als Class Selector Code Point (CSC) ausgewertet. Die Bit 3-5 bedeuten entsprechend die relative Priorität innerhalb der Klasse. Es ist zu beachten, dass die Class Selector Code Points immer auf die PHBs, die die untere Grenze der Dienstgüte innerhalb der Klasse bedeuten, abgebildet werden. Weiterhin werden Code Points mit einem höheren numerischen Wert einfach größer bezeichnet und umgekehrt. Bei der Wahl der Code Points müssen folgende Bedingungen erfüllt werden:

- Für die acht CSC muss es mindestens zwei unabhängige PHBs geben. Es müssen also mindestens 2 Datenstromklassen existieren.
- Ein numerisch größerer CSC muss auf ein PHB, der einen genauso guten oder besseren Dienst bereitstellt, abgebildet werden. Mit der Bezeichnung „besserer Dienst“ ist eine höhere Wahrscheinlichkeit dafür gemeint, dass bei sonst gleichen Verhältnissen ein zugehöriges Datenpaket rechtzeitig verarbeitet wird.
- Außerdem muss ein PHB, der dem CSC 11x000 entspricht, gegenüber den PHBs, auf die der Code Point 000000 abgebildet wird, eine vorteilhafte Verarbeitung und Weiterleitung der zugehörigen Pakete erfahren. Das stellt sicher, dass die Pakete mit Routing-Daten nach RFC-791 vor den Best-effort-Daten bevorzugt werden.
- Pakete, die mit unterschiedlichen CSC markiert werden, werden gemäß der zugehörigen PHB direkt weitergeleitet. Werden also Pakete eines Micro-

flows[58] mit unterschiedlichen CSCs markiert, so können sie in falscher Reihenfolge am Ziel ankommen.

Der Wertebereich des DSCP beinhaltet 64 Werte und ist deshalb in drei Bereiche unterteilt:

- 32 Werte, die standardisiert und in Netzen für den allgemeinen Einsatz implementiert werden können.
- 16 Werte, die für experimentelle Zwecke bzw. zum lokalen Einsatz verwendet werden.
- 16 Werte, die für den Fall reserviert worden sind, dass der erste Wertebereich ausgeschöpft wird.

Der Codepoint-Raum des Wertebereichs DSCP ist dabei wie folgt unterteilt: [NBBB98]

- Bereich 1: xxxxx0
- Bereich 2: xxxx11
- Bereich 3: xxxx01

Architektur

Eine DiffServ-Domäne (DS-Domain) ist eine Menge benachbarter DS-Knoten. Normalerweise besteht eine DS-Domain aus einem oder mehreren Netzen, die gemeinsam administriert werden, wie beispielsweise von einem Unternehmen oder einem Internet Service Provider (ISP). Die innerhalb der DS-Domain verfügbaren Dienste können über Service Level Agreements (SLA) angeboten und abgestimmt werden. Dazu werden die Dienstparameter für einen entsprechenden Datenstrom in einer so genannten Service Level Specification (SLS) zusammengefasst. Ein Bestandteil davon ist die Traffic Conditioning Specification (TCS), in der die einzelnen Klassifizierungsregeln und Verkehrsprofile beschrieben sind. Wenn allerdings in einer Domäne ein nicht DS-fähiger Knoten vorhanden ist, so kann nicht gewährleistet werden, dass die mit dem Dienstnutzer vereinbarten SLAs eingehalten werden können. Dies muss deshalb aus Sicht eines Service Providers unbedingt vermieden werden.

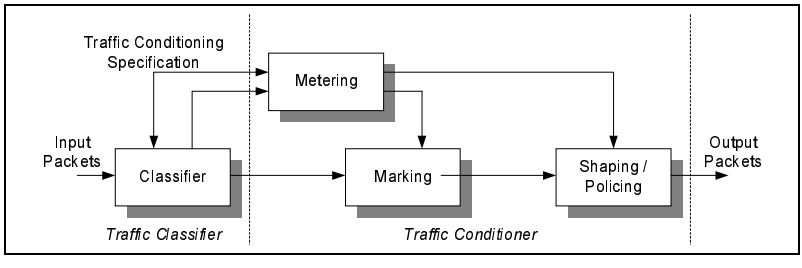


Abb. 3.18 Traffic Classifier und Traffic Conditioning

58 Eine einzelne Instanz eines Ende-zu-Ende-Datenstroms, gekennzeichnet durch die Sender IP-Adresse und Port-Nummer, Empfänger-IP-Adresse und Port-Nummer und die IP-Protokoll-ID

Bei Eintritt des Datenstroms in eine DS-Domain werden die SLAs überprüft. Weiterhin muss der DS-Codepoint gesetzt werden, falls er nicht schon vor dem Eintritt gesetzt wurde. Ist in den SLA vereinbart, dass der Datenstrom vor dem Eintritt markiert wird (Pre-marking[59]), so muss die Verkehrscharakteristik überprüft und bei Bedarf mit einem Pre-marking versehen werden. Unter Umständen können die Pakete am Eingang auch verworfen werden. Diese Aufgaben übernimmt die Einheit Traffic Classification und Traffic Conditioning (siehe auch Abb. 3.18). Es ist zu beachten, dass nicht zwangsläufig alle einzelnen Komponenten des Traffic Conditioner implementiert werden müssen.

Die einzelnen Prozesse zur Klassifizierung und Veränderung des Datenstroms nach Abb. 3.18 werden wie folgt unterteilt: [SEUM01]

- **Classifier:** Ankommende Datenpakete werden durch die Paketkopfinformationen und entsprechend den definierten Regeln den einzelnen Datenströme zugeordnet.
- **Metering:** Die temporär vom Classifier vergebenen Eigenschaften eines Datenstroms (max. Paketrate) werden gemessen. Dies beeinflusst direkt die Aufgabe des Marking und Shaping.
- **Marking:** Die Datenpakete werden durch das Setzen des DSCP auf Grundlage der definierten Regeln markiert. Dies wird mittels Pre-marking und Re-marking[60] durchgeführt.
- **Shaping:** Die Pakete eines Datenstroms werden manuell verzögert, um die Einhaltung der TCS gewährleisten zu können.
- **Policing:** Die Pakete eines Datenstroms werden verworfen, um die Einhaltung der TCS gewährleisten zu können.

Eine Domäne besteht aus Grenzknoten und internen Knoten. Alle DS-Knoten müssen die ankommenden Pakete bezüglich der definierten PHBs weiterleiten. Weiterhin müssen Grenzknoten die ankommenden Datenpakete nach den vereinbarten TCAs prüfen bzw. mit Hilfe eines Traffic Conditioner (TC) verarbeiten. Ein Ingress Node muss sicherstellen, dass der einlaufende Datenverkehr dem zwischen den Domänen vereinbarten TCA entspricht. Sie kennen somit die vereinbarten SLS- und TCS-Spezifikationen. Im Gegensatz dazu muss ein Egress Node sicherstellen, dass der ausgehende Datenverkehr dem mit der Nachbardomäne vereinbarten TCA entspricht. Dafür kann auf den ausgehenden Verkehr auch Traffic Conditioning angewandt werden. Ein interner Knoten[61] kann optional ein Re-marking durchführen. Das kann an kritischen Stellen der DS-Domain eingesetzt werden, beispielsweise beim Eintritt auf eine transozeanischen Verbindung zur Glättung des Datenburst-Aufkommens.

59 Setzen des DSCP der weitergeleiteten Datenpakete vor dem Verlassen einer DS-Domäne
60 Ändern des DSCP zur Erfüllung der TCS
61 Interior Node

Das Architekturmodell von DiffServ funktioniert so, dass ein Paket, welches ein DiffServ-Netz erreicht, einen Classifier passiert, in dem es einem bestimmten Aggregat zugeordnet wird. Das Resultat wird als Behavior Aggregate (BA) bezeichnet. Ankommende Pakete werden mittels des Traffic Classifier modifiziert. Dabei muss man zwischen zwei Klassifizierungsarten unterscheiden:

- **Behavior Aggregate (BA) Classifier:** Mit Hilfe des BA-Classifier werden alle Aggregates klassifiziert. Diese Zuordnung findet nur anhand des DSCP statt. Eine solche Klassifizierung wird beim Eingang des Datenstroms aus einer anderen DS-Domäne vorgenommen.
- **Multifield (MF) Classifier:** Durch den MF-Classifier wird der Datenstrom anhand mehrerer Felder des IP-Headers klassifiziert.

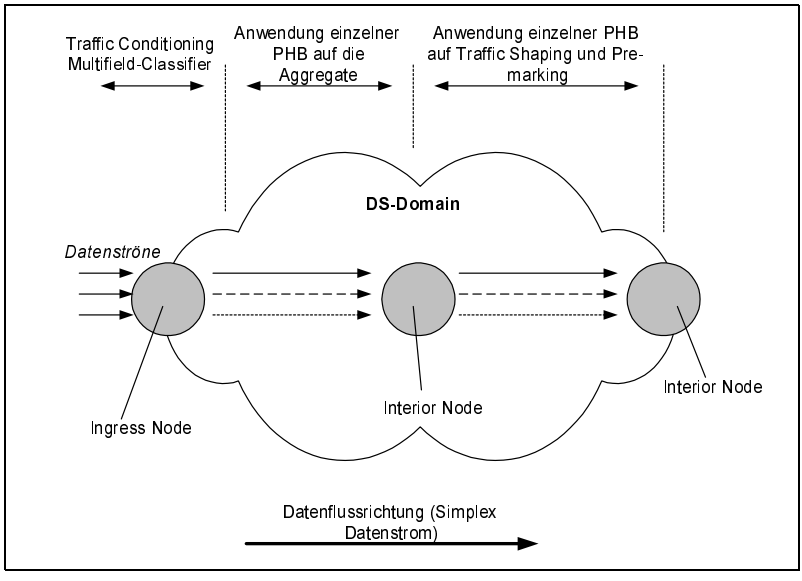


Abb. 3.19
DS-Domainarchitektur

Zur Aufgabe des Traffic Classifiers gehört auch die Authentisierung einlaufender Datenpakete. Er wird bei dem Ansatz von DiffServ genauso wie bei IntServ verwendet. Ein Dropper ist ein Grenzfall eines Shapers. Hier wird die Puffergröße auf Null bzw. auf einen sehr kleinen Wert gesetzt.

Vom Classifier wird es an den Traffic Conditioner weitergereicht. Spätestens am Ausgang des Conditioners wird das Datenpaket mit einem DS-Codepoint versehen, mit dem es durch die DS-Domäne weitergeleitet wird. Jeder DS-Knoten prüft bei dessen Empfang den DSCP und leitet das Paket gemäß dem zugehörigen PHB und der Routing-Tabelle weiter. Die Zuordnung DSCP zu PHB wird anhand der global definierten Mapping-Tabelle oder anhand der in der DS-Domäne geltenden lokalen Mapping-Regeln durchgeführt.

Traffic Conditioner und MF-Classifier können innerhalb einer Source-Domain oder an einem Edge-Router eingesetzt werden. Die Source-Domain enthält Knoten, die den Datenverkehr erzeugen. Es kann hier vereinbart werden, dass diese Knoten beim Aussenden der Daten auf dem Interface den DS-Codepoint auf den gewünschten Wert setzen. Dieses Verfahren wird als Pre-marking bezeichnet. Obwohl diese Pakete beim Eintreten in die DS-Domäne noch einmal durch den Traffic Conditioner müssen, kann es durchaus sinnvoll sein, das Pre-marking durchzuführen. Dies wird aus zwei Gründen vorgenommen:

- Es können dem Quell-Knoten bestimmte Grenzen für unterschiedliche QoS-Klassen zugewiesen werden. In diesem Fall kann der Quellhost bzw. die Anwendung am besten entscheiden, welche Datenpakete mit welcher Priorität gesendet werden sollen. Dann würde am Eingang in die DS-Domain lediglich geprüft, ob die gesamte SLA eingehalten wird, und wenn das zutrifft, die Markierung einfach übernommen werden.
- Sendet ein Knoten auf mehreren Ports gleichzeitig, was besonders häufig der Fall sein würde, wenn ein ganzes Netz als Quelldomäne agiert, so wird beim Einlauf in die DS-Domäne nicht mehr auf der Microflow-Basis, sondern vielmehr auf Aggregat-Basis markiert. Damit wird Rechenleistung der Router am Rande des DS-Netzes eingespart.

Im Edge-Router hin zu einer anderen DS-Domain bzw. an beiden Seiten des Boundary Link[62] kann ebenfalls ein Traffic Conditioner und MF-Classifier eingesetzt werden. In den SLA zwischen zwei Domänen muss dabei vereinbart werden, ob die Upstream- oder die Downstream-Domäne bzw. beide Domänen für des Traffic Conditioning zuständig sind. [BBCD98]

Service Level Agreement (SLA)

Damit eine DS-Domain gebildet und erfolgreich verwaltet wird, ist es wichtig, dass sie tatsächlich als eine administrative Einheit existiert. Trotzdem möchte man eine bestimmte Dienstgüte auch außerhalb dieser Domäne bereitstellen. Dafür werden aus einzelnen Domänen DiffServ-Regionen gebildet. Eine solche Region ist eine Zusammensetzung mehrerer DS-Domains, innerhalb welcher ein einheitlicher DS-Dienst angeboten wird. Ein solcher Dienst kann demnach nur auf der Basis mehrerer Vereinbarungen zwischen einzelnen Domänen gut funktionieren. Diese Vereinbarungen umfassen sowohl wirtschaftliche, kommerzielle als auch technische Details der Interoperabilität zwischen einzelnen Domänen. Diese Vereinbarungen werden allgemein als SLA[63] bezeichnet. Hier stehen die technischen Aspekte dieser Vereinbarungen im Vordergrund, die auch SLS[64] genannt werden. Ähnlich zur Beschreibung der PHB, kann ein Service in absoluten Werten der QoS-Parameter angegeben werden, z.B. maximale

62 Ein Link bei z.B. DiffServ, der zwei angrenzenden Knoten aus benachbarten Domänen verbindet.

63 Service Level Agreement

64 Service Level Specification

Jitter, maximale Verzögerung usw. Dabei spricht man von einem quantitativen Service. Legt man lediglich eine relative Priorität der Datenströme fest, so spricht man von einem qualitativen Service.

Eine wichtige Untermenge der SLS bilden die Traffic Conditioning Agreements (TCA). TCA spezifiziert die Serviceparameter für jede definierte Dienstklasse[65]. Sie muss Folgendes enthalten:

- Detaillierte Service-Performance-Parameter wie Durchsatz (verfügbare Bandbreite), Verzögerung und Verlustwahrscheinlichkeit[66]
- Verkehrsprofile für einen Dienst
- Gültigkeitsbereiche (Scope) für einen Service
- Behandlung der Pakete, die den vereinbarten Profil überschreiten
- Eine Vereinbarung, wo und wer das Marking und Shaping durchführt

Optional kann man weitere Merkmale in den SLAs vereinbaren:

- Verfügbarkeit und Erreichbarkeit für den Dienst
- Einsatz von Verschlüsselungsverfahren
- Routing-Einschränkungen
- Mechanismen zum Monitoring und Auditing
- Verantwortung, falls einige Parameter nicht eingehalten werden
- Preise und Abrechnungsmechanismen

Um zu erkennen, ob eine Regel eingehalten wird, werden so genannte Verkehrsprofile eingeführt. Diese legen fest, ob ein Datenpaket als profilkonform oder nicht profilkonform eingestuft werden kann. Das kann beispielsweise folgendermaßen aussehen:

$$DSCP = X; \quad TokenBucket = \{r, b\}$$

Dies kann so interpretiert werden, dass alle Pakete mit dem Codepoint X mit einem Token-Bucket-Filter mit den Parametern r und b gefiltert werden müssen. Ein Paket wird dabei als nicht profilkonform behandelt, falls dafür keine Token im Bucket vorhanden sind. Sind in einem TCA Verkehrsprofile vereinbart worden, so müssen auch Regeln zur Behandlung der dazu konformen und nicht konformen Pakete vereinbart werden. Es kann z.B. vereinbart werden, dass alle nicht zum Verkehrsprofil konformen Pakete verworfen werden oder ihr DSCP auf den Default-Codepoint gesetzt wird. Die profilkonformen Pakete können beispielsweise unverändert in die Ausgangswarteschlange weitergereicht werden oder es kann deren Drop-Precedence heruntergesetzt werden.

65 genauer, für jede PHB

66 Ähnlich den Paketverlustbetrachtungen bei IntServ wird hier als Paketverlustwahrscheinlichkeit lediglich ein Verlust infolge eines Überlaufs der Puffers bzw. infolge einer Filterung mit Hilfe eines Dropper (Policing) betrachtet.

Der vereinbarte Dienst muss nicht zwangsläufig auf alle Daten des Kunden, die am betrachteten Eintrittspunkt in die DS-Domain einlaufen, angewandt werden. Es kann in den TCA der Gültigkeitsbereich[67] für den Dienst vereinbart werden. Das DiffServ-Rahmenwerk legt folgende Beschreibung für den Gültigkeitsbereich fest: [NICA01]

1. Der gesamte Datenverkehr vom Einlaufknoten zum beliebigen Austrittsknoten
2. Der gesamte Datenverkehr vom Einlaufknoten zum Austrittsknoten
3. Der gesamte Datenverkehr vom Einlaufknoten zu mehreren explizit aufgelisteten Austrittsknoten etc. Hierbei wird ein Zielhost innerhalb der DS-Domain genauso als Austrittspunkt aus der DS-Domain gesehen.

DiffServ-Dienste

Die PHBs, die innerhalb einer DS-Domain implementiert sind, besitzen eine lokale Bedeutung. Die Interoperabilität zwischen mehreren DS-Domains basiert dabei auf den SLAs, die zwischen den beiden Kommunikationsteilnehmern ausgemacht wurden. Aus diesem Grund ist es für die DiffServ-Realisierung nicht unbedingt notwendig, spezielle PHB im weltweiten Internet zu standardisieren. Allerdings kann sich die Interoperabilität zwischen den einzelnen DS-Domains stark verbessern, wenn ähnliche PHBs festgelegt wurden. Ein weiterer Vorteil wäre, dass größere DS-Regionen mit demselben Dienst abgedeckt werden können. Es bleibt dabei zu beachten, dass ein Netzknoten dem DiffServ-Referenzmodell entsprechen muss, um DiffServ-fähig zu sein.

Aus diesem Grund gibt es zwei Spezifikationen[68] einer PHB bzw. einer PHB-Group, die von der IETF[69] herausgegeben wurden. Eine Spezifikation eines Dienstes innerhalb des DiffServ besteht aus einer Beschreibung der PHB bzw. PHB-Group und der Festlegung der entsprechenden DSCP[70]. Die zwei definierten Dienste sind Expedited Forwarding PHB (EFP) und Assured Forwarding PHB Group (AFPG). Beides sind qualitative Dienste. Die Expedited Forwarding PHB soll ähnlich zum Controlled Load Service lediglich einen besseren Dienst als den Best-effort anbieten. Er wird oft auch als Premium Service bezeichnet. Die Assured Forwarding PHB Group bietet eine stärkere Differenzierung der Datenströme und eine höhere Dienstgüte.

Expedited Forwarding PHB (EFP)

Das Expedited Forwarding PHB (EFP) wurde in der RFC-2598 spezifiziert und wird dazu verwendet, einen so genannten Premium Service zur Verfügung zu stellen. Es wird ein Ende-zu-Ende-Dienst mit geringen Paketverlusten, Verzögerungszeiten und Jitter mit zugesicherter Bandbreite über eine DS-Domain geliefert. Dabei sichert EFP allerdings lediglich die Datenrate für die weiterge-

67 Scope-of-Service
68 RFC-2598 und RFC-2597
69 Internet Engineering Task Force
70 Differentiated Services Code Point

leiteten Daten zu. Die zustande gekommene so genannte Punkt-zu-Punkt-Verbindung wird als Virtual Leased Line bezeichnet und durch absolute Priorität gegenüber den restlichen Daten erreicht. Der Durchsatz für EFP-Daten muss unabhängig von der Belastung auf der Verbindung gewährleistet werden. Das Traffic-Profil wird als Token-Bucket-Parameter angegeben. Die zugesicherte Datenrate (bzw. auch die Burst-Größe) muss vom Netzadministrator einstellbar sein.

Bekommen die EFP-Daten eine hohe Priorität gegenüber den restlichen Datenströme, z.B. durch einfaches Priority Queueing, so muss sichergestellt werden, dass die Best-effort-Daten nicht vollständig verdrängt werden. Kommen EFP-Pakete an, die die vereinbarten Parameter überschreiten, so müssen sie verworfen werden. Das Verwerfen dient hierbei nicht nur dem Schutz des Best-effort-Verkehrs, sondern stellt auch sicher, dass kein Überholen der Pakete stattfindet! Innerhalb der DS-Domain dürfen die markierten EFP-Pakete auf andere Dienstarten ummarkiert werden. Der Codepoint für den EFP ist 101110, woran man ihn erkennen kann. [JNP99]

Die Queueing-Mechanismen sind bei EFP und AFPG entscheidend. Diese sind in der Regel innerhalb eines Routers implementiert, um Verzögerungszeiten und Jitter möglichst gering zu halten. Hierbei sind in erster Linie die Methoden wichtig, die die internen Warteschlangen (Queues) optimieren, um die Eingangs- und Ausgangsspeicher zu optimieren. Eine Möglichkeit, die Auslastung der Eingangsspeicher zu optimieren, ist das Verfahren Random Early Discard (RED), mit dem man den Durchsatz unter Überlast optimieren kann. TCP[71] nutzt einen Rate-Control-Mechanismus, um eine kontinuierliche Netzlast zu erzeugen. Hierfür wird angenommen, dass ein Sender ein Datenpaket auf das Netzwerk weiterleitet und zur selben Zeit ein Empfänger ein Datenpaket vom Netzwerk entfernt. Sobald der Empfänger signalisiert, dass eine Überlastsituation vorhanden ist, muss der Sender die Sendedatenrate verkleinern. Dieses Verfahren wird durch den Slow-Start-Algorithmus für eine exponentiell steigende Datenrate und durch den Mechanismus Congestion Avoidance zur Minimierung der Datenrate durch Halbierung der TCP Window Size ermöglicht. RED wirkt dem TCP-Slow-Start entgegen, indem ein zufällig ausgewähltes Paket am Eingangs-Port verworfen wird. Durch das Setzen eines Schwellwerts wird nun verhindert, dass der entsprechende Warteschlagenspeicher überläuft. Gleichzeitig kommt es zu einer Halbierung der Window Size aller TCP-Datenflüsse. [FEHU98]

Dieser RED-Mechanismus bedeutet aber auch, dass umso mehr Pakete verworfen werden, je stärker die Warteschlangen (Queues) gefüllt werden. In leistungsfähigen Routern kommt es hier auf die Speichertiefe pro Queue an. Eine Erweiterung von RED ist das Weighted Random Early Detection (WRED).

71 Transmission Control Protocol

Hier können bestimmte Prioritäten beim Verwerfen der Pakete vorgegeben werden. Zusätzlich werden den Prioritäten innerhalb einer Warteschlange entsprechend minimale und maximale Speichertiefen zugeordnet.

Das Rate Limiting ist ein weiteres Verfahren, um den Datenverkehr kontrollieren zu können. Man kann dadurch pro Port und Datenfluss bestimmte Bandbreiten zuweisen, die nicht überschritten werden dürfen (ähnlich einer Peak Cell Rate bei ATM). Dadurch können Überlastsituationen bereits im Vorfeld vermindert werden, wodurch das Überlaufen des Eingangs- und Ausgangsspeichers vermieden wird. In einem verbindungslosen, paketvermittelnden Netzwerk werden die Pakete einem bestimmten Datenfluss zugeordnet und an entsprechende Ausgangsspeicher einer Schnittstelle weitergegeben. Diese Übertragungsart wird als Link Scheduling Discipline bezeichnet und ist entscheidend für die Zuweisung von QoS in IP-Netzwerken.

Normalerweise arbeiten Router nach dem Prinzip des First-In-First-Out (FIFO). Dabei werden die Pakete in derselben Reihenfolge weitergeleitet, in der sie eingetroffen sind. Deshalb muss ein Paket, welches auf einem Ausgangsinterface auf das Forwarding wartet, warten, bis alle Pakete, die ebenfalls über diese Schnittstelle weitergeleitet werden wollen, abgearbeitet sind. Wenn die Warteschlange überfüllt ist, müssen die Pakete verworfen werden. Mechanismen wie RED oder Rate Limiting können dem entgegenwirken, es aber nicht gänzlich verhindern.

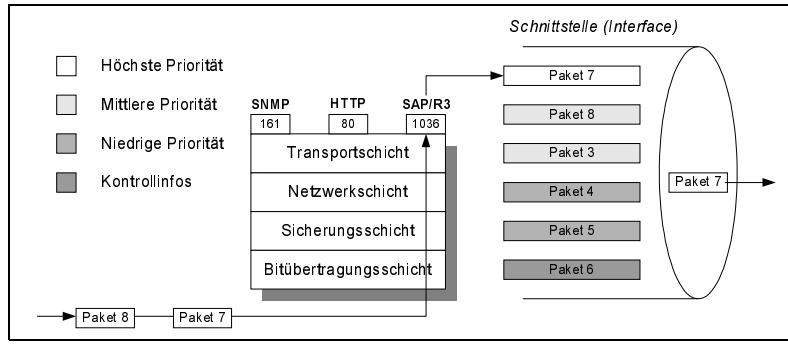


Abb. 3.20 Strict Priority Queueing (SPQ)

Damit eine Differenzierung des Datenverkehrs ermöglicht werden kann, sind weitere Varianten notwendig. Dazu werden die Datenpakete entsprechend den eingestellten Prioritäten vorgeordnet und anschließend an bestimmte Ausgangsspeicher weitergeleitet. Dabei hat idealerweise jede Prioritätsklasse ihre eigene Queue. Die Abarbeitung dieser Queues erfolgt dann wiederum nach definierten Prioritäten (von der höchsten zur niedrigsten Priorität). Dadurch kann man selbst bei höheren Lastbedingungen die Weiterleitung der Pakete gewährleisten, ohne Verluste notwendigerweise in Kauf zu nehmen. Das Verfahren wird als Strict Priority Queueing (SPQ) bezeichnet.

Ein alternatives Verfahren zur Verteilung der Datenpakete nach Prioritäten ist das Weighted Fair Queueing (WFQ). Hierfür werden die Datenflüsse im ersten Schritt ebenfalls verschiedenen Prioritätsklassen zugeordnet. Durch WFQ besteht nun die zusätzliche Möglichkeit, die verfügbare Bandbreite pro Schnittstelle aufzuteilen. Der Vorteil gegenüber SPQ ist, dass jede Priorität eine garantierte Mindestbandbreite auf der Ausgangsschnittstelle bekommt, die bei niedriger Last auch überschritten werden kann. Das Expedited-Forwarding-Verhalten kann mit Hilfe des Priority Queueing (PQ) oder Weighted Fair Queueing (WFQ) realisiert werden. Der Warteschlange muss dabei ein Token Bucket vorgeschaltet werden, der die Funktionen einer Policy-Einheit übernimmt. [FEHU98]

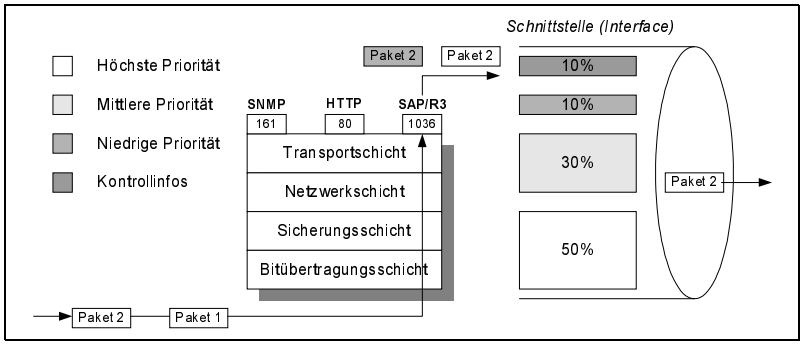


Abb. 3.21 Weighted Fair Queueing (WFQ)

Assured Forwarding PHB Group (AFPG)

Die Assured Forwarding PHB Group (AFPG) wurde in der RFC-2597 spezifiziert und definiert vier unabhängige AFPG-Klassen. Innerhalb dieser Klassen gibt es drei Stufen zum Verwerfen des Pakets, um jedem Datenpaket eine relative Verlustwahrscheinlichkeit (Drop Precedence) zuzuweisen. Die Drop Precedence (DP) stellt den Vorrang dar, mit dem die Paktete bei Überlast im Netz verworfen werden. Die AFPG wird auch als Olympic Service bezeichnet, da man die unterschiedlichen Klassen mit den Bezeichnungen Gold, Silver, Bronze beschreibt.

Jeder DS-Knoten soll sicherstellen, dass Datenpakete eines Microflows ihre Reihenfolge behalten, und zwar unabhängig davon, ob diese Pakete das vereinbarte Traffic Profil einhalten oder nicht. Es soll demnach eine Überschreitung des vereinbarten Traffic Profile zugelassen werden können. Dabei kann aber eine geringere Dienstgüte gewährleistet werden. Durch die unterschiedliche Konfiguration der einzelnen Hops innerhalb einer DS-Domain kann es daher im Gegensatz zu EFP allerdings zu nicht vorhersagbaren Verzögerungen und Jitter kommen.

Jede AFPG-Klasse legt einen bestimmten Anteil der Ressourcen im Knoten[72] fest. Die Pakete werden entweder beim Kunden oder beim Dienstanbieter mit der vom Nutzer gewünschten Klasse und der gewünschten DP vormarkiert. Diese Vormarkierung könnte vom Dienstanbieter z.B. als Mehrwertdienst angeboten werden. Kommt es zur Überlast im Netz, so werden Pakete mit höherem Wert der Drop Precedence zuerst verworfen. Die Wahrscheinlichkeit (Sicherheit), mit der ein Paket in einem Knoten weitergeleitet wird, hängt von folgenden Faktoren ab:

- Wie viele Ressourcen im Netz von der AFPG-Klasse definiert worden sind.
- Wie hoch die momentane Belastung im Netz ist.
- Mit welcher Verlustwahrscheinlichkeit das Paket markiert ist.

Tab. 3.5 DSCP-Werte für die Dienstklassen von AFPG

Drop Precedence	Klasse AF1	Klasse AF2	Klasse AF3	Klasse AF4
Niedrige DP	001010	010010	011010	100010
Mittlere DP	001100	010100	011100	100100
Hohe DP	001110	010110	011110	100110

Ein Paket, welches zu einer AFPG-Klasse nach Tab. 3.5 gehört, wird mit dem DSCP Af_{ij} markiert. Dabei bezeichnet i ($1 \leq i \leq 4$) die AFPG-Klasse und j ($1 \leq j \leq 3$) die Drop Precedence. Der Wertebereich kann für die lokale Nutzung erweitert werden. Es ist allerdings kein Merging mehrerer Klassen möglich, weshalb die Daten jeder AFPG-Klasse unabhängig voneinander weitergeleitet werden müssen. Ein DS-Knoten muss innerhalb jeder Klasse alle drei DSCP für die Drop Precedence zulassen. Innerhalb der Knoten müssen wiederum mindestens zwei unterschiedliche Drop Precedences realisiert werden. Es werden keine quantitativen Werte für die Verzögerung bzw. den Jitter festgelegt. Die AFPG-PHB kann zur Implementierung eines Ende-zu-Ende-Dienstes bzw. für einen Service zwischen zwei Domänengrenzen eingesetzt werden. Am Ingress Node wird mit Hilfe eines Traffic Conditioners der Datenverkehr geprüft und die Verlustwahrscheinlichkeit herauf- bzw. heruntergesetzt. Diese Aktionen müssen auch sicherstellen, dass keine Vertauschung der Reihenfolge der Pakete stattfinden kann.

Es werden im Schnitt härtere Anforderungen an das Queueing-Mangagement bei der Implementierung des AFPG-Service als beim EFP-Service gefordert. Dies betrifft zum einen die dauerhafte Belastung, deren Zeispannen die Burst-Time übertreffen. Das Verwerfen von Paketen wird dabei dadurch ver-

72 Z.B. Puffer und Bandbreite auf der Netzwerkschnittstelle

mieden, dass kurzzeitige Bursts in den Traffic Shapern geglättet werden. Weiterhin muss die Filterung bei dauerhaften Überlastungen unabhängig von der Burstiness erfolgen. Wenn also zwei Microflows mit unterschiedlicher Burstiness ankommen, so muss die Wahrscheinlichkeit für das Verwerfen der Pakete für beide Microflows identisch sein. Ebenfalls darf die Häufigkeit des Droppens von Paketen nicht sprunghaft, sondern nur kontinuierlich sein, wenn sich die Überlast im Netz erhöht. Als passendes Warteschlangen-Management-Verfahren wird hierfür RED eingesetzt. [HBWW99]

Security

DiffServ muss ebenso wie IntServ bestimmte Sicherheitsmerkmale unterstützen. Da beide Ansätze Daten im Netz bevorzugen können, muss man davon ausgehen, dass Versuche unternommen werden, dies zu manipulieren. Bei DiffServ bedeutet dies, dass der DSCP verfälscht werden könnte, um die Dienste zu beeinflussen bzw. nutzen zu können.

Um DiffServ sicher einzusetzen, kann man aber die vorhandenen Mechanismen der Spezifikation ausnutzen. Dabei muss jeder Knoten einer DS-Domain, der Daten aus anderen Domänen sowie von Hosts empfängt, als Ingress Node betrachtet werden. Ein Ingress Node muss bei allen einkommenden Datenpaketen sicherstellen, dass die DSCP gemäss der TCA und der Service Provisioning Policy gesetzt sind. Im Zweifelsfall wird der Codepoint auf den Standard-Codepoint gesetzt. Es kann für Daten mit einigen DSCP[73] eine besondere Authentisierung (z.B. mittels IPsec) verlangt werden. Ein Link, der nicht gegen Manipulationen des Codepoint sicher ist oder auf dem Dateninfluenzierung möglich ist, wird als Boundary-Link betrachtet. Folglich müssen alle Daten, die auf einem solchen Link ankommen, wie Daten in eine DS-Domäne ankommend behandelt werden.

Die Verantwortung für das Setzen des DSCP und das Prüfen der Policies kann in den SLA dem Datenerzeuger auferlegt werden. Das kann vor allem an der Grenze zwischen zwei DS-Domänen sinnvoll sein. Dabei kann im Ingress Node lediglich die Entsprechung der gesamten Aggregate der TCAs überprüft werden.

Bei Verwendung von IPsec und Tunneling kann weiterhin die BA-Classification auf den Tunnel- bzw. IPsec-Datenstrom angewendet werden. Das TOS- bzw. TCO-Byte des IP-Header bleibt dabei unverändert. Eine MF-Classification ist aber nicht möglich, da die Port-Nummern und ggf. die IP-Adressen im Netz nicht zugänglich sind. Man muss also eine MF-Classification vor dem Eingangspunkt in einen Tunnel durchführen. Beim Einpacken des Headers ist darauf zu achten, dass der äußere Header mit dem richtigen DSCP versehen wird. [SIEM99]

73 z.B. für besonders bevorzugte Daten

Ressourcenzuweisung

Ein wichtiger Aspekt in der DiffServ-Konfiguration ist die Ressourcenzuweisung in den Netzknoten. Um DiffServ zu realisieren, muss in jedem Netzknoten eine getrennte Queue pro Datenstromaggregat eingerichtet werden. Dabei müssen für jede Warteschlange die Parameter des entsprechenden Packet-Scheduling-Verfahrens, beispielsweise durch die Gewichtung der Prioritäten beim Einsatz von WFQ[74], so justiert werden, dass die in den SLA definierte Vereinbarungen für die einzelnen Datenflüsse eingehalten werden können. Dafür werden in den Ingress Nodes zusätzliche Traffic Conditioner eingesetzt. In den Egress Nodes sowie in bestimmten Bereichen innerhalb der DS-Domain kommen Traffic Shaper zum Einsatz. Bei der DiffServ-Realisierung werden drei unterschiedliche Stufen der Ressourcenzuweisung verwendet:

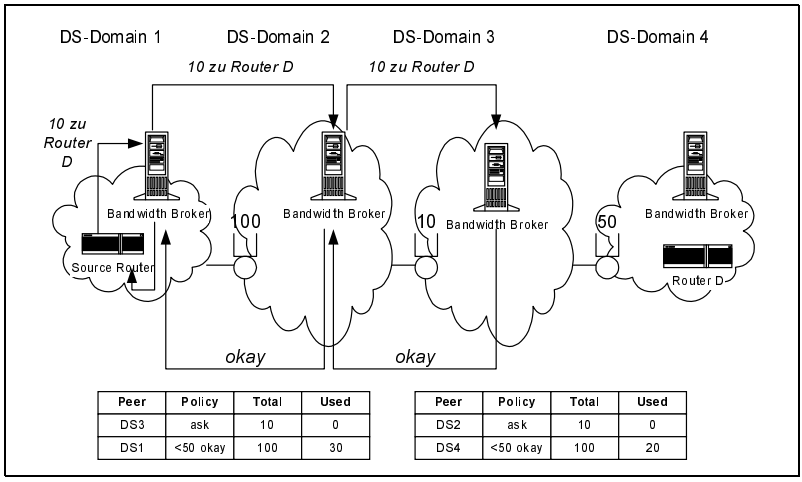
1. Statische Zuweisung ohne Bandwidth Broker
2. Statische Zuweisung mit Hilfe von Bandwidth Broker
3. Dynamische Zuweisung mit Hilfe der Bandwidth Broker

Die statische Zuweisung ohne Bandwidth Broker ist für die erste DiffServ-Realisierung in bereits bestehenden Netzen geeignet. Dabei müssen alle Netzwerkschnittstellen sowie die Traffic Shaper und Traffic Conditioner manuell vom Netzadministrator konfiguriert werden. Der Netzadministrator darf dabei nicht den Überblick über das gesamte Geschehen im Netz verlieren. Aufgrund der manuellen Konfiguration und der fehlenden dynamischen Anpassung an die aktuelle Belastung ist dieses Verfahren auf kleine Netze beschränkt. Anhand der IP-Adressen und Port-Nummern der Datenpakete entscheiden Multifeld-Klassifizierer am Eingang in das DiffServ-Netz, welche Klassifizierungsregeln auf den einlaufenden Datenstrom angewendet werden. Man ist somit nur in der Lage Teilnehmer mit statischen IP-Adressen sowie Anwendungen mit vordefinierten Port-Nummern zu priorisieren. Viele multimediale Anwendungen (z.B. NetMeeting) belegen allerdings die Port-Nummern dynamisch bzw. wechseln sie während einer Session.

Als Erweiterung des ersten Verfahrens wird in der Spezifikation RFC-2638 ein Agent definiert, der in der DiffServ-Architektur als Bandwidth Broker (BB) bezeichnet wird. Dieser soll den gesamten Überblick über das Geschehen in der eigenen DS-Domäne haben. Dafür kommuniziert er mit den Ingress- und Egress-Nodes einer DS-Domäne und konfiguriert deren DiffServ-Komponenten. Weiterhin muss der Bandwidth Broker Ressourcenanforderungen mit den BBs der Nachbardomänen austauschen bzw. auf solche Anforderungen antworten. In diesem Kommunikationsmodell muss der Netzadministrator zur Netzkonfiguration nur mit einem Bandwidth Broker kommunizieren. Dem BB wird eine Policy-Tabelle übermittelt, in welcher die Berechtigungen einzelner Teilnehmer sowie die vereinbarten Verkehrsprofile der Boundary Links gespeichert werden.

74 Weihted Fair Queueing

Wenn beim Netzadministrator eine Reservierungsanfrage eintrifft[75], so muss er nur die Policy-Tabelle des lokalen BB anpassen. Dafür muss sich der Administrator beim BB authentifizieren. Die Kommunikation findet verschlüsselt statt. Wenn noch weitere DiffServ-Netze zwischen dem BB und der Zieldomäne vorhanden sind, wird die Anforderung von einem BB zum anderen bis zum BB der Zieldomäne gesendet. In Gegenrichtung findet dann ebenfalls eine entsprechende Kommunikation statt, um die Anforderung zu bestätigen. Dieses Modell ist zwar leistungsfähiger als das erste, genügend aber durch die manuelle Anfrage auch nicht den Anforderungen größerer Netze. [NJZ99]



Peer	Policy	Total	Used
DS3	ask	10	0
DS1	<50 okay	100	30

Peer	Policy	Total	Used
DS2	ask	10	0
DS4	<50 okay	100	20

Abb. 3.22 Statische Ressourcenzuordndung

Die dritte Möglichkeit bietet eine weitere Automatisierung der Ressourcenverwaltung. Es werden dynamische Ressourcenzuweisungen ermöglicht, die mit Hilfe von BB entwickelt werden. Dabei konfiguriert ein BB nicht nur die Schnittstellen und Traffic-Control-Einheiten einzelner Router, sondern versendet zusätzlich Statusabfragen an die einzelnen Router seiner Domäne und wertet die Antworten darauf aus. Ist das Netz schwach belastet, so kann der BB selbsttätig in den Policy-Tabellen zusätzliche Subklassen bilden, damit zusätzliche Datenströme priorisiert werden bzw. die Dienstgüte existierender Datenströme erhöht wird. Somit sind die TCA[76] nicht mehr statisch, sondern dynamisch!

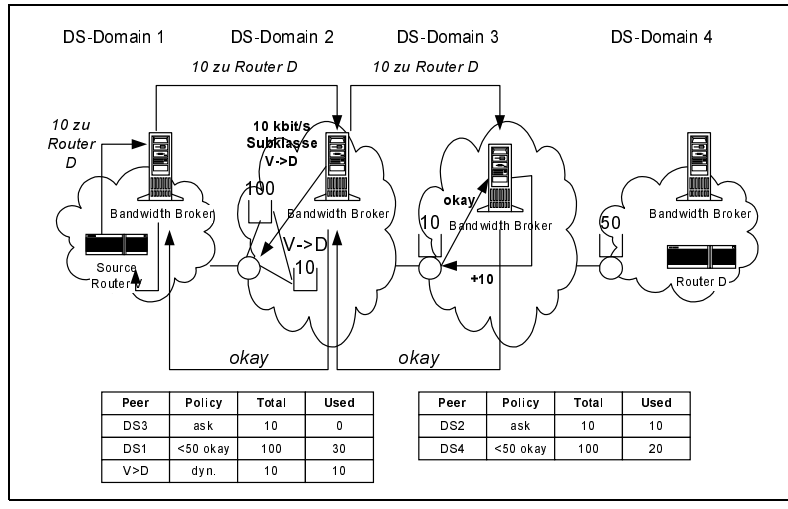
Zur Steuerung der Ressourcenzuweisung werden zusätzliche Kommunikationsmittel benötigt. Erstens ist ein weiteres Protokoll notwendig, mit dem ein BB einzelne Netzelemente steuert. Hierfür wird das Protokoll Common Open Policy Service (COPS) nach RFC-2748 verwendet. Weiterhin wird ein Protokoll

75 Z.B. per E-Mail, Telefon von einem Mitarbeiter

76 raffic Conditioning Agreement: ein Abkommen zwischen einem Diensterbringer und Dienstnutzer bzgl. Klassifizierungsregeln

für die Kommunikation zwischen den BBs unterschiedlicher Domänen benötigt. Für diese Funktion können SNMP[77] oder RSVP[78] angepasst werden. Dabei sind natürlich andere Anforderungen bei der Kommunikation zwischen zwei benachbarten BBs (Authentisierung und Verschlüsselung) zu beachten, wie wenn Nachrichten innerhalb einer Domäne verschickt werden. Als letzte Erweiterung müssen die Anwendungen in der Lage sein, Ressourcen von dem Ingress Router anfordern zu können. RSVP ist dafür geeignet, muss aber um zusätzliche Objekte für die DiffServ-Architektur erweitert werden. Als Alternative kann ein Pre-Marking der Datenpakete in den Hosts erfolgen und so als Ressourcenanforderung gelten. Diese Art der Signalisierung würde allerdings nur unidirektional verlaufen, sodass vom Host zum Netz und nicht vice versa signalisiert würde. [DBCH00]

Abb. 3.23 Dynamische Ressourcenzuordnung



Peer	Policy	Total	Used
DS3	ask	10	0
DS1	<50 okay	100	30
V>D	dyn.	10	10

Peer	Policy	Total	Used
DS2	ask	10	10
DS4	<50 okay	100	20

Ein Beispiel der statischen Ressourcenzuordnung mit Hilfe von Bandwidth Brokern zeigt Abb. 3.22. Dabei ist der Source Router in der DS-Domain 1. Die Daten werden in das Netz DS-Domain 4 gesendet. Dazwischen liegen zwei weitere DS-Domains. Im statischen Fall prüft der lokale BB im Austrittsknoten der Domäne lediglich, ob die Summe aller abgehenden Datenströme die Vorgabe in der Policy-Tabelle nicht überschreitet. Ist diese Bedingung erfüllt, so wird der BB der benachbarten DS-Domäne auf Verfügbarkeit der Ressourcen abgefragt. Dabei ist zu beachten, dass nicht benachbarte Knoten, sondern benachbarte Netze miteinander kommunizieren. Ein Policy-Eintrag im Netz DS-Domain 3 ermöglicht bis zu 50 Kbit/s ohne zusätzliche Anforderung an das Netz der DS-Domain 4. Es werden zuerst 20 Kbit/s an diesem Schnittpunkt übertragen,

77 Simple Network Management Protocol
78 Resource Reservation Protocol

wodurch weitere 10 Kbit/s an Datenpaketen ohne weitere Anfragen weitergeleitet werden. Sind alle Anfragen der BBs erfolgreich, so wird bis zum Eingangsrouter in der DS-Domain 1 von allen BBs eine explizite Bestätigung gesendet. Weist ein Netz die Anforderung ab, so bekommt der Quellrouter eine negative Bestätigung. In diesem Fall werden alle Pakete vom Quellhost zum Best-effort-DSCP mit einem Re-Marking versehen.

Beim Einsatz dynamischer Zuweisungsverfahren werden im ersten Schritt die Policy-Tabellen auf die gleiche Weise abgefragt. Diese Tabellen sind nun aber dynamisch und können in Abhängigkeit von der Netzlast angepasst werden. Ist die in der Policy-Tabelle eingetragene Menge der Ressourcen belegt, so kann der BB den Eintrittsknoten auf Verfügbarkeit zusätzlicher Ressourcen abfragen. Sind Ressourcen frei, so wird für den zugesagten Datenfluss eine Subklasse in der Policy-Tabelle gebildet. Dabei muss der Ingress Router einer solchen Domäne die Fähigkeit besitzen, eine Multifeld-Klassifizierung vornehmen zu können. [SIEM99]

3.2.6 Fazit

An dieser Stelle werden die Ansätze IntServ und DiffServ zusammengefasst, verglichen und eine mögliche Interoperabilität zwischen den beiden Verfahren abgeschätzt.

IntServ

Das Grundkonzept der IntServ ist verbindungsorientiert. Man versucht eine feste Bandbreite auf den Übertragungsstrecken für einzelne Teilnehmer zu reservieren. Eine Ersparnis der Ressourcen erhofft man sich dabei durch eine prädiktive Schätzung des ankommenden Verkehrs. Dabei geht man davon aus, dass die angemeldete Datenrate nicht die ganze Zeit ausgenutzt wird. Zusätzlich bekommt man beim CLNE-Dienst durch Anmeldung der Sender einen Überblick über die Anforderung an das Netz, was das Management der Netzknoten wesentlich erleichtert[79].

Dazu bieten die Integrated Services gegenüber einer physikalischen Verbindung eine bessere Granularität, da die Bandbreite in 1-Byte-Schritten variiert werden kann. Man ist hier nicht auf 64-Kbit/s-Schritte festgelegt, die aus der Pulse Code Modulation (PCM) heraus stammen. Dies wird gerade bei typischen VoIP[80]- oder Videokonferenzszenarien deutlich, bei dem die Reservierung mehrerer Kanäle mit unterschiedlichen Verkehrscharakteristiken signifikant sind. Integrated Services bieten auch einen hohen Grad an Flexibilität, denn die Ressourcenreservierung kann während der Datenübertragung angefordert, geändert oder aufgelöst werden.

79 Z.B. Aufbau neuer ATM SVC-Verbindungen in Zeiten hoher Anforderung an das Netz

80 Voice-over-IP

Auf der anderen Seite ist zu beachten, dass der Unterhalt von PATH- und Reservierungszuständen einen nicht zu vernachlässigenden Management-Aufwand in den Routern und zusätzlichen Management-Datenstrom verursacht! Zur Abschätzung der Belastung eines heutigen Routers im Kernnetz kann man von 2 GBit/s ausgehen, die an IntServ-Daten verarbeitet werden müssen. Als Anwendung wird dabei eine Videokonferenz genommen, da sie sowohl Bild- als auch Sprachdaten enthält. Reservierungen müssen demnach für die Videoübertragung nach H.263 und die Audioübertragung nach PCM[81] vorgenommen werden. Dabei werden Samples von 20 ms in ein Datenpaket zusammengefasst und übertragen. Vortragsfolien werden parallel zur Audio- und Videoübertragung alle 100 Sekunden mit einer Burst Time von 10 s transportiert. Die Folien werden als Bilder mit RGB-Kodierung (ca. 4,8 MByte pro Folie) weitergeleitet. Ein Telepointer wird zehnmal pro Sekunde mit 4 Bytes übertragen (x-y-Position).

Tab. 3.6 Datenraten einer Videokonferenzübertragung

Datenstrom	Datenrate [Kbit/s]	Paketgröße [Byte] (inkl. Header)	Paketrate [Pakete/s]
Videostream	384	1500	33
Audiostream	80	200	50
Folien	3750	1500	313
Telepointer	3,5	44	10
Insgesamt	*4217,5*	-	*406*

Ein Router muss in der Lage sein, bis zu 500 solcher Konferenzen zu verwalten. Somit müssten alle Router entlang der Route permanent 4000 PATH-State-Einträge und genauso viele RESV-State-Einträge in Echtzeit verwalten können. Hinzu kommt, dass in bestimmten Zeitabständen im Sekundenbereich Refresh-Nachrichten für jede Reservierung gesendet, empfangen und verarbeitet werden müssen. Mit der Annahme, dass eine Statustabelle 2 KByte groß ist, braucht ein Router nur für die Unterhaltung der Statustabellen ein Speichervolumen von 16 MByte. Die Paketankunftsrate beträgt dabei 205.000 Pakete/s, somit muss jedes Paket im Mittel innerhalb von 4,9 µs verarbeitet werden. Das heißt, in jedem Paket ist nicht nur der IP-Header, sondern auch der Dateninhalt auf die Sender- und Empfänger-IP-Adresse und auf die Port- und Protokoll-Nummer zu überprüfen. Anschließend wird anhand dieser Kriterien in den RESV-State-Tabellen gesucht werden, ob ein entsprechender Reservierungszustand existiert. Existiert für ein Datenpaket ein Reservierungszustand, so wird das Datenpaket in die entsprechende Warteschlange weitergereicht.

81 Pulse Code Modulation

Zusätzlich zu diesem großen Verarbeitungsaufwand kommt noch der Aufwand zum Management der 4.000 einzelnen Warteschlangen (Queues). Anhand dieser Abschätzung wird ersichtlich, dass ein IntServ-Netz im Weitverkehrsbereich große Skalierungsprobleme haben wird, da man End-to-end-Verbindungen etablieren möchte. Daher ist der IntServ-Ansatz eher im lokalen Bereich einsetzbar!

DiffServ

Der DiffServ-Ansatz stellt genauso wie IntServ eine weitgehend garantierte Dienstgüte bereit. Das Priorisieren von Datenpaketen nach verschiedenen Dienstklassen ist bei DiffServ allerdings mit höheren Anforderungen spezifiziert worden. Trotzdem lassen sich diese Dienste (EFP und AFPG) letztendlich nur im Kernnetz einsetzen und nicht durchweg am Rand eines Netzwerks. Der Vorteil gegenüber IntServ ist die bessere Skalierbarkeit, da nicht verbindungsorientiert gearbeitet wird. Allerdings fehlt die direkte Zuordnung zu Applikationen, die wiederum durch den IntServ-Ansatz umgesetzt werden können.

Bezüglich Multicasting ist bei dem noch relativ jungen Ansatz DiffServ ebenfalls noch nichts berücksichtigt worden. Deshalb bringt die Nutzung von DiffServ in Verbindung mit Multicast-Datenströmen einige Komplikationen mit sich. Ankommend in einem Ingress-Node, kann ein Datenstrom in einem Knoten verzweigen und über mehrere Egress-Nodes die Domäne verlassen. Außerdem ist die Anzahl der Teilnehmer einer Multicast-Gruppe im Allgemeinem dynamisch. Damit wird die Verwaltung der Ressourcen innerhalb der Domäne wesentlich komplizierter. Es ist problematisch, eine stabile Dienstgüte zu gewährleisten. Das nächste Problem tritt auf, falls ein Multicast-Datensrom nach der Verzweigung in unterschiedliche Domänen weiterfließt. Die vereinbarten Service Level Agreements (SLAs) gelten jeweils nur für die beteiligten Seiten. Die verfügbaren Kapazitäten sowie die vereinbarten Traffic-Spezifikationen dafür können unterschiedlich sein. Es wird im RFC-2475 lediglich vorgeschlagen, bei konkreten Service-Spezifikationen zusätzlich Code Points für Multicast-Daten vorzusehen, damit man die Möglichkeit einer Isolation der Unicast- von den Multicast-Datenströmen hat. Weitere Lösungswege stehen zur Zeit aus. [BBCD98]

Zusammenspiel IntServ und DiffServ

Das DiffServ-Modell besitzt seinen größten Vorteil in der guten Skalierbarkeit der Dienste auf große Netze. Dabei werden im inneren Netz nicht einzelne Datenströme, sondern einige wenige Datenstromaggregate unterschieden. Auf der anderen Seite bietet das IntServ/RSVP-Modell eine wesentlich bessere Granularität des Dienstes als das einfache DiffServ-Modell. Die Anforderungen einer Anwendung sind dabei an keine Verkehrsprofile gebunden. Man kann also nahezu beliebige Werte der Verzögerung und der Bandbreite verlangen. Außerdem bietet RSVP ein sehr leistungsfähiges Signalisierungsprotokoll zur Aushandlung der QoS-Parameter zwischen einem Anwender und dem Netz an.

Um die Vorteile beider Modelle zu nutzen, ist beim IETF ein Architekturmodell zur Interoperabilität zwischen IntServ und DiffServ in der Spezifikation RFC-2998 entwickelt worden.

Danach liegt am Rande des gesamten Netzes ein IntServ-Netz und in der Mitte ein DiffServ-Netz vor. Das IntServ-Netz tritt hier als Dienstnutzer (Kunde) der DiffServ-Region auf. Abb. 3.24 zeigt dafür das Architekturmodell. Zur Vereinfachung ist auf beiden Seiten der DS-Domain nur eine IntServ-Domäne eingezeichnet. In der Realität befinden sich an beiden Seiten der DiffServ-Wolke viele IntServ-Domänen. Da die Router in einer DS-Domain auch RSVP-fähig sein können und die Knoten der IntServ-Domäne DS-fähig, ist die Einteilung des Netzes in DiffServ- und IntServ-Regionen nicht immer eindeutig. Man unterscheidet IntServ- von DiffServ-Regionen nur anhand der Behandlung der Datenflüsse. Werden in einer Region die Datenflüsse auf Microflow-Basis behandelt, so betrachtet man diesen Bereich als eine IntServ-Region. Wenn hingegen die Datenflüsse auf einer Aggregat-Basis behandelt werden, so betrachtet man diese als eine DiffServ-Region.

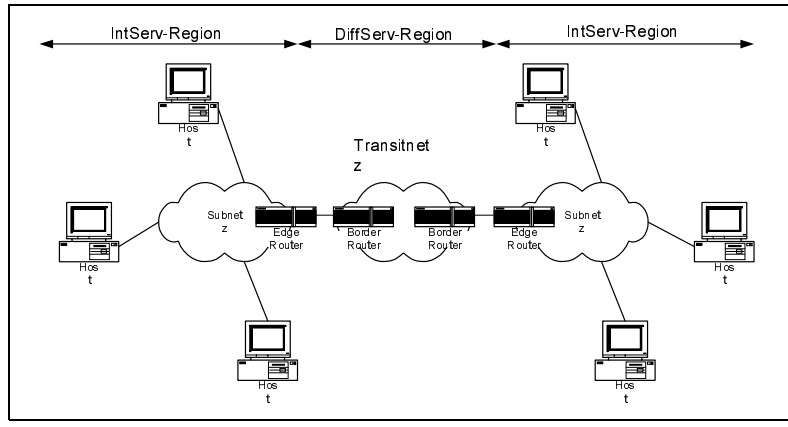


Abb. 3.24 Architekturmodell der IntServ/DiffServ-Interoperabilität

Die Grenze zwischen der IntServ- und DiffServ-Region ist vom Netzadministrator frei wählbar. Zwei Extremfälle sind hierbei denkbar:

- Alle Router im Netz behandeln die Datenflüsse auf Microflow-Basis. Lediglich auf einer Verbindung im Netz werden die Flüsse zu Datenflussaggregaten zusammengefasst. In diesem Fall hat das Netz im Grunde die Eigenschaften einer IntServ-Region. Dabei werden aber Router, in denen extrem viele Microflows zusammenlaufen, wesentlich entlastet.
- Alle Router im Netz sind DiffServ-fähig und lediglich im Quell- und Ziel-Host werden die Daten auf Microflow-Basis behandelt. Dabei wird das gesamte Netz von der Per-Flow-RSVP-Signalisierung entlastet. Die Hosts haben aber die Möglichkeit, Ressourcen explizit vom Netz anzufordern sowie das Ergebnis der Anforderung zu erfahren.

Bei der Interoperation zwischen IntServ und DiffServ sind unterschiedliche Bereiche zu berücksichtigen:

- Behandlung der Datenpakete in den Hosts
- Ende-zu-Ende Signalisierung
- Aufgaben der Edge-Router
- Aufgaben der Border Router
- Service Mapping

Jeder einzelne Host in diesem Modell muss alle Fähigkeiten eines IntServ-Host besitzen. Zusätzlich können die Datenpakete in den Hosts bezüglich der für die DS-Domain vereinbarten TCA vormarkiert (Pre-Marking) werden. Alternativ dazu kann man die Datenpakete auch in einem Router der IntServ-Region vormarkieren. Allerdings wäre aus Gründen der Performance ein Host der optimale Punkt für die Markierung. Im Allgemeinen werden die Knoten innerhalb der DS-Domain keine (per Flow) RSVP-Signalisierung (Ende-zu-Ende) unterstützen. Somit müssen die RSVP-Nachrichten zwischen den Border-Routern anhand der definierten Regeln zur Überbrückung von nicht RSVP-fähiger Teilstrecken ausgetauscht werden. Alle Router innerhalb der DS-Region müssen somit in der Lage sein, Datenpakete mit der RSVP-Kennung im IP-Header transparent weiterzuleiten. Bei der Berechnung der *ADSPEC* wird die Strecke zwischen den Border-Routern als einzelne Verbindung mit bestimmbaren C-/D-Parametern betrachtet.

Edge-Router sind Grenzrouter der IntServ-Region, deren Aufgaben von der konkreten Aufteilung in die IntServ- und DiffServ-Bereiche abhängen. Sind die Border-Router nicht RSVP-fähig, so agieren die Edge-Router als Admission-Control- und Policy-Control-Agenten für die DiffServ-Region. Sie übertragen Signalisierungsinformationen zwischen Sender und Empfänger. Die Admission-Control-Meldungen in den Knoten der Edge-Router basieren auf der Verfügbarkeit der Ressourcen in der DiffServ-Region. Weiterhin kann zur Erhöhung der Effizienz in den Edge-Routern Traffic Shaping für alle Daten zwischen den Edge-Routern[82] und dem Border-Router durchgeführt werden. Sind die Border-Router nicht RSVP-fähig, so agieren sie als normale Boundary-Router einer DS-Domain. Das Policing wird dabei auf Aggregatbasis durchgeführt. Sind die Border-Router RSVP-fähig, so werden die Admission-Control-Aufgaben in den Border-Routern durchgeführt. Die Verlagerung der Admission-Control-Aufgaben in die Edge-Router bietet eine bessere Skalierbarkeit der Architektur. Werden diese dagegen in die Border-Router verlagert, so kann das DiffServ-Netz Einfluss auf das Policy-Control-Processing nehmen. Werden Ressourcen innerhalb der DS-Region statisch zugeordnet, so wird jedem defi-

82 Falls mehrere Edge-Router (IntServ-Regionen) auf eine einzelne DiffServ-Region treffen und an einem Border-Router zusammentreffen.

nierten IntServ-Dienst eine in der SLA vereinbarte PHB zugewiesen. Der Edge-Router der Upstream IntServ-Region bzw. der Ingress-Router der DiffServ-Region muss dabei die einzelnen *ADSPEC*-Parameter für die gesamte DS-Region kennen und an das IntServ-Netz exportieren. Am Ingress-Punkt wird das BA[83]-Policing durchgeführt. Wird die Ressourcenzuweisung mit Hilfe von BBs durchgeführt, so können diese die Markierungsregeln in Abhängigkeit von der Belastung des Netzes dynamisch anpassen und diese Änderungen in den Ingress- und Egress-Routern eintragen.

Um sich das Zusammenspiel zwischen den Ansätzen IntServ und DiffServ vergegenwärtigen zu können, wird hier noch einmal das Szenario nach Abb. 3.24 schrittweise durchgegangen. Zuerst wird der RSVP-Prozess im Sender-Host ganz links im Bild durch eine PATH-Nachricht erzeugt, die den vom Sender angebotenen Datenstrom beschreibt. Die PATH-Nachricht wird hop-by-hop regelkonform durch die IntServ-Region durchgereicht. Im Edge-Router des IntServ-Netzes wird der Eintrag PATH-State gespeichert. Die PATH-Mitteilung wird durch das DiffServ-Netz, welches als reines Transitnetz fungiert, ohne weitere Verarbeitung bis zum nächsten Edge-Router durchgereicht. Die PATH-Nachricht wird von diesem Router weiter durch das zweite IntServ-Netz am rechten Rand hop-by-hop durchgereicht. Wenn die Nachricht beim Empfänger-Host ankommt, entscheidet dieser, ob eine Reservierung für diesen Datenfluss vorgenommen werden soll. Ist eine Reservierung gewünscht, so generiert das Betriebssystem des Empfänger-Hosts über eine Schnittstelle zum RSVP-Prozess eine RESV-Nachricht. Diese RESV- Nachricht wird in Richtung des DiffServ-Netzes wieder hop-by-hop durchgereicht. Dabei werden in jedem Router anhand der Standard-Behandlungsregeln der IntServ die Einheiten Admission Control und Policy Control auf Verfügbarkeit der Ressourcen und die Berechtigung zur Reservierung abgefragt. Erreicht die RSVP- Nachricht einen Edge-Router, der nicht RSVP-fähig ist, so wird eine modifizierte Nachricht direkt an den ersten Edge-Router gesendet. Im ersten Edge-Router (ganz links) prüft der Prozess Admission Control, ob ausreichend Ressourcen für den entsprechenden DiffServ-Dienst in der DiffServ-Region zur Verfügung stehen. Dabei werden die aktuelle SLS[84] und die Ressourcenbelegung berücksichtigt. Zusätzlich kann hier die Policy Control die Teilnehmerberechtigungen zur Nutzung der DiffServ-Dienste prüfen. Ein DCLASS-Objekt mit dem vorgegebenen DSCP kann der RESV-Nachricht beigefügt werden. Wird die RESV-Anfrage von keiner Admission Control und Policy Control im Netz abgewiesen, so wird sie weiter zum Sender-Host durch das IntServ-Netz durchgereicht. Der Sender-Host empfängt die RESV-Message und interpretiert diese als Zulassung des

83 Behavior Aggregate: ein Bündel von Verkehrsflüssen bei DiffServ, die vom Netz auf gleiche Weise behandelt werden.

84 Service Level Specification

Dienstes sowohl in der IntServ- als auch in der DiffServ-Region. Der Sender-Host kann die abgehenden Datenpakete mit einem DSCP versehen. Dabei kann zur Bestimmung des DSCP eine Standard-Mapping-Methode angewendet oder in der RESV-Nachricht der zugewiesene DSCP explizit mitgeteilt werden. [BFYB+00]

Vorteile der Interoperabilität

An dieser Stelle sollen noch einmal die Vorteile einer Interoperabilität beider Ansätze hervorgehoben werden. Es ist durchaus sinnvoll, wie auch eben dargestellt wurde, beide Ansätze gleichzeitig und in großen Netzen zu verwenden. Der größte Vorteil eines kombinierten IntServ-/DiffServ-Netzes gegenüber einem reinen IntServ-Netz ist natürlich die bessere Skalierbarkeit des Modells bezogen auf große Netze. In den Kernknoten werden dabei nur wenige Zustandsinformationen ausgewertet und unterhalten. Ebenfalls ergeben sich Vorteile gegenüber einer reinen DiffServ-Nutzung. Beim Einsatz von RSVP am Rande des Gesamtnetzes wird eine so genannte Explicit Admission Control (EAC) angewendet. Dabei wird jede einzelne Ressourcenanforderung explizit angefordert und bestätigt, wodurch man eine höhere Ressourcenausnutzung erreicht.

Anhand eines VoIP[85]-Beispiels wird der Nutzen deutlich. Vorgegeben sei eine DS-Domain, die am Eintrittspunkt 100 Kbit/s für Expedited-Forwarding-Daten zulässt und diesen Dienst für VoIP nutzt. 10 Teilnehmer möchten den Dienst IP-Telefonie nutzen, wobei jeder Teilnehmer 20 Kbit/s benötigt. Alle Daten werden von den Anwendern mit den zum EFP-Service entsprechenden DSCP markiert. Der EFP-Service verlangt, dass alle Daten, die das Traffic Profil überschreiten, verworfen werden. Da aber die Datenpakete am Einlaufrouter auf Aggregatbasis behandelt werden, können Pakete von allen Teilnehmern verworfen werden. Somit kann im ungünstigsten Fall kein einziges Telefonat stattfinden. Würde man allerdings auf der Teilnehmerseite IntServ einsetzen, so würden die Teilnehmer ihre Anforderungen explizit signalisieren und auf eine Bestätigung der Anforderung warten. Somit könnten fünf Teilnehmer ihre Telefonate ungestört führen. Die restlichen 5 Teilnehmer wären explizit abgewiesen worden, könnten aber immerhin noch durch Markierung der Datenpakete mit dem Best-effort-DSCP ein IP-Telefonat auf Basis von Best-effort vornehmen.

Weiterhin kann beim Zusammenspiel zwischen IntServ und DiffServ eine Policy-basierte Admission Control angewendet werden. Dabei werden Ressourcenanforderungen von bestimmten RSVP-fähigen Knoten im Netz abgefangen und anhand von Policies überprüft. Der Einsatz von Policy-basierter Admission Control ermöglicht eine Vergabe personenbezogener Berechtigungen zur Nutzung von Netzressourcen. Dagegen kann ein reines DiffServ-Netz seine Policy-Entscheidungen nach der gegenwärtiger Spezifikation nur anhand der IP-Adresse und Port-Nummer treffen.

85 Voice-over-IP

Ebenfalls kann die Zuweisung des DSCP dynamisch erfolgen. Wird ein DSCP dynamisch einem Datenstrom zugewiesen, so kann mit Hilfe von RSVP dieser dem Sender mitgeteilt werden. Damit kann die Markierung der Pakete komplett vom Sender vorgenommen werden. Diese Maßname entlastet die Router im Netz. Zur Übertragung des DSCP mittels RSVP müssen zusätzliche Objekte im RSVP definiert werden. Dies wurde in der Spezifikation RFC-2996 auch unternommen.

In den IntServ-fähigen Netzelementen wird für jeden Datenfluss eine separate Warteschlange eingerichtet. Diese Vorgehensweise erleichtert eine Unterstützung von quantitativen Diensten. Das heißt, dass bei den zurzeit spezifizierten DiffServ-Diensten noch keine quantitativen Dienste möglich sind. Zur Definition quantitativer DiffServ-Dienste fehlen noch praktische Erfahrungen in der Realisierung von DiffServ-Netzen. [BERN00]

4

Traffic Engineering

In den letzten Jahren hat sich die Telekommunikations- und Datenlandschaft rapide gewandelt. Das geht hauptsächlich auf das Internet zurück, dessen Protokollfamilie immer mehr Einzug in andere Netze hält. Dabei wandelt sich auch das Internet selbst immer mehr, da es neue Anforderungen wie Echtzeitfähigkeit, Dienstgütegarantie, Sicherheit und Skalierbarkeit bewältigen muss, für die es eigentlich nicht entwickelt wurde. Entstanden ist diese Entwicklung durch den Wunsch „Anything-over-IP" oder „IP-over-Anything" umzusetzen, wodurch auch Sprache und Video besser unterstützt werden sollen. Hinzu kommt der alte Wunsch nach Sprach- und Datenkonvergenz, damit nur noch ein Netz verwaltet werden muss, um Kosten einzusparen und flexibler auf Innovationen reagieren zu können.

Im Bereich heterogener Netzstrukturen im WAN[1] hat sich aufgrund der Plattformunabhängigkeit und der leicht zu implementierenden Software auf unterschiedlichen Rechnersystemen die TCP/IP-Protokollfamilie gegenüber anderen Protokollen durchsetzen können. Aufgrund der anwachsenden Datenraten im Weitverkehrsbereich unter Einbeziehung neuer Anwendungsmöglichkeiten, wie Computer Supported Co-operative Work (CSCW), ist die Effektivität von IP-Protokollen von entscheidender Bedeutung. Da TCP/IP-Protokolle eigentlich für geringe Datenraten im Internet entwickelt wurden, spielt die Anpassung und Integration von IP auf Hochgeschwindigkeitsnetzen wie ATM[2] oder Gigabit Ethernet (GE) für das LAN[3] oder WAN eine wichtige Rolle.

ATM ist für das B-ISDN-Referenzmodell als Übertragungsprotokoll eingeführt worden. Dabei haben der große Erfolg und Verbreitungsgrad des Internet dazu geführt, das ATM als Transportmedium einzusetzen, welches neben hoher Bandbreite auch Skalierbarkeit, Dienstgüte und Erweiterbarkeit anbietet. Deshalb müssen die beide Protokolle IP und ATM reibungslos zusammenarbeiten. Darüber hinaus sind die Anwender an der Integration der eigenen traditionellen Netzwerke interessiert, um bestehende und zukünftige Investitionen absichern zu können. Allerdings haben IP-Protokolle und ATM eine völlig unterschied-

1 Wide Area Network
2 Asynchronous Transfer Mode
3 Local Area Network

liche Funktionsweise. Die Standardisierungsgremien IETF[4], ITU[5] und ATM-Forum haben Standards spezifiziert, die IP auf ATM anpassen bzw. integrieren. Dabei ist das endgültige Ziel, die Vorteile von ATM unter Einbeziehung des IP-Protokoll nutzen zu können. Als wichtige Elemente sind in diesem Zusammenhang LAN Emulation (LANE), Multi-Protocol-over-ATM (MPOA) und Multi-Protocol-Label-Switching (MPLS) zu nennen. Ebenfalls wird über direkte IP-Verbindungen über SDH nachgedacht, um so den Overhead gering zu halten.

Aus Gründen der Ökonomie war die optimale Netznutzung für einen Netzbetreiber schon immer ein Ziel, dessen Realisierung eine Anforderung an die Netztechnologie darstellt. Aufgrund der Dienstvielfalt erhält diese Anforderung aber einen neuen Aspekt. Das Vermeiden von Überlastsituationen von einzelnen Netzwerkkomponenten ist auch für die Übertragung von zeitkritischen Daten vorteilhaft, da sich die Netzperformance verbessert. Traffic Engineering beschreibt die Fähigkeit, durch entsprechende Maßnahmen den Verkehr so zu lenken, dass das Netz optimal ausgenutzt wird. Es soll vermieden werden, dass freie Ressourcen ungenutzt bleiben und gleichzeitig andere Bereiche überlastet sind. Erste Ansätze, um Traffic Engineering in IP-Netze zu implementieren, wurden über das Load Sharing realisiert, das von Routing-Protokollen wie Open Shortest Path First (OSPF) und Intermediate System to Intermediate System Protocol (IS-IS) unterstützt wird. Dieser Mechanismus findet dann Anwendung, wenn mehrere Pfade zwischen Quelle und Senke existieren, welche die gleichen Kosten aufweisen. Kosten definieren die Verbindung zwischen zwei Netzelementen anhand von Parametern und erlauben den Routing-Protokollen so, eine Entscheidung über den Verbindungsweg zu treffen.

Werden beim Load Sharing mehrere Pfade gefunden, die mit gleichen Kosten verbundenen sind, wird der Datenstrom gleichmäßig auf die Pfade aufgeteilt. Dieser Mechanismus eignet sich, um Traffic Engineering in kleinen Netzen zu betreiben. In großen Netzwerkumgebungen wird es aber kaum möglich sein, mehrere Pfade mit gleichen Kosten zu finden. Eine weitere Möglichkeit, den Verkehr zu steuern, bietet das Explicit Routing[6] (ER). Dabei werden Informationen der Netzbelastung verwendet, um einen alternativen Pfad zu bestimmen, der über mehr Ressourcen verfügt. Einige Datenpakete werden zur Entlastung eines überlasteten Bereichs über diesen Pfad umgeleitet. Zu diesem Zweck wird den Headern dieser Pakete eine ER-Information angefügt. Alle Netzelemente entlang der Explicit Route werten diese ER-Information aus und leiten das Paket entsprechend weiter. Der Nachteil dieser Methode besteht darin, dass zusätzliche Informationen im Header transportiert werden müssen, was die Nutzdatenrate herabsetzt.

4 Internet Engineering Task Force
5 International Telecommunication Union
6 Wird auch als Source Routing bezeichnet.

Prinzipiell wäre es möglich, durch genügend Bandbreite im Netz eine ausreichende Qualität bereitzustellen und das Best-effort-Prinzip auch weiterhin beizubehalten. Dieser Ansatz, der besonders von der Fraktion IP-over-Wave-Division-Multiplexing favorisiert wird, ist aber anzuzweifeln, da nie ausreichend Bandbreite vorhanden sein wird. Jede Erweiterung des Netzes wird auch unweigerlich neue Dienste und Anwendungen hervorbringen, die diese neuen Ressourcen wieder aufbrauchen. Um das zu erkennen, muss man sich nur die Entwicklung in der Rechnertechnologie ansehen, deren Leistungsfähigkeit sich fortlaufend verdoppelt, ohne allerdings wesentlich mehr Spielraum für die vorhandenen Applikationen zu schaffen.

Es soll also ein Netz geschaffen werden, welches die Qualität von Echtzeit- und Datenanwendungen garantieren kann. Um dies umzusetzen, gibt es unterschiedliche Verfahren sowie die Möglichkeit, Traffic Engineering im Kernnetz einzuführen. Das Hauptverfahren heißt heute Multi-Protocol Label Switching (MPLS), welches sich aktuell in der Standardisierungsphase befindet. Dieses Kapitel widmet sich den Basistechnologien[7] von MPLS, der technischen Darstellung des Protokolls und seinen Funktionen. Darüber hinaus wird eine Bewertung von MPLS in Hinblick auf seinen Einsatz im WAN-Bereich vorgenommen. Zu diesem Zweck werden die vornehmlich bestehenden WAN-Protokolle ATM und Packet-over-SONET (PoS) vorgestellt und mit MPLS anhand verschiedener Parameter verglichen. Es findet somit eine Bewertung statt, die für den Einsatz von Quality-of-Service (QoS) im Netzwerk entscheidend ist, da MPLS sowohl auf ATM als auch PoS basieren kann.

4.1 Asynchronous Transfer Mode (ATM)

Als die International Telecommunication Union (ITU) die Übermittlungstechnik ATM vorgeschlagen und definiert hat, wurde ursprünglich die Realisierung des Broadband Integrated Services Digital Network (B-ISDN) verfolgt. ATM ermöglicht hohe Datenraten bis zu mehreren Gbit/s bei geringen und definierbaren Verzögerungen und zeichnet sich auch durch eine flexible Bandbreitenzuordnung je Verbindung und Bedarf der Anwendung aus. Diese Vorteile des ATM-Prinzips führten zu weiteren Anwendungsbereichen. Es war deshalb der Einsatz nicht nur im B-ISDN vorgesehen, sondern u.a. auch in Backbone-Netzen, in LANs und in Endeinrichtungen. Auch wenn sich ATM nicht in allen Bereichen durchsetzen konnte, so hat sich diese Übermittlungstechnik im WAN-Segment erfolgreich etabliert. Immerhin werden ungefähr 75% des Internetverkehrs über ATM-Netze transportiert. [8]

7 MPLS basiert auf verschiedenen Layer-3-Switching-Verfahren wie Tag-Switching, ARIS, IP-Switching und Cell Switch Router (CSR) .

8 http:///www.atmforum.com/pages/mediarelationsfs1.html

4.1.1 Funktionsweise

Bei ATM werden auf jedem Übertragungsabschnitt Zellen fester Länge (53 Byte) übertragen. Die Zellen bestehen aus 48 Byte für Nutzinformationen und 5 Byte für den Zellkopf (Header), der unter anderem die Kanal- oder Pfadadressierung enthält (siehe Abb. 4.2). Ist gerade keine Nutzinformation zu senden, so werden spezielle Leerzellen gesendet. Hierbei ist die Nettobitrate sehr klein. Werden hingegen fast nur Nutzzellen gesendet, so nähert sich die Nettobitrate der Transportbitrate[9]. Dadurch ermöglicht ATM Verbindungen mit beliebiger Nettobitrate. Jede Zelle wird durch eine Kennung einem bestimmten virtuellen Übertragungspfad und einem darin geführten virtuellen Kanal zugeordnet. Da die Zellen klein sind, kommt es nur zu sehr geringen Verzögerungszeiten. Dadurch können synchrone und asynchrone Dienste gleichermaßen realisiert werden.

Das ATM-Netz arbeitet verbindungsorientiert, behält also die Reihenfolge der Zellen für jede Verbindung bei. Beim Verbindungsaufbau teilt der Netzbenutzer dem Netz über einen virtuellen Signalisierungskanal die gewünschte Bitrate mit und das Netz reserviert anschließend auf allen Übertragungswegen die entsprechende Bandbreite. Alle Netzknoten sind über eine oder mehrere so genannte ATM-Schalteinheiten (ATM-Switches) miteinander verbunden, welche die Zellen an ihren jeweiligen Bestimmungsort vermitteln. Dies kann aufgrund der festen Zellenlänge gleichzeitig für mehrere Zellen erfolgen. Die Netzteilnehmer teilen sich also nicht ein gemeinsames Übertragungsmedium, wie bei Shared-Media-Technologien, sondern sie entledigen sich an der ATM-Schalteinheit ihrer Zellen. Auf Zugriffsalgorithmen (Media Access) muss keine Rücksicht genommen werden. Die insgesamt zur Verfügung stehende Übertragungsbandbreite wird von der ATM-Schalteinheit nach Bedarf verteilt.

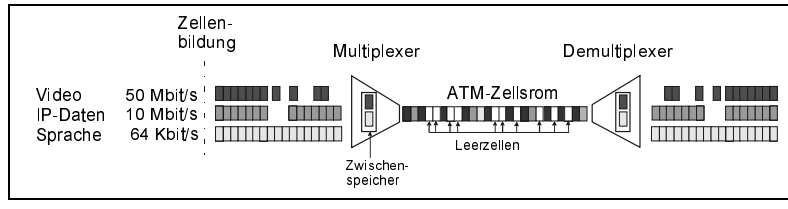


Abb. 4.1 ATM-Multiplexing/-Demultiplexing

ATM-basierte Netzwerke können durch neue Benutzer erweitert werden, ohne dass die bisherige Bandbreite der jeweiligen Teilnehmer eingeschränkt wird. In die vermittelnde ATM-Schalteinheit werden lediglich weitere Anschlussmodule mit Bandbreiten von beispielsweise 155 Mbit/s eingesetzt. Jedes Anschlussmodul stellt dann seinem Teilnehmer immer die volle Bandbreite von 155 Mbit/s zur Verfügung. Die einzige Begrenzung stellt die Verarbeitungsgeschwindigkeit

9 Bei 155 Mbit/s beträgt die Transportbitrate etwa 149,76 Mbit/s.

der Switches dar, wobei diese allerdings heute bereits Datenmengen von 40 Gbit/s und mehr verarbeiten können. Deshalb kann dieser Übertragungsmechanismus in nahezu allen Bereichen der Datenkommunikation eingesetzt werden. ATM ist damit gleichermaßen für den lokalen wie für den Weitverkehrsbereich geeignet.

Weiterhin ist ATM als eine der wenigen Techniken für Hochgeschwindigkeitsnetze in der Lage, Datenströme für unterschiedliche Datenraten flexibel zu übertragen und zu übermitteln. Es kann somit nicht nur Dienste, sondern auch Netze vollständig integrieren. Damit stellt ATM eine vereinfachte Paketvermittlung dar, wobei die ATM-Zelle keine Fehlerkorrektur vornimmt. Die Fehlererkennung und evtl. -korrektur wird durch die Anpassungsschicht AAL-Layer durchgeführt. Die ATM-Switches kontrollieren den Zellen-Header. Im Fehlerfall wird das entsprechende Paket verworfen. Wiederaufsetzen nach Fehlern bleibt den Endsystemen überlassen. Ebenfalls findet keine kontinuierliche Flusssteuerung statt. Wenn das Netzwerk überlastet ist, kommt es nur zu einer Verbindungsabweisung. Bestehende Verbindungen werden weder unterbrochen noch deren Datenrate gesenkt. Die Leistungsfähigkeit eines ATM-Netzes wird demnach ausschließlich von der Netzauslegung beeinflusst und nicht vom Übertragungsprotokoll. Da bei heutigen Netzen von wesentlich weniger Bitfehlern auszugehen ist, müssen im Grunde auf der Ebene 2 der Sicherungsschicht auch keine Fehlerkorrekturen durchgeführt werden. Diesen Umstand macht sich ATM zunutze. Weiterhin arbeitet ATM verbindungsorientiert, da virtuelle Kanäle verwendet werden. Man unterscheidet im ATM zwei Formen der Verbindung[10]:

1. **Permanent Virtual Connections** (**PVC**) sind virtuelle Festverbindungen, die vom Netzbetreiber eingerichtet werden müssen und über konstante Dienstgüteparameter verfügen.
2. **Switched Virtual Connections** (**SVC**) sind Wählverbindungen. Für eine Kommunikation werden hier drei Phasen unterschieden: Verbindungsaufbau, Nachrichtenübertragung und Verbindungsabbau. Die Eigenschaften der Verbindung werden beim Verbindungsaufbau festgelegt.

Beim ATM-Übermittlungsprinzip wird jede virtuelle Verbindung durch eine logische Kanalnummer[11] und die Identifizierung des Kanalbündels[12] gekennzeichnet. Über die Felder VCI und VPI des Headers werden die Zellen einer bestimmten Verbindung zugewiesen. Die PVC und SVC sind zwischen den Netzknoten unterteilt in viele aneinander gereihte virtuelle Kanäle. Die virtuellen Kanäle sind wiederum zu Pfaden zusammengefasst, um ein einfacheres Routing zu gewährleisten. Weiter unterscheiden sich die Verbindungsarten in

10 Virtual Connection (VC)
11 Virtual Channel (VC)
12 Virtual Path (VP)

Punkt-zu-Punkt, Punkt-zu-Mehrpunkt sowie Mehrpunkt-zu-Mehrpunkt. Zudem kann die Bandbreite asymmetrisch sein, wenn die Bandbreite in Sende- und Empfangsrichtung unterschiedlich ist.

Wie auch andere Übertragungsverfahren basiert ATM grundsätzlich auf einer Paketübertragungstechnik. Hierbei sind jedoch folgende wesentliche Änderungen bzw. Ergänzungen zu anderen Paketübertragungsverfahren zusammenfassend festzuhalten:

- **Feste Paketlänge:** ATM nutzt zur Übertragung ausschließlich Pakete fester Länge, die als Zellen bezeichnet werden
- **Qualitätsparameter der Verbindung:** ATM unterstützt eine garantierte Dienstgüte. Die dazu notwendigen Parameter werden beim Aufbau einer Verbindung festgelegt und zugewiesen.
- **Verbindungsorientierte Betriebsweise:** ATM agiert immer verbindungsorientiert. Die Umsetzung verbindungsloser Dienste ist in höheren Schichten vorgesehen.
- **Verzicht auf abschnittsweise Fehlersicherung:** ATM besitzt eine sehr geringe Bitfehlerwahrscheinlichkeit (10^{-12}), die durch den Einsatz heutiger Glasfaser und implementierter Flusssteuerung möglich wurde.
- **Verzicht auf abschnittsweise Flusssteuerung:** Durch die hohen Übertragungs- und Verarbeitungsgeschwindigkeit ist eine abschnittsweise Flusssteuerung nicht möglich.
- **Signalisierung:** Wie im ISDN wird die Signalisierung durch Nutzung separater Signalisierungskanäle vom Nutzdatenstrom entkoppelt.

4.1.2 ATM-Zellenformat

Die 53-Byte-Zellen bestehen aus einem fünf Byte großen Header zur Steuerung und Kontrolle der Zellen und einem 48 Byte großen Payload, der Nutzinformationen oder Signalisierungsnachrichten aufnimmt. Aufgrund der unterschiedlichen Aufgabenstellung unterscheiden sich die Header der Zellen, die an der Teilnehmer-/Netzschnittstelle[13] und an der Schnittstelle zwischen ATM-Vermittlungsstellen[14] verwendet werden, geringfügig, wie Abb. 4.2 zeigt.

Die ATM-Zelle besteht aus den folgenden Feldern, die unterschiedliche Funktionen beinhalten:

- **Generic Flow Control (GFC):** Flusskontrolle des User Network Interface (UNI), bestehend aus 4 Bit. Der Datenfluss zwischen ATM-Endstation und ATM-Vermittlungsknoten kann dadurch gesteuert werden. Zwei Verbindungsarten sind bekannt: nicht gesteuerte Verbindungen, die alle vier Bit auf null setzen und gesteuerte Verbindungen, die durch Endstationen aus-

13 User Network Interface (UNI)
14 Network Node/Network Interface (NNI)

gelöst werden und ihre Berechtigung dazu durch das Netzwerk erhalten. Innerhalb des ATM-Netzes werden diese Bits vom Network Network Interface (NNI) für einen erweiterten VPI verwendet.

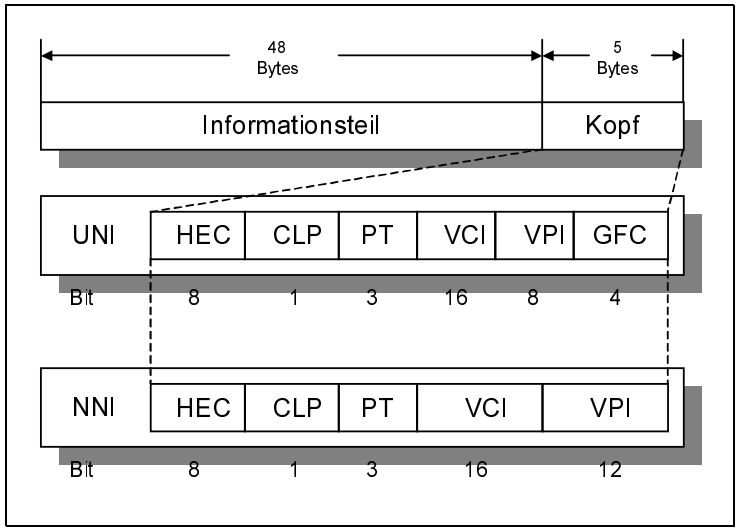


Abb. 4.2 ATM-Zellenkopf-Formate

- **Virtual Path Identifier (VPI):** Der VPI kennzeichnet den virtuellen Pfad zwischen zwei benachbarten Netzteilnehmern. Der VPI ist an der UNI-Schnittstelle 8 Bit lang. Innerhalb des ATM-Netzes kommen 12 Bit zur Geltung, da die NNI-Schnittstelle das GFC-Feld überschreibt. Dadurch lassen sich zwischen einer Endstation und dem Netzwerk 256 Pfade verwenden, während innerhalb eines ATM-Netzes 4096 Pfade geschaltet werden könnten. Der VPI-Wert 0 ist für die Managementfunktionen reserviert.
- **Virtual Channel Identifier (VCI):** Das Feld VCI ordnet der Verbindung neben dem Pfad einen virtuellen Kanal zu. Zusammen mit dem VPI-Feld ist somit eine eindeutige Kennzeichnung eines virtuellen Pfades möglich. Das VCI-Feld ist immer 16 Bit lang, wodurch 65 536 Kanäle in einem einzelnen Pfad unterschieden werden können. Somit lassen sich an der UNI-Schnittstelle insgesamt 16 777 216 Verbindungen schalten, während die NNI-Schnittstelle sogar 268 435 456 Verbindungen ermöglicht. Die VCI-Werte 0-15 und 17-31 sind bereits für Managementfunktionen der ATM-Schicht reserviert.
- **Payload Type (PT):** Das Feld PT belegt 3 Byte und gibt Aufschluss über die Zellenart. Hierbei lassen sich unterscheiden: Benutzerzelle mit/ohne Überlast, OAM[15]-Zelle und reserviert.

15 Operation and Maintenance

- **Cell Loss Priority (CLP):** Durch CLP lassen sich Zellen nach hoher (CLP=0) und niedriger Priorität (CLP=1) unterscheiden. Zellen mit niedriger Priorität werden bei Auftreten einer Überlast zuerst verworfen. Zunächst erhalten alle Zellen im Netz eine hohe Priorität. Die Zellen niedriger Priorität werden dabei durch das ATM-Netz so lange nicht transportiert, wie nicht genug Bandbreite zur Verfügung steht. Ändert sich dies, werden die Zellen zuerst verworfen. Das CLP-Feld wird von der User Parameter Control Unit überwacht und gesetzt.
- **Header Error Control (HEC):** Das Prüfsummenfeld gibt Aufschluss über die korrekte Übertragung des Kopfes. Bei bestimmten Übertragungsformen der physikalischen Schicht wird das HEC-Feld auch zur Synchronisation des Zellenstroms verwendet.

4.1.3 ATM-Schichten

Die ATM-Schichten basieren auf dem B-ISDN-Referenzmodell, welches eine Weiterentwicklung des OSI-Schichtenmodells darstellt. Beim B-ISDN unterscheidet man zunächst zwischen Benutzer-, Steuer- und Managementebene. Die Benutzerebene ist für den benutzerorientierten Informationsfluss zuständig. Die Steuerebene sorgt für den Auf- und Abbau sowie die Überwachung der Verbindungen, während die Managementebene sozusagen orthogonal zu allen anderen Ebenen und Schichten steht und unterschiedliche Informationsflüsse kontrolliert.

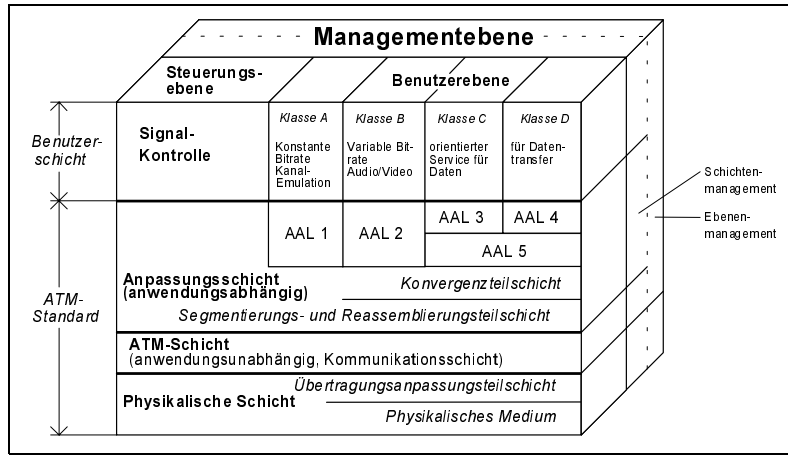


Abb. 4.3 B-ISDN-Referenzmodell

Da die Aufgaben für die Signalisierung komplizierter als bei herkömmlichen Netzen sind, ist das Schichtenmanagement zusätzlich für die Meta-Signalisierung oder den OAM-Informationsfluss zuständig. Als Meta-Signalisierung wird die Signalisierung der Signalisierung bezeichnet, weshalb auch ein eigener

Informationskanal zur Steuerung der verschiedenen Abläufe vorhanden ist. Die OAM-Daten werden zur Überwachung der Netzwerkleistungsfähigkeit und für das Fehlermanagement auf der ATM-Ebene benötigt. Hierzu sind besondere Zellen erforderlich.

Das B-ISDN-Referenzmodell enthält in vertikaler Richtung weiterhin die ATM-Anpassungsschicht[16], die ATM-Schicht und eine physikalische Schicht, welche die Kernfunktionalitäten bereitstellen und hier kurz näher erläutert werden.

Physikalische Schicht

Die Aufgabe der Bitübertragungsschicht ist es, den höheren Schichten Funktionen für die Übertragung von Signalen auf einem physikalischen Medium zur Verfügung zu stellen. Je nach verwendetem Medium, wie Glasfaser oder Koaxialkabel, muss die Bitübertragungsschicht angepasst werden. Die Funktionen der Bitübertragungsschicht beinhalten unter anderem die Umsetzung auf den Leitungscode sowie Synchronisierung und Fehlererkennung über das HEC-Feld des Zellenkopfes. Die physikalische Schicht wird unterteilt in eine medienabhängige Teilschicht und eine vom Übertragungsmedium abhängige Teilschicht:

- Physical Medium (PM)
- Transmission Convergence (TC)

Die PM-Teilschicht legt die Eigenschaften der Übertragungsmedien fest. Das heißt, hier werden Steckertyp, Kabelcharakteristika und Bitkodierung spezifiziert. Dabei können prinzipiell zwei Arten der Übertragung unterschieden werden: die zellenbasierte Übertragung sowie die Nutzung von bestehenden Netzarchitekturen. Die Aufgaben der TC-Teilschicht lassen sich hingegen wie folgt beschreiben:

- Generierung eines kontinuierlichen Zellenstroms bei gleichzeitiger Entkopplung der Zellrate durch Idle-Zellen von der zur Verfügung stehenden Bandbreite
- Erzeugung der Prüfsumme des Zellenkopfes, wodurch das Erkennen von Fehlern im Header gewährleistet ist
- Empfangsseitige Synchronisation des Zellenstroms und das Erkennen der Header
- Erzeugung der zugrunde liegenden Übertragungsrahmen bzw. der Zellenkodierung

Durch diese Aufteilung wird die ATM-Schicht vollständig vom verwendeten Übertragungsverfahren entkoppelt. Damit das physikalische Übertragungsmedium für die Übertragung von ATM-Zellen geeignet ist, muss eine entsprechende Definition der TC-Teilschicht vorhanden sein. Die Entkopplung der Zellrate geschieht dabei durch das Einfügen von Idle-Zellen (Leerzellen), in den

16 ATM Adaptation Layer (AAL)

von der ATM-Schicht generierten Zellenstrom. Die von der Konvergenzteilschicht sendeseitig erzeugten Leerzellen werden beim Empfänger entfernt und nicht an die lokale ATM-Schicht weitergeleitet. Da sich Virtual Channel Indication (VCI) und Virtual Path Indication (VPI) entlang einer virtuellen Verbindung an jeder ATM-Vermittlungsstelle ändern können, wird durch die physikalische Schicht die Header Error Control (HEC) durchgeführt. Sie ermöglicht das Erkennen von Fehlern und die Korrektur von einzelnen Bitfehlern. Die HEC wird ebenfalls für die empfangsseitige Synchronisation der Endstation auf den Zellenstrom verwendet.

Der Mechanismus der Synchronisation lässt sich als endlicher Automat mit drei Zuständen darstellen, wie auch Abb. 4.4 zeigt:

- **HUNT:** Der empfangene Bitstrom wird auf ein 5-Byte-Wort hin überprüft, dessen letztes Byte eine korrekte Prüfsumme enthält. Ist ein solches Wort gefunden, geht der Empfänger in den Zustand PRESYNCH über.
- **PRESYNCH:** Der Empfänger bleibt so lange in diesem Zustand, bis entweder eine Anzahl von *d* korrekten Zellen empfangen oder zuvor eine fehlerhafte Zelle erkannt wurde. In letzterem Fall wird erneut der Zustand HUNT eingenommen.
- **SYNCH:** Wurden *d* aufeinanderfolgende korrekte Zellen empfangen, wird der Zustand SYNCH eingenommen. Kommt es hier zum Empfang von mehr als *a* fehlerhaften Zellen, wird erneut in den Zustand HUNT übergegangen.

Die zulässigen Werte für *a* und *d* liegen zwischen 6 und 8. Zur leichteren Erkennung eines Headers wird außerdem der Payload der Zelle sendeseitig verschlüsselt. Die Verschlüsselung wird dabei erst durch den Empfänger wieder aufgehoben. [DETK98a]

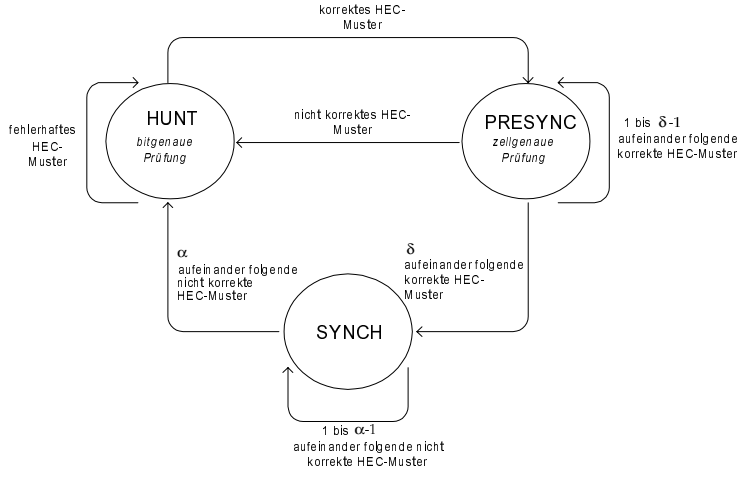


Abb. 4.4 Zustandsdiagramm der Zellensynchronisation

Die Rahmenanpassung erfolgt grundsätzlich auf drei verschiedene Arten:

- **SDH/SONET:** Anpassung der physikalischen Eigenschaften auf SDH/SONET-Rahmen (häufigste Methode der Übertragung bei ATM)
- **Physical Layer Convergence Procedure (PLCP):** Anpassung an plesiochrone Netze (PDH)
- **Direkte Zellenübertragung:** direkte Übertragung von ATM-Zellen ohne Berücksichtigung von anderen Schicht-2-Verfahren

ATM-Schicht

Die ATM-Schicht ist für den Transport der Zellen durch das Netz verantwortlich. Die Schicht überprüft die Umsetzung der Verbindung innerhalb der Switches, setzt die Identifikationen (VPI/VCI) für die virtuellen Verbindungen um und übernimmt das Multiplexen sowie das Demultiplexen der Datenströme. Die ATM-Schicht erhält die 48-Byte-Nutzdaten von den übergeordneten Schichten, generiert den Header und fügt diesen hinzu. Zu den Aufgaben der ATM-Schicht gehört zum einen die Flusskontrolle. Dadurch kann die Datenrate des Senders kontrolliert und gesteuert werden. Das Multiplexen und Demultiplexen von ATM-Zellen in den Zellenstrom wird durch die korrekte Verteilung der Zellen auf die einzelnen physikalischen Übertragungswege, die unter der ATM-Schicht liegen, vorgenommen. Das Switching wird durch Umsetzung der VPI/VCI-Eingangskennung in die entsprechende Ausgangskennung erreicht. Dabei ändert sich jedes Mal der Header, wodurch die Header Error Control (HEC) jedes Mal neu berechnet werden muss.

Das Traffic Policing ist ebenfalls eine wichtige Aufgabe dieser Schicht. Jedes ATM-Endsystem geht mit dem ATM-Netz einen Traffic Contract ein. Dieser beschreibt die Beschaffenheit der Datenkommunikation. Der Verkehrsvertrag enthält eine Beschreibung anhand von Eigenschaften, wie die Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) oder die maximal mögliche Burst-Länge. Die Überprüfung der Einhaltung dieses Traffic Contract, der einen Quality-of-Service (QoS) garantiert, und das Ergreifen von Gegenmaßnahmen bei Verletzung wird Traffic Policing genannt.

Mit Traffic Shaping ist hingegen der Prozess gemeint, den eine Endstation durchführen kann, um einen Datenstrom zu erzeugen, der auf den Traffic Contract angepasst wird. Das heißt, kurze Überlasten werden zugelassen und erst dann verworfen, wenn das Netz längere Zeit überlastet ist. Dies wird durch das CLP-Feld realisiert.

Zellen, die fehlgeleitet wurden, Fehler im Header oder unbekannte VCI/VPI enthalten, werden von der ATM-Schicht verworfen. Zellen, die mit reserviertem Payload Type markiert sind, werden ebenfalls verworfen. Kann eine Zelle aufgrund von Überlastsituationen nicht weitergeleitet werden, dann wird sie ebenfalls verworfen. Es existieren unterschiedliche Möglichkeiten, den Zellverlust aufgrund von Überlast so gering wie möglich zu halten. Aufgrund von

höheren Schichten (AAL), können zusammengehörige Zellen beispielsweise durch Partial Packet Discard (PPD) gestoppt werden, wenn eine Überbelastung eingetreten ist. Zellen kann man auch verwerfen, wenn eine bestimmte Grenze erreicht wurde, obwohl sich der Switch noch nicht in einer Überlastsituation befindet, diese aber demnächst erreicht wird. Diese Möglichkeit wird durch das Early Packet Discard (EPD) ausgenutzt. Beide Methoden kann man daher als Überwachungsmechanismen der Verkehrslast bezeichnen. EPD ist dabei das effektivere Verfahren, da es vor einer Überlast bereits reagiert. Es wird in Verbindung mit der Dienstklasse Unspecified Bit Rate (UBR) eingesetzt.

Anpassungsschichten (AAL-Typen)

Die Anpassungsschicht stellt die Schnittstelle zwischen den höheren Anwendungsschichten und der ATM-Schicht dar. Sie ist dienstspezifisch und lässt sich in zwei Teilschichten gliedern:

- **Segmentation and Reassembly (SAR):** Sie ist das Bindeglied zur ATM-Schicht und tauscht mit dieser auf der Benutzerebene 48-Byte-Nutzdaten aus. Diese werden in Abhängigkeit der Dienstklasse bearbeitet und an die jeweilige Konvergenzschicht weitergeleitet.
- **Convergence Sublayer (CS):** Die Konvergenzteilschicht ist für die Bereitstellung der dienstspezifischen Dienstparameter zuständig. Hierfür wird sie je nach Dienstklasse in eine dienstspezifische Teilschicht[17] und eine gemeinsame Teilschicht[18] unterteilt.

Die Aufgabe der Adaptionsschicht ist es, die Nutzdaten der höheren Schichten und die eigenen, dienstspezifischen Protokollelemente, wie beispielsweise Informationen zur Datensicherung, auf der Senderseite in 48-Byte-Zellen zu segmentieren. Diese Zellen werden von der ATM-Schicht transparent übertragen und im Empfänger in der Anpassungsschicht wieder zum ursprünglichen Nutzdatenstrom zusammengefügt.

Tab. 4.1 ATM-Dienste bezogen auf Dienstklassen und AAL-Typen

Dienste	Klasse A	Klasse B	Klasse C	Klasse D
Zeitbezug	isochron		nicht isochron	
Bitrate	konstant	variable		
Verbindungsart	verbindungsorientiert			verbindungslos
Dienstklasse	CBR	rt-VBR	nrt-VBR	UBR, ABR
AAL-Typ	AAL-1	AAL-2	AAL-3/4 AAL-5	

17 Service Specific Convergence Sublayer (SSCS)
18 Common Part Convergence Sublayer (CPCS)

Um nicht für jeden Dienst eine eigene ATM Adaptation Layer (AAL) definieren zu müssen, wurden alle Dienste in vier Dienstklassen abgebildet. Für die Aufteilung wurden die Übertragungskriterien Zeitbezug, Bitrate und Verbindungsart herangezogen. Die AAL stellt für jede Dienstklasse einen Diensttyp zur Verfügung: [DETK98a]

- **AAL-Typ 1:** verbindungsorientierter, zeitsynchroner Dienst mit konstanter Bitrate; Anwendung in der Sprachübertragung
- **AAL-Typ 2:** Dieser Dienst erlaubt es, Informationen unterschiedlicher Quellen über Mikrozellen in einer ATM-Zelle zu transportieren; Anwendung in der Sprachübertragung mit geringer Datenrate.
- **AAL-Typ 3/4:** verbindungsorientierter oder verbindungsloser, zeitasynchroner Dienst mit variabler Bitrate; Anwendung in der Datenübertragung. Dieser Diensttyp verwendet vier Oktett für Protokollelemente.
- **AAL-Typ 5:** entspricht Diensttyp 3/4 und ist aus der Forderung nach geringerem Overhead entstanden. Aus diesem Grund enthält der Rahmen keine Protokollelemente der Anpassungsschicht. AAL-5 wird ebenfalls für den Transport von Signalisierungsinformationen und IP-over-ATM-Verfahren verwendet.

4.1.4 Anpassung von IP und ATM

Um im lokalen Netzwerk oder Weitverkehrsbereich TCP/IP-Protokolle über ATM-Netzwerke übertragen zu können, mussten einige Spezifikationen und Anpassungen an ATM vorgenommen werden. Das liegt an den völlig unterschiedlichen Strukturen und Eigenschaften dieser beiden Übertragungsarten.

ATM ist aufgrund seiner virtuellen Verbindungen verbindungsorientiert aufgebaut, besitzt eine eigene Adressstruktur und Routing-Funktionen. Dadurch besitzt ATM den Vorzug der universellen Skalierbarkeit gegenüber anderen Netzstrukturen. Kleine 53-Byte-Zellen transportiert ATM über diese virtuellen Verbindungen. Die ATM-Signalisierung übernimmt dabei den Auf- und Abbau der virtuellen Verbindungen (VCC/VPC). Das heißt, es können Verbindungen manuell über Permanent Virtual Circuits (PVC) oder automatisch über Switched Virtual Circuits (SVC) konfiguriert werden. Dafür wird ein spezieller Verkehrsvertrag ausgehandelt und der QoS festgelegt. Der QoS bei ATM definiert u.a. Bandbreite, Verlustrate und Jitter, um QoS zu garantieren. Für den Verbindungsaufbau wird die Signalisierung benötigt.

Das Internetprotokoll (IP) ist hingegen verbindungslos realisiert. TCP/IP leitet Datenpakete auf Hop-by-hop-Basis (Netzwerk-zu-Netzwerk), unabhängig vom darunter liegenden Netzwerk, weiter. Fehlererkennung und -korrektur sowie Erhaltung der Sendereihenfolge und Duplikatserkennung werden nicht durch IP angeboten und müssen durch Protokolle höherer Schichten wie über das Transmission Control Protocol (TCP) vorgenommen werden. TCP/IP ist

inzwischen für fast alle Rechnerplattformen und Betriebssysteme verfügbar und hat sich aufgrund seiner Verbreitung durch das Internet als Quasi-Standard durchgesetzt. Weiterentwicklungen in den letzten Jahren haben das Protokoll immer weiter verbessert.

MAC[19]-Protokolle für Shared-Medium-LANs arbeiten ebenfalls nicht verbindungsorientiert. Somit sind keine Quittierungsmechanismen beim Empfang implementiert worden. Verlorene Datenpakete müssen durch höhere Protokollebenen wiederholt angefordert werden, wie das bei dem TCP-Protokoll geschieht. LAN-Netze übertragen Daten über ein physikalisches Medium, wodurch die Information durch Zugriffsmechanismen für jede angeschlossene Station zur Verfügung steht. Broadcast-Funktionen sind nötig, da jede Station mit den anderen angeschlossenen Stationen kommunizieren muss. Dies geschieht durch eine spezielle Angabe der Zieladresse, wodurch sich eindeutige Kommunikationsbeziehungen zu anderen Netzteilnehmern aufbauen lassen.

Diese Eigenschaften stehen im starken Gegensatz zu den ATM-Mechanismen. Es lassen sich aber Broadcast-Nachrichten auch über viele virtuelle Verbindungspfade realisieren. Dies wird durch Signalisierungsmechanismen mit Hilfe von Zuordnungstabellen ermöglicht. Zusätzlich zu den verschiedenen Eigenschaften treten Probleme im Bereich unterschiedlicher Übertragungsgeschwindigkeiten und inkompatibler Paket- bzw. Zellenformate auf. Diese Schwierigkeiten können heutzutage jedoch durch leistungsfähige Router überwunden werden.

Zwei Institutionen sind bei der Entwicklung von Anpassungsmöglichkeiten zwischen IP und ATM entscheidend. Die Internet Engineering Task Force (IETF) arbeitet in der Arbeitsgruppe Internetworking Over NBMA (ION) an Verfahren, IP an NBMA[20]-Netze wie ATM, X.25 und Frame Relay anzupassen. Weiterhin ist das ATM-Forum[21], ein Zusammenschluss aus Herstellern, Universitäten, Entwicklern, Dienstanbietern und Nutzern, gegründet worden. Ziel dieses Forums ist die schnellere Spezifizierung von ATM-Standards mit Hilfe der Industrie, um die Arbeiten der International Telecommunication Union (ITU) zu beschleunigen. Um die vermittlungstechnisch völlig unterschiedlichen Welten von IP und ATM miteinander zu verbinden, sind aus diesen Arbeitsgruppen mittlerweile die folgenden Verfahren hervorgegangen:

- Classical IP (RFC-2225)
- LAN-Emulation (LANE)
- Multi-Protocol-over-ATM (MPOA)

19 Media Access Control
20 Non-Broadcast Multiple Access
21 http://www.atmforum.com

Einkapselung von IP-Paketen

Die IETF veröffentlichte 1993 die erste verfügbare Spezifikation zur Übertragung von LAN-Daten über ATM-Netzwerke. Dadurch wurde es erstmals möglich, TCP/IP-Protokolle für ATM-Netzwerke zu nutzen. Das Dokument RFC-2684 definierte, wie unterschiedliche Protokolle in ATM-Zellen gepackt und verschickt werden können. Dabei werden in dieser Spezifikation zwei grundsätzlich verschiedene Methoden zur Einkapselung von LAN-Datenpaketen beschrieben:

Logical Link Control (LLC) Encapsulation: Unterschiedliche Protokolle können über eine einzelne Verbindung transportiert werden. Die eingekapselten Pakete lassen sich dabei durch einen LLC/SNAP[22]-Header identifizieren. Eine weitere Auswirkung der LLC/SNAP-Einkapselung ist es, dass alle Verbindungen, die eine solche Einkapselung verwenden, an der LLC-Schicht innerhalb der Endsysteme enden, da hier das Paketmultiplexen stattfindet.

VC Based Multiplexing: Bei der VC-Multiplexmethode wird nur ein einzelnes Protokoll über eine ATM-Verbindung inklusive des an der Verbindungskonfiguration identifizierten Protokolls, übertragen. Folglich ist kein zusätzliches Multiplex- oder Pakettypenfeld erforderlich, obwohl dem eingekapselten Paket ein Blockfeld vorangestellt werden könnte.

Das LLC-Verfahren ist im Gegensatz zum VC-Multiplexing für IP-over-ATM-Verfahren entscheidend, weshalb es hier genauer erläutert wird. Hier werden alle Datenpakete der LAN-Protokolle durch einen zusätzlichen Header ergänzt. Die Vorgehensweise wird in Abb. 4.5 am Beispiel der Einkapselung eines IP-Datenpakets verdeutlicht. Das LLC-Feld ist für die Einkapselung aller möglichen Schicht-3-Protokolle vorgesehen. Das SNAP-Feld wird nur in bestimmten Fällen verwendet und muss durch einen entsprechenden Wert für das LLC-Feld angekündigt werden. Für den Transport von IP-Datenpaketen ist diese SNAP-Erweiterung fest definiert. Weitere LLC-Einkapselungs-Verfahren für andere Protokolle sind vorhanden und können in der Spezifikation RFC-2684 nachgelesen werden.

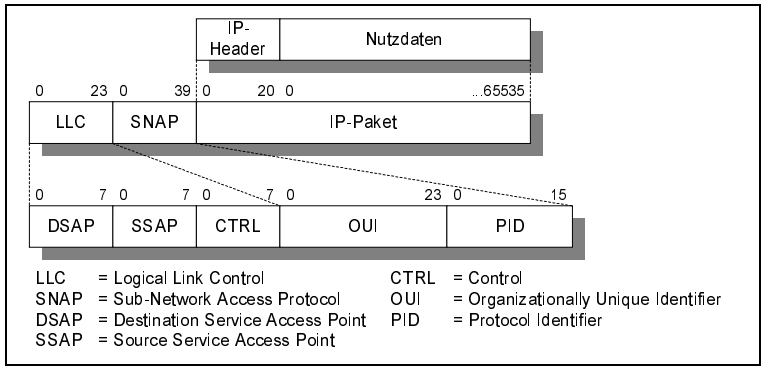


Abb. 4.5 IP-Einkapselung

22 Sub-Network Access Protocol (SNAP)

Das LLC-Feld hat eine Länge von 24 Bit und lässt sich in drei Oktett-Felder unterteilen. Für geroutete Protokolle muss das LLC-Feld auf 0xAA-AA-03 gesetzt werden, um so einen Unnumbered Information (UI) PDU[23] zu kennzeichnen und anzuzeigen, dass ein SNAP-Header folgt. Das SNAP-Feld wiederum besteht aus 40 Bits, die sich auf zwei Felder aufteilen. Die Werte des 24 Bit Organizationally Unique Identifier (OUI) geben an, um welches Netzwerk es sich handelt. Der 16 Bit Protocol Identifier (PID) bezeichnet die Protokollart. Für IPv4-Daten muss für den gesamten SNAP-Header der Wert 0x00-00-00-08-00 gesetzt werden. Wurde der LLC/SNAP-Header angefügt, so kann das gesamte IP-Datenpaket an die AAL-Schicht übergeben werden. Hier werden die LLC/SNAP-Pakete in einen AAL-5-Rahmen verpackt, für die ATM-Schicht segmentiert und dann in ATM-Zellen übertragen. [GRHE99]

Classical IP (CLIP) Für die Übertragung von IP-Daten über ATM müssen die IP-Adressen auf ATM-Adressen abgebildet werden. RFC-2225 definiert das CLIP-Verfahren als eine Möglichkeit zur Adressumsetzung. Bei CLIP werden Logical IP Subnets (LIS) auf einem ATM-Netz definiert. Ein LIS besteht aus den angeschlossenen Endeinrichtungen und Routern in diesem Netz. Abb. 4.6 verdeutlich die CLIP-Architektur und den notwendigen Einsatz von Routern, auch wenn alle LIS in einem einzelnen physikalischen ATM-Netzwerk beheimatet sind.

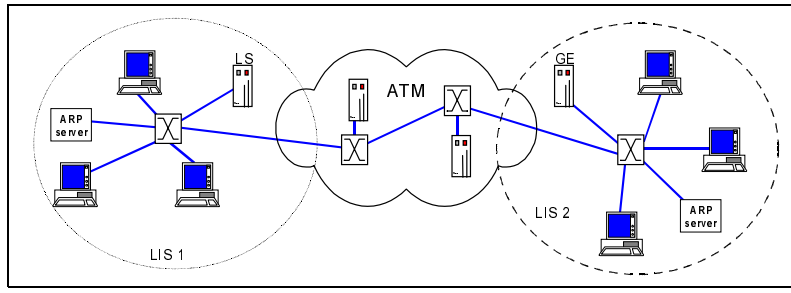


Abb. 4.6 CLIP-Datenaustausch zwischen zwei logischen Subnetzen

Für die eigentliche Adressauflösung, die sich an dem im LAN-Bereich eingesetzten Address Resolution Protocol (ARP) orientiert, muss zwischen den beiden Verbindungstypen im ATM unterschieden werden:

- **Switched Virtual Circuit (SVC):** Werden im ATM-Netz SVC-Verbindungen verwendet, so muss jedes LIS zusätzlich mit einem ATMARP[24]-Server ausgestattet werden. Dieser Server dient ausschließlich der Adressauflösung. Will ein Endsystem Daten an eine IP-Adresse senden, wird zunächst die eigene ATMARP-Tabelle nach einem entsprechenden Eintrag durchsucht. Ist kein Eintrag vorhanden, wird eine Request-Mitteilung an den

23 Packet Data Unit
24 Asynchronous Transfer Mode Address Resolution Protocol

Server gesendet, der mit der ATM-Adresse des Zielsystems antwortet. Nun kann eine Punkt-zu-Punkt-Verbindung mittels ATM aufgebaut werden.

- **Permanent Virtual Circuit (PVC): W**erden ausschließlich PVC-Verbindungen eingesetzt, so sind keine weiteren Elemente im LIS notwendig. Zur Adressauflösung wird jedem PVC-Ende eine IP-Adresse zugeordnet. Dafür werden über alle PVCs InATMARP[25]-Nachrichten an die angeschlossenen Endeinrichtungen gesendet, die dann mit der eigenen IP-Adresse über die gleichen PVCs antworten. Auf diese Weise kann jedes Endsystem für die eigenen PVCs eine ATMARP-Tabelle aufbauen (siehe Abb. 4.7).

Alle Elemente innerhalb eines LIS besitzen identische Netzwerk-IDs und Subnetzmasken. So wird sichergestellt, dass alle Endeinrichtungen einer LIS direkt über Shortcuts miteinander kommunizieren können. Subnetzübergreifender Verkehr muss allerdings über Router abgewickelt werden. Hierzu müssen die ATM-Zellen im Router wieder zu IP-Datenpaketen zusammengesetzt werden. Der Router wertet dann die IP-Zieladressen aus und vermittelt die erneut eingekapselten IP-Pakete ins nächste LIS weiter. Um auch für subnetzübergreifenden Datenverkehr eine Shortcut-Verbindung aufbauen zu können, wurde das Next-hop Resolution Protocol (NHRP) entwickelt. Es stellt eine Ergänzung zu CLIP dar, mit dem Ziel, den hohen Bearbeitungsaufwand in den LIS-Routern zu umgehen und die Übertragungsraten zu erhöhen.

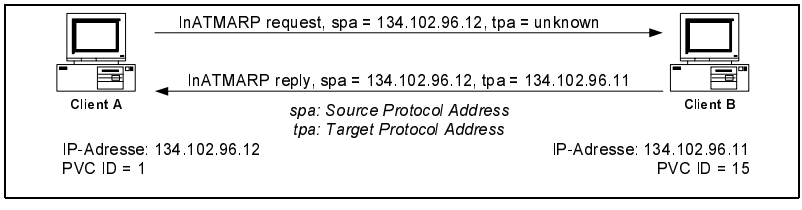


Abb. 4.7 CLIP-Adressauflösung bei PVC

CLIP ist nach der Spezifikation RFC-2225 um eine ATMARP-Serverliste erweitert worden, die es gestattet, mehrere ATMARP-Server in einem LIS-Subnetz aufzubauen. Dadurch lassen sich redundante Strukturen realisieren, was vorher nicht möglich war. Tab. 4.2 zeigt die aktuellen Spezifikationen von CLIP. Dabei hat seit Mitte 1998 keine Weiterentwicklung mehr stattgefunden, da aufgrund des zu hohen Verwaltungsaufwands, fehlender Dienstgütegarantien sowie der eingeschränkten Nutzungsmöglichkeiten CLIP eine Zwischenlösung geblieben ist. [LAHA98]

25 Inverse ATMARP

Die Eigenschaften für IP-Clients in einer ATM-LIS-Konfiguration lassen sich abschließend wie folgt zusammenfassen:

- Alle Clients besitzen die gleiche IP-Netzwerk-/Subnetz-Nummer sowie die gleiche Adressmaske.
- Alle Clients innerhalb eines IP-Subnetzes sind direkt mit dem ATM-Netz verbunden.
- Alle Clients außerhalb eines IP-Subnetzes können nur über einen Router erreicht werden, was durch die Erweiterung mittels NHRP allerdings geändert werden kann.
- Alle Clients eines IP-Subnetzes wandeln mittels ATMARP-Adressen IP-Adressen zu ATM-Adressen um und umgekehrt.
- Alle Clients eines IP-Subnetzes können über ATM mit allen anderen Clients direkt kommunizieren.
- Die Formung virtueller LANs erfolgt bei CLIP auf der OSI-Schicht 3 (logische IP-Subnetze).

Tab. 4.2 Aktuelle CLIP-Spezifikationen der IETF

RFC-Nummer	Titel	Autoren	Datum	Status
RFC-2336	Classical IP to NHRP Transition	J. Luciani	Juli 1998	Informational
RFC-2320	Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB)	M. Greene, J. Luciani, K. White, T. Kuo	April 1998	Standard
RFC-2225	Classical IP and ARP over ATM	M. Laubach, J. Halpern	April 1998	Standard
RFC-1919	Classical versus Transparent IP Proxies	M. Chatel	März 1996	Informational

Next-hop Resolution Protocol (NHRP)

NHRP nach RFC-2332 ordnet jedem LIS einen NHRP-Server (NHS) zu. Jeder NHS besitzt Next-hop Resolution Cache Tables, welche die IP-/ATM-Adressenabbildung für die bekannten IP-Knoten, also die Elemente im eigenen LIS, vornehmen. Im Fall einer SVC-Umgebung kann so auf einen ATMARP-Server verzichtet werden. Des Weiteren verfügt der NHS über Einträge der IP-Präfixe von Routern, die über diesen NHS erreicht werden können. Will ein Endsystem IP-Pakete an eine andere Station übermitteln, so wird eine Request-Nachricht an den eigenen NHS gesendet. Gehört das Ziel zu den vom NHS verwalteten Knoten, wird mit der ATM-Adresse geantwortet. Es kann sofort ein Shortcut von der Quelle zum Ziel aufgebaut werden. Kennt der NHS das Ziel nicht, so wird der NHRP-Request unter Verwendung der IP-Präfix-Einträge zum nächs-

ten NHS in Zielrichtung weitergeleitet. Diese Vorgehensweise wird so lange wiederholt, bis der NHS erreicht wurde, der das Ziel in seinem LIS verwaltet. Die ATM-Adresse des Ziels wird über die aufgebaute NHS-Kette an die Quelle gesendet und ein Shortcut kann über mehrere LIS hinweg aufgebaut werden, wie Abb. 4.8 verdeutlicht. [LKPC+98]

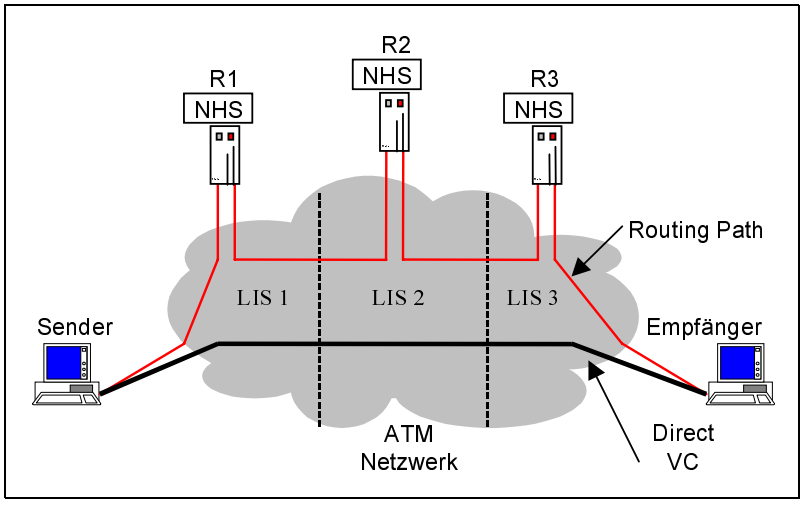


Abb. 4.8
NHRP-Funktionalität

NHRP ist innerhalb jeder administrativen Region anwendbar, erlaubt aber direkte Verbindungen nur am Zutrittspunkt einer anderen administrativen Region. Bei NHRP handelt es sich um eine Client-/Server-Architektur, bei dem der Server den Dienst der Adressauflösung den Clients anbietet. Statt eines ARP-Servers wurde beim NHRP die Vorstellung von einem NHRP-Server (NHS) verwirklicht. Die Knoten sind mit der ATM-Adresse ihres NHS konfiguriert und melden anschließend ihre eigenen ATM- und IP-Adressen beim NHS mit Hilfe von Registrierungspaketen an, sodass das NHS seine Cache-Tabellen erweitern kann. Dafür hat man das ARP-Protokoll zu dem NBMA Address Resolution Protocol (NARP) erweitert, wodurch auch klassische ARP-Dienste weiter Verwendung finden. Weiterhin sind NHS in der Lage, auch andere Protokolle zu unterstützen, um beispielsweise Routing-Informationen über das Subnetz und jenseits seiner Grenzen zu verteilen. Als Client ist der NHRP Client (NHC) vorhanden, der die NHRP-Anfragen zum nächsten NHS formuliert. Der NHC unterhält dabei ebenso einen Cache mit Adresspaaren. Dieser Speicher wird durch NHRP-Antwortpakete, durch manuelle Konfiguration oder weitere Mechanismen außerhalb der NHRP-Spezifikation initialisiert und aktualisiert. Der so genannte Single-Point-of-Failure (SPOF) wird dadurch vermieden, dass der NHC einer Domäne sich gleichzeitig bei mehreren NHS registrieren lassen. Bei Ausfall eines Servers bleibt somit die Information erhalten.

Die Hauptfunktion von NHRP ist also, die Adressauflösung effizienter und vor allem protokollunabhängig zu bewerkstelligen, als das bisher durch vergleichbare Verfahren ermöglicht wurde, sowie direkte Shortcut-Verbindungen über Subnetze hinweg aufbauen zu können. Dadurch wird Endgeräten und Routern der direkte Datenaustausch über SVCs gestattet, die sich anhand der ATM-Adressen identifizieren. Dazwischen liegende IP-Router können umgangen werden. NHRP ist jedoch kein Routing-Protokoll. Es benötigt Standard-IP-Routing-Protokolle wie beispielsweise OSPF[26], um seine NHRP-Nachrichten durch das Netzwerk senden zu können. [DETK00a]

LAN-Emulation (LANE)

Die LAN-Emulation (LANE) ist eine Möglichkeit, ATM-Produkte und ATM-Netzwerke in vorhandene LAN-Topologien zu integrieren. Sie stellt die universellere Methode der Netzanpassung dar, da sie die MAC-Schicht eines lokalen Netzwerks vollständig emuliert. Dadurch wird bestehenden LAN-Anwendungen, ohne zusätzliche Einstellungen über ATM-Netzwerke hinweg, eine LAN-zu-LAN-Kommunikation vorgetäuscht. Im Januar 1995 wurde die erste LANE-Spezifikation vom ATM-Forum vorgelegt, wodurch die Anbindung von lokalen Netzwerken über Protokolle wie IPX, AppleTalk, IP usw. möglich wurde. Folgende Einsatzfälle werden dabei durch LANE abgedeckt:

- Verknüpfung traditioneller LANs über ein ATM-Netzwerk
- Verbindung zwischen Endsystemen an traditionellen LANs mit denen an ATM-Netzwerken
- Einsatz von LAN-Anwendungen zwischen Endsystemen, die direkt am ATM-Netzwerk angeschlossen sind

Weiterhin hat LANE die Aufgabe, vorhandene LAN-Treiber zu benutzen, da diese für traditionelle LANs auch in Zukunft weiter verwendet werden. Somit würde ein Endsystem in der Lage sein, über eine standardisierte MAC-Schnittstelle, wie Network Driver Interface Specification (NDIS) und Open Data-link Interface (ODI), auf ein bestimmtes LAN zuzugreifen, ohne eine Anpassung von Protokollen und Programmen vornehmen zu müssen. Aus diesem Grund setzt LANE unterhalb der Schicht 3 an und emuliert den MAC-Treiber eines herkömmlichen LANs. Die Verbindungssysteme zwischen den traditionellen und den emulierten LANs (ELAN) erfüllen dabei die Brückenfunktion: Transparent oder Source Routing. Ein ELAN besteht aus vier wesentlichen Komponenten, die in Abb. 4.9 dargestellt sind:

- LAN Emulation Clients (LEC)
- LAN Emulation Server (LES)
- LAN Emulation Configuration Server (LECS)
- Broadcast and Unknown-Server (BUS)

26 Open Shortest Path First

Das bedeutet, dass LANE auf einem Client-Server-Prinzip basiert, wobei es einen LAN Emulation Service gibt, der die zentralen Dienste der LANE bereitstellt. Jedes Endsystem, welches die LANE nutzen möchte, muss einen LAN Emulation Client (LEC) besitzen. Dieser ist meistens innerhalb entsprechender Treibersoftware auf einer ATM-Karte implementiert. Dabei besteht die Hauptaufgabe des LEC in der Bereitstellung der MAC-Schnittstelle für höhere Protokolle. Das heißt, er muss die MAC-Adressen auflösen, die Verbindungen abbauen und danach die Daten über ATM versenden. Für den LEC sind für diesen Fall zwei Adressarten vorhanden: 6-Byte-MAC-Adressen und 20-Byte-ATM-Adressen.

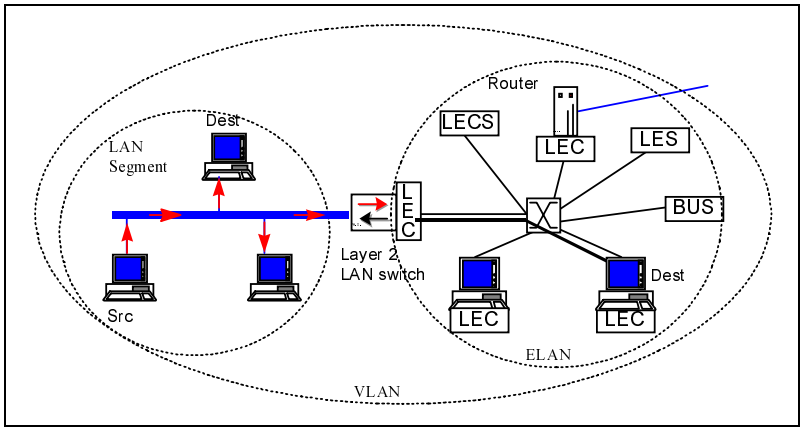


Abb. 4.9
LANE-Architektur mit ELAN und VLAN

Abb. 4.9 zeigt die drei wesentlichen Komponenten des LAN Emulation Services. Dabei dient der LECS zur automatischen Konfiguration der LECs und ermöglicht die einfache Umsetzung virtueller LANs von einer zentralen Stelle aus. Das heißt, es wird die Zugehörigkeit der LAN Emulation Server zu verschiedenen LANs zentral verwaltet. Die Konfigurationsinformationen werden in einer Datenbank gespeichert und mit Hilfe der Clients einem emulierten LAN zugeordnet. Der LAN Emulation Server ist die so genannte Schaltzentrale der LANE. Seine Aufgabe ist es, die LECs zu unterstützen, indem ATM-Adressen für vorgegebene MAC-Adressen ermittelt werden. Zusätzlich registriert er alle vorhandenen LECs. Der jeweilige LEC sendet dazu die eigene LAN-MAC-Adresse, die dazugehörige ATM-Adresse sowie Wegewahlinformationen an den LES. Dieser untersucht daraufhin die eigene Adresstabelle nach der ATM-Adresse. Findet er die ATM-Zieladresse, kann die Datenübertragung beginnen. Zur Ermittlung der ATM-Adresse wird das LAN Emulation Address Resolution Protocol (LE-ARP) verwendet, da ein LES nicht die Aufgabe einer Brücke übernehmen soll und deshalb nicht alle Adressen kennt. Wird die ATM-Zieladresse nicht gefunden, so wird die Adresszuordnung durch eine Broadcast Message

über den Broadcast and Unknown Server (BUS) vorgenommen. Der BUS ist demnach für die Verteilung der Broadcast- und Multicast-Pakete zuständig. Dabei stellt er auch die Pakete zu, denen noch keine ATM-Verbindung vom LEC zugeordnet wurde. Zusätzlich sind Router-Mechanismen implementiert, die zur Ermittlung des optimalen Weges dienen. Die empfangenen Datenpakete werden vom BUS in der korrekten Reihenfolge an die angeschlossenen LECs weitergeleitet, um eine Überschneidung mit AAL[27]-5-Datenpaketen von anderen Sendern vermeiden zu können. AAL-1 bis AAL-4 stellen kein Problem dar, weil LANE ausschließlich mit AAL-5 arbeitet. Die logischen Einheiten des LAN Emulation Service lassen sich getrennt sowie gemeinsam in bestehende Systeme integrieren. Als Zielsysteme kommen dabei Workstations, ATM-Switches und Router in Frage. [ATMF95]

LANE bietet eine einfache Möglichkeit, um VLANs[28] zu bilden und ATM mit Legacy LANs kombinieren zu können. Praktisch alle Netzwerkhersteller implementieren LANE-Client-/Server-Funktion in diversen Bridges und Switches. Die Einschränkung von LANE liegt darin, dass ein ELAN immer nur von einem Typ sein kann, also entweder Ethernet oder Token-Ring. Weiterhin erlaubt LANE keine Subnetze und damit naturgemäß keine Integration von Routern beziehungsweise Routing-Funktionen. ELANs sind unstrukturierte, flache LANs, ähnlich wie die VLAN-Spezifikation auf OSI[29]-Schicht 2 nach IEEE 802.1q.

LANE ist in der Version 2.0 erst im Laufe des Jahres 1999 endgültig spezifiziert worden. Die Schnittstellen zwischen Client und Server[30] wurde dadurch endlich mit der Schnittstelle LANE Network-to-Network Interface (L-NNI) in der Version 2.0 komplettiert. L-UNI ist kompatibel zu der bisherigen Version und enthält weitere Funktionen: [ATMF97a]

- Multiplexen von ELANs über ATM-Verbindungen
- Verbesserte Unterstützung von Multicast-Paketen
- Eingeschränkte Unterstützung von QoS
- LECS wird zu einem allgemeinen Konfigurationsserver umfunktioniert

L-NNI liegt in der Version 2.0 von LANE zum ersten Mal vor. Hauptziel war es, verteilte Server zu ermöglichen und die Kommunikation zwischen diesen zu verbessern. Dadurch können redundante ELANs aufgebaut werden, der BUS begrenzt das Gesamtsystem nicht weiter und unterschiedliche Hersteller können eingesetzt werden. Das Problem dieses Overlay-Ansatzes ist es allerdings, dass beide Netze (IP und ATM) völlig unabhängig voneinander agieren und sich die unterschiedlichen Mechanismen (z.B. Routing) gegenseitig behindern

27 ATM Adaptation Layer
28 Virtual Local Area Networks
29 Open System Interconnection
30 LAN-Emulation User Network Interface (L-UNI)

können. Der Einsatz von garantierten Dienstgüteparametern spielt hier ebenfalls keine Rolle und kann nicht eingesetzt werden. Es steht nur Unspecified Bit Rate (UBR) zur Verfügung. [ATMF99a]

Multi-Protocol-over-ATM (MPOA)

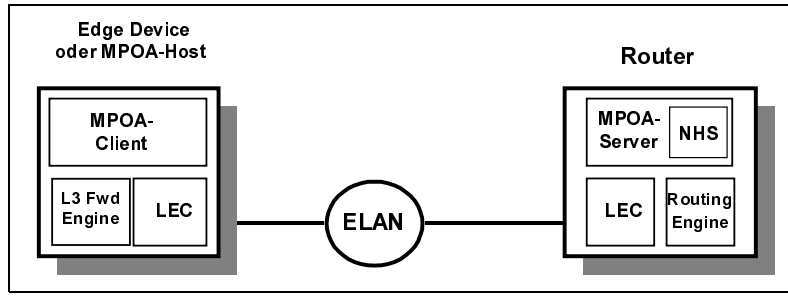
Die vorhandenen Möglichkeiten von IP-over-ATM-Verfahren wie CLIP oder LANE ermöglichten bisher nur den Aufbau einzelner Subnetze[31]. Wenn mehrere Subnetze auftreten, sind Router für die Kommunikation erforderlich. Dadurch kann die Effizienz erheblich beeinflusst werden, da Router den Datendurchsatz begrenzen und Verzögerungen durch Protokollbearbeitung hervorrufen. Im Gegensatz zu Bridges, die auf OSI-Schicht 2 (Datensicherungsschicht, Media Access Control Layer) arbeiten und ebenfalls die traditionellen LANs und LAN-Segmente verbinden, besitzen die Router auf OSI-Schicht 3 (Vermittlungsschicht, Network Layer) zusätzlich Filter- oder auch Firewall-Funktion. Router legen somit fest, welcher Datenverkehr sich innerhalb einer bestimmten Domäne befinden soll und welcher nach außen gehen beziehungsweise von dort kommen darf. Innerhalb der Internet-Protokoll-Umgebung bedeutet dies, dass Applikationsdaten von einem Endgerät zuerst in IP-Pakete unterteilt werden (20 Byte - 65 KByte) und danach in Ethernet-Rahmen, falls es sich um ein solches LAN handelt. Befinden sich dabei Sender und Empfänger in derselben Domäne[32], wird der Empfänger direkt per Ethernet/MAC- und IP-Adresse angesprochen und die Ethernet-Rahmen werden zuletzt wieder zu IP-Paketen zusammengefügt. Bei unterschiedlichen Domänen muss dabei die Kommunikation über mindestens einen Router erfolgen. In diesem Fall bestimmt nur noch die IP-Adresse den Empfänger. Die MAC-Adresse wählt nur den nächsten Router auf dem Routing-Pfad aus. Dabei speichert jeder Router die Ethernet-Rahmen in seinem Eingangspuffer, setzt diese zu IP-Paketen zusammen und stellt durch die IP-Adresse fest, über welchen Router-Port der Empfänger erreicht werden kann. Adresstabellen im Router beinhalten die IP-Adressen und Portnummern und müssen durch Routing-Protokolle (OSPF, RIP etc.) laufend aktualisiert werden bzw. sind bei der Weiterleitung der Pakete behilflich. Letztendlich werden die IP-Pakete wieder in MAC-Rahmen zerlegt, mit einer neuen MAC-Empfangsadresse versehen und über den entsprechenden Ausgangsport am Router in Richtung Empfänger weitergeleitet. Dieser Vorgang wird bei allen Routern auf dem Übertragungspfad wiederholt, bis schließlich die richtige Domäne vom IP-Paket erreicht wurde, in der MAC- und IP-Adresse wieder übereinstimmen. Hinzu kommt noch der Spanning-Tree-Algorithmus, um die aktiven und nicht aktiven Ports zu überwachen, falls der Router auch noch die Funktionen einer Bridge übernimmt. Dieser komplexe Vorgang hat natürlich erhebliche Verzögerungszeiten zur Folge, die gerade bei

31 Eine Ausnahme stellt CLIP nur dar, wenn das Protokoll NHRP zusätzlich integriert wird.

32 Auch IP-Subnetz genannt.

der heutigen Anforderung nach Echtzeitfähigkeit der Netze nicht mehr hingenommen werden können. Latenzzeiten von 100 ms pro Router können so leicht entstehen. Durch das verwendete Store-and-Forward-Verfahren ist der Datendurchsatz von Routern auf Werte zwischen 100 000 und 1 Millionen Pakete pro Sekunde begrenzt[33].

Abb. 4.10 MPOA-Komponenten



Multi-Protocol-over-ATM (MPOA) versucht diese Nachteile durch neue Konstruktionen, die aus vielen Subnetzen bestehen, und so genannten Shortcuts (direkte Verbindungen) zwischen unterschiedlichen IP-Subnetzen auszuschalten. Da MPOA zwischen der Netzwerkschicht 2 und 3 aufsetzt, wird die Ausnutzung von ATM-Eigenschaften ermöglicht. Durch das neue Szenario eines virtuellen Routers und durch Integration bestehender Ansätze versucht MPOA, die Router-Funktionen Forwarding und Routing räumlich zu trennen. Bisherige Router-Funktionen ermöglichten den Informationsaustausch über die Wegewahl zum Empfänger (Routing) und der anschließenden Übertragung der Daten (Forwarding). Beim Ansatz des virtuellen Routers werden so genannte Forwarder über ein standardisiertes Protokoll zentral von Router-Servern gesteuert. Die Routing-Funktionen werden von den Router-Servern ausgeführt. Durch diese Trennung sind erhebliche Kosteneinsparungen durch Managementvereinfachungen über die Zentralisierung der Router-Funktionen möglich. Zusätzlich lassen sich einfache Forwarder mit einem geringeren technischen Aufwand realisieren.

MPOA setzte somit zuerst den Grundsatz des Layer-3-Switchings um: Switching wenn möglich, Routing wenn nötig. MPOA liegt seit Mitte 1997 in der Version 1.0 über das ATM-Forum als Spezifikation vor. Der Standard LANEv2.0 (L-UNI Spezifikation) des ATM-Forums sowie die Paketeinkapselung nach RFC-2684 und das Next-hop Resolution Protocol (NHRP) der IETF sind weitere Spezifikationen, aus denen MPOA besteht. LANE ermöglicht die Einbindung traditioneller Ethernet- und Token-Ring-LANs in ein ATM-Backbone/Campusnetz. LANE beschreibt, wie Domänen über ein ATM-Netz eingerichtet werden

33 Das haben Messungen unterschiedlicher Messlabore nachweisen können bzw. wurde durch eigene Tests belegt.

können, ATM also als Ersatz für herkömmliche LAN-Segmente dient. Die Paketeinkapselung stellt eine Mehrfachprotokollverpackung über eine ATM-Verbindung dar, während das NHRP-Protokoll die Shortcut-Benutzung beim Versenden von IP-Paketen über ATM erlaubt. Dadurch ist es möglich, IP-Pakete direkt über wirkliche End-to-end-Verbindungen durch Switched Virtual Circuits (SVC) an andere Systeme in anderen Netzen weiterzuleiten, wodurch effiziente Methoden zur Verbindung virtueller Netze geschaffen wurden.

Die Grundarchitektur von MPOA besteht ebenfalls aus einer Client-/Server-Umgebung und enthält MPOA-Client (MPC), MPOA-Router (MPR) und MPOA-Server (MPS), die über das ATM-Netzwerk miteinander verbunden sind. Der MPC kann sowohl ein Router mit einer speziellen ATM-Schnittstelle als auch ein Layer-2-Switch sein und besitzt die folgenden Funktionen:

- MPC-Funktionen auf einem Edge Device oder ATM-Host stellen Anfangs- und Endpunkt für Shortcut-Verbindung dar.
- Ermittelt Traffic Flows: Fordert von dem MPOA-Server Informationen über das Zielsystem und die Informationen zum Aufbau eines Shortcut (falls er möglich oder sinnvoll ist).
- Setzt Shortcut-Verbindung auf und realisiert Layer-3-Forwarding.
- Speichert Shortcut-Informationen im lokalen Cache (Alterung über Time-Outs).

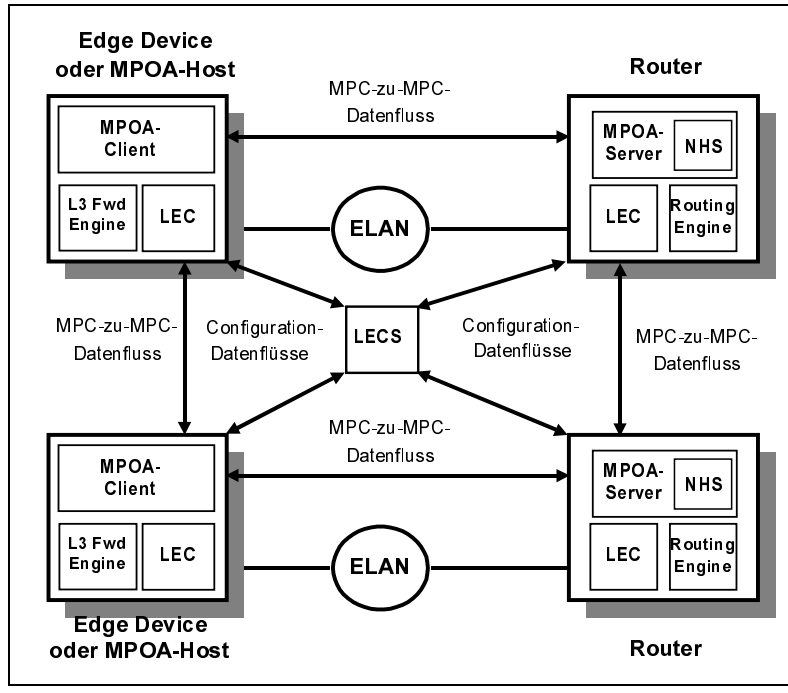
Der MPOA-Router (MPR) besitzt hingegen Funktionen, um Subnetze auf Netzwerkebene (Layer 3) auf ATM-Netze abbilden zu können. Dafür wird zwischen den MPOA-Servern das Protokoll NHRP[34] zur Adressauflösung verwendet. Zusätzlich werden Adressinformationen bezüglich Netzwerk-, MAC[35]- und ATM-Adressen verwaltet sowie bekannte Routing-Protokolle wie OSPF[36] und RIP[37] zur Kommunikation mit traditionellen Routern weiterverwendet. Dabei kann der MPOA-Router sowohl Bestandteil eines ATM-Switches oder als eine getrennte Komponente am ATM-Netz implementiert werden. Es können auch mehrere MPR in einem ATM-Netz implementiert werden. Weiterhin realisiert der Router auch einige zusätzliche Funktionen, zu denen u.a. der MPS gehört. Der MPS ist die logische Komponente eines MPR und hat folgende Merkmale:

- Beinhaltet einen Next-hop Server (NHS).
- Identifiziert im Zusammenspiel mit NHS und Routing Engine einen Pfad zur ATM-Zieladresse.
- Stellt MPCs Layer-3-Forwarding-Informationen zur Verfügung (inkl. Layer-2-Encapsulation).

34 Next-hop Resolution Protocol
35 Medium Access Control
36 Open Shortest Path First
37 Routing Information Protocol

Die logischen Komponenten werden über permanente ATM-Verbindungen miteinander verbunden. Dadurch entsteht ein logisches Routing-Netz, in dem die MPS als Knoten und die MPC als Endsysteme arbeiten. Die Steuerungsinformationen werden dabei aufgrund der Protokolladresse übermittelt. Diese können dann den MPCs zugeordnet werden, die mit den MPOA-Systemkomponenten verbunden sind. Deshalb kann man die vernetzten MPS als logisches Routing-Netz bezeichnen, welches in ein physikalisches ATM-Netz eingebettet ist. Abb. 4.11 zeigt den Informationsfluss innerhalb eines MPOA-Netzes auf und verdeutlicht die Funktion. [ATMF97b]

Abb. 4.11 Informationsfluss in einem MPOA System



Bisher litten die klassischen IP-over-ATM-Verfahren an Begrenzungen, die ihnen von Host-Anforderungen auferlegt werden. Diese schließen Cut-Through-Routing aus und umgehen Router-Hops zwischen Knoten auf demselben ATM-Netz, die aber innerhalb von zwei verschiedenen IP-Subnetzen (LIS) vorhanden sind. Eine Arbeitsgruppe der IETF, welche sich Routing Over Large Clouds (ROLC) nennt, arbeitet an Protokollen, die diese Begrenzungen aufheben sollen. Nach der Betrachtung von zahlreichen unterschiedlichen Ansätzen hat die Gruppe die Arbeit an dem Next-hop Resolution Protocol (NHRP) abgeschlossen.

NHRP ist eine Grundfunktionalität von MPOA und baut auf klassischen IP-Modellen auf. Es ersetzt die LIS-Begriffsvorstellung durch ein logisches Non-Broadcast-Multi-Access-(NBMA)-Netzwerk. Das heißt, dass eine Netztechnologie (wie ATM, Frame Relay oder X.25), die den Zugriff vieler Geräte auf ein Netz ermöglicht, aber nicht die Verwendung von Broadcast-Mechanismen erlaubt, praktisch auf einem LAN angeordnet ist. Solch ein Netzwerk besteht aus einer Gruppe von Knoten, die in demselben NBMA-Netz vorhanden sind. Dabei findet die Kommunikation zwischen den unterschiedlichen Knoten auf direktem Wege statt. Ein einzelnes NBMA-Netz kann mehrfache administrative Domänen unterstützen, wobei jede über direkte Verbindungen mit der anderen kommunizieren kann. Dabei können auch bestimmte Verbindungen ausgeschlossen werden.

MPOA liegt seit 1999 in der Version 1.1 vor. Es bietet zum ersten Mal eine effiziente Möglichkeit im LAN, ATM-Vorzüge direkt auf IP zu übertragen. Dabei nimmt die Idee des virtuellen Routers konkretere Strukturen an. MPOA setzt auf LANEv2.0 auf und erweitert diesen Ansatz im Wesentlichen um NHRP und QoS, was allerdings nur bei direkter ATM-Verbindung an das Kernnetz möglich ist. Dadurch harmonisieren IP und ATM miteinander und können sich gegenseitig ergänzen. Beispielsweise ist das dynamische Routing-Verfahren P-NNI[38] von ATM dem dynamischen IP-Routing OSPF sehr ähnlich und kann adaptiert werden. Da MPOA immer auf ATM aufsetzt, sind langfristig Mechanismen vorhanden, die in der Lage sind, eine effiziente Integration zu gewährleisten. Hinzugekommen sind in der Version 1.1 ebenfalls Sicherheitsmechanismen, die vorher noch offen gelassen wurden. Nachteilig ist allerdings die Unflexibilität hinsichtlich der Layer-2-Technologie bezüglich ATM, da die heutige Entwicklung oftmals in Richtung Ethernet und SDH bzw. Packet-over-SONET zeigt. Weiterhin ist MPOA schlecht zu skalieren und ist daher nur für LAN-/MAN-Umgebungen geeignet. Im LAN allerdings wird MPOA keine Zukunft mehr eingeräumt, da sich hier inzwischen der simplere Ansatz Gigabit-Ethernet (GE) durchsetzen konnte. Aus diesem Grund und wegen des relativ komplexen Ansatzes haben sich die Hersteller auch weitgehend von MPOA zurückgezogen, bieten diese Funktionalität allerdings in heutigen modernen Switches durchaus an. [ATMF99b]

4.2 Packet-over-SONET (PoS)

Eine weitere Möglichkeit, ein WAN aufzubauen, bietet das Point-to-Point Protocol (PPP). Das von der Internet Engineering Task Force (IETF) standardisierte Protokoll nach RFC-1661 wird durch ständige Erweiterungen und Ergän-

38 Public Network-to-Network Interface

zungen für den Einsatz über unterschiedliche Medien optimiert. An dieser Stelle wird zunächst, ähnlich wie dies vorher bei ATM vorgenommen wurde, ein genereller Überblick über den Aufbau und die Funktionsweise von PPP gegeben. Für den Einsatz von PPP im WAN beschränkt sich die Darstellung auf die Variante PoS, da PoS eine spezielle Form des PPP darstellt und hier explizit mit ATM verglichen werden soll.

4.2.1 Point-to-Point Protocol (PPP)

Der Standard RFC-1661 definiert das Protokoll PPP als eine Methode, um Datenpakete unterschiedlicher Netzwerkprotokolle wie beispielsweise IP oder IPX über Punkt-zu-Punkt-Verbindungen zu übertragen. Diese Punkt-zu-Punkt-Verbindungen können nach den Festlegungen der ITU-T synchrone Bit/Byte-orientierte oder asynchrone Verbindungen sein. In jedem Fall muss die Verbindung Fullduplex-Fähigkeit aufweisen. Als zugrunde liegende Technologien kommen somit beispielsweise ISDN, ATM oder Frame Relay (FR) in Frage. Eine solche Verbindung stellt für den Einsatz des Transportprotokolls PPP die einzige Voraussetzung dar. [SIMP94a]

Das Protokoll PPP gliedert sich in drei Hauptkomponenten:

- Eine Methode, um Datenpakete einzukapseln.
- Ein Steuerungsprotokoll für die Verbindung Link Control Protocol (LCP)
- Ein Steuerungsprotokoll für die Netzwerkschicht Network Control Protocol (NCP)

Einkapselung der Datenpakete

Die Einkapselung von Datenpaketen bei PPP ermöglicht es, mehrere Netzwerkprotokolle gesichert über eine Verbindung zu übertragen. Die Einkapselung besteht hierbei aus zwei Teilen. Im ersten Schritt werden die Datenpakete der Netzwerkprotokolle in einen PPP-Rahmen eingepackt. Abb. 4.12 verdeutlicht den Ablauf der Einkapselung am Beispiel eines IP-Datenpakets. Im zweiten Schritt werden die PPP-Rahmen zur Absicherung gegen Übertragungsfehler in einen weiteren Rahmen eingepasst. Hierfür empfiehlt der Standard RFC-1661 den Rahmen High Level Data Link Control (HDLC). Das Abbilden von PPP-Rahmen in einen HDLC-Rahmen ist in der Spezifikation RFC-1662 definiert.

Das Protokollfeld umfasst ein oder zwei Oktetts (Bytes) und identifiziert das eingekapselte Datenpaket, welches sich im Informationsfeld des PPP-Rahmens befindet. Es ermöglicht auf diese Weise den Transport verschiedener Protokolle über eine Verbindung. Die Struktur des Protokollfelds basiert auf dem Standard ISO3309. Zulässige Werte sind in der Spezifikation RFC-1340 definiert. Für das Einkapseln von IP-Paketen wird der Wert 0021 gesetzt. [REPO92]

Das Payload-Feld besteht aus keinem oder mehreren Oktetts und beinhaltet das im Protokollfeld spezifizierte Datenpaket. Die maximale Länge des Informationsfelds wird durch die Maximum Transmission Unit (MTU) festgelegt. Der Standardwert für die MTU beträgt 1500 Byte, kann aber zwischen

Quelle und Senke vom Standardwert abweichend ausgehandelt werden. Das Padding-Feld ist hingegen notwendig, um das Informationsfeld um die fehlende Anzahl von Bytes bis zur MTU zu ergänzen. Dabei liegt es im Aufgabenbereich der Protokolle, zwischen Daten im Informationsfeld und den Padding-Bytes für die Auswertung zu unterscheiden. [SIMP94a]

Der HDLC-Rahmen beginnt mit dem Flag-Feld, welches ein Byte (Oktett) umfasst und immer die Bitkombination 01111110 aufweist. Jeder HDLC-Rahmen beginnt und endet mit diesem Flag-Muster, wodurch eine Rahmensynchronisation erreicht wird. Zwischen zwei direkt aufeinanderfolgenden Rahmen ist nur ein Flag erforderlich. Da dieses Flag-Muster zufällig auch innerhalb der Nutzdaten vorkommen kann, beschreibt der Standard RFC-1662 eine Methode, um zu verhindern, dass es in einem solchen Fall zu fehlerhaften Rahmenerkennungen kommt.

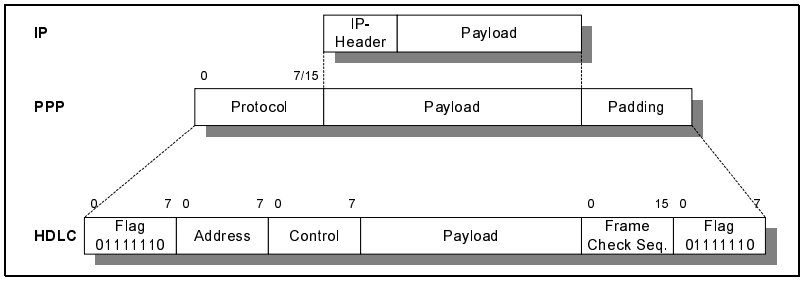


Abb. 4.12 Einkapselung von Paketen bei PPP

Das Adressfeld beinhaltet ebenfalls ein Byte und wird üblicherweise verwendet, um Befehle[39] und Meldungen[40] zu unterscheiden. Im PoS sind allerdings keine individuellen Adressen zugewiesen. RFC-1662 schreibt die Bitsequenz 11111111 vor, welche die All Stations Address (ASA) identifiziert. Abweichende Werte dürfen nicht verwendet werden. Da es sich um Punkt-zu-Punkt-Verbindungen handelt, sind andere Werte auch nicht notwendig. Das Feld Control besteht auch aus einem Byte und kennzeichnet die Art des Blocks. Auch hier ist die binäre Sequenz für PPP vorgeschrieben. Es wird der Wert 00000011, welcher einen UI-Block identifiziert, verwendet. Die Frame Check Sequence (FCS) besteht standardmäßig aus zwei Byte. Das Verwenden der ebenfalls definierten vier Oktetts großen FCS, muss ausgehandelt werden. Die FCS ermöglicht dem Empfänger festzustellen, ob der Rahmen fehlerfrei übertragen wurde. Das FCS-Feld wird beim Sender erzeugt, indem der komplette Rahmen, ausgenommen Flags und FCS, durch ein Generatorpolynom geteilt, und der invertierte Rest als FCS übertragen wird. Der Empfänger kann eine eigene Berechnung des Rahmens vornehmen und die errechnete FCS mit der empfangenen FCS vergleichen. Für die

39 Zieladresse
40 Eigene Adresse

Übertragung von Netzwerkprotokollen sind so nur acht zusätzliche Byte erforderlich. In bandbreite-kritischen Umgebungen können die durch die komplette PPP-Einkapselung zusätzlich benötigten Bytes auf zwei bis vier reduziert werden. [SIMP94b]

Link Control Protocol (LCP)

Das LCP wird eingesetzt, um die Übertragungsstrecke auf- und abzubauen, zu konfigurieren und zu testen. Hierfür verwendet das LCP verschiedene Nachrichten. RFC-1661 unterscheidet drei Klassen von LCP-Nachrichten:

- Nachrichten zum Aufbau und zur Konfiguration der Verbindung
- Nachrichten zum Beenden der Verbindung
- Nachrichten zum Testen der Verbindung und zur Fehlerbeseitigung

Tab. 4.3 LCP-Nachrichten [ORLA00]

Klasse	Code	Typ	Bedeutung
1	1	Configure_Request	Öffnen einer Verbindung
	2	Configure_Acknowledge	Bestätigung des Configure_Request
	3	Configure_Not-Acknowledge	Der Configure_Request und die gewünschten Optionen wurden zwar vollständig erkannt, einige Optionen werden aber nicht akzeptiert
	4	Configure_Reject	Der Configure_Request bzw. einige Optionen wurden nicht erkannt
2	5	Terminate_Request	Schließen der Verbindung
	6	Terminate_Acknowledge	Bestätigung des Terminate_Request
3	7	Code_Reject	Eine LCP-Nachricht mit unbekanntem Code wurde empfangen
	8	Protocol_Reject	Es wurde versucht, ein Protokoll zu initiieren, das nicht unterstützt wird
	9	Echo_Request	Erlaubt Schleifenprüfungen in beide Übertragungsrichtungen für Test- und Performance-Messungen
	10	Echo_Reply	
	11	Discard_Request	Ebenfalls für Testzwecke

Zur Übertragung wird ebenfalls die PPP-Einkapselung verwendet. Hierbei wird genau ein LCP-Paket in das Informationsfeld eines PPP-Rahmens abgebildet. Das Protokollfeld des PPP-Rahmens trägt dann den Wert für LCP: c021. Der Aufbau eines LCP-Pakets wird in Abb. 4.13 gezeigt. Dabei umfasst das Code-Feld ein Byte und identifiziert die einzelnen LCP-Nachrichten der Tab. 4.3. Der

Identifier beinhaltet ebenfalls ein Byte und dient der Zuordnung von Anfragen und Antworten. Das Längenfeld Length gibt die Länge des gesamten LCP-Pakets an, da das Datenfeld unterschiedliche Längen annehmen kann. Außerhalb liegende Bytes werden als Padding-Bytes betrachtet. Das Format des Datenfeldes wird durch die Art des LCP-Nachrichtentyps bestimmt, welcher durch den Code gekennzeichnet wird. Der Aufbau des Datenfeldes eines LCP-Pakets wird in RFC-1661 für alle LCP-Nachrichten eingehend beschrieben. [SIMP94a]

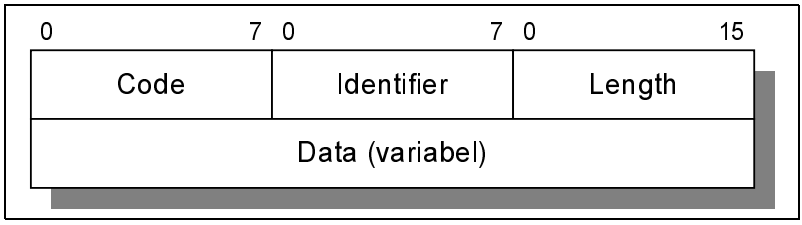


Abb. 4.13 LCP-Paket

Network Control Protocol (NCP)

Das Protokoll NCP wird verwendet, um die Netzwerkprotokolle auf- und abzubauen und zu konfigurieren. Für jedes Schicht-3-Protokoll existiert ein eigenes NCP. Die Liste der verschiedenen NCPs kann man in der Spezifikation RFC-1340 nachschlagen. Der Aufbau eines NCP-Pakets entspricht grundsätzlich dem des LCP. Für NCP werden allerdings nur die ersten sieben Nachrichtentypen der Tab. 4.3 verwendet. Diese Nachrichten beziehen sich beim NCP auf die Netzwerkschicht. Für jedes NCP werden eigene, spezifische Erweiterungen vorgenommen. An dieser Stelle ist das Internet Protocol Control Protocol (IPCP) zu nennen, welches für das Netzwerkprotokoll IP verwendet wird. Da die NCP-Pakete auch über die PPP-Einkapselung transportiert werden, müssen entsprechende Werte für das Protokollfeld des PPP-Rahmens gesetzt werden. Für IPCP beträgt der Wert 8021. Auch beim IPCP besteht die Möglichkeit, über verschiedene Konfigurationsoptionen bestimmte IP-Parameter für Klasse-1-Nachrichten festzulegen, was IPCP spezifiziert. Aufgrund der unterschiedlichen Aufgabenbereiche stehen natürlich unterschiedliche Konfigurationsoptionen zur Verfügung. Ein aktueller Überblick über mögliche Konfigurationsoptionen im IPCP kann in der Spezifikation RFC-1340 nachgeschlagen werden.

Die Hauptkonfigurationsoptionen sind dabei das IP Compression Protocol und IP Address. Die Konfigurationsoption bietet die Möglichkeit, das Verwenden eines speziellen Komprimierungsprotokolls auszuhandeln. Standardmäßig ist allerdings keine Komprimierung vorgesehen. Das entsprechende Feld von zwei Bytes gibt das gewünschte Komprimierungsprotokoll an und enthält eine Liste der definierten Protokolle. Es ist dabei zu beachten, dass im Fall einer Komprimierung im Protokollfeld des PPP-Rahmens nicht mehr der Wert 0021 für IP, sondern der Wert für das Komprimierungsprotokoll gesetzt werden

muss! Die Konfigurationsoption IP Address wird verwendet, um die Zuweisung einer IP-Adresse auszuhandeln. Standardmäßig wird eine Adresse nicht zugewiesen. Das entsprechende Feld enthält vier Byte. Der Sender kann über die Nachricht Configure_Request seine gewünschte IP-Adresse in diesem Feld übermitteln. Wird vom Sender das Feld auf null gesetzt, so fordert er die Zuweisung seiner Adresse vom Empfänger an. Die Adresse erhält der Sender, indem der Empfänger mit einer Configure_Not-Acknowledge antwortet und eine gültige IP-Adresse überträgt. [MCGR92]

Kommunikationsablauf

Der Kommunikationsablauf nach RFC-1661 wird anhand einer zeitlichen Abfolge durch einen Zustandsgraphen beschrieben. Die Verbindung beginnt und endet mit der Link-Dead-Phase. Durch ein externes Signal, wie Carrier Detection, geht das PPP in die Link-Establishment-Phase über. In dieser Phase wird das LCP verwendet, um die Verbindung über den Austausch von Konfigurationspaketen herzustellen. Werden in dieser Phase Pakete empfangen, die keine LCP-Nachricht beinhalten, so müssen diese verworfen werden. Werden Konfigurationspakete in anderen Phasen der Kommunikation empfangen, so wechselt das PPP direkt in die Link-Establishment-Phase zurück. Der Link-Establishment-Phase kann sich bei Bedarf eine Authentication-Phase anschließen. Diese ist standardmäßig nicht vorgesehen. In dieser Phase sind ausschließlich die Pakete LCP, Authentication Protocol und Link Quality Monitoring zugelassen. Andere Pakete müssen verworfen werden. Sind diese Phasen abgeschlossen, muss jedes Netzwerkprotokoll, das anschließend für die Übertragung verwendet werden soll, in der Network-Layer-Protocol-Phase über entsprechende NCP-Nachrichten konfiguriert werden. In dieser Phase besteht der Datenverkehr über den Link aus LCP- und NCP-Paketen. Darüber hinaus werden auch Pakete der bereits konfigurierten Netzwerkprotokolle übertragen. Sobald alle gewünschten Daten übertragen sind, kann die Verbindung getrennt werden. Hierfür werden bestimmte LCP- oder NCP-Pakete gesendet, woraufhin das PPP in die Link-Termination-Phase wechselt. Diese Phase kann auch ungewollt erreicht werden, indem beispielsweise die Verbindung zu Schicht 1 verloren geht (Loss-of-Carrier). Anschließend geht PPP wieder in den Ausgangszustand, die Link-Dead-Phase, über. Das Ablaufdiagramm verdeutlicht den Kommunikationsverlauf. [FROMM01]

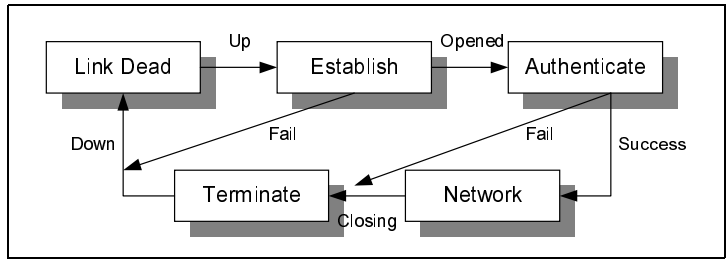


Abb. 4.14 PPP-Kommunikationsablaufdiagramm

4.2.2 SDH/SONET

Um das Protokoll IP über das WAN[41] übertragen zu können, wird auf der einen Seite das Protokoll PPP benötigt. Hinzu kommt die Layer-2-Schicht, die das PPP-Protokoll in einem Übertragungsrahmen einpackt und weitervermittelt. Bei der PoS-Technologie wird das PPP-Protokoll direkt auf SDH/SONET, einer byteorientierten, synchronen Verbindung, aufgesetzt. Die Spezifikation RFC-2615 von PoS befasst sich mit dieser Anpassung auf die Schicht 2.

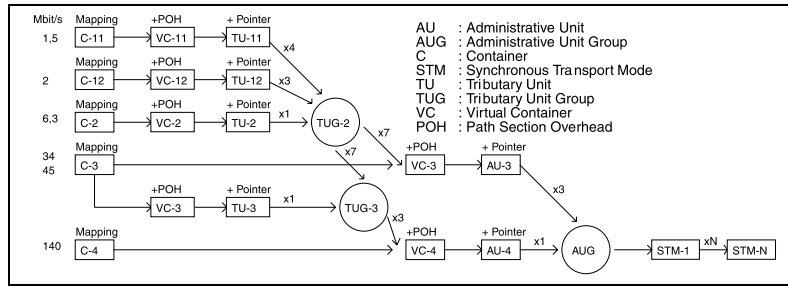
Als die digitale Sprachübertragung unter Verwendung der Pulse Code Modulation (PCM) eingeführt wurde, definierte die ITU-T als Übertragungstechnik eine Multiplex-Hierarchie, die als Basis den PCM-64Kbit/s-Sprachkanal verwendete. Diese Hierarchie wird als Plesiochrone Digitale Hierarchie (PDH) bezeichnet. Allerdings stellte die fehlende Synchronität der einzelnen Netzelemente untereinander und die Tatsache, dass ein direkter Zugriff auf einen bestimmten Kanal nur durch komplettes Demultiplexen der höherwertigen Übertragungsrahmen erreicht werden kann, einen wesentlichen Nachteil der PDH-Technologie dar. Dieser Nachteil führte zur Einführung der Synchronous Digital Hierarchy (SDH). 1985 definierte das American National Standardization Institute (ANSI) das Synchronous Optical NETwork (SONET). Um die Probleme des PDH zu lösen, enthält der Standard SONET ein Konzept zur Verwendung eines zentralen Referenztaktes. Auch ermöglicht SONET den direkten Zugriff auf Signale innerhalb eines höherwertigen Übertragungsrahmens. Kernpunkt der Transporttechnik stellt dabei die Nutzung der Glasfaser für hohe Übertragungsraten dar. Die ITU-T übernahm 1988 den SONET-Standard unter der Bezeichnung SDH. Allerdings wurde bei SDH als Basis der 155,52-Mbit/s-Übertragungsrahmen gewählt. SONET definiert als Basis einen 51,84-Mbit/s-Übertragungsrahmen. Dieser Unterschied von Faktor drei zieht sich durch sämtliche Hierarchiestufen, bleibt aber für PoS, ebenso wie weitere Unterschiede, irrelevant. PoS kann beide Übertragungstechnologien verwenden. Weitere Betrachtungen beschränken sich auf SDH. [WILD99]

SDH ist somit ein europäisches Transportnetz mit weltweit einheitlicher Normung, welches eine transparente Übertragung von jeglichen Informationen, wie IP-Daten, ermöglicht. Hierzu werden zwischen zwei Netzelementen konstant Übertragungsrahmen gesendet, welche die Informationen aufnehmen. Das SDH realisiert Punkt-zu-Punkt-Verbindungen. Eine Vermittlungsfunktion ist im SDH nicht implementiert und liegt im Aufgabenbereich der jeweils aufsetzenden Protokolle. Der Basisrahmen bei SDH wird Synchronous Transfer Module-1 (STM-1) genannt und besteht aus verschiedenen Feldern. Die kleinste Einheit eines STM-1 bildet der Container (C). Der kleinste verfügbare Container im SDH ist ein 2,048-Mbit/s-System (C12). Jeder *C* trägt einen

41 Wide Area Network

Path Overhead (POH). Der POH enthält Angaben zur Struktur des *C* und Steuerinformationen. *C* und POH bilden zusammen einen virtuellen Container (VC). Diese VCs besitzen Substrukturen, die von der zu transportierenden Nutzlast abhängig sind. Besteht die Payload wiederum aus einzelnen VCs, so werden sie in Tributary Unit Group (TUG) unterteilt, welche ihrerseits wieder in Tributary Units (TU) unterschieden werden. Jeder TU stellt die Übertragungskapazität für einen VC-Kanal dar, während die TUG die Lage der TU im VC höherer Ordnung bestimmt. Die TU, die sich unmittelbar unter der Hierarchie des STM-1-Rahmens befindet, heißt Administrative Unit (AU). Sie werden zur Administrative Group (AUG) zusammengefasst. Der AU-Pointer verweist auf den Beginn eines VC. Abb. 4.15 zeigt die unterschiedlichen Multiplexmöglichkeiten auf, die ein STM-1-Rahmen bei SDH besitzt.

Abb. 4.15 SDH-Multiplexschema



Dabei muss eine Phasenkompensation der einzelnen gemultiplexten Signale im SDH-Netzwerk durchgeführt werden, da die Container durch Laufzeitunterschiede und Taktverteilungsstörungen Phasenverschiebungen erfahren. Die notwendige Kompensation findet durch Pointer statt. Die Beschreibung der Containerlage wird durch einen Pointer vorgenommen, der sich in der nächsthöheren Hierarchie befindet. Durch Modifikationen der Pointer können so die Phasenverschiebungen ausgeglichen werden. [DETK98a]

4.2.3 IP über PoS

Die notwendigen Schritte, damit PoS eine Übertragung des IP-Protokolls durchführen kann, werden ebenfalls in der Spezifikation RFC-2615 beschrieben. Dabei werden beide Übertragungssysteme SONET und SDH berücksichtigt. Abb. 4.16 zeigt den Ablauf der IP-Einkapselung über PoS. Auf der Empfangsseite sind die einzelnen Schritte in umgekehrter Reihenfolge zu durchlaufen.

Das Verfahren, bei dem IP-Pakete zunächst in PPP- und anschließend in HDLC-Rahmen abgebildet werden, wurde bereits beschrieben. Es ist für PoS ebenso gültig, wie das Verwenden der bereits beschriebenen Steuerprotokolle LCP[42] und NCP[43] bzw. IPCP[44]. Abweichungen bei PoS bestehen nur in der verwendeten Frame Check Sequence (FCS) im HDLC-Rahmen und richten sich

nach dem für den Transport verwendeten Container. Nach der Spezifikation RFC-2615 werden bei PoS nur die VC-4, VC-4-4c/16c/64c-SDH-Container unterstützt. Allen anderen Containern, mit Ausnahme des VC-4, sind im Gegensatz zu PPP nicht die standardmäßige 16-Bit-FCS, sondern die 32-Bit-FCS vorgeschrieben. Für den VC-4-Container ist auch die 16-Bit-FCS möglich, aber wird nicht explizit empfohlen.

Es schließt sich das Byte-Stuffing an. Hierbei werden Stopfinformationen in den Container eingefügt, um Taktschwankungen auszugleichen. Während die Daten dann byteweise in den Container eingeschrieben werden, erfolgt aus Sicherheitsgründen eine Verwürfelung (Scrambling). Hierfür wird ein 43-Bit-Schieberegister verwendet. Es ist ebenfalls in RFC-2615 definiert und beschrieben worden. Weitere Informationen zum Scrambling können dort nachgelesen werden. Im Container muss nun abschließend im POH das Path Signal Label (C2), welches den Inhalt des Containers definiert, auf den Wert 0x16 bei Einsatz des Scramblers und auf 0xCF ohne Verwendung des Scramblers gesetzt werden. Der Multiframe Indicator (H4) bleibt unbenutzt und muss auf null gesetzt werden. [MASI99]

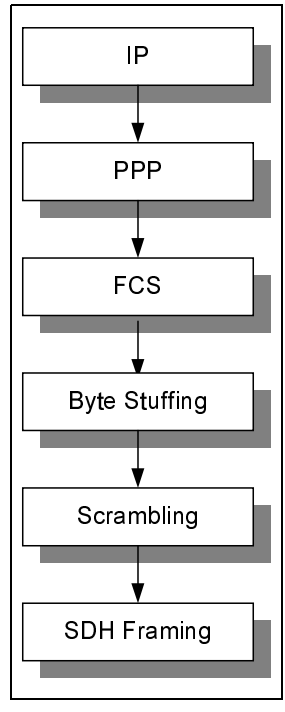


Abb. 4.16
IP-over-PoS

42 Link Control Protocol
43 Network Control Protocol
44 Internet Protocol Control Protocol

4.3 Label Switching

Das Multi-Protocol Label Switching (MPLS) besitzt als Grundlage den Mechanismus des Label Switching (LS). Kern des LS stellt dabei die Trennung in Forwarding- und Control-Komponente dar. Für die Entwicklung von LS-Technologien lassen sich verschiedene Gründe anführen: [DARE00]

1. **Layer-3-Switching:** Aufgrund steigender Bandbreite-Anforderungen, ausgelöst durch neue Anwendungen und höhere Teilnehmerzahlen, bildet die Steigerung der Performance einen wichtigen Faktor, der zur Entwicklung des LS führte. Das Ziel war es, den in Bezug auf Performance effektiveren Switches einige Routing-Funktionen zu übertragen und so auf konventionelle Router zu verzichten. Diese in der Funktion erweiterten Switches werden Label Switching Router (LSR) bei LS genannt. Darüber hinaus sind Switches im Vergleich zu Routern auch günstiger und die entstehende Kosteneinsparung sprach zusätzlich für die LS-Entwicklung.
2. **Interworking:** Durch die Koexistenz von IP und ATM im Internet ist Interworking notwendig. Die komplexen Mechanismen, die entwickelt wurden, um IP und ATM anzupassen bzw. zu integrieren (z.B. CLIP), weisen alle Schwachpunkte auf. Aus diesem Grund wurde ein Verfahren gesucht, das diese Integration effektiv und kostengünstig vereinfacht.
3. **Skalierbarkeit:** Ein weiterer wichtiger Grund für die Entwicklung des LS stellt die Skalierbarkeit dar. Die Skalierbarkeit beschreibt das Aufbrauchen von Ressourcen, wenn ein Netzwerk wächst. Da durch steigende Teilnehmerzahlen die Netze immer größer werden, ist es daher unbedingt erforderlich, die Skalierbarkeit zu optimieren.
4. **Traffic Engineering:** In konventionellen Routern wird ausschließlich Destination-based Routing betrieben. Um zusätzliche Routing-Funktionen, wie das Explicit Routing mit dem Ziel des Traffic Engineering einzuführen, wurde ein neues Verfahren gesucht und mit der Entwicklung des LS gefunden.

4.3.1 LS-Topologie

Label Switching kombiniert wie auch MPLS die Performance des Schicht-2-Switchings mit der Intelligenz des Schicht-3-Routings. Aus diesem Grund lässt sich das LS im OSI-Referenzmodell zwischen der Sicherungsschicht und der Vermittlungsschicht einordnen.

Eine LS-Domäne besteht aus Label Edge-Routern (LER), welche die Schnittstelle zu angeschlossenen IP-Netzen bilden. Diese LERs befinden sich am Rand der LS-Domäne und müssen sowohl IP- als auch LS-fähig sein. Im Kern der LS-Domän werden die Label Switching Router (LSR) eingesetzt. Die LSRs beschränken sich auf eine Paketvermittlung über das LS. Eine Vermittlung anhand des IP-Headers ist nicht notwendig.

Abb. 4.17 zeigt auch den prinzipiellen Vermittlungsvorgang im LS, wodurch die notwendigen Funktionen der einzelnen LS-Netzelemente noch einmal verdeutlicht werden. Der Vermittlungsvorgang beginnt mit dem Empfangen eines IP-Pakets aus einem angeschlossenen Netz in einem LER. Der LER wertet den IP-Header aus und fügt dem Paket ein bestimmtes Label an. Anhand des Labels wird das MPLS-Paket (IP-Paket mit angefügtem Label) in den LSRs weitervermittelt. In einem zweiten LER wird das Label entfernt und das IP-Paket konventionell auf Schicht-3 in ein angeschlossenes Netz geroutet. Die Vermittlung im LS erfolgt hierbei immer verbindungsorientiert. Um IP-Pakete über die LS-Domän zu transportieren, muss zwischen den LERs eine Verbindung, die im LS Label Switched Path (LSP) genannt wird, aufgebaut werden. Ein LSP ist eine unidirektionale Verbindung, beginnt immer in einem so genannten Ingress und endet immer in einem so genannten Egress. Ein Ingress bzw. Egress muss ein LER sein.

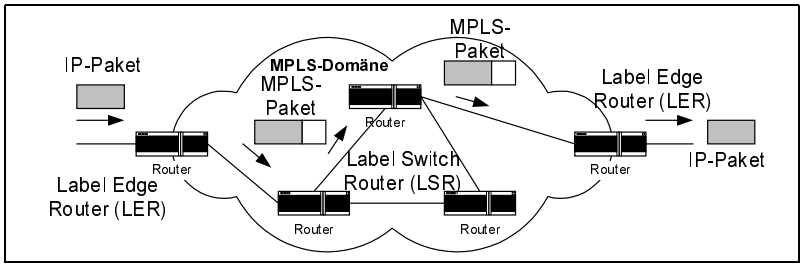


Abb. 4.17 LS-Domäne

4.3.2 Forwarding-Komponente

Die Aufgabe der Forwarding-Komponente besteht im Weiterleiten der Pakete vom Eingangsport zum Ausgangsport des LSRs. Dies beinhaltet eine Reihe von Algorithmen, welche den Vermittlungsvorgang auf Basis des Label Swapping ermöglichen. Zur Vermittlungsentscheidung werden das Label und die LS-Forwarding-Table als Informationsquellen verwendet. Folgende Komponenten können aufgelistet werden:

- Label
- LS-Forwarding Tabelle
- LS-Forwarding Algorithmus

Das Label ist Teil des MPLS-Pakets und befindet sich im Header. Es handelt sich um eine kurze Bitsequenz fester Länge, welche keine Daten kodiert, sondern aus einem frei gewählten Wert besteht. Das Label hat ausschließlich lokale Bedeutung. Um das Label mit dem Datenpaket zu transportieren, lassen sich abhängig von der zugrunde liegenden Schicht-2-Technologie zwei Fälle unterscheiden:

- **Zufügen neuer Felder an das Datenpaket:** Wird für Schicht-2-Technologien wie PPP oder Ethernet verwendet. Die Rahmen dieser Protokolle bieten keine Möglichkeit, das Label zu transportieren. Aus diesem Grund werden die Header der einzelnen Protokolle um einen so genannten Shim-Header zwischen Link und Network Layer erweitert, welcher das Label beinhaltet.
- **Verwendung bestehender Felder im Header:** Wird von Schicht-2-Technologien wie ATM oder Frame Relay (FR) verwendet. Diese Dienste können das Label in ihrem Header transportieren, indem sie bestehende Felder des Headers umdefinieren. Die Bits dieser Felder stehen vollständig für das Label zur Verfügung. Abhängig von der Schicht-2-Technologie und der damit verbundenen Größe der verwendeten Felder, ergibt sich die Limitierung für die Anzahl der verfügbaren Labels.

Das Forwarding in konventionellen ATM-Switches basiert auf einem ähnlichen Prinzip wie das LS und verwendet das VCI- und VPI-Feld. Aus diesem Grund bietet sich der Transport des Labels in diesen Feldern an, da nur geringfügige Modifikationen in Form eines Software-Updates an den konventionellen ATM-Switches notwendig sind, um sie im MPLS weiterzuverwenden. Aus dem gleichen Grund ist für FR der Labeltransport im DLCI[45]-Feldes FR-Headers vorgesehen. Durch die flexiblen Festlegungen des Labeltransports kann die LS-Technologie auf einer Vielzahl bestehender Schicht-2-Protokolle aufsetzen.

Die LS-Forwarding-Tabelle ist in jedem LSR enthalten. Für jedes Label existiert ein eigener Eintrag, bestehend aus einem oder mehreren Subeinträgen. Ein Subeintrag beinhaltet die Informationen, die benötigt werden, um ein ankommendes Paket nach dem unten beschriebenen LS-Forwarding-Algorithmus weiterzuleiten. Hierzu gehören das Outgoing Label, das Outgoing Interface und die Adresse des Next-Hops. Ein LSR kann dabei entweder nur eine einzige LS-Forwarding Table oder eine für jede seiner Schnittstellen unterhalten. Im zweiten Fall wird die Vermittlungsentscheidung dann nicht nur anhand des Labels getroffen, sondern auch anhand des Eingangsports, an dem das Paket empfangen wurde.

Um die begrenzte Anzahl der Labels optimal zu nutzen und auf diese Weise die Skalierbarkeit zu erhöhen, wurden so genannte Forwarding Equivalence Classes (FEC) eingeführt. Eine FEC wird durch bestimmte Metriken festgelegt. Üblicherweise wird der Adresspräfix der Zieladresse verwendet. Alle Pakete, deren Metriken innerhalb einer FEC liegen, werden vom LSR gleich behandelt und beispielsweise zum gleichen Next-hop weitergeleitet. Auf diese Weise werden alle möglichen Pakete, die der LSR zu verarbeiten hat, anhand ausgewählter Metriken in eine endliche Anzahl von FECs abgebildet. Abb. 4.18 verdeutlicht den Zusammenhang von Paket, Metrik und FEC.

45 Data Link Connection Identifier

Ein Label wird einer FEC zugeordnet. Alle Pakete, deren Metriken innerhalb einer FEC liegen, verwenden aus diesem Grund das gleiche Label. Die Zuordnung von FEC und Label stellt einen wesentlichen Teil des LS dar. Eine FEC zeichnet sich durch die Granularität aus. Die Granularität bestimmt, wie viele Flows anhand ihrer Metriken zu einer FEC zusammengefasst werden. Eine relativ grobe Granularität wird beispielsweise erreicht, indem nur der Adresspräfix der Zieladresse für die Abbildung der Pakete in eine FEC verwendet wird. Werden stattdessen die gesamte IP-Adresse des Ziels und zusätzlich noch die Ursprungsadresse und die Portadressen für die FEC verwendet, so ist die Granularität vergleichsweise fein. Eine grobe Granularität bietet den Vorteil der Ressourceneinsparung, da weniger Labels benötigt werden. Eine feine Granularität hingegen weist eine höhere Flexibilität auf, welche dadurch begründet ist, dass die Festlegung eines QoS nur pro FEC zu realisieren ist. Je feiner die Aufteilung der Flows über deren Metriken in FECs vorgenommen wird, desto flexibler und effektiver kann eine Ressourcenzuweisung erfolgen.

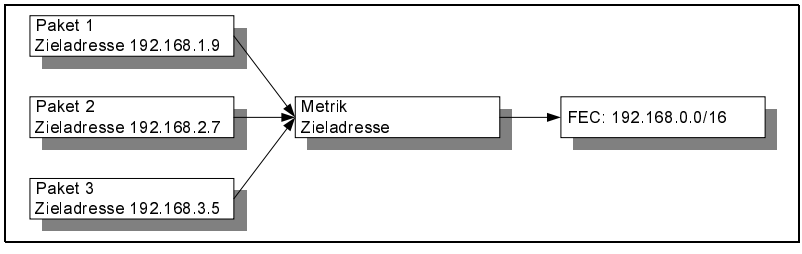


Abb. 4.18
FEC-Funktionalität

Unterhält der LSR mehrere LS-Forwarding Tables, so besteht der erste Schritt des LS-Forwarding-Algorithmus darin, anhand des Eingangsports die entsprechende LS-Forwarding Table zu bestimmen. Anschließend bzw. im ersten Schritt, wenn der LSR nur eine Tabelle unterhält, bestimmt der Algorithmus das Label des empfangenen Pakets (Incoming Label). Dieses Label wird als Index verwendet, um den entsprechenden Eintrag in der LS-Forwarding Table auszulesen. Der Algorithmus ersetzt das Label, mit dem das Paket empfangen wurde, durch das Outgoing Label des ersten Subeintrags in der Tabelle. Dies ist erforderlich, da die LS-Forwarding Tables lokal aufgebaut werden.

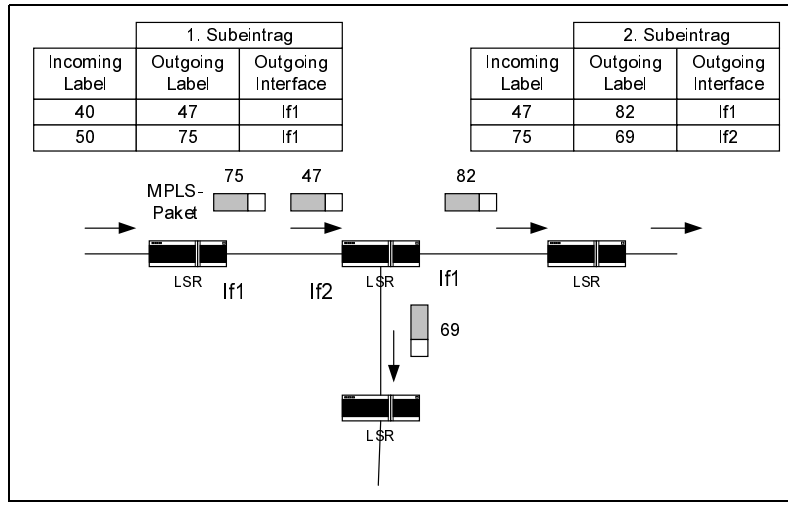
Aus diesem Grund hat der folgende LSR den FECs andere Labels zugeordnet, was durch das Label Swapping berücksichtigt wird. Anschließend bestimmt der LS-Forwarding-Algorithmus anhand der weiteren Informationen des Subeintrags wie Next-hop Address oder Outgoing Interface den Ausgangsport, über den der Next-hop der FEC erreicht werden kann, und vermittelt das Paket entsprechend weiter. Sind für das Incoming Label mehrere Subeinträge vorhanden, so wird die Prozedur des Label Swappings für jeden Subeintrag ein-

zeln durchgeführt. Auf diese Weise ist das LS Multicast-fähig, da das empfangene Paket über mehrere Ausgangsports[46] weitergeleitet wird.

Der LS-Forwarding-Algorithmus im LS ist für alle Forwarding-Funktionen wie Unicast oder Multicast identisch und basiert auf einfachem Auslesen. Konventionelles Routing hingegen verwendet je nach Anwendungsfall unterschiedliche Algorithmen und muss einen bestimmten Eintrag in der Routing-Tabelle über eine Suche bestimmen:

- Unicast-Routing verwendet den Longest-Match-Algorithmus für die Zieladresse.
- Unicast-Routing mit TOS-Feld verwendet den Longest-Match-Algorithmus für die Zieladresse und der Exact-Match-Algorithmus für das TOS-Feld.
- Multicast-Routing verwendet eine ganze Reihe von Algorithmen, welche die Ursprungs- und Zieladressen auswerten.

Abb. 4.19
LS-Forwarding



Aus diesem Grund bietet der Forwarding-Algorithmus des LS im Vergleich eine geringere Komplexität. Zusätzlich ermöglicht die Verwendung des Forwarding-Algorithmus im LS erweiterte Forwarding-Funktionen wie beispielsweise das Explicit Routing.

Der LS-Forwarding-Algorithmus wird von allen LSRs innerhalb einer LS-Domäne verwendet. Die Randgeräte (LER) sind aufgrund eines erweiterten Aufgabengebiets auf zusätzliche Algorithmen angewiesen. Die LERs stellen die Schnittstelle zwischen LS- und IP-Netzen dar. Aus diesem Grund müssen sie in

46 Ein Ausgangsport pro Subeintrag

der Lage sein, Datenpakete der an die LS-Domain angeschlossenen IP-Netze mit einem Label zu versehen, falls diese in das LS-Netz vermittelt werden sollen, worauf später noch eingegangen wird.

4.3.3 Control-Komponente

Die Aufgabe der Control-Komponente ist der Aufbau und der Unterhalt der LS-Forwarding-Tabelle. Hierzu muss die Control-Komponente Routing-Informationen zwischen den Geräten LER/LSR austauschen und Algorithmen bereitstellen, um anhand dieser Informationen die LS-Forwarding Table zu erstellen. Für das Austauschen der Routing-Informationen verwendet die Control-Komponente konventionelle IP-Routing-Protokolle wie Open Shortest Path First (OSPF), Border Gateway Protocol (BGP) oder Protocol Independent Multicast (PIM). Auf diese Weise ist es in einem LER/LSR möglich, einer FEC einen Next-Hop zuzuordnen. Das bedeutet, dass der Pfad, über den ein LSP für eine bestimmte FEC aufgebaut wird, durch konventionelle Routing-Protokolle bestimmt wird. Die Funktionen, welche diese Routing-Protokolle bieten, reichen aber für das LS nicht aus. Es müssen zusätzliche Mechanismen bereit gestellt werden, um FEC und Labels einander zuzuordnen. Die Kombination beider Zuordnungen[47] bietet alle Informationen, die für das Erstellen der LS-Forwarding-Tabelle gebraucht werden. Darüber hinaus müssen die Label-FEC-Zuordnungen in irgendeiner Form dem benachbarten LSR mitgeteilt werden. Die Zuordnungen werden auch als Label Binding bezeichnet und beinhalten unterschiedliche Optionen:

- **Local und Remote Binding:** Bestimmt ein LSR selbst, welches Label aus seinem Pool freier Labels einer FEC zugeordnet wird, so handelt es sich um Local Binding. Remote Binding ist gegeben, wenn die Zuordnungen in einem anderen LSR festgelegt werden. Somit wählt ein anderer LSR das zu verwendende Label aus und ordnet es einem FEC zu.
- **Upstream und Downstream Binding:** Jeder LSR verwendet Local und Remote Binding gleichzeitig: ein Verfahren für das Binding seines Incoming Label und das andere für das Binding seines Outgoing Label. Wird für die Zuordnung des Incoming Label die Methode des Local Binding und dementsprechend für das Outgoing Label das Remote Binding verwendet, so handelt es sich um Downstream Binding. Upstream Binding ist für den umgekehrten Fall[48] gegeben. Das Binding ist relevant für die Kommunikation zwischen zwei LSRs. Durch die ausschließlich lokale Bedeutung des Binding wird die Zuordnung von einem der beiden beteiligten LSRs festgelegt. Abhängig davon, ob der bestimmende LSR Downstream- oder Upstream-orientiert[49] ist, erhält das Binding-Verfahren seinen Namen.

47 FEC à Next-hop und FEC à Label

48 Incoming Label à Remote Binding und Outgoing Label à Local Binding

49 In Bezug auf den Flow der Datenpakete

- **Data-Driven Binding**: Damit die Label Bindings von den LSRs generiert oder aufgehoben werden, muss man unterscheiden zwischen Data-Driven und Control-Driven. Beim Data-Driven Binding ist das Empfangen von Datenpaketen das auslösende Ereignis, das die LSR zum Generieren von Label Bindings veranlasst. Das bedeutet, dass bei diesem Verfahren die Bindings nur bei Bedarf durchgeführt werden. Die ideale Umgebung, die für dieses Verfahren spricht, zeichnet sich durch unendlich lange Flows aus. Somit ist der Aufwand des Label Binding[50] im Vergleich zu der Gesamtmenge der zu transportierenden Daten zu vernachlässigen. In realen Umgebungen werden sowohl kürzere als auch längere Flows auftreten. Da kürzere Flows den Aufwand des Label Binding nicht rechtfertigen, besteht die Möglichkeit, erst nach einer gewissen Anzahl von Paketen ein Binding zu generieren. Flows, die kürzer als die festgelegte Anzahl sind, müssten komplett konventionell geroutet werden. Je mehr Pakete abgewartet werden, bis ein Label Binding generiert wird, desto weniger Flows werden anhand eines Label geswitcht und desto weniger Labels werden verbraucht. Auf der anderen Seite müssten mehr Pakete konventionell geroutet werden, was mit hohen Leistungseinbußen verbunden ist. Ein optimaler Wert muss dem Netz und seinen durchschnittlichen Datenflows angepasst werden.
- **Control-Driven Binding:** Eine zweite Möglichkeit um das Label Binding auszulösen, besteht im Control-Driven-Verfahren. Die Label-Zuweisung basiert hierbei nicht auf dem Empfang von Datenpaketen sondern auf Kontrollinformationen. So können beispielsweise auf Grund von Routing-Informationen Label Bindings generiert werden, unabhängig davon, ob gerade ein Datenstrom dieses Label benötigt oder nicht. Die ideale Umgebung für dieses Verfahren besteht in einem stabilen Netz. Auf diese Weise bleiben die Routing-Informationen unverändert, wodurch auch keine neuen Bindings generiert werden müssen. Da die LS-Forwarding-Tabelle mit ihren Bindings bereits vor dem Empfangen von Datenpaketen besteht, kann sofort das erste Paket anhand des Label geswitcht werden. Auf diese Weise verzichtet dieses Verfahren im Gegensatz zum Data-Driven Modell während des Forwarding vollständig auf die Paketvermittlung in der Vermittlungsschicht und erhöht so die Performance.

Wenn ein LSR Label Bindings vornimmt oder auflöst, müssen die anderen LSRs darüber informiert werden. Hierzu besteht die Möglichkeit, die Label-Binding-Informationen über die von der Kontrollkomponente verwendeten Routing-Protokolle zu versenden. Auf diese Weise lässt sich die Label Distribution beispielsweise über BGP[51] oder PIM[52] realisieren. Hierdurch kann auf die Einfüh-

50 Binding inklusive Distribution
51 Border Gateway Protocol
52 Protocol Independent Multicast

rung eines zusätzlichen Protokolls verzichtet werden. Dennoch sind weitere Punkte zu beachten, wie die Formatänderung der unterschiedlichen Routing-Pakete oder die Reaktion eines konventionellen Routers auf ein modifiziertes Routing-Paket. Eine zweite Möglichkeit andere LSRs über die Bindings zu informieren, besteht mit der Einführung eines weiteren und speziell für diese Anwendung entwickelten Protokolls: dem Label Distribution Protocol (LDP). Auf diese Weise ist das System nicht von den Fähigkeiten des Routing-Protokolls abhängig, um die Label-Binding-Informationen zu transportieren. Der Nachteil dieser Lösung besteht in einer weiteren Erhöhung der Komplexität des gesamten Systems, da ein zusätzliches Protokoll eingeführt und verarbeitet werden muss. Abb. 4.20 zeigt einen Label Switch Router (LSR), bestehend aus den jeweiligen Komponenten Forwarding und Control. Sie verdeutlicht auch das Zusammenspiel der einzelnen Elemente im LSR. [DARE00]

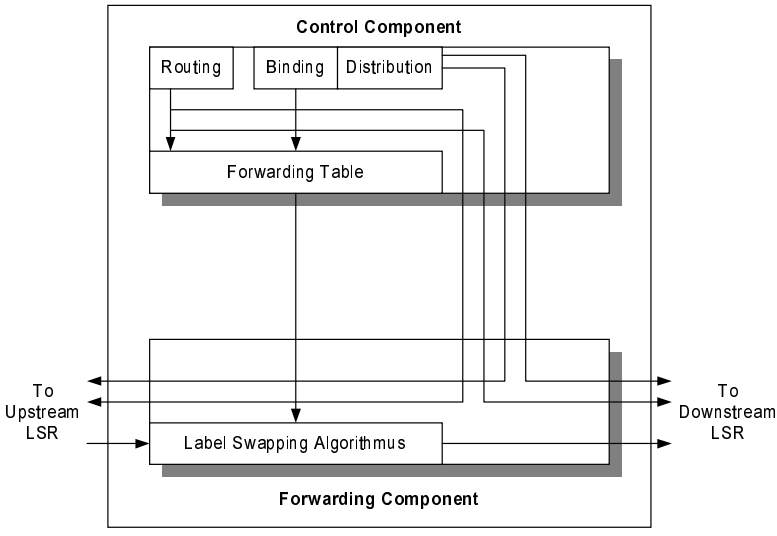


Abb. 4.20
LSR-Architektur

4.3.4 Implementierungen

Vor einigen Jahren entwickelten einige Hersteller eigene Label-Switch-Technologien, die alle auf dem hier vorgestellten LS-Prinzip basieren. Als Vorreiter der LS-Technologie galt das IP-Switching von Ipsilon, welches das LS-Verfahren auf dem Markt einführte und bekannt machte. Einige Monate später präsentierte Cisco Systems das Tag Switching. Dieses LS-Verfahren hat große Bedeutung für die Entwicklung des MPLS, da viele Elemente übernommen wurden. Ebenfalls haben Toshibas CSR-Verfahren und IBMs ARIS-Verfahren Einfluss auf die Entwicklung von MPLS gehabt. Wegen der entwicklungsgeschichtlichen Bedeutung für MPLS werden diese LS-Implementierungen hier kurz vorgestellt.

Ipsilon IP-Switching Ipsilon IP-Switching nach RFC-1954 ist von der Firma Ipsilon Networks, die inzwischen von Nokia aufgekauft wurde, entwickelt worden, um eine Kombination von IP und ATM herzustellen. Dabei wird ein hybrider IP-Switch angeboten, der einen ATM-Switch mit einem Gigabit-Router kombiniert. Die Datenpakete werden von einer ATM-Switching-Fabric weitergeleitet, während der Switch-Controller sie mit Hilfe traditioneller Router-Software zu ihrem Bestimmungsort führt. Daraus lassen sich folgende Vorteile ableiten:

- ATM-Switching-Technologie kann preiswert und effektiv eingesetzt werden.
- Signalisierungssoftware (UNI, P-NNI, MPOA) für IP-over-ATM entfällt.
- Es wird nur IP-Software (IFMP, GSMP) eingesetzt, die einfacher in der Handhabung ist.

Bei der Kombination von ATM-Hardware und IP-Software (Kontrollkomponente) handelt es sich im Prinzip um die Realisierung eines LSR, der in der Implementierung von Ipsilon IP-Switch genannt wird. Durch die Flussklassifikation kann die Belastung des IP-Switch-Controller begrenzt werden. Die Funktionsweise ist dabei die folgende: Vor einer Übertragung aktiviert ein IP-Knoten jede physikalische ATM-Verbindung als einen Übertragungskanal. Werden auf dem Eingangsport Daten von einem anderen Upstream-Gerät wahrgenommen, wird das Datenpaket zu der intelligenten Routing-Software des IP-Switch-Controllers geschickt. Wenn das der Fall ist, wird der ATM-Switch lediglich als I/O[53]-Erweiterung der Routing-Software verwendet. Anschließend schickt der IP-Switch das Datenpaket über den Standardübertragungskanal weiter. Ein Datenstrom ist eine Sequenz von Paketen zwischen einem Sender und Empfänger, die durch Parameter wie Quell- und Zieladresse sowie Portnummern charakterisiert werden können. Beim Ipsilon IP-Switching wird zwischen den Datenströmen Host Pair Flow auf der Ebene der IP-Adressen und Port Pair Flow auf der Ebene der IP-Ports unterschieden. [NEHH+96a]

Ein wichtiger Bestandteil des IP-Switch-Controllers stellt die Komponente Flow Classification and Control dar. Die Aufgabe dieses Moduls besteht darin, den ankommenden IP-Verkehr zu untersuchen und die Flows zu bestimmen, für die sich die Vermittlung anhand von LS als effektiv erweist (Data-Driven). Hierzu können beispielsweise die Portnummern des Transmission Control Protocol (TCP) oder des User Datagram Protocol (UDP) herangezogen werden. FTP-Sessions, die in der Regel lange Flows produzieren und sich aus diesem Grund für eine Label-basierte Vermittlung eignen, können so erkannt werden. Im Flow Classifier liegt die Intelligenz des Systems.

Eine weitere Funktion des IP-Switch-Controllers besteht in konventionellem Routing. Dies ist sowohl für LS-Flows erforderlich als auch für die Flows,

53 Input/Output

die nach Ermessen der Flow Classification konventionell auf Schicht 3 geroutet werden sollen. Für das Austauschen sowohl der Routing-Informationen als auch der Schicht-3-Pakete werden Default VCs verwendet. Diese VCs verbinden benachbarte ATM-Switches, benötigen aber keine Signalisierung. Sollen bestimmte Flows nach der Flow Classification anhand eines Labels geswitcht werden, so müssen die anderen IP-Switche über das Binding informiert werden. Das Ipsilon IP-Switching verwendet zur Mitteilung von Labelinformationen ein eigenes LDP, welches in RFC-1953 definiert wurde. Hierbei handelt es sich um das Ipsilon Flow Management Protocol (IFMP). Nach den Festlegungen ist das IFMP ein Soft-State-Protokoll, welches das Downstream-Binding unterstützt. Zur Übertragung der IFMP-Nachrichten werden die Default VCs verwendet. Nachdem alle Informationen ausgetauscht wurden, können dann die Flows über Data VCs zum Bestimmungs-IP-Switch vermittelt werden.

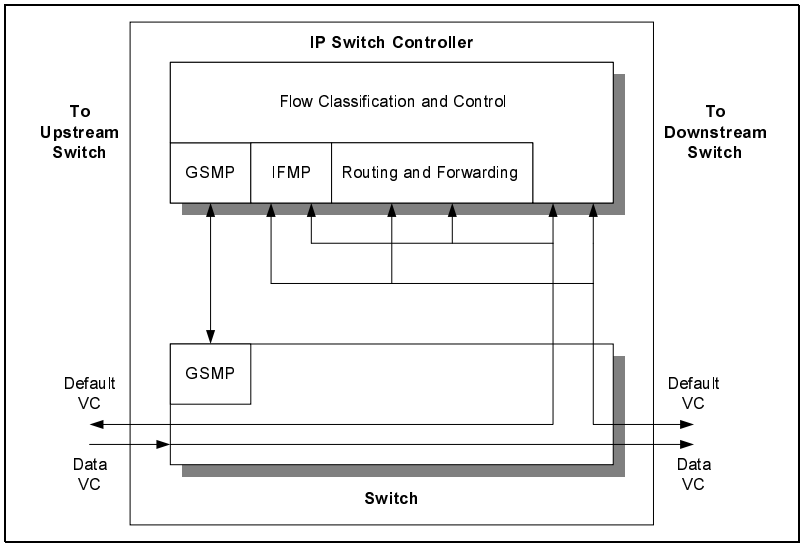
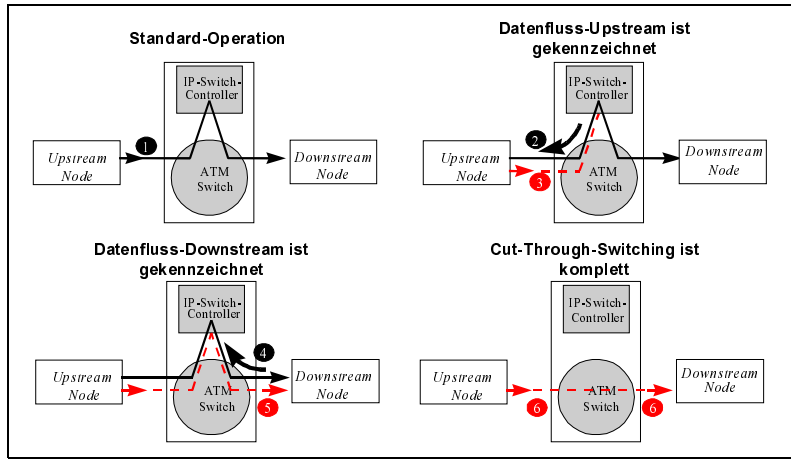


Abb. 4.21 Ipsilon IP-Switch-Architektur

Nachdem ein Datenfluss identifiziert wurde, wird über IFMP die Information im Netz verteilt. Diese enthält den Datenflusstyp und das Label. Wenn es sich um einen IP-Datenfluss handelt, schickt der Switch-Controller die Aufforderung an den Knoten, von dem das Paket kam, alle zu diesem Flow gehörenden Datenpakete zu markieren und ihn über einen neu aufgebauten virtuellen ATM-VC-Kanal zu schicken (Abb. 4.22). Hierfür wird ein freies Label (VCI = x) aus dem Label Space des eingehenden Ports sowie ein freies Label (VCI = x) auf dem Controller-Port gesucht. Diese Werte werden dem Switch über General Switch Management Protocol (GSMP) mitgeteilt und im ATM-Header wird das entsprechende Bit im VCI-Feld gesetzt, das die zu übertragenden Daten als homogenen Datenstrom kennzeichnet (siehe Abb. 4.22, Punkt 2).

Anschließend wird über das Protokoll IFMP dem vorhergehenden Switch diese Information mitgeteilt. Dieser wählt für den Eingangsport ebenfalls ein noch freies Label aus, nimmt den Eingang in der Switching-Tabelle vor und sendet eine IFMP-Nachricht (IFMP Redirection Message) an den vorhergehenden Switch. Falls der Upstream-Knoten damit übereinstimmt, wird ein neuer virtueller Kanal ausgewählt und der Datenstrom fließt über diesen VC (siehe Abb. 4.22, Punkt 3). Unabhängig davon kann der Downstream-Knoten ebenfalls den IP-Switch-Controller nach einem VC befragen und eine Markierung setzen (siehe Abb. 4.22, Punkt 4). Wenn der Datenfluss (IP-Flow) zu einem einzelnen Eingangs- und Ausgangskanal isoliert wurde, nimmt der IP-Switch-Controller die entsprechenden Anpassungen (Port Mapping, Umgehen der Routing-Software und dem damit verbundenen Overhead) vor (siehe Abb. 4.22, Punkt 5). Das heißt, ein auf dem Port ankommendes Datenpaket, geteilt in einzelne AAL-5-Pakete) wird wieder zusammengesetzt und an den Controller gesendet. Dieser trifft dann eine Forwarding-Entscheidung, welche durch einen Cache beschleunigt werden kann.

Abb. 4.22
Cut-Through-Switching



Abschließend wird beim IP-Switching eine festgeschaltete Verbindung zwischen Sender und Empfänger im ATM-Netz erzeugt (siehe Abb. 4.22, Punkt 6). Alle Datenpakete werden über diesen virtuellen Kanal transportiert. Das Verfahren wird als Cut-Through-Switching bezeichnet. Cut-Through-Switching zeigt, dass bei festgelegter Verbindung zwischen zwei Punkten ein direkter Transport über den ATM-Switch erfolgt. Die übliche und zeitaufwendigere Store-and-Forward-Methode wird nur noch bei alleinstehenden Paketen angewandt, d.h., wenn kein Flow vorliegt. [NEHH+96b]

Um den ATM-Switch zu steuern, wird vom IP-Switch-Controller ein bestimmter VC zu ATM-Switch geschaltet, wie eben beschrieben. Dieser enthält

Signalisierungsinformationen, die durch GSMP nach RFC-2297 verpackt und an den Switch gesendet werden. Es handelt sich hierbei um ein Master-/Slave-Protokoll, das kein fester Bestandteil des IP-Switching ist, sondern eine Systemoptimierung darstellt. Folgende Informationen lassen sich dabei unterscheiden:

- **Nachrichten für das Verbindungsmanagement:** Der Switch ist dadurch in der Lage, Verbindungen auf- und abzubauen sowie zu bestätigen oder abzulehnen. Veränderungen werden ebenfalls aufgenommen.
- **Nachrichten für das Management der Ports:** Die einzelnen Ports des ATM-Switches werden kontrolliert. Der IP-Switch-Controller kann dadurch die Zustände (aktiv, inaktiv, Reset, Looped Back) der Ports beeinflussen.
- **Nachrichten für Statistiken der Switches:** Der IP-Switch-Controller kann dadurch statistische Informationen der Input-/Output-Ports sowie die Anzahl der aktiven Verbindungen abfragen.
- **Nachrichten zur Konfiguration:** Der verwendete VCI/VPI-Adressraum kann dadurch definiert werden sowie andere Parameter, die für die Einstellung der ATM-Switches notwendig sind.
- **Nachrichten zur Ereignissteuerung:** Dem IP-Switch-Controller werden aktuelle Informationen über bestimmte Ereignisse wie neue oder gestörte Ports sowie ungültige VCI/VPI-Werte mitgeteilt.
- **Nachrichten zur Synchronisation:** Der IP-Switch-Controller wird mit dem ATM-Switch hierüber synchronisiert. Das heißt, die Entitäten, die über die GSMP-Verbindungen miteinander verbunden sind, werden gleichgeschaltet. Weiterhin werden Veränderungen und die Identität der Entitäten festgestellt.

Ein End-to-end-QoS ist prinzipiell in heterogener geswitchter Netzumgebung umsetzbar. Jedoch ist dies nur machbar, wenn eine Priorisierung der Datenströme erfolgen kann. Weiterhin ist ein großer Nachteil von Ipsilon IP-Switching, dass nur das Netzwerk einen QoS anfordern kann, nicht aber eine Applikation, da das Netzwerk die VC-Verbindung aufbaut. QoS wird demnach über lokale Einstellungen, über die ATM-Hardware oder im Zusammenspiel mit RSVP unterstützt. [NEHH+98]

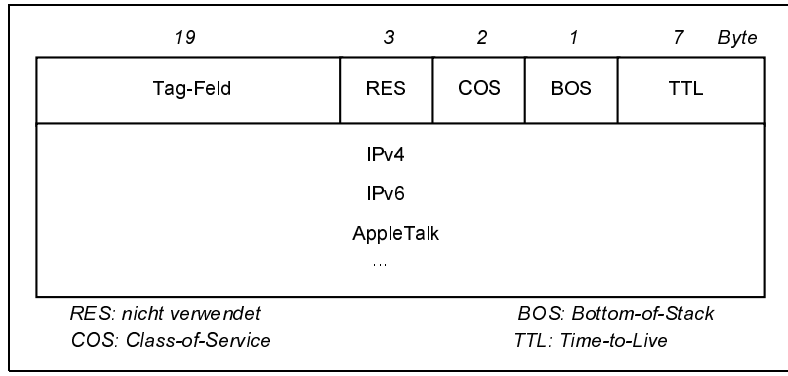
Tag-Switching

Im Vergleich zum Ipsilon IP-Switching ist das Cisco Tag Switching nach RFC-2105 wesentlich weiter gefasst. So beschränkt sich das Tag Switching nicht nur auf ATM als zugrunde liegende Technologie, sondern kann auch auf anderen Schicht-2-Protokollen wie Ethernet, FDDI oder Token Ring aufsetzen. Darüber hinaus bietet das Tag Switching eine Reihe zusätzlicher Funktionen. Ziel ist es, die Router-Performance in der WAN-Umgebung zu erhöhen. Dies kann durch Reduzierung der Paketkomplexität beim Forwarding erreicht werden. Verbesserte Skalierbarkeit und höhere Funktionalität des Netzwerk-Schicht-Routing sind die erweiterten Merkmale. Unicast-Pakete, die durch IP-Router weiterge-

leitet werden, müssen bisher in IP-Adresspräfixtabellen nach der übereinstimmenden Präfix suchen. Das führt zu Verzögerungszeiten. Tag-Switching ersetzt diesen Vorgang so gut wie möglich durch feste Längenadressverweise in der Hardware, wie das bereits bei ATM oder Frame Relay der Fall ist. Dies verbessert die Performance des Forwarding und ermöglicht neue Funktionen.

Das Tag Binding[54] beim Tag Switching basiert auf dem Control-Driven-Ansatz. Hierzu verwendet jeder Tag Switching Router[55] (TSR) herkömmliche Routing-Protokolle wie OSPF oder BGP. Diese Protokolle werden anders als beim Ipsilon IP-Switching ausschließlich für das Erstellen der Tag Forwarding Information Base[56] (TFIB) verwendet. Wie bereits erwähnt wurde, ist beim Control-Driven Label Binding kein konventionelles Routing der Datenpakete mehr notwendig. Darüber hinaus wird beim Tag Switching das Tag Binding nach dem Downstream-Verfahren erstellt.

Abb. 4.23 Tag-Einbettung in bestehende Protokolle



Das Tag Switching realisiert neben dem Destination-Based-Routing auch Source-Based-Routing. Hierbei kann der Sender für die Datenpakete eine bestimmte Route festlegen, um bestimmte Teile des Netzwerks zu entlasten. Das Tag Switching verwendet hierzu das Protokoll RSVP[57]. Das Tag Switching zeichnet sich zusätzlich durch Multicast-Funktionalität und eine effektive Unterstützung von LSP-Tunnel aus. Diese Vorzüge wurden für MPLS übernommen.

Tag-Switching besteht aus zwei Komponenten: der Forwarding-Komponente, die die Tag-Information in den Paketen und die Tag-Informationsbasis in den Switches verwendet, und die Kontrollkomponente, die den Tag erzeugt und in das Netz verteilt. Tags besitzen eine kurze und fest definierte Adresslänge, wobei nicht ein eigenes Paketformat definiert wurde, sondern man Tags

54 Das Tag entspricht dem Label.
55 Entspricht dem LSR.
56 Entspricht der LS-Forwarding Tabelle.
57 Ressource Reservation Protocol

zum bestehenden Paketformat hinzufügt (siehe Abb. 4.23). Es gibt verschiedene Wege, die Tag-Information in die Paketstruktur einfließen zu lassen. Beispielsweise kann ein 32-Bit-Tag auf ein Netzschichtpaket gesetzt werden, wodurch man IPv4, IPv6, AppleTalk und andere Formate verwenden kann. Abb. 4.23 zeigt das Tag-Format, welches auf Schicht-2-Mechanismen beruht. Die Merkmale von Tag-Switching bei ATM lassen sich dabei wie folgt zusammenfassen:

- Tags können im VCI- und/oder VPI-Feld eingebettet werden.
- Downstream nach Bedarf wird bei der Tag-Zuordnung verwendet.
- ATM-Switches müssen Kontrollkomponenten von Tag-Switching implementiert haben, um Routing und Forwarding als Peer in einem Netz zu unterstützen.
- ATM-Tag-Switches können nur über konventionelle ATM-Switches miteinander verbunden werden, wenn man VC-Verbindungen benutzt.
- Um Zellen-Interleaving bei ATM-Tag-Switches zu vermeiden, müssen mehrere Tags in einem Pfad zugeordnet werden.
- Tag-Kontroll- und ATM-Signalisierungskomponenten arbeiten innerhalb eines ATM-Tag-Switches unabhängig voneinander.

Durch die verschiedenen Möglichkeiten, Tag-Informationen zu transportieren, unterstützt Tag-Switching jedes physikalische Medium. Tags kann man dabei wahlweise stapeln. Dies ermöglicht eine Ansammlung von Datenflüssen, die die Paketverarbeitung im Backbone beschleunigen kann. Zusätzlich kann ein Reservierungsmechanismus festgelegt werden. Tags werden dabei entweder zwischen dem Layer 2 und 3, als Teil des Layer-2-Headers (VCI/VPI-Kombination) oder Teil des Layer-3-Headers (IP-Quellen/-Empfänger/Port-Kombination) eingesetzt. Zum Informieren der anderen TSRs über die Tag Bindings wird beim Tag Switching der Transport über Routing-Protokolle bevorzugt. Für den Fall, dass dies aufgrund des verwendeten Routing-Protokolls nicht möglich ist, hat Cisco Systems das Tag Distribution Protocol (TDP) entwickelt. Es entspricht dem LDP[58]. Dies geschieht durch das Tag Distribution Protocol (TDP).

Die Forwarding-Komponente eines Tag-Switches basiert auf der Adress-Auslagerungsfunktion bzw. Label-Swapping. Jeder Tag-Switching-Knoten verwaltet dafür eine Tag Forwarding Information Base (TFIB). Wenn ein hereinkommendes Paket gekennzeichnet wird, sucht TFIB nach dem passenden Eintrag. Falls dieser Eintrag gefunden wird, zeigt die herausgehende Schnittstelle, wohin das Datenpaket weitergeleitet werden soll. Im Gegensatz zum ATM-Switching, wird die Netzwerkschicht-Information im Datenpaket verwendet, falls kein Eintrag besteht. Dabei verhält sich die Forwarding-Komponente unabhängig von der Netzwerkschicht. Drei mögliche Methoden für Tag-Zuweisungen und TFIB-Management sind:

58 Label Distribution Protocol

- **Downstream-Tag-Zuweisung:** Ein Tag-Router am Rand eines Netzes erzeugt für jede Route, die er kennt, einen Eintrag in seiner TFIB. Dabei wird für jede Route ein Eingangs-Tag festgelegt. Diese Bindung wird anschließend den anderen Switches oder Routern mittels Tag Distribution Protocol (TDP) bekannt gegeben.
- **Downstream-Tag-Zuweisung nach Bedarf:** Ein Downstream Switch stellt eine Verbindung zwischen einem bestimmten Datenstrom und einem Tag auf Verlangen des Upstream-Switches/-Routers her. Der vorgelagerte Switch hat bereits eine Zuweisung durchgeführt und überträgt nun über TDP die Zuweisungsparameter an den nachgelagerten Switch. Bei Durchführung der Zuweisung wird eine Nachricht zurückgesendet. Erst dann wird ein Eintrag in der TIB erzeugt und der Upstream-Tag wird auf den Wert gesetzt, den er von dem Downstream-Switch erhalten hat.
- **Upstream-Tag-Zuweisung:** Bei jeder Route, die ein Edge-Router kennt, wird ein Tag in der TIB zugewiesen. Anschließend macht der Router durch das TDP die Tags bei allen nachgelagerten Switches bekannt. Diese erzeugen ebenfalls den Eintrag Eingangs-Tag in der TIB mit dem Tag, den sie erhalten haben.

Bei den ersten beiden Möglichkeiten ist der Switch für die Erstellung der Tag-Bindings zuständig. Dies betrifft hereinkommende Datenströme und empfangene Tag-Bindings für ausgehende Datenpakete an die Switch-Nachbarn. Upstream-Zuweisung verhält sich hingegen umgekehrt. Zusätzlich kann man die Tag-Verteilung in explizite Reservierung[59] und bezogen auf die Zieladressen[60] unterteilen. Die erste Möglichkeit ist sehr eng mit dem Verbindungsaufbau bei ATM verbunden. Für das Routing müssen die Tag-Switches auch in der Lage sein, Routing-Funktionen für die unterstützten Protokolle (IPv4, IPv6, AppleTalk usw.) anzubieten. Dabei sind Routing-Protokolle zum Schreiben der Präfixeinträge notwendig, die anschließend den Tags zugeteilt werden. Routing-Updates können die Tags huckepack nehmen (Distance/Path Vector Protocol) oder separat das Tag Distribution Protocol verwenden. Binding Tag Distribution in Verbindung mit Routing vorzunehmen ist wesentlich einfacher als Anpassungen IP-over-ATM zu verwenden. Die Gegenwart des TDP-Protokolls vermeidet außerdem noch die Schleifenbildung (Loops). [RDKR+97]

Tag-Switching ist ein sehr effizienter Weg, um Cell-Switching-Technologien mit einer einfachen Adressen- und Routing-Struktur des Frame-Switching zu kombinieren. Aufgrund der Forwarding-Unterteilungen kann man viele Routing-Funktionen unterstützen (empfängerbasiertes, Multicast-, QoS-basiertes Routing). Somit ist sogar ein QoS-basiertes Routing ohne Unterstützung der

59 Z.B. durch das Hinzunehmen von RSVP, indem der Tag-Wert Teil einer RSVP-Nachricht ist.

60 Tags werden durch das Routing-Protokoll verteilt.

Applikationen realisierbar. Falls Tag-Switching mit IP und ATM verwendet wird, kann man die gesamte ATM-Kontrollebene (UNI, P-NNI usw.) durch die wesentlich einfacheren Kontrollkomponenten von Tag-Switching ersetzen. Falls zusätzlich noch RSVP eingesetzt wird, ist sogar eine garantierte Dienstgüte End-to-end zwischen Ingress- und Egress-Router möglich, die sich auch auf Anwendungen beziehen lässt. Hierzu gibt es bisher aber noch keine kommerziellen Produkte. Tag-Switching ist hauptsächlich eine Backbone-Technologie, die für ISPs zum effektiven Routing durch ein High-Speed Network geeignet ist. Weiterhin besteht auch hier die Möglichkeit, Multicast-Pakete über mehrere Untereinträge in der TIB zu verarbeiten. Zukünftig soll auch QoS, flexible Routing-Verarbeitung und Verarbeitung von hierarchischen Routing-Informationen von Exterior Gateway Protokollen (z.B. BGP) integriert werden. [DETK00b]

Cell Switch Router

Ein weiterer Ansatz kommt von Toshiba. Hier wird die Idee des Cell Switch Routers (CSR) verfolgt. In den Dokumenten RFC-2098 und RFC-2129 der IETF wird der Label-Switching-Ansatz von Toshiba beschrieben. Dabei ist es wie bei Tag- und Ipsilon IP-Switching nicht notwendig, dass alle Geräte in einem CSR-Netzwerk Cell Switch Router sein müssen. Normale ATM-Switches können ebenfalls genutzt werden. Das Verfahren ähnelt dabei sehr dem Ipsilon IP-Switching. Allerdings beherrscht ein CSR die normalen ATM-Funktionen wie UNI oder P-NNI. Zusätzlich sind spezielle CSR-Funktionen implementiert, die folgende Aufgaben besitzen:

- **Default VC:** Normales Routing wird für den IP-Verkehr standardmäßig angewendet. Die Datenpakete werden über einen Default VC gesendet, der beim Start des CSR zwischen diesen aufgebaut wird. Der nächste Hop, über den das IP-Datenpaket geschickt wird, wird vom CSR durch die Routing-Parameter des verwendeten Routing-Protokolls (RIP[61], OSPF) ermittelt.
- **Dedicated VC:** Eine direkte Weiterleitung der Pakete wird ermöglicht, wenn der CSR Datenpakete eines Datenstroms erhält, die er verarbeiten kann, ohne auf die IP-Schicht zu gehen. Ist das der Fall, wird sofort eine dedizierte Verbindung für diesen Datenstrom aufgebaut.

Zum Aufbau eines Dedicated VC wird die Zieladresse des IP-Pakets durch das ATMARP[62]-Verfahren nach RFC-2225 in eine ATM-Adresse aufgelöst. Danach wird über die ATM-Signalisierung eine direkte Virtual Connection (VC) aufgebaut. Die Zieladresse muss dabei ebenfalls ein CSR sein, der alle VCs zu einer bestimmten IP-Adresse wieder konzentriert.

61 Routing Information Protocol
62 ATM Address Resolution Protocol

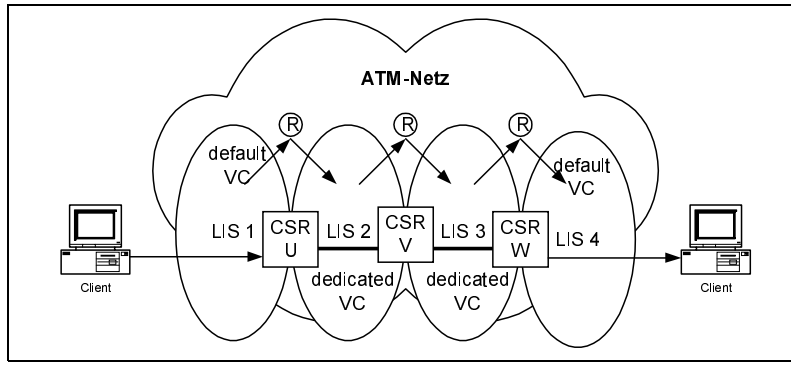


Abb. 4.24 Cell Switch Routing von Toshiba

Ein CSR-Netzwerk kann man zu den Shortcut-Verfahren zählen und arbeitet ähnlich dem Next-hop Resolution Protocol (NHRP). Unterschiede sind dennoch vorhanden. So wird zwar das IP-Routing verwendet, um IP-Subnetze miteinander zu verbinden. Um innerhalb eines Subnetzes die Verbindung aufzubauen, wird allerdings der Verbindungsaufbau von der ATM-Schicht durchgeführt. Dadurch ist es möglich, dass die gefundene Gesamtverbindung nicht genauso effizient ermittelt wird, wie mit dem Protokoll NHRP. Weiterhin können Wege durch ein Netzwerk während einer bestehenden Verbindung dynamisch umgeleitet werden. Dies kann durch Fehler auf der physikalischen Schicht notwendig sein. CSR unterstützt zusätzlich Multicast-Pakete. Innerhalb eines Subnetzes werden dazu Punkt-zu-Mehrpunkt-Verbindungen über ATM realisiert, während zwischen den Subnetzen mehrere Verbindungen durch die CSR ausgenutzt werden. Da die Flussklassifizierung von CSR vorgenommen wird, kann man schnell wenige Pakete über den normalen Routing-Weg leiten, während man für lang andauernde Datenströme einen dedizierten Kanal anfordert. Dieser Kanal kann dann auch QoS-Parameter unterstützen, da der VC auch die Dienstart enthält. [KNE97] [NKSM+97]

Aggregate Route IP Switching (ARIS)

Das Aggregate Route IP Switching (ARIS) wurde von der Firma IBM entwickelt, um ebenfalls den Durchsatz von IP und anderen Layer-3-Protokollen auf Wire-Speed zu erhöhen. Dafür hat man Integrated Switch Router (ISR) definiert, die neben Switching- auch Routing-Funktionen beinhalten. Die ISRs stellen am Anfang untereinander eine Verbindung her, die zu bestimmten Eingangs- oder Ausgangspunkten des ARIS-Netzwerks führen. Diese Punkte lassen sich durch die üblichen Routing-Protokolle ermitteln. Die Switching-Pfade werden durch den Austausch von ARIS-Nachrichten aufgebaut. Die Topologie wird dabei in Baumstruktur ausgeführt. Erhält ein Ingress-ISR ein Datenpaket, sucht er in seiner Forwarding Information Base (FIB) nach dem passenden Präfix. Die Information wurde hierbei aus der Routing-Tabelle der verwendeten Routing-Verfahren gewonnen. Jetzt ist er in der Lage, das passende Switched

Path Label[63] (SPL) zuzuordnen. Anschließend sendet der ISR die Daten auf dem geswitchten Pfad. Kann kein Eintrag gefunden werden, dann wird das Datenpaket konventionell geroutet.

ARIS verwendet zwei Datenbasen, um geswitchte Wege erkennen zu können. Dies sind die FIB und die Routing Information Base (RIB). Mit Hilfe der FIB ermittelt der ISR, auf welchem geswitchten Pfad die Layer-3-Pakete gesendet werden können. Er enthält eine Tabelle, die folgende Einträge beinhaltet:

- Ziel-Präfix
- Ausgehende Schnittstelle
- IP-Adresse des nächsten Routers
- Ausgangsidentifikator
- Label des geswitchten Pfades

Den Zusammenhang zwischen der IP-Adresse des nächsten Routers und dem Ausgangsidentifikator kann der ISR aus den Daten des Routing-Protokolls ermitteln. Nur der Zusammenhang zwischen dem Ausgangsidentifikator und dem Label des geswitchten Pfades wird über das ARIS-Protokoll hergestellt. Die RIB entspricht eher einer konventionellen Routing-Tabelle. Sie wird ebenfalls von ISR verwendet, um Kommunikationspunkte nach außerhalb feststellen zu können. Dadurch kann ein Ausgangsidentifikator erkannt werden, um diesen dann in anderen Datenbasen zu verwenden.

ARIS verwendet unterschiedliche Nachrichten, um eine Kommunikation in geswitchter Umgebung zu etablieren:

- **Acknowledgement:** Diese Nachricht wird als positive oder negative Antwort auf alle ARIS-Meldungen gesendet.
- **Establish:** Diese Nachricht wird als periodische Nachricht allen ISRs zugesendet, durch die ein geswitchter Pfad aufgebaut werden kann bzw. gehalten wird.
- **Init:** Ein ISR gibt seinen Nachbarn hiermit seine Existenz bekannt.
- **Keep Alive:** Ein ISR zeigt seinen Nachbarn an, dass er noch funktionsfähig ist. Diese Nachricht wird nur dann gesendet, wenn in einem bestimmten Zeitintervall noch keine andere Nachricht gesendet wurde.
- **Teardown:** Diese Nachricht ist wichtig, wenn ein ISR die Verbindung zu einem Ausgang verliert, wenn also bestimmte Routing-Gebiete außerhalb des ARIS-Netzwerks nicht mehr vorhanden sind.
- **Trigger:** Wenn ein ISR die Verbindung zu einem benachbarten Switch unterbricht, sendet er anschließend diese Nachricht an den nächsten ISR. Dadurch wird ein erneuter Verbindungsaufbau signalisiert, der durch die Establish-Nachricht umgesetzt wird.

63 Entspricht dem allgemein definierten Label.

Durch die Establish-Nachricht wird zusätzlich eine Schleifenerkennung durchgeführt. Ein ISR überprüft nämlich, wenn er eine Establish-Nachricht bekommt, zuerst, ob der Pfad fehler- und schleifenfrei ist. Falls dies nicht der Fall ist, löscht der ISR den alten Pfad und setzt einen neuen, fehlerfreien Pfad auf. Weiterhin definieren die in den Establish-Nachrichten enthaltenen Ausgangsidentifikatoren einen gerouteten Weg durch das ARIS-Netzwerk. Je nach Routing-Information wird dabei zwischen den Eigenschaften unterschieden. Der IP-Empfänger-Präfix ordnet jeder IP-Route einen eigenen Switching-Pfad zu. Hierfür wird das Routing-Protokoll RIP verwendet. IP-Adressen für den Ausgang verwenden hingegen BGP oder OSPF.

Das ARIS-Verfahren ist ebenfalls dem Tag-Switching von Cisco sehr ähnlich. Auch ARIS verwendet eine Kennzeichnung, die Switched Path Label (SPL) genannt wird. Zusätzlich ist es jedoch auch möglich, bereits vorhandene Layer-2-Markierungen zu verwenden, die ARIS speziell interpretieren kann. Ein Vorteil von ARIS ist, dass es unabhängig von dem verwendeten Protokoll arbeitet. Die Nachrichten werden einfach in das jeweilige Paket gepackt und verschickt. Vorteilhaft gegenüber Tag-Switching ist jedoch, dass Schleifen innerhalb des geswitchten Pfades vermieden werden. Ebenfalls ist es möglich, Multicast-Pakete auszunutzen. Allerdings ist bislang keine Dienstgüte beachtet worden, auch wenn das Zusammenspiel mit RSVP untersucht wurde. [DETK00b]

4.4 Multi-Protocol Label Switching (MPLS)

Die Aufgabe, die Anzahl der nötigen Router-Hops auf einer Übertragungsstrecke so gering wie möglich zu halten, ist heute noch nicht optimal gelöst. So wird beispielsweise in heutigen IP-Netzen jedes einzelne Paket von einem Router zum nächsten weitergereicht. Diese müssen dazu die Header der Datenpakete analysieren und anhand der darin abgelegten Informationen eine Wegewahl treffen (Hop-by-hop-Routing). Das Verfahren ist relativ aufwendig und erhöht die Verzögerungszeiten. Aus diesem Grund wurde das Label-Switching-Verfahren entwickelt, welches als Grundlage für einen gemeinsamen Standard-MPLS gilt.

Die MPLS Working Group treibt die Standardisierung stark voran. Wesentliche Punkte von MPLS wie seine Architektur oder das Label Distribution Protocol (LDP) wurden bereits festgelegt. Allerdings fehlt es u.a. noch an einer gemeinsamen Signalisierung, da hier unterschiedliche Verfahren im Einsatz sind. Trotz fehlender Standards ist MPLS allerdings bereits von einigen Router-Herstellern implementiert worden und wird in einigen Netzen bereits eingesetzt. Die Vorteile von MPLS lassen sich wie folgt zusammenfassen:

- **Virtual Private Network (VPN):** Durch das Label-Konzept ist MPLS sehr gut dazu geeignet, VPNs zu realisieren.

- **Multi-Protocol- und Multi-Link-Unterstützung:** Obwohl hauptsächlich für IP Version 4 und Version 6 genutzt, bietet MPLS auch die Möglichkeit, andere Protokolle wie IPX zu übertragen. Weiterhin ermöglicht es die Kommunikation und Ressourcenreservierung über verschiedene Techniken, wie zum Beispiel Ethernet, PoS, ATM und WDM.
- **Quality-of-Service:** Mit MPLS ist es möglich, Ressourcenreservierung im Netz vorzunehmen. Hierfür gibt es speziell angepasste Verbindungsaufbauprotokolle, die diese Reservierungen durchführen.
- **Verschiedene Service-Klassen:** MPLS ermöglicht dem Service Provider, seinen Kunden verschiedene Dienstklassen anzubieten. Der Kunde profitiert von dieser Wahlmöglichkeit. Er wählt den Dienst nach seiner Qualität und den damit verbundenen Kosten.
- **Einsatz mit ATM:** Kombiniert mit ATM bietet MPLS auch harte Dienstgüte wie Bandbreitengarantien, Zusicherung von Verzögerungszeiten und Laufzeiten der Pakete. Vorhandene ATM-Switches können in ein MPLS-Netz integriert und weitergenutzt werden, auch wenn sie selbst kein MPLS unterstützen.

MPLS funktioniert nach dem Prinzip des Label Swapping und verhält sich demnach sehr ähnlich wie Tag-Switching und ARIS. Allerdings spezifiziert MPLS deutlich mehr, als in beiden Verfahren beschrieben ist. Das heißt, MPLS skaliert kleine Netzwerke und hohe Vergabezahl der Label bis hin zu großen Netzwerken, in denen die Labels für große Routing-Gebiete vergeben werden. MPLS macht sich den Umstand zunutze, dass Layer-3-Protokolle Informationen besitzen, die nicht für die Bestimmung des nächsten Routers bzw. Hops nötig sind. Für die einfache Weiterleitung von Paketen reichen wesentlich weniger Informationen aus.

Die Vergabe eines Labels wird bei MPLS immer durch eine Netzkomponente durchgeführt. Nachdem dies geschehen ist, wird der ihr vorgelagerte Nachbar über diese Entscheidung informiert. Gleichzeitig wird mitgeteilt, welches Label für den Datenstrom vergeben wurde. Dabei besitzt MPLS eine hohe Skalierbarkeit, da Label für ganze Routing-Gebiete sowie einzelne Datenströme vergeben werden können. Dabei wird den topologieorientierten Verfahren der Vorzug gegeben. Switching-Pfade durch das Netz werden durch ein Protokoll festgelegt, welches die Verteilung der Label im Gesamtnetzwerk steuert. Zusätzlich können aber auch einzelne Switching-Pfade zusammengefasst werden. Dadurch entsteht ein vom Engress-LSR aus gesehener Switching-Baum.

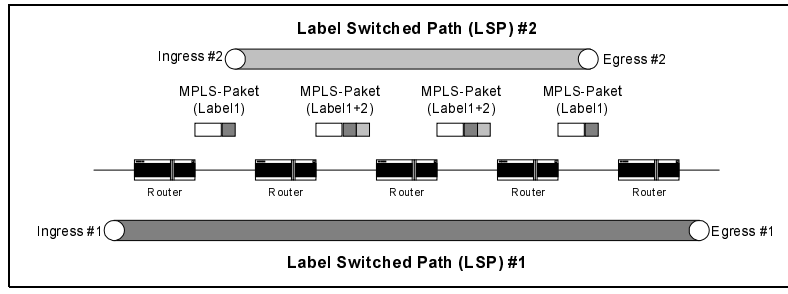
4.4.1 Forwarding-Komponente

Im MPLS-Verfahren wird nicht nur ein einzelnes Label definiert, vielmehr können in einer Hierarchie mehrere Labels an ein Datenpaket angefügt werden. Das Festlegen von Labels wird in der Spezifikation RFC-3031 beschrieben. Der

Stack arbeitet nach der Methode Last-In, First-Out (LIFO). Wird in einem LSR einem Paket, das bereits ein Label enthält, ein weiteres Label angefügt, so erfolgt das Switching in den folgenden LSR anhand des zweiten Labels. Wird dieses Label entfernt, so werden die Vermittlungsentscheidungen der folgenden LSR[64] wieder auf Basis des ursprünglichen Labels getroffen. Zur Unterstützung dieser Funktion verfügt das Label über ein Bit, welches für den Wert 1 den „Bottom-of-Stack" anzeigt. Dieses Bit wird ausschließlich für das erste Label auf den Wert 1 gesetzt. Werden weitere Labels an das MPLS-Paket angefügt, so werden diese durch den Wert 0 gekennzeichnet. Auf diese Weise kann ein LER[65] entscheiden, ob er das Paket nach Entfernen eines Labels konventionell auf Schicht 3 weiterhin routen (für „Bottom-of-Stack" = 1) oder es anhand eines weiteren Labels switchen muss (für „Bottom-of-Stack" = 0).

Aufgrund der Einführung des Label-Stacks erweitert sich die Definition des LSP: Ein LSP beginnt im Ingress, endet im Egress und verwendet nur eine Hierarchiestufe des Labels. Des Weiteren kann ein Ingress/Egress auch in einem LSR und nicht nur in einem LER bestehen.

Abb. 4.25 Label Switched Path mit unterschiedlichen Labels



Die Anwendung eines Label-Stacks besteht in LSP-Tunnel oder der Unterstützung von Routing-Hierarchien. Dadurch lassen sich relativ einfach Virtual Private Networks (VPN) aufbauen. Für den Transport des Labels wurden bereits zwei Verfahren unterschieden: Transport im Shim-Header und Transport über umdefinierte Felder des Schicht-2-Headers. Welches der Verfahren Einsatz in einer MPLS-Domain findet, ist abhängig von der verwendeten Schicht-2-Technologie. Werden von den LSR zur Übertragung der MPLS-Pakete PPP- oder LAN-Verbindungen verwendet, so wird der Transport im Shim-Header vorgenommen. In der Spezifikation RFC-3031 wird dieses Verfahren auch als Generic MPLS Encapsulation bezeichnet. Ein LSR hingegen, der beispielsweise in einer ATM- oder FR-Umgebung arbeitet, verwendet für den Transport des Labels das zweite Verfahren. Die folgenden Abschnitte befassen sich mit einer ausführlichen Darstellung beider Transportlösungen. [RVC01]

64 Label Switch Router
65 Label Edge-Router

Generic MPLS Encapsulation

Auf die Funktion des Shim-Headers wurde bereits im Rahmen der Darstellung des LS-Prinzips erläutert. An dieser Stelle folgt nun eine Betrachtung des Shim-Header-Aufbaus für MPLS. Die Struktur wird in der Spezifikation RFC-3032 beschrieben und stellt sich wie in Abb. 4.26 gezeigt dar.

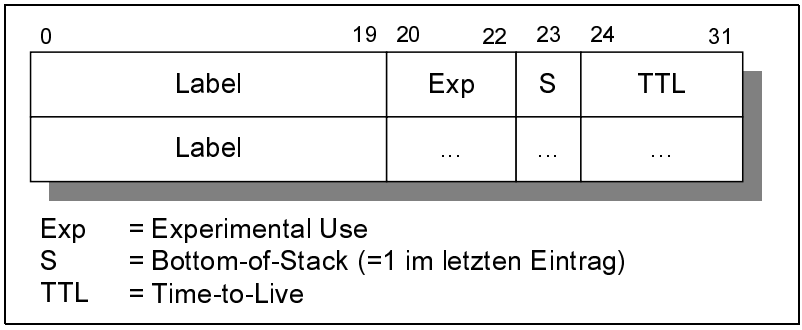


Abb. 4.26
Shim-Header bei MPLS

Die unterschiedlichen Felder haben folgende Eigenschaften:

1. **Label-Feld:** Dieses Feld umfasst 20 Bit und trägt das eigentliche Label für die Pakete einer bestimmten FEC[66]. Die Werte 0-15 wurden reserviert. RFC-3032 definiert die Bedeutung der Werte 0-3, die Werte 4-15 stehen für zukünftige Festlegungen zur Verfügung.
2. **Exp-Feld:** Dieses Feld beinhaltet drei Bits und ist reserviert. Mögliche Anwendungen werden noch an anderer Stelle betrachtet.
3. **S-Feld:** Dieses Feld ist 1 Bit groß und kennzeichnet für den Wert 1 den „Bottom-of-Stack".
4. **TTL-Feld:** Das Time-to-Live-Feld besteht aus 8 Bit und wird zur Verhinderung von Schleifenbildungen eingesetzt.

Werden mehrere Labels verwendet, so besteht jeder Eintrag des Label-Stack aus den in Abb. 4.26 gezeigten Feldern (vollständiger Shim-Header). Die Wertigkeit der Label-Stack-Einträge nimmt in Richtung der Sicherungsschicht (Data Link Layer) zu.

Durch die zusätzlichen Bytes, die einem Paket durch einen Shim-Header (bzw. mehrere Shim-Header im Label-Stack) angefügt werden, erhöht sich die Paketgröße. Dies kann zu Problemen bei der Übertragung führen. Wie in konventionellen Netzwerken auch, kann in den Netzelementen einer MPLS-Domäne die Situation eintreten, dass ein empfangenes Paket zu groß ist, um es über den Ausgangsport weiterzuleiten. Die Ursache dieses Problems liegt in der Regel darin, dass verschiedene Netzwerke unterschiedliche maximale Paketgrößen unterstützen. Der Sender versucht, die Paketgröße möglichst groß zu halten, um den relativen Overhead-Anteil zu minimieren, berücksichtigt aber

66 Forwarding Equivalence Class

nicht die maximal zulässigen Paketgrößen der Netzwerke[67], die auf dem Weg ins Zielnetz durchquert werden müssen. [RTFR+01]

Bei MPLS kann darüber hinaus der Fall auftreten, dass ein empfangenes Paket erst durch das Anfügen eines Shim-Headers die kritische Paketgröße überschreitet. Um eine Fragmentierung, die eine zusätzliche Bearbeitung der Pakete erfordert, zu vermeiden, muss im Netzelement, welches das Paket generiert, sichergestellt werden, dass das Paket in keinem der zu durchquerenden Netzen die Schicht-3-Maximum Transfer Unit (MTU) überschreitet. Die MTU definiert die maximale Größe des Pakets, bei der es in einem Netz nicht fragmentiert werden muss. Aus diesem Grund wurde in der Spezifikation RFC-1191 ein Mechanismus festgelegt, um die kleinste MTU aller Netze auf der gesamten Übertragungsstrecke zu bestimmen. Das Path-MTU-Discovery-Verfahren wird aber nicht von allen Netzelementen unterstützt. Hosts, die dieses Verfahren nicht anwenden können, generieren in der Regel nur Pakete, die in ihrer Größe 576 Byte nicht überschreiten. Da die MTU in den meisten Schicht-2-Technologien 1500 Byte oder mehr beträgt, können die Pakete selbst nach Anfügen mehrerer Label-Stack-Einträge ohne Fragmentierung übertragen werden. Dennoch ist nicht auszuschließen, dass in diesen Hosts auch Pakete generiert werden, welche die MTU überschreiten und eine Fragmentierung auf dem Weg ins Zielnetz erfordern. Eine Fragmentierung in der MPLS-Domain ist ebenfalls notwendig, wenn der Host das Path-MTU-Discovery-Verfahren unterstützt und anhand dessen die Paketgröße exakt auf die MTU festlegt. Durch Anfügen des Shim-Headers wird die MTU dann überschritten. [MODE90]

Die Spezifikation RFC-3032 definiert für IPv4-Mechanismen, wie Pakete beim Anfügen eines Shim-Headers fragmentiert werden müssen. Jeder LSR, der in der Lage ist einen Shim-Header anzufügen[68], muss über diese Mechanismen verfügen. Die Prozeduren für ein Paket, das die MTU überschritten hat, werden am Beispiel eines IPv4-Datenpakets dargestellt.

Empfängt ein LER ein IPv4-Datenpaket, welches nach Anfügen des Shim-Headers die MTU überschreitet, so wird zunächst das DF[69]-Bit des IP-Headers überprüft. Ist dieses Bit nicht gesetzt, kann der LER das Paket entweder verwerfen oder fragmentieren. Bei einer Fragmentierung wird zunächst der Shim-Header wieder entfernt und das IPv4-Datenpaket in Fragmente zerlegt. Die Fragmentgröße muss die MTU mindestens um die Größe des Shim-Headers unterschreiten. Es sollte zusätzlich bei der Wahl der Fragmentgröße beachtet werden, dass eventuell in folgenden LSRs weitere Shim-Header angefügt werden. Um eine weitere Fragmentierung in der MPLS-Domain zu vermeiden, ist

67 Maximum Segment Size (MSS): Es handelt sich dabei um die Rahmenlänge, die eine Schicht 4 (z.B. die TCP-Schicht) senden soll, damit das Paket ohne segmentiert werden zu müssen, optimal gefüllt werden kann.

68 Wie beispielsweise ein LER

69 Don't Fragment.

es vorausschauend, die Fragmentgröße klein genug zu wählen, um mehrere Label-Stack-Einträge transportieren zu können. Nachdem das Datenpaket zerlegt wurde, wird jedes Fragment mit dem gleichen Shim-Header versehen und weitergeleitet. Das Zusammensetzen des ursprünglichen IPv4-Datenpakets erfolgt im Ziel-Host. [RTFR+01]

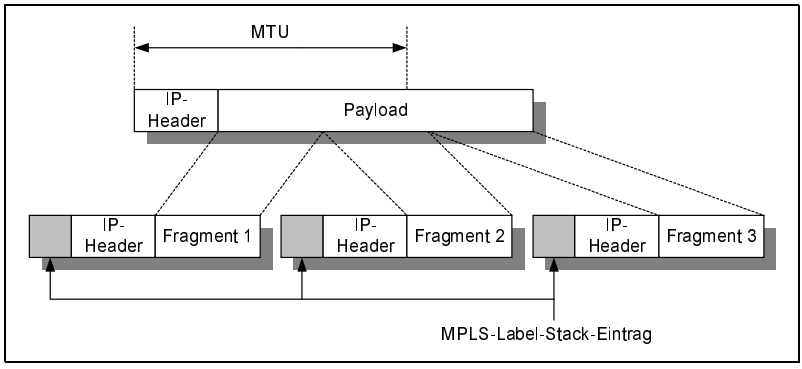


Abb. 4.27 Fragmentierung eines IP-Pakets

Empfängt ein LER ein IPv4-Datenpaket mit gesetztem DF-Bit, welches nach Anfügen des Shim-Headers die MTU überschreitet, so muss der LER das Paket verwerfen und eine ICMP-Mitteilung an den Sender weiterleiten. Die ICMP-Mitteilung informiert den Sender über die MTU der MPLS-Domäne[70], wodurch ein neues Datenpaket generiert werden kann, welches die aktuelle MTU nicht überschreitet.

Labeltransport über undefinierte Schicht-2-Felder

Für die Betrachtung des Label-Transports über umdefinierte Felder des Schicht-2-Headers beschränkt sich die Betrachtung auf ATM-basierte MPLS-Netzwerke. Definitionen zu Frame-Relay-basierten Netzwerken sind ebenfalls vorhanden und können in der Spezifikation RFC-3034 nachgelesen werden. Sie sind bezogen auf die Echtzeitfähigkeit nicht relevant, weshalb hier nicht darauf eingegangen wird. [CDM01]

Für den Label-Transport in einem ATM-Header werden das VCI- und VPI-Feld verwendet. Dabei wird in diesen Feldern ausschließlich der Label-Wert transportiert. Zusätzlich muss auch immer der Shim-Header transportiert werden. Der Shim-Header wird vor der Segmentierung des Pakets in ATM-Zellen eingefügt. Da sich die Exp-, S- und TTL-Felder nicht im ATM-Header befinden, können sie auch nicht in den LSR verarbeitet werden. Dennoch ist es notwendig, sie im Shim-Header zu transportieren, damit die Informationen auch nach Durchqueren des ATM-basierten MPLS-Netzwerks zur Verfügung stehen.

70 Entspricht der MTU der zugrunde liegenden Schicht-2-Technologie abzüglich der Bytes für die Label-Stack-Einträge.

Es gibt drei Verfahren, um den Labelwert im VCI- und VPI-Feld zu transportieren: [RVC01]

- **SVC Encoding:** schreibt das Top-Label des Label-Stacks in das VCI- und VPI-Feld.
- **SVP Encoding:** schreibt das Top-Label des Label-Stacks in das VPI-Feld. Falls im Label-Stack weitere Labels vorhanden sind, wird der zweite Label-Wert in das VCI-Feld geschrieben.
- **SVP Multipoint Encoding:** schreibt das Top-Label des Label-Stacks in das VPI-Feld. Falls im Label-Stack weitere Labels vorhanden sind, wird der zweite Label-Wert in einen Teil des VCI-Felds geschrieben. Der andere Teil des VCI-Felds wird für die Kennzeichnung des Ingress für diesen LSP verwendet. Auf diese Weise können Zellen einem Flow zugeordnet werden, wodurch es möglich ist, Label Merging zu betreiben.

Insgesamt gilt, dass für den Transport von MPLS-Paketen über ATM-Zellen immer der AAL-Typ 5 verwendet wird, wie dies auch bei den anderen IP-over-ATM-Verfahren der Fall ist. [RTFR+01]

Verhinderung von Schleifenbildung

Routing-Protokolle stellen in einem stabilen Netzwerk durch ihre Algorithmen sicher, dass Schleifenbildung unterbunden wird. Fällt aber eine Verbindung oder ein Netzelement aus, so kann dies unter ungünstigen Bedingungen dennoch zu einer temporären Schleife führen. Durch Austausch von Routing-Informationen werden die Routing-Tabellen aktualisiert und die Schleife durch Rerouting beseitigt. Dieser Vorgang nimmt eine gewisse Zeit in Anspruch. Werden in dieser Zeit Pakete in die Schleife vermittelt, so brauchen die Pakete in der Schleife unnötig Ressourcen auf. Diese Ressourcen stehen dann nicht mehr für die dringende Vermittlung der Routing-Informationen zur Verfügung.

Um dem Problem der Schleifenbildung entgegenzuwirken, können zwei Strategien angewendet werden: [DARE00]

- **Loop Prevention:** verhindert, dass Pakete in eine Schleife vermittelt werden.
- **Loop Mitigation:** verhindert, dass Pakete in einer Schleife die Ressourcen aufbrauchen.

MPLS verwendet im Shim-Header über das TTL-Feld das Verfahren Loop Mitigation, welches auch im IPv4-Header über das TTL-Feld bzw. im IPv6-Header über das Feld Hop-limit Anwendung findet. Das TTL-Feld wird von der Quelle auf einen bestimmten Wert gesetzt und in jedem Netzelement auf dem Weg zur Senke um den Wert 1 dekrementiert. Beträgt das TTL-Feld in einem Netzelement 0, so wird es nicht mehr weitergeleitet und verworfen. Wurde ein Paket in eine Schleife vermittelt, so wird es auf diese Weise nur um eine begrenzte Anzahl von Hops weitervermittelt[71]. Werden im MPLS mehrere Labels verwendet, erfolgt eine Dekrementierung in den LSRs[72] nur im TTL-Feld des Top-

Labels. Eine Unterstützung der IPv4- bzw. des IPv6-Headers wird dadurch realisiert, dass der entsprechende Wert aus dem Schicht-3-Header beim Eintritt in die MPLS-Domäne im LER in das TTL-Feld des Shim-Headers kopiert wird. Beim Verlassen der MPLS-Domäne kopiert der LER das TTL-Feld des Label wieder in das entsprechende Feld des Schicht-3-Headers zurück.

In bestimmten Situationen kann es vom Netzbetreiber gewünscht sein, das TTL-Feld des IP-Headers nur um den Wert 1 zu dekrementieren, wenn die Pakete eine MPLS-Domäne durchqueren. Auf diese Weise können Informationen über die Topologien geheim gehalten werden. Wird MPLS in einem ATM-Netzwerk implementiert, so erfolgt das Switching anhand des ATM-Headers. Der ATM-Header enthält allerdings kein TTL-Feld. Aus diesem Grund müssen andere Mechanismen verwendet werden, um dem Problem der Routing-Schleifen entgegenzuwirken. Aus diesem Grund ist für ATM das Verfahren Hop-Count entwickelt worden, welches in der Spezifikation RFC-3035 enthalten ist. Es findet Anwendung, wenn im MPLS Ordered Downstream-on-Demand Binding betrieben wird. Hierbei fordert der Ingress ein Binding über alle LSR des LSP[73] vom Egress an. Das Modell Ordered Downstream-on-Demand stellt eine Form der Binding Distribution dar. Dieses Modell ist für den Einsatz von MPLS in einem ATM-Netzwerk vorgeschrieben. Das Hop-Count-Verfahren realisiert für Datenpakete die Strategie Loop Prevention, indem es verhindert, dass Datenpakete in die Schleife vermittelt werden.

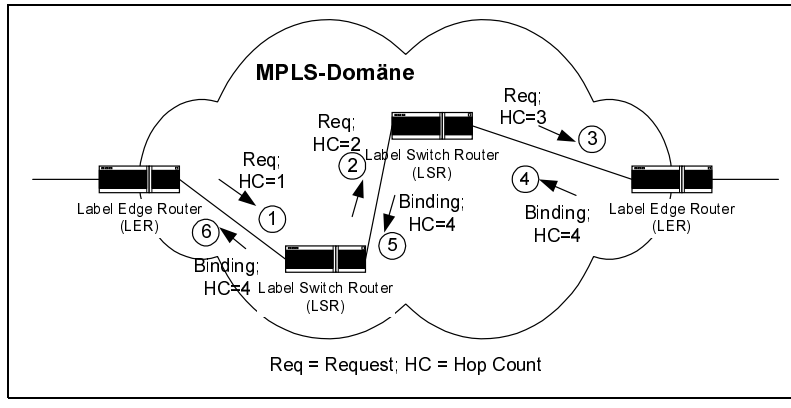
Empfängt ein LER ein Paket aus einem angeschlossenen Netz, das über die MPLS-Domäne vermittelt werden soll, sendet er zunächst eine Label-Request-Nachricht über alle LSRs des LSP zum Egress. Diese Nachricht enthält ein Feld für den Wert Hop-Count, welches vom LER, dem Ingress dieses LSP, auf 1 gesetzt wird. Diese Nachricht ist Bestandteil des LDP. Der folgende LSR inkrementiert den Wert Hop-Count und prüft vor dem Weiterleiten, ob der Maximalwert erreicht wurde. Ist dieser Wert erreicht, so muss die Label-Request-Nachricht verworfen werden. In Implementierungen von Cisco Systems beträgt der Standard-Maximalwert 254 Hops, kann aber auf jeden anderen Wert gesetzt werden. Diese Prozedur setzt sich bis zum Egress fort.

Eine Label-Request-Nachricht, die in eine Schleife vermittelt wurde, wird nur bis zum Erreichen des Maximalwerts des Hop-Counts weitergeleitet. Da sie in diesem Fall den Egress nicht erreicht, erhält der Ingress vom Egress nicht die benötigte Binding-Information, um die Datenpakete weiterzuleiten. Der Ingress muss eine neue Label-Request-Nachricht senden. Auf diese Weise können keine Datenpakete in eine Schleife vermittelt werden. Ist der LSP schleifenfrei, so empfängt der Egress die Label-Request-Nachricht mit einem bestimm-

71 Bis das TTL-Feld den Wert 0 beträgt.
72 Label Switch Router
73 Label Switched Path

ten Hop-Count-Wert. Er inkrementiert diesen Wert und sendet den Hop-Count mit der Binding-Information über den LSP zurück. Der Hop-Count wird in den LSRs nicht mehr verändert. Er trägt die Anzahl aller Hops des LSP und wird für die Manipulation des TTL-Feldes des im Ingress empfangenen Pakets verwendet.

Abb. 4.28 Das Hop-Count-Verfahren



Empfängt der Ingress die Binding-Nachricht mit dem Hop-Count, so ersetzt er den Wert des TTL-Feldes des empfangenen IP-Pakets durch die Differenz des ursprünglichen TTL-Wertes und des Hop-Counts. Beträgt der Wert des TTL-Feldes nach Abzug des Hop-Counts den Wert 0, so bestehen für den Ingress zwei Möglichkeiten:

- Verwerfen des Pakets und generieren einer entsprechenden ICMP-Nachricht
- TTL-Wert nur um den Wert 1 dekrementieren und das Paket auf Schicht 3 weiterleiten. Hierzu wird die Non-MPLS-Verbindung verwendet.

Zusätzlich zum Hop-Count-Verfahren wurde in der Spezifikation RFC-3035 ein weiteres Verfahren beschrieben, welches ebenfalls das Problem von Routing-Schleifen adressiert. Es handelt sich hierbei um das Verfahren Loop Detection via Path Vectors (LDPV), welches alternativ verwendet werden kann.

Das LDPV basiert auf dem Path Vector Object, welches eine Liste von durchlaufenen LSRs darstellt und der Request- oder Binding-Nachricht angefügt wird. Der Ablauf dieses Verfahrens beginnt mit dem entsprechenden LER[74], der seine Adresse in das Path Vector Object schreibt. Jeder weitere LSR auf dem Weg zum Ziel-LSR fügt die eigene Adresse hinzu. Hat sich eine Schleife gebildet, so wird zu einem bestimmten Zeitpunkt ein LSR die eigene Adresse in der Liste erkennen. Auf diese Weise wird die Schleife entdeckt und der LSR kann damit beginnen, entsprechende Maßnahmen zur Beseitigung einzuleiten,

74 Ingress für Request-Nachrichten und Egress für Binding-Nachrichten

bevor Datenpakete in die Schleife vermittelt werden. Das LDPV stellt eine Realisierung der Loop-Prevention-Strategie dar.

Im Gegensatz zum Hop-Count-Verfahren, das in der Regel 254 Hops benötigt, um eine Schleife zu erkennen, bietet die LDPV-Lösung eine wesentlich schnellere Schleifenerkennung. Sobald ein Hop zweimal durchlaufen wird, ist die Schleife erkannt. Nachteilig wirkt sich der durch das LDPV-System bedingte höhere Overhead aus. In jedem Hop wird eine weitere Adresse angehängt, die mit übertragen werden muss. [DLRS+01]

Schicht-3-Protokoll-Festlegung

Die Festlegungen, die für MPLS getroffen wurden, setzen für den Einsatz kein bestimmtes Schicht-3-Protokoll voraus. Bisherige Festlegungen fokussieren den Einsatz von IPv4. MPLS kann theoretisch aber auch andere Protokolle wie beispielsweise IPv6, IPX und Apple Talk unterstützen. Hieraus ergibt sich die Notwendigkeit, dass ein LER das Schicht-3-Protokoll bestimmen kann, welches er in einem MPLS-Paket transportiert. Dies ist beispielsweise dann erforderlich, wenn ein MPLS-Paket aus der MPLS-Domäne in ein nicht MPLS-fähiges Netz vermittelt wird. Der LER muss nach dem Entfernen des Shim-Headers in der Lage sein, das Paket auf Schicht 3 zu routen, wofür die Kenntnis des verwendeten Protokolls unbedingte Voraussetzung ist. Da der Shim-Header Bestandteil beider Transportverfahren ist, ermöglicht eine Kodierung des Schicht-3-Protokolls im Shim-Header eine einheitliche Lösung für alle MPLS-Netze. Nach den Festlegungen sieht der Shim-Header aber in seinem Aufbau kein Feld für die Kennzeichnung des transportierten Schicht-3-Protokolls vor. Aus diesem Grund müssen andere Mechanismen bereitgestellt werden, um das verwendete Schicht-3-Protokoll festzulegen.

Nach der Spezifikation RFC-3032 wird dies über das Label-Feld sichergestellt, indem bestimmte Labelwerte nur für bestimmte Schicht-3-Protokolle verwendet werden dürfen. Dies muss auch beim Ersetzen der Labels innerhalb der MPLS-Domäne in den LSR berücksichtigt werden. Auf diese Weise kann ein LER anhand des Labels das transportierte Schicht-3-Protokoll bestimmen. [RTFR+01]

4.4.2 Forwarding-Tabelle

Die Forwarding-Tabelle eines LSR bei MPLS besteht aus zwei Komponenten:

- Next-Hop Label Forwarding Entry (NHLFE)
- Incoming Label Map (ILM)

Der NHLFE wird für das Vermitteln eines MPLS-Pakets verwendet und entspricht prinzipiell einem Subeintrag. Er enthält die Informationen, die für das Forwarding erforderlich sind. Dazu gehören die Angabe des Next-Hops für das Paket und die Operation, die auf den Label-Stack des Pakets angewendet werden muss. Hierzu gehören beispielsweise:

- Ersetzen des Top-Labels durch ein bestimmtes Outgoing-Label
- Ersetzen des Top-Labels durch ein bestimmtes Outgoing-Label und das Hinzufügen eines weiteren Outgoing-Labels
- Entfernen eines Labels

Die Funktion der ILM besteht darin, jedes empfangene MPLS-Paket anhand seines Labels auf einen oder mehrere NHLFEs abzubilden. Sind mehrere NHLFEs für das entsprechende Paket vorhanden, so muss in der Regel ein NHLFE für das Forwarding des Pakets ausgesucht werden. Mehrere NHLFEs werden für das Label verwendet, wenn Anwendungen wie das Load Balancing betrieben werden.

Tab. 4.4 Beispiel einer Forwarding-Tabelle bei MPLS [FROMM01]

ILM	**1. NHLFE**			**2. NHLFE**		
Incoming-Label	Next-Hop	Outgoing-Label	Operationen	Next-Hop	Outgoing-Label	Operationen
41		46	Swap Label		77	Swap Label
42		86	Push Label	X	X	X
…	…	…	…	…	…	…

Zusätzlich zur ILM verfügt ein LER über die FEC-to-NHLFE (FTN) Map. Die FTN ist für die LER erforderlich, da diese im Gegensatz zu den LSR nicht nur MPLS-, sondern auch konventionelle Schicht-3-Pakete verarbeiten müssen. Die FTN entspricht in ihrer Funktion der ILM, sie bildet aber ein Paket nicht anhand des Labels, sondern über die FEC auf einen oder mehrere NHLFEs ab. [DARE00]

4.4.3 Forwarding-Algorithmus

Der Forwarding-Algorithmus bei MPLS basiert auf dem Label-Swapping, das in seinem Prinzip bereits im Zusammenhang mit den LS-Grundlagen behandelt worden ist. Hierbei wurde auch auf die Option zur Unterhaltung einer Forwarding-Tabelle pro LSR oder pro Schnittstelle eingegangen. Die Spezifikation RFC-3031 beschreibt den Ablauf der Paketvermittlung im MPLS auf der Basis des Label-Swappings.

Empfängt ein LSR ein MPLS-Paket, so bestimmt er zunächst das Top-Label. Über dieses Label und die Incoming Label Map (ILM) ermittelt der LSR dann den zugehörigen Next-Hop Label Forwarding Entry (NHLFE). Der NHLFE beinhaltet die Informationen über den Next-Hop des MPLS-Pakets, über den er das Paket weiterleiten muss. Anschließend führt er die Operationen für den Label-Stack durch, die ebenfalls im NHLFE für dieses Paket eingetragen sind. Soll das Label beispielsweise durch ein bestimmtes Outgoing-Label ersetzt

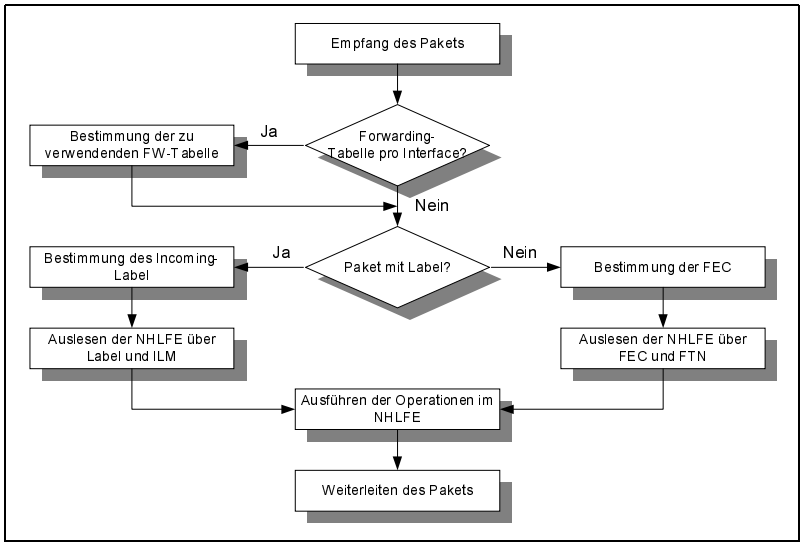


Abb. 4.29 Forwarding-Algorithmus von MPLS

werden, so wird das neue Label in das entsprechende Feld[75] des MPLS-Pakets eingeschrieben. Nachdem der LSR alle Operationen abgeschlossen hat, vermittelt er das Paket weiter. Empfängt ein LER ein Schicht-3-Paket, so besteht der einzige Unterschied im Algorithmus darin, dass zunächst der Schicht-3-Header ausgewertet wird, um die FEC des Pakets zu bestimmen. Über die FEC und die FTN-Map ermittelt der LER anschließend den NHLFE und verfährt an dieser Stelle wie bereits dargestellt weiter. Abb. 4.29 zeigt die Arbeitsweise des Forwarding-Algorithmus und verdeutlicht die Funktion.

In einem LSR kann die Situation eintreten, dass das MPLS-Paket über kein gültiges Label verfügt. Ein Label ist dann ungültig, wenn kein Binding für dieses Label besteht. In einem solchen Fall darf der LSR das Label nicht entfernen, um das Paket konventionell zu routen, da es unter Umständen eine Schleife bilden könnte. Wird beispielsweise ein MPLS-Paket, das einem Explicit Path folgt, im ersten LSR konventionell geroutet, so wird es zum LER zurückgesendet, wenn weniger Hops über den Conventional Path vorliegen. Der LER wertet den Schicht-3-Header aus, fügt das ursprüngliche Label wieder an und sendet es erneut über den Explicit Path, wodurch sich der Vorgang wiederholt. Aus diesem Grund müssen Pakete, die mit einem ungültigen Label empfangen wurden, sofort verworfen werden. Diese Vorgehensweise gilt auch für andere Situationen, in denen ein LSR das Label nicht in einen NHLFE abbilden kann. Mögliche Gründe könnten eine fehlerhafte ILM sein oder eine falsche Vermittlung zu einem LSR, der nicht Element des LSP ist.

75 Label-Wert im Shim-Header oder VCI/VPI im ATM-Header

Eine Optimierung des Forwarding-Algorithmus bietet das Penultimate Hop Popping. Hierbei verwendet der vorletzte LSR eines LSP ebenfalls den Forwarding-Algorithmus, entfernt aber das Label bevor er das Paket zum Egress weiterleitet. Der Egress empfängt das Paket und kann es direkt auf Schicht 3 weitervermitteln. Auf diese Weise müssen im Egress keine Label-Swapping-Operationen mehr durchgeführt werden. Da ein LSR nicht bestimmen kann, ob er der vorletzte Hop im LSP ist, muss dieses Verfahren vom Egress beim vorletzten LSR beantragt werden. [FROMM01]

4.4.4 Control-Komponente

Die Kontrollkomponente von MPLS beinhaltet grundsätzlich das Label Binding und die Label Distribution. Das Label Binding ist dabei die Zuordnung von Label und FEC[76]. In den Label-Switching-Grundlagen wurden grundlegende Konzepte wie Down- und Upstream Binding oder Control- und Data-Driven Binding bereits vorgestellt. Für MPLS wurde nach der Spezifikation RFC-3031 die Realisierung über das Downstream Binding gewählt. Eine Festlegung über die Verwendung von Control- oder Data-Driven Binding wurde von der MPLS-Working Group aber nicht getroffen. Dennoch ergibt sich aus der Darstellung der MPLS-Architektur in der RFC-3031, dass die Control-Driven-Lösung bevorzugt wird. Bei der Label Distribution unterstützt MPLS ausschließlich das Downstream Binding, um in der Regel benachbarte LSR über seine Label-Zuweisungen zu informieren. Werden LSP[77]-Tunnel verwendet, so ist eine Binding Distribution auch zwischen Ingress und Egress des Tunnels erforderlich.

Independent und Ordered Control

Neben dem festgelegten Downstream Binding existieren im MPLS zusätzliche Binding-Optionen, deren Verwendung von der MPLS-Implementierung in den LSR bestimmt wird. Für FEC, die anhand von Adresspräfixen definiert werden, unterstützen die LSRs im Destination-Based-Routing, je nach Implementierung der Hersteller, den Aufbau eines LSP entweder über Independent oder Ordered Control:

- **Independent Control:** erlaubt einem LSR, eigenständig FEC zu bestimmen und ihnen ein Label zuzuweisen.
- **Ordered Control:** versetzt ausschließlich Ingress oder Egress eines LSP in die Lage, FEC zu bestimmen und die Labelzuweisung durchzuführen.

Über die Binding Distribution wird den benachbarten LSR die Information über das Binding mitgeteilt. Erst nach Empfang einer solchen Nachricht darf ein LSR selbst ein Binding vornehmen und anschließend über die Distribution

76 Forwarding Equivalence Class
77 Label Switched Path

den folgenden LSR informieren. Dieser Vorgang setzt sich bis zum Ingress bzw. Egress[78] fort, mit dessen Binding der LSP dann vollständig aufgebaut ist.

Die Vorteile der Independent Control liegen in der Geschwindigkeit, mit der ein LSP aufgebaut werden kann. Durch dieses Verfahren sind die LSR nicht gezwungen, auf das Binding des jeweiligen Downstream LSR zu warten, bevor sie selbst Label-Zuweisungen vornehmen. Ordered Control hingegen zeichnet sich durch eine einheitliche Einteilung der FECs aus. Um diesen Vorteil zu verstehen, muss das Aufteilen des Verkehrs in FECs betrachtet werden. Eine einfache Verkehrsaufteilung lässt sich dadurch erreichen, dass jedem Präfix einer Routing-Tabelle eine FEC zugeordnet wird. Dabei kann in einer MPLS-Domäne die Situation eintreten, dass eine Reihe von Präfixen der gleichen Route folgen und über den gleichen Egress die MPLS-Domain verlassen.

Label Aggregation

MPLS bietet weiterhin die Möglichkeit, die FECs in einer solchen Situation zu einer übergeordneten FEC zusammenzufassen (Aggregation). Auf diese Weise wird der Label-Verbrauch und der Label-Distribution-Verkehr reduziert. Dabei können mehrere FECs, die der gleichen Route folgen, zu einer übergeordneten FEC, einer Reihe von übergeordneten FECs oder gar nicht zusammengefasst werden. Abb. 4.30 zeigt das Zusammenfassen von FECs zu einer übergeordneten FEC. Dabei stellt die Ordered Control sicher, dass die Entscheidung über das Zusammenfassen im Ingress oder Egress getroffen und von allen anderen LSRs über die Label Distribution übernommen wird. Verwenden die LSRs hingegen das Independent-Control-Verfahren, so entscheidet jedes Netzelement unabhängig von den anderen über die FEC-Aggregation. Dabei besteht die Möglichkeit, dass die LSRs verschiedene FEC unterschiedlich zusammenfassen. Haben zwei benachbarte LSRs unterschiedliche FECs gebildet, so sind zwei Situationen zu betrachten:

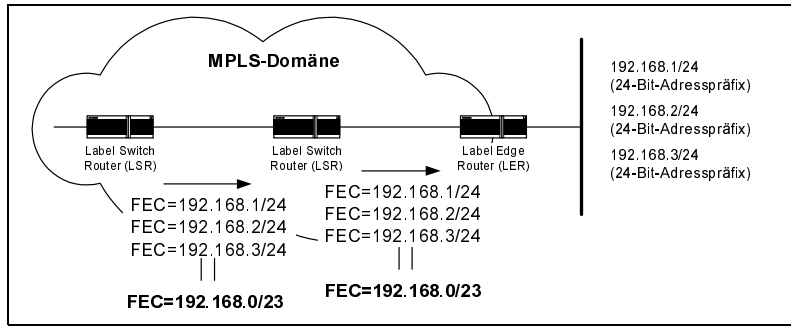
- **Feine Granularität:** Der Upstream LSR verfügt über eine feinere Granularität der FEC als der Downstream LSR[79]. Diese Situation ist beispielsweise dann gegeben, wenn der Upstream LSR jedem Präfix eine FEC zuweist, während der Downstream LSR Aggregation betreibt.
- **Grobe Granularität:** Der Upstream LSR verfügt über eine gröbere Granularität der FEC als der Downstream LSR.

Der erste Fall hat nur geringe Auswirkungen auf das Forwarding. Der Upstream LSR muss lediglich seine FECs in die FECs des Downstream LSR abbilden. Im zweiten Fall ist dies nicht ohne weiteres möglich, da die FECs des Upstream LSR mehrere FECs des Downstream LSR beinhalten und aus diesem Grund nicht auf eine einzige FEC abgebildet werden können. Die Spezifikation RFC-3031 empfiehlt für diese Situation, dass der Upstream LSR die eigenen Bindings ver-

78 Abhängig davon, wer das Binding vorgenommen hat.
79 Next-hop für den Upstream LSR

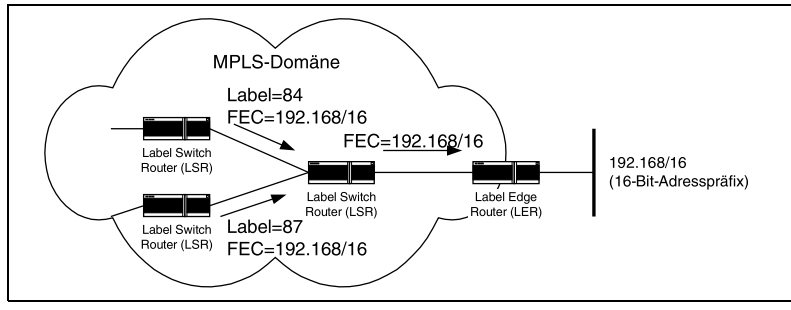
wirft und die Zuordnungen des Downstream LSR übernimmt. Diese Probleme und der damit verbundene Aufwand, neue Bindings in der MPLS Forwarding-Tabelle zu generieren, werden über die Ordered-Control-Lösung vermieden. Beide Verfahren sind vollständig kompatibel. Um die Vorteile der Ordered Control zu nutzen, müssen alle LSRs dieses Verfahren unterstützen. [RVC01]

Abb. 4.30 Aggregation von ähnlichen FEC-Zuweisungen



Label Merging Aggregation wird betrieben, um den Label-Verbrauch zu reduzieren. Dieses Ziel verfolgt auch das Label Merging. Es kann verwendet werden, wenn ein LSR mehrere Incoming Labels einer FEC zugewiesen hat. Abb. 4.31 stellt diese Situation anhand von zwei verschiedenen Incoming Labels für einen LSR dar[80]. Dieser LSR kann für die Vermittlung zum LER für jedes Incoming Label der FEC=192.168/16 je ein Outgoing Label verwenden. Hierzu müsste der LER der FECs zwei Labels zuweisen[81], wie in Abb. 4.32 in der oberen Darstellung gezeigt wird.

Abb. 4.31 Ausgangszustand für das Label Merging



Ist der mittlere LSR in der Lage Label Merging zu betreiben, so benötigt er nur ein einziges Outgoing Label für die Vermittlung zum LER. Jedes Incoming Label wird durch das gleiche Outgoing Label ersetzt, wie in Abbildung 47 dargestellt.

80 Die Incoming Labels können auch den gleichen Wert haben, was für den LSR jedoch voraussetzt, dass er eine MPLS-Forwarding-Tabelle für jedes seiner Interfaces führt.

81 Downstream Binding

Die Festlegungen der MPLS Working Group setzen nicht die Fähigkeiten des Label Merging eines LSR für dessen Einsatz in einer MPLS-Domäne voraus. Allerdings ergeben sich dadurch die Vorteile, dass die Anzahl der Outgoing Labels pro FEC immer 1 ist. Somit ist man durch das Label Merging unabhängig von der Anzahl aller Upstream LSRs in Bezug auf die FECs. Somit muss ein MPLS-Netzwerk nicht einheitlich sein, sondern kann auch unter Verwendung beider LSR-Typen[82] aufgebaut werden. Die Spezifikation RFC-3031 hat auch diesen Zustand berücksichtigt und beschreibt die notwendigen Festlegungen für das Interworking.

Der Nachteil des Label Mergings besteht darin, dass ein Teil der Informationen verloren geht. Der LER in der unteren Darstellung von Abb. 4.32 kann beim Empfang von MPLS-Paketen mit dem Label 94 nicht mehr entscheiden, ob sie ursprünglich von dem einen oder anderen LSR stammen. Ein paketbasiertes MPLS-Netzwerk, das beispielsweise auf Ethernet oder PPP aufsetzt, ist auf diese Information nicht angewiesen. MPLS-Netzwerke auf ATM-Basis hingegen benötigen diese Informationen zur Übertragung. Kann auf diese Information nicht zurückgegriffen werden, so kann dies zum Problem des Cell Interleave führen.

In ATM-basierten Netzen werden die Datenpakete in ATM-Zellen segmentiert und übertragen. Diese Segmentierung erfolgt im LER. Es kann hierbei die Situation eintreten, dass in einem LSR abwechselnd Zellen verschiedener Pakete von zwei verschiedenen Upstream LSRs empfangen werden. Gehören die Zellen bzw. das Paket der gleichen FEC an, so werden sie beim Label Merging mit dem gleichen Outgoing Label weitergeleitet. Dabei empfängt der nachfolgende LSR Zellen zweier Datenpakete in unbestimmter Reihenfolge, was als Cell Interleave bezeichnet wird. Die Zellen können nicht mehr einem Paket zugeordnet werden. Das Zusammensetzen der ursprünglichen Pakete im LER ist ohne weitere Mechanismen nicht mehr möglich.
Um Label Merging in ATM-basierten MPLS-Netzwerken zu betreiben, werden folgende Verfahren benötigt:

- **VC-Merge:** Dies basiert auf der Speicherung aller Zellen eines Pakets vor dem Weiterleiten. Erst wenn die letzte Zelle eines Pakets anhand des AAL-Typ-5-Trailers erkannt wurde, werden die Zellen zum Next-Hop gesendet. Auf diese Weise wird sichergestellt, dass alle empfangenen Zellen einem Paket angehören.
- **VP-Merge:** Dies ermöglicht einem LSR die eindeutige Paketzuordnung jeder empfangenen Zelle, trotz des angewandten Label Merging. Dies wird über die Festlegungen des Label-Transports realisiert.

82 Merging und Non-Merging

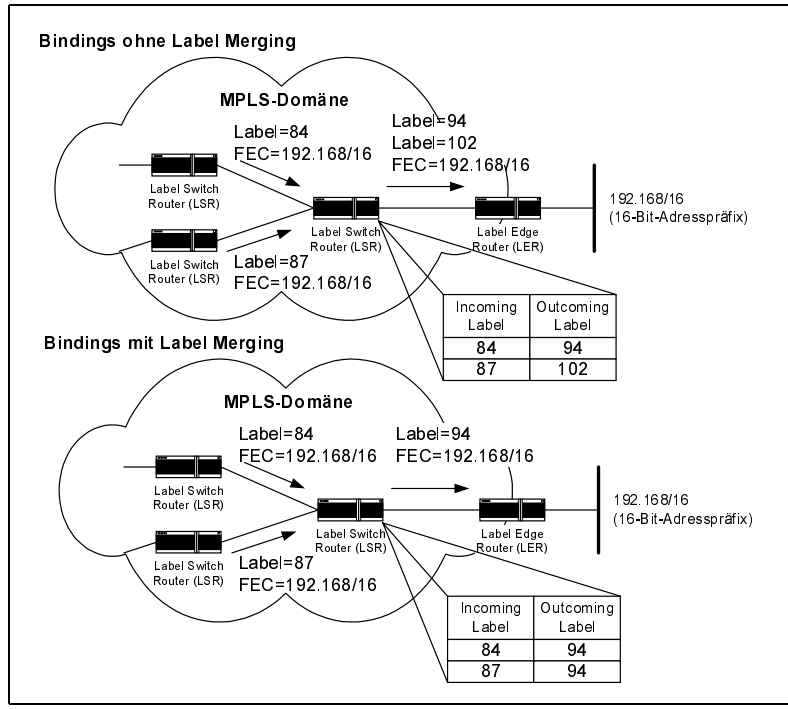


Abb. 4.32 Bindings mit und ohne Label Merging

Beide Verfahren haben ihre Vorteile. VC-Merge benötigt keine Modifikation der Labelzuweisung. VP-Merge weist eine höhere Kompatibilität mit bestehenden ATM-Switches auf, die im MPLS in abgewandelter Form als LSR eingesetzt werden können. Diese benötigen im VP-Merge keine zusätzlichen Speicher. Des Weiteren werden Verzögerungen durch die Zwischenspeicherung vermieden. [RVC01]

Label Distribution MPLS unterstützt ausschließlich das Downstream Binding. Dabei stehen für die Label Distribution grundsätzlich zwei Optionen zur Verfügung:

- **Downstream-on-Demand:** Hier sendet ein LSR nur dann Binding-Informationen, nachdem er eine explizite Label-Request-Nachricht empfangen hat.
- **Unsolicited Downstream:** Es wird den LSR erlaubt, sofort nach Zuordnung eines Labels die entsprechenden Binding-Informationen zu senden.

Da bei dem Unsolicited Downstream keine Request-Nachricht eines bestimmten LSR vorausgeht, werden die Binding-Informationen an alle benachbarten LSRs vermittelt. Dabei kann die Situation eintreten, dass ein LSR Binding-Informationen von einem Downstream LSR empfängt, welcher nicht den Next-Hop für diesen LSR darstellt. In einem solchen Fall werden die Binding-

Informationen vom LSR nicht gebraucht und können je nach Label Retention Mode entweder verworfen[83] oder gespeichert[84] werden.

Der Conservative Label Retention Mode (CLRM) reduziert die Anzahl der in den LSRs zu verwaltenden Labels. Der Liberal Label Retention Mode (LLRM) ermöglicht eine schnellere Anpassung an Netzwerkänderungen. Fällt ein Netzelement oder eine Verbindung aus, so müssen die Next-Hops betroffener LSRs neu bestimmt werden. Da alle Bindings gespeichert wurden, stehen diese ohne Verzögerungen zur Verfügung. Die Wahl des Label Retention Mode ist eine Implementierungsoption. Auch wenn einige LSR beide Verfahren unterstützen, so müssen sie sich dennoch auf einen Label Retention Mode festlegen.

Ein wesentlicher Teil der Label Distribution besteht in dem für den Transport der Label-Binding-Informationen verwendeten Protokolle. Es wurde bereits erwähnt, dass der Transport über ein eigenes Label-Distribution-Protokoll oder über Routing-Protokolle realisiert werden kann. Die MPLS Working Group arbeitet aktuell an der Möglichkeit, Label-Mapping-Informationen über BGPv4[85] zu transportieren. Hierfür soll das BGPv4 Multi-Protocol Extension Attribute, welches in RFC-2283 spezifiziert ist, verwendet werden.

Die Label-Binding-Informationen werden im NLRI[86]-Feld des Multi-Protocol Extension Attributes transportiert. Das SAFI[87]-Feld zeigt den Inhalt des NLRI an und wird für den Transport von Label-Binding-Informationen auf den Wert 4 gesetzt. Die Label-Binding-Informationen werden im NLRI-Feld in einem oder mehreren Einträgen kodiert, wobei jeder Eintrag aus mindestens drei Feldern besteht. Folgende Felder umfasst der Aufbau eines Eintrags des NRLI-Feldes: [BCKR98]

- **Length:** Dieses Feld umfasst ein Byte und gibt die Länge des bzw. der Label-Felder und des Präfix an.
- **Label:** Dieses Feld umfasst drei Byte, wobei die höherwertigen 20 Bit den Label-Wert und das niederwertigste Bit den „Bottom-of-the-Stack" des Shim-Headers anzeigen. Die restlichen drei Bit werden auf Null gesetzt. Es können abhängig von der Anzahl der Einträge im Label-Stack mehrere Label-Felder transportiert werden.
- **Prefix:** Die Länge dieses Feldes ist abhängig von dem darin transportierten Adresspräfix, dem das Label zugeordnet ist. Das Feld wird durch Padding-Bits auf das nächste Byte aufgefüllt.

83 Conservative Label Retention Mode
84 Liberal Label Retention Mode
85 Border Gateway Protocol
86 Network Layer Reachability Information
87 Subsequent Address Family Identifier

Mit dem LDP[88] wurde in RFC-3036 von der MPLS Working Group ein spezielles Protokoll definiert, um Label-Binding-Informationen zu transportieren. Das LDP unterscheidet vier Klassen von Nachrichten:

- **Discovery-Nachrichten:** werden von den LSRs verwendet, um andere LSRs im Netzwerk für die Label Distribution zu finden. Diese Nachrichten werden als UDP-Pakete übertragen.
- **Session-Nachrichten:** werden verwendet, um die LDP-Session zwischen zwei LSRs einzurichten, zu unterhalten und aufzulösen. Aus Gründen der Zuverlässigkeit findet der Transport ausschließlich über TCP innerhalb einer LDP-Session statt.
- **Advertisement-Nachrichten:** werden verwendet, um die eigentlichen Label-Binding-Informationen, welche für den Unterhalt der MPLS-Forwarding-Tabelle benötigt werden, zu übertragen. Aus Gründen der Zuverlässigkeit findet der Transport ausschließlich über TCP innerhalb einer LDP-Session statt.
- **Notification-Nachrichten:** werden benötigt, um Fehlermeldungen zu übertragen oder den LSR zu Beginn der LDP-Session zu informieren, dass die Festlegungen in der Initialization-Nachricht nicht verarbeitet werden können. Aus Gründen der Zuverlässigkeit findet der Transport ausschließlich über TCP innerhalb einer LDP-Session statt.

Der Discovery-Mechanismus für benachbarte LSRs basiert auf der Hello-Message, die über einen Well-known-UDP-Port an die „all Routers on this Subnet" Multicast-Gruppe gesendet wird. Empfängt ein LSR über diesen Port eine Hello-Message, so wird eine TCP-Verbindung für die weiteren Abläufe der Label Distribution aufgebaut. Über diesen Discovery-Mechanismus kann ein LSR über die TCP-Verbindung LDP-Sessions mit allen benachbarten LSRs unterhalten. Für die Verwendung von beispielsweise LSP-Tunnel ist ein zusätzlicher Discovery-Mechanismus notwendig, der es erlaubt, LDP-Sessions auch zwischen nicht benachbarten LSRs zu unterhalten. Hierbei wird ebenfalls eine Hello-Message über einen Well-known-UDP-Port gesendet. Der Empfänger dieser Nachricht ist aber keine Multicast-Gruppe, sondern ausschließlich der LSR, mit dem der Sender der Hello-Message Label-Binding-Informationen austauschen möchte. Hierzu benötigt der Sender die IP-Adresse des anderen LSRs, welche ihm beispielsweise über die Konfiguration zur Verfügung gestellt werden muss.

Bei der Session-Nachricht wird zu Beginn die Initialization Message der LDP-Session gesendet. Über diese Nachricht einigen sich die beteiligten LSRs über bestimmte Parameter und Optionen. Hierzu gehören beispielsweise der zu

88 Label Distribution Protocol: Im Folgenden ist mit LDP ein spezielles Protokoll gemeint, welches Label-Binding-Informationen transportiert; es darf nicht mit dem allgemeinen Ausdruck, der für ein beliebiges Transportprotokoll verwendet wird, verwechselt werden.

verwendende Label Retention Mode, der Zeitwert für Keep Alive Messages oder der für die Verbindung zur Verfügung stehende Label-Range. Der Label-Range ist beispielsweise davon abhängig, ob der LSR je eine MPLS-Forwarding-Tabelle pro Schnittstelle führt oder ob sich alle Ports eine Forwarding-Tabelle und die damit verbundenen Labels teilen. Ist der Empfänger der Initialization Message mit den Festlegungen für die Verbindung einverstanden, so bestätigt er über eine Keep Alive Message. Diese Nachrichten werden in bestimmten Zeitabständen gesendet, um die LSR über den Status der Verbindung zu informieren. Bei Empfangen einer Keep Alive Message oder einer anderen LDP Message wird ein Timer auf einen bestimmten Wert zurückgesetzt. Werden keine Nachrichten empfangen, da beispielsweise die Verbindung unterbrochen ist, läuft der Timer ab und die LDP-Session wird beendet.

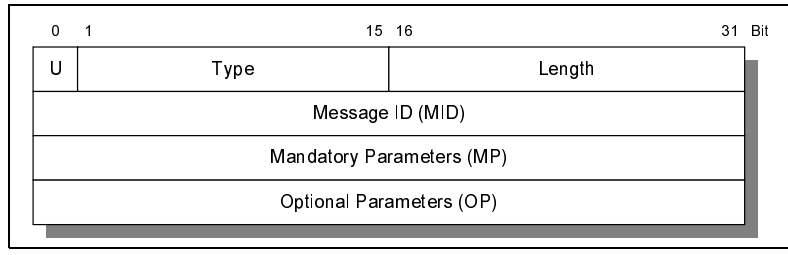
Advertisement-Nachrichten stellen die dritte Klasse von LDP-Nachrichten dar. Ein Teil der Advertisement-Nachrichten ist die Label Mapping Message. Diese Nachricht wird verwendet, um andere LSR über das eigentliche Binding von Label und FEC zu informieren. Sie stellt den Kern der Label Distribution dar. Weitere Nachrichtentypen dieser Nachrichtenklasse sind beispielsweise die Request-Nachricht für das Downstream-on-Demand-Binding oder die Withdrawal-Nachricht, um Bindings wieder aufzulösen. Alle LDP-Nachrichten verfügen über ein einheitliches Format, welches in Abb. 4.33 dargestellt wird und folgende Felder beinhaltet:

- **Unknown Message Bit (U):** Ist dieses Bit bei Empfangen einer unbekannten Nachricht gesetzt, so wird die Nachricht ignoriert. Ist das Bit gelöscht, so muss eine Notification-Nachricht an den Sender der unbekannten Nachricht gesendet werden.
- **Type:** Dieses Feld umfasst 15 Bit und kennzeichnet den Typ der Nachricht. Über das Type-Feld wird beispielsweise zwischen Hello-, Initialization- oder Label-Mapping-Nachrichten unterschieden.
- **Length:** Dieses Feld umfasst zwei Byte und gibt die Länge der Felder Message ID, Mandatory Parameters und Optional Parameters in Byte an.
- **Message ID (MID):** Dieses Feld umfasst vier Byte und wird für die Zuordnung von Notification-Nachrichten verwendet. Eine Notification-Nachricht beinhaltet die MID der Nachricht, auf die sie sich bezieht.
- **Mandatory Parameters (MP):** Dieses Feld hat eine variable Länge und transportiert, je nach Nachrichtentyp erforderliche, Parameter.
- **Operational Parameters (OP):** Dieses Feld hat ebenfalls eine variable Länge und transportiert je nach Nachrichtentyp zusätzliche Parameter.

Die LDP-Nachrichten verwenden für die zu transportierende Information im MP- bzw. im OP-Feld eine Kodierung namens Type Length Value (TLV). Abb. 4.34 zeigt den Aufbau einer TLV für LDP-Nachrichten, welche die folgenden Felder enthält:

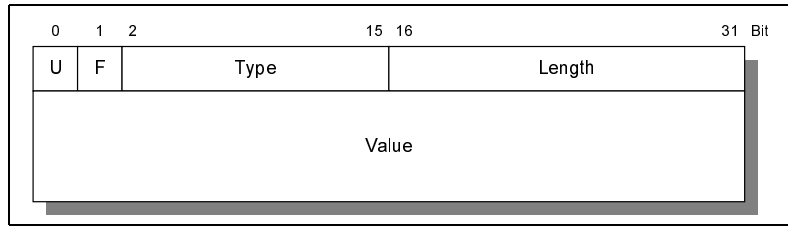
- **Unknown TLV Bit:** Ist dieses Bit bei Empfangen einer unbekannten TLV gesetzt, so wird die TLV ignoriert. Ist das Bit gelöscht, so muss eine Notification-Nachricht an den Sender der unbekannten TLV gesendet werden.
- **Forward Unknown TLV Bit:** Ist dieses Bit gesetzt, so wird die unbekannte TLV weitergeleitet. Ist das Bit gelöscht, so erfolgt keine Weitervermittlung des TLV-Typs. Dieses Feld umfasst 14 Bit und bestimmt den Inhalt des Value-Felds.
- **Length:** Dieses Feld umfasst zwei Byte und gibt die Länge des Value-Felds in Byte an.
- **Value:** Dieses Feld hat eine variable Länge und transportiert die eigentlichen Informationen.

Abb. 4.33 LDP-Nachrichtenformat



In der Spezifikation RFC-3036 sind ein Reihe von TLVs definiert, beispielsweise Prefix FEC, Generic Label, ATM Label oder Hop-Count TLV. Durch Definition neuer TLVs kann dabei die Funktionalität des LDP bei Bedarf erweitert werden. [ADFF+01]

Abb. 4.34 TLV-Format



4.4.5 QoS-Funktionen

Die bisherigen Betrachtungen von MPLS verdeutlichen die Grundfunktionen. Die Darstellung der MPLS-Architektur fokussierte den Datentransport über Hop-by-hop-routed LSPs nach dem Best-effort-Prinzip. Dieses Prinzip erlaubt für den Transport von Paketen keine Einteilung in verschieden Qualitätsklassen. Hop-by-hop-routed LSP werden aufgebaut, indem jeder LSR seinen Next-Hop eigenständig und anhand von Routing-Informationen bestimmt. Die IP-Routing-Protokolle, welche den LSR die benötigten Routing-Informationen

bereitstellen, basieren auf der Optimierung einer einzelnen Metrik, wie beispielsweise der Minimierung der Hop-Anzahl. Die Route, der die MPLS-Pakete in einem LSP folgen, entspricht der gleichen Route, der sie bei konventionellem Routing folgen würden. Der wesentliche Unterschied besteht darin, dass die Festlegung der Route im MPLS nicht pro Paket, sondern pro LSP getroffen wird. Die MPLS Working Group arbeitet an der Möglichkeit, die vorgestellten IP-QoS[89]-Modelle IntServ und DiffServ im MPLS zu unterstützen. Allerdings ist dies zu diesem Zeitpunkt noch nicht standardisiert.

IntServ

Die Unterstützung des IntServ-Modells soll im MPLS durch eine Modifikation des Protokolls RSVP realisiert werden. Aus diesem Grund entwickelte die MPLS Working Group das Protokoll Resource Reservation Protocol – Traffic Engineering (RSVP-TE). Das RSVP-TE kann durch seine MPLS-spezifischen Erweiterungen im Gegensatz zum herkömmlichen RSVP die Funktion des Label Distribution Protocol (LDP) übernehmen. Die Erweiterungen bestehen in der Definition neuer RSVP-Objekte. Für die Unterstützung des IntServ-Modells soll hierbei das neu definierte Label Object verwendet werden. Das Label Object wird mit der RESV-Nachricht transportiert und ermöglicht es, dem Flow, für den die Ressourcenzuweisung vorgenommen werden soll, ein Label zuzuweisen. Anhand des Labels können die LSR den entsprechenden Flow erkennen und über die reservierten Ressourcen weiterleiten. Dieser Flow stellt in gewisser Weise eine neue FEC dar. Es ergibt sich aus dem Reservierungsablauf im RSVP, dass im MPLS Ordered Downstream-on-Demand betrieben werden muss.

Der Vorgang der Ressourcenreservierung über eine MPLS-Domäne und das RSVP-TE beginnt ebenfalls mit der PATH-Nachricht, die vom Sender transparent über die MPLS-Domäne zum Ziel gesendet wird. Der Empfänger antwortet mit der RESV-Nachricht, welche eine Liste der zu reservierenden Ressourcen beinhaltet. Empfängt der Egress der MPLS-Domäne die RESV-Nachricht, so wählt er ein Label aus seinem Pool freier Labels aus und generiert einen Eintrag in seine Forwarding-Tabelle. Dieser Eintrag beinhaltet zusätzlich zu den üblichen Informationen auch die Angaben zu den reservierten Ressourcen. Anschließend wird das Label innerhalb des Label Object, mit der RESV-Nachricht zum Upstream-LSR weitervermittelt, der einen eigenen Eintrag generiert und den Aufbau des LSP fortsetzt. Auf diese Weise kann ein Pfad über die MPLS-Domäne vom Sender zum Empfänger aufgebaut werden, der über bestimmte Ressourcen verfügt und einen Soft-State-QoS garantiert. Abb. 4.35 gibt die Ressourcenreservierung mittels RSVP-TE wieder.

89 Quality-of-Service

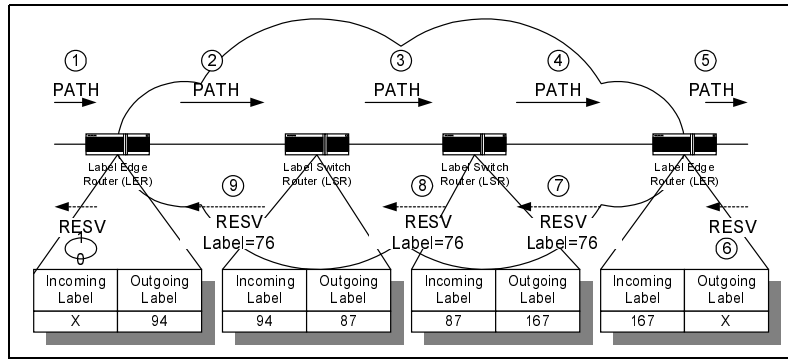


Abb. 4.35 RSVP-TE-Signalisierung

DiffServ Das DiffServ-Modell wurde bereits vorgestellt und basiert im Gegensatz zum IntServ-Modell auf der Einführung bestimmter Qualitätsklassen. Jede dieser Klassen definiert für die zugehörigen Pakete ein Per Hop Behavior (PHB). Die Abbildung von Paketen in Qualitätsklassen wird in IP-Netzen über das Feld Differentiated Services Codepoint (DSCP) im modifizierten IP-Header realisiert. Im MPLS werden Schicht-3-Header in den LSR nicht ausgewertet, sodass ein anderes Verfahren für die Bestimmung des PHB eines Pakets gefunden werden muss. Momentan werden zwei Verfahren von der MPLS Working Group untersucht, um ein MPLS-Paket einer Klasse zuzuordnen und in den LSRs entsprechend zu verarbeiten:

- **Shim-Header (Exp-Bits):** Das erste Verfahren verwendet die Exp-Bits des Shim-Headers, um Pakete in Klassen aufzuteilen, und setzt aus diesem Grund die Generic Encapsulation voraus. Die drei Exp-Bits erlauben die Kodierung von acht Qualitätsklassen. Sind diese Qualitätsklassen ausreichend, so kann das DiffServ-Modell ohne Modifikationen des MPLS-Verfahrens verwendet werden. Die LSPs, die das Verfahren über die Exp-Bits unterstützen, werden als E-LSP bezeichnet.
- **Labelwert:** Das zweite Verfahren verwendet den Labelwert, um MPLS-Pakete in eine bestimmte Qualitätsklasse einzuordnen. Auf diese Weise ermöglicht es im Gegensatz zum E-LSP auch eine Unterstützung des DiffServ-Modells in ATM-basierten MPLS-Netzen. Zusätzlich findet es Anwendung, wenn in einer MPLS-Domäne mehr als acht Qualitätsklassen gefordert sind, was über das erste Verfahren nicht zu realisieren ist. Die LSPs, die das Verfahren über den Labelwert unterstützen, werden als L-LSPs bezeichnet. Für die Realisierung von L-LSPs sind Modifikationen in der Label Distribution erforderlich, da zusätzliche Informationen mit dem Labelwert assoziiert werden müssen. Diese Modifikationen bestehen in der Erweiterung des verwendeten LDP. Das LDP muss für den Aufbau von L-LSP in der Lage sein, dem Label nicht nur eine FEC, sondern auch ein PHB zuzuweisen.

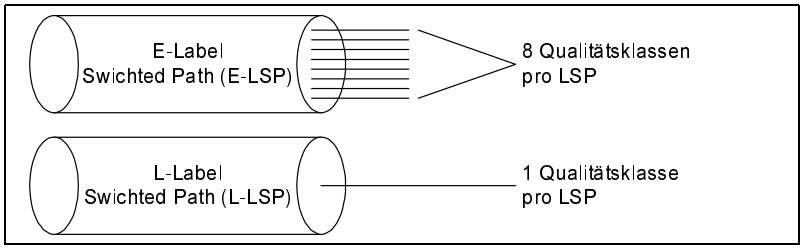


Abb. 4.36 Gegenüberstellung von E- und L-LSP

Die Vorteile des E-LSP liegen in der Reduzierung der Anzahl der benötigten LSPs. Über einen LSP können bis zu acht Qualitätsklassen realisiert werden. Des Weiteren fordert dieses Verfahren zur Unterstützung des DiffServ-Modells keine Modifikationen in der Label Distribution. Die Vorteile des L-LSP liegen in der Unterstützung von mehr als nur acht Qualitätsklassen. Da für jedes PHB ein eigener LSP aufgebaut werden muss, besteht zusätzlich die Möglichkeit, für verschiedene PHB, verschiedene Routen zum Ziel zu wählen. Ein LSP für ein zeitkritisches PHB könnte beispielsweise über Low Delay Links aufgebaut werden. Die Anwendungen, die durch die Unterstützung der IP-QoS-Modelle ermöglicht werden, liegen im optimierten Transport von Daten mit besonderen Anforderungen. Die Implementierungsmöglichkeiten werden noch behandelt. [FROMM01]

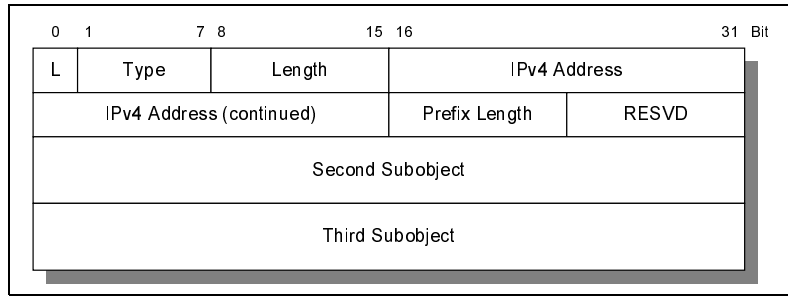
4.4.6 RSVP-TE

Um die MPLS-Funktionen in RSVP zu integrieren, wurden die Nachrichtentypen PATH und RESV, wie bereits erwähnt, um zwei neue Objekttypen ergänzt. Um ein Label anzufragen, werden Objekte der Klasse Label Request Object in eine PATH-Nachricht eingebettet und in Richtung des Egress-Router gesendet. Das Gegenstück ist ist die Objektklasse Label Object. Fügt der Egress-Router der RESV-Nachricht ein solches Objekt hinzu, wird von einer erfolgreichen Vergabe aller zu passierenden Netzknoten ausgegangen. Der Ingress-Router am Eingang des MPLS-Netzes versieht den Datenstrom mit dem entsprechenden Label und überträgt die IP-Pakete. Fehler in der Kommunikation signalisieren die Router mit den aus der Spezifikation RFC-2205 bekannten PATH- und RESV-ERR-Nachrichten.

Für das Traffic Engineering (TE) von RSVP sind die zwei Objekte Explicite Route und Record Route hinzugekommen. Mit ihnen lassen sich Pfade im Netz ermitteln und etablieren. Sie bilden die Basis für das so genannte Smooth Rerouting und ermöglichen die Anweisung „Make before break“, wenn die reservierte Bandbreite erhöht wird. In Überlastsituationen etabliert das System einen neuen Pfad, auf den es umschalten kann, ohne existierende Reservierungen aufzuheben.

Zur Unterstützung des TE soll das RSVP-TE das Explicit Route Object (ERO) verwenden, welches derzeit nur als Draft [ABBB+01b] spezifiziert wird. Das ERO enthält die Adressen der zu durchlaufenden LSRs und stellt aus diesem Grund eine wesentliche Komponente des Explicit Routings dar. Das ERO besteht aus einer Reihe von Subobjekten, deren Aufbau sich nach dem Netzwerkprotokoll richtet. Mit Hilfe von expliziten Routen soll sich die Lastverteilung im Netz steuern lassen. Abb. 4.37 zeigt den Aufbau eines IPv4-Subobjekts im ERO.

Abb. 4.37
IPv4-Subobjekt im ERO



Dabei haben die Felder die folgende Bedeutung:

- **L-Feld:** Das L-Bit definiert für das Subobjekt für den Wert 1 einen Loose-Hop und für den Wert 0 einen Strict-Hop.
- **Type:** Dieses Feld umfasst 7 Bit und ist für IPv4-Subobjekte auf den Wert 0x01 gesetzt.
- **Length:** Dieses Feld umfasst zwei Byte und gibt die Gesamtlänge des Subobjekts in Byte an. Das Length-Feld trägt immer den Wert 8.
- **IPv4 Address:** Dieses Feld umfasst vier Byte und enthält eine IPv4 Adresse. Bits, die außerhalb der Präfix-Länge liegen, werden ignoriert.
- **Prefix Length:** Dieses Feld umfasst ein Byte und bestimmt den Präfix der IPv4-Adresse.
- **RESVD:** Dieses Feld umfasst ein Byte und wird im Empfänger ignoriert. Es trägt den Wert 0.

Das ERO wird mit der PATH-Nachricht, die einem Label-Request entspricht, transportiert und bestimmt die Route, der die PATH-Nachricht folgt. Der Ingress-LER generiert diese PATH-Nachricht mit einem ERO, welches den Pfad in Form einer Liste von so genannten Subobjekten enthält. Die Subobjekte geben die zu passierenden Router mit der jeweiligen IP-Adresse an. Durch den Pfad der PATH-Nachricht wird der Pfad der RESV-Nachricht festgelegt. Da über die RESV-Nachricht der LSP aufgebaut wird, orientiert sich auch der LSP am Pfad der PATH-Nachricht und damit verbunden am ERO. Auf diese Weise sollen Explicitly-routed LSP realisiert werden. [ABBB+01b]

Die Record Route Objects (RRO) vermeiden zum einen Routing-Schleifen, zum anderen unterstützen sie die Wegefindung durch ein MPLS-Netz. Bei einem RRO handelt es sich somit vereinfacht gesagt um eine Liste von IP-Adressen, die den Pfad zwischen Routern beschreibt. Ist dieses Objekt in einer PATH- oder RESV-Nachricht vorhanden, vergleicht der empfangene Router alle Einträge mit der eigenen IP-Adresse. Findet er seine Adresse in der Liste, bedeutet dies, dass er die Nachricht schon einmal erhalten hat. In diesem Fall generiert er einen PATH-ERR mit dem Fehlercode „Routing Problem“ und dem Fehlerwert „Loop Detected“.

Für die Priorisierung von Datenströmen führt die TE-Erweiterung von RSVP das Objekt „Session Attribute“ ein, welches zwei Felder zur Angabe der Setup- und Holding-Priorität zur Verfügung stellt. Diese Werte regeln in Überlastsituationen das Aussortieren von Pfaden mit niedriger Priorität aus der Reservierung. Dadurch wären Service Provider in der Lage, ihren Kunden unterschiedliche Bandbreitengarantien anbieten zu können. Bei Engpässen verdrängen beispielsweise die höher priorisierten Datenströme bereits bestehende Verbindungen. Diese Verbindungen verlegt MPLS durch das Rerouting dynamisch auf andere physikalische Strecken. Ebenfalls wäre es möglich, die bereits genutzte Bandbreite wiederum anderen Kunden zur Verfügung zu stellen. [SCHR01]

4.4.7 Constraint-based-Routed LSP

Die MPLS-Architektur bietet die Option, LSP über das Constraint-based Routing (CR) aufzubauen. Das CR bestimmt einen Pfad im Gegensatz zum konventionellen Routing nicht nur anhand der Optimierung einer einzelnen Metrik, sondern auch über weitere Parameter. Diese Parameter werden im Ingress festgelegt und können beispielsweise eine bestimmte Bandbreitenanforderung oder eine Reihe von Verbindungen, die verwendet bzw. vermieden werden müssen, sein. Auch dieser Ansatz ist momentan nur als Draft [ABBB+01a] verfügbar und wird noch innerhalb der IETF diskutiert.

Zur Realisierung des Constraint-based Routing müssen Mechanismen bereitgestellt werden, damit der Ingress einen Pfad bestimmen kann, der alle Anforderungen erfüllt. Die Bestimmung eines Pfades muss ausschließlich und vollständig im Ingress erfolgen, da es nur so möglich ist, den Konfigurationsaufwand der Parameter auf die Ingress Router zu beschränken. Da nur der Ingress den Pfad festlegt, muss er zusätzlich in der Lage sein, den gesamten LSP über diesen Pfad aufzubauen. Sind mit der Bestimmung des Pfades Bandbreitenanforderungen verbunden, so sind zusätzliche Mechanismen zur Ressourcenreservierung notwendig, wie

- Pfadbestimmung
- LSP-Aufbau
- Ressourcenreservierung

Sie sind die Bestandteile des Constraint-based Routing. Allerdings sind sie ebenfalls noch nicht endgültig standardisiert. Die Bestimmung des Pfades eines LSP kann im einfachsten Fall über manuelle Konfiguration erfolgen. Diese Lösung eignet sich aber nur bei einer kleinen und übersichtlichen Netztopologie. Eine zweite Möglichkeit bietet die Offline Computation. Die Berechnung des Pfades wird hierbei durch ein geeignetes Offline-Tool durchgeführt und stützt sich auf die Kenntnis der gesamten Netztopologie. Die Berechnungen des Offline-Tools müssen anschließend in den Ingress geladen werden. Eine weitere Form der Pfadbestimmung besteht in der Verwendung des Algorithmus Constraint Shortest Path First (CSPF). Es handelt sich hierbei um einen modifizierten Algorithmus Shortest Path First, der bei der Pfadbestimmung auch die zusätzlichen Parameter berücksichtigen kann. Eine detaillierte Darstellung der Modifikationen kann in [DARE00] nachvollzogen werden.

Für den Aufbau der LSP über die vom Ingress festgelegten Hops besteht die Voraussetzung zur Unterstützung des Explicit Routing[90]. Beim Explicit Routing wird die Folge der LSR teilweise[91] oder vollständig[92] in einem einzigen LSR festgelegt. Üblicherweise erfolgt die Bestimmung der Route im Ingress oder Egress. Liegt die Anwendung des Explicit Routing in der Unterstützung des Constraint-based Routing, so muss der Pfad vollständig im Ingress festgelegt werden, da nur dieser Router die Parameter kennt. Für die Realisierung des Explicit Routing muss im MPLS Ordered Downstream-on-Demand betrieben werden. Das Explicit Routing erfordert Modifikationen im verwendeten LDP.

Es verwendet die gleichen LDP-Nachrichten, definiert aber zusätzlich eine Reihe neuer Objekte Type Length Value (TLV), die dem geänderten Aufgabenbereich des CR-LDP entsprechen. Zu den neuen TLV gehört unter anderem die Explicit Route TLV (ER-TLV). Die ER-TLV beinhaltet analog zum ERO im RSVP-TE die Adressen der zu durchlaufenden LSR. Jede Adresse ist in einer Explicit Route Hop TLV (ER-Hop TLV) eingefügt, welche die Komponenten des ER-TLVs bilden. Die ER-HOP TLV entspricht einem Subeintrag im ERO des RSVP-TE. Abb. 4.38 zeigt den Aufbau der ER-Hop TLV im ER TLV.

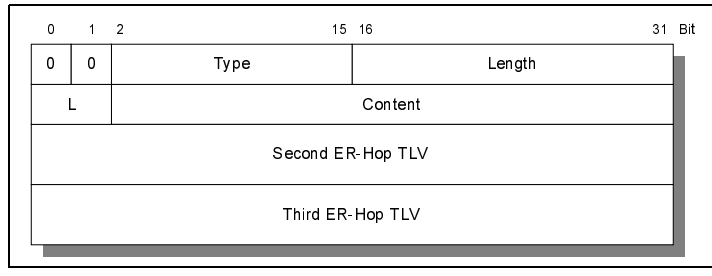


Abb. 4.38 Explicit Route Hop TLV im ER-TLV

90 Wird auch als Source Routing bezeichnet.
91 Loosely Explicitly Routed
92 Strictly Explicitly Routed

Dabei haben die Felder die folgende Bedeutung:

- **Type:** Dieses Feld umfasst 6 Bits und ist für einen IPv4-Präfix auf den Wert 0x0801 gesetzt.
- **Length:** Dieses Feld umfasst zwei Byte und gibt die Gesamtlänge des L-und Content-Feldes in Byte an.
- **L-Feld:** Das L-Bit definiert für die Explicit Route Hop TLV für den Wert 1 einen Loose-Hop und für den Wert 0 einen Strict-Hop.
- **Content:** Dieses Feld hat eine variable Länge, die durch das Lentgh-Feld bestimmt wird und enthält die Adresse eines einzelnen oder auch mehrerer LSRs.

Die Explicit Route TLV wird der Label-Request-Nachricht angefügt und bestimmt entsprechend dem ERO im RSVP-TE den Pfad, dem diese Nachricht und letztendlich auch der LSP folgt.

Wird eine Ressourcenreservierung in Verbindung mit dem Constraint-based Routing gefordert, so muss diese durch das verwendete LDP durchgeführt werden. Während das RSVP-TE die Ressourcenreservierung im Rahmen der Unterstützung des IntServ-Modells anbietet, wird bei CR-LDP eine erneute Erweiterung des Protokolls notwendig. Dies soll durch die Definition der Traffic Parameters TLV in [ABBB+01a] realisiert werden. Abb. 4.39 zeigt den Aufbau der Traffic Parameters TLV.

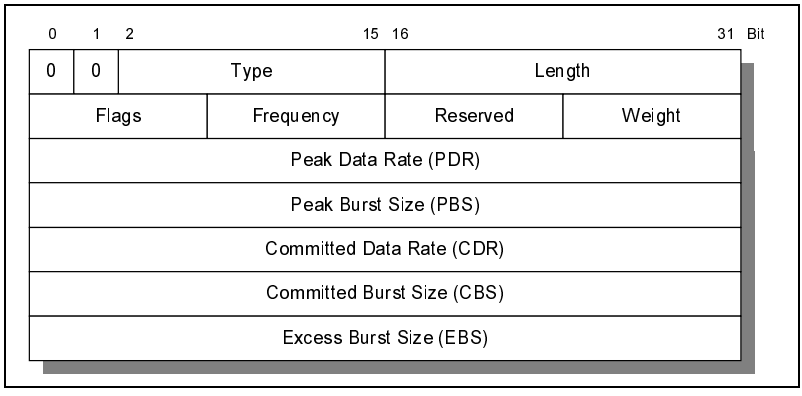


Abb. 4.39
Traffic Parameters TLV

Dabei haben die Felder die folgende Bedeutung:

- **Type:** Dieses Feld umfasst 14 Bit und ist für die Traffic Parameters TLV auf den Wert 0x0810 gesetzt.
- **Length:** Dieses Feld umfasst zwei Byte und gibt die Gesamtlänge der folgenden Felder in Byte an. Dieses Feld trägt immer den Wert 24.
- **Flags:** Das Feld Flags beinhaltet ein Byte und kennzeichnet, welche Parameter ausgehandelt werden können.

- **Frequency:** Dieses Feld beinhaltet ebenfalls ein Byte und definiert die Granularität der CDR.
- **Reserved:** Dieses Feld umfasst ein Byte und ist für spätere Anwendungen vorgesehen.
- **Weight:** Dieses Feld ist ein Byte groß und bestimmt die Verteilung der eventuell über der Commited Data Rate liegenden Bandbreite.

Die Traffic Parameter enthalten jeweils vier Byte und tragen einen entsprechenden Wert. Die Hauptanwendung des Constraint-based Routing liegt im TE. Das TE ermöglicht eine effektive Netzauslastung, indem es gezielt überlastete Verbindungen durch gezielte Einrichtung von Datenströmen entlastet. Die IP-QoS-Modelle bieten einen QoS, indem sie Ressourcen für bestimmte Flows im IntServ bzw. für bestimmte Qualitätsklassen im DiffServ reservieren. Der Pfad, auf dem diese Reservierungen vorgenommen werden, basiert auf konventionellem IP-Routing und wird aus diesem Grund völlig losgelöst von QoS-Aspekten bestimmt. Ist ein Pfad überlastet, so kann ein QoS nicht mehr realisiert werden. Durch das Traffic Engineering bzw. das Constraint-based Routing nimmt das Routing eine aktive Rolle in der Realisierung des QoS ein, da die Routing-Entscheidungen auch auf bestimmten QoS-Parametern basieren. Durch die Wahl alternativer Pfade kann ein QoS effektiver in einem Netz implementiert werden. [ABBB+01a]

4.5 Vergleich der Layer-2-Verfahren

In diesem Kapiteln wurde Traffic Engineering (TE) hervorgehoben und unterschiedliche Verfahren vorgestellt, die auf der einen Seite eine Anpassung von IP auf unterschiedliche Netzwerkinfrastrukturen vornehmen und auf der anderen Seite Netzmechanismen einführen, um IP mehr Intelligenz im Kernnetz zu verleihen. Dabei wurden als zentrale Layer-2-Technologien ATM und PoS hervorgehoben.

Mit MPLS ist der Ansatz gegeben, die vorhandenen Label-Switching-Verfahren miteinander zu kombinieren. Dabei wird aber hauptsächlich Tag-Switching und ARIS einbezogen, da Cisco und IBM die größte Einflussnahme auf die Standardisierung haben. Hinzu kommen aber auch Ipsilon IP-Switching und Toshibas Cell Switch Router (CSR). MPLS ist für Weitverkehrsnetze geeignet und kann über Router mit MPOA ergänzt werden. Da einem IP-Paket am Netzrand ein Label zugeordnet wird, können spezifizierte CoS-/QoS-Parameter vergeben werden. Eine garantierte Dienstgüte ist somit möglich. Allerdings leidet MPLS darunter, dass man die reservierten Ressourcen, die für die Wege genutzt werden, nur selten alle nutzen kann. Hinzu kommt, dass MPLS bereits seit drei Jahren sehr kontrovers diskutiert wird und daher eine ähnliche Entwicklungszeit wie MPOA durchlaufen wird. Da sich momentan aber eine Reihe

von Herstellern MPLS verschrieben haben, kann man davon ausgehen, dass es auch abschließend standardisiert wird und zum Einsatz kommt. Erste Piloten sind bereits vorhanden und erfolgreich getestet worden.

MPLS wird momentan in zwei verschiedenen Ansätzen stark diskutiert und weiterentwickelt:

1. **Traffic Engineering (TE):** Die Grundidee von MPLS basiert auf festlegbaren Routen, die durch das Internet hindurchgeschaltet werden, um Verzögerungen reduzieren oder gar garantieren zu können. Mittels ATM kann MPLS dies erreichen, da virtuelle Pfade durch das IP-Netz geschaltet werden. MPLS wird heute allerdings als eigenständige Protokollschicht definiert, die unabhängig von Layer-2-Protokollen Aufgaben übernimmt, die ursprünglich ATM vorbehalten waren. Zur Qualitätsgarantie wird daher der Ansatz IntServ oder DiffServ innerhalb von IP-Netzen eingesetzt.
2. **Virtual Private Network (VPN):** Der Aufbau eines VPN kann ebenfalls sehr effizient mittels MPLS durchgeführt werden. Mit MPLS lassen sich Tunnels zwischen den Edge-Routern der ISP schalten, die verschiedene Standorte eines VPN verbinden. Hierdurch wird eine höhere Sicherheit sowie Qualität durch festverschaltete virtuelle Verbindungen erreicht. Dieser Ansatz wird in einem späteren Kapitel noch erläutert.

Das MPLS-Verfahren mit seiner Architektur und seinen Funktionen wurde vorgestellt und eingehend behandelt. Die Darstellung seiner Funktionen alleine gibt aber keinen Aufschluss darüber, in welcher Form das MPLS die Internettechnologie optimiert. In dieser Optimierung liegt das Ziel seiner Entwicklung und das Argument für seine Implementierung. Das MPLS-Verfahren stellt ein neues Vermittlungskonzept dar und muss für seine Bewertung mit bestehenden Vermittlungskonzepten verglichen werden. Da der Hauptanwendungsbereich des MPLS in den WAN-Netzen liegt, werden für die Gegenüberstellung das ATM und das PoS herangezogen. Diese Protokolle eigenen sich für den Vergleich, da sie derzeit wichtige WAN-Lösungen darstellen. Es wird hierbei geprüft, in welcher Form die einzelnen Protokolle in der Lage sind, die Anforderungen, die an die WAN-Technologie gestellt werden, zu realisieren.

4.5.1 MPLS contra ATM

Dieser Abschnitt vergleicht ein konventionelles ATM-Netz mit einem auf ATM-Technologie aufsetzendem MPLS-Netzwerk für den IP-Transport. Zu diesem Zweck werden in einem ersten Schritt die Funktionen beider Protokolle einander im Hinblick auf die Realisierung der WAN-Anforderungen gegenübergestellt. Anschließend folgt eine Bewertung des MPLS, welche sich auf die Effizienz der betrachteten Funktionen stützt. Im Folgenden ist deshalb mit einem MPLS-Netz ein ATM-basiertes MPLS-Netz gemeint.

Anpassung/Integration von IP auf ATM

ATM wurde ursprünglich nicht in Hinblick auf den IP-Transport entwickelt. Bei ATM und IP handelt es sich sogar um zwei grundsätzlich verschiedene Protokolle, was die Anpassung bzw. Integration erschwert. Dennoch konnten mehrere Verfahren entwickelt werden, um die geforderte IP-Anpassung bei ATM zu realisieren. Beispiele für diese Verfahren bestehen in CLIP, LANE und Multi-Protocol-over-ATM (MPOA), die auch in diesem Kapitel behandelt wurden.

Das CLIP-Verfahren bestimmt den notwendigen Mechanismus, um IP-Pakete auf ATM-Zellen abzubilden und regelt die Adressauflösung. Für diese Adressabbildung muss im CLIP zwischen den beiden Verbindungstypen im ATM unterschieden werden. Hierbei handelt es sich um

- Permanent Virtual Connections (PVC)
- Switched Virtual Connections (SVC)

PVC-Verbindungen müssen manuell konfiguriert werden. Die Adressauflösung erfolgt über lokale ATMARP-Tabellen. SVC-Verbindungen werden hingegen bei Bedarf automatisch über ein ATM-Routing-Protokoll aufgebaut. Für die Abbildung der IP-Adressen auf ATM-Adressen werden zusätzliche Server benötigt. Da das CLIP-Verfahren grundsätzlich keine Multicast-Fähigkeiten bietet, muss es durch die Einführung des Multicast Address Resolution Servers (MARS) ergänzt werden. Auch das Next-Hop Resolution Protocol (NHRP) ist eine sinnvolle Ergänzung CLIP, welche die Performance erhöht, da direkte Verbindungen möglich geworden sind.

LANE ist in der Version 2.0 erst im Laufe des Jahres 1999 endgültig spezifiziert worden, wird aber ähnlich wie CLIP nur noch wenig vorangetrieben. Die Schnittstellen zwischen Client und Server (L-UNI[93]) liegen in der Version 2.0 ebenfalls vor. Dies gilt ebenso für die Schnittstelle zwischen den Servern (L-NNI[94]). L-NNI liegt in der LANE-Version 2.0 zum ersten Mal vor. Hauptziel war es, verteilte Server zu ermöglichen und die Kommunikation zwischen diesen zu verbessern. Dadurch können redundante ELANs aufgebaut werden, der BUS begrenzt das Gesamtsystem nicht weiter und unterschiedliche Hersteller sollten einsetzbar sein. Das Problem dieses Overlay-Ansatzes[95] besteht allerdings darin, dass beide Netze (IP und ATM) völlig unabhängig voneinander agieren und sich die unterschiedlichen Mechanismen (z.B. Routing) gegenseitig behindern können. Die Dienstgüte spielt hier ebenfalls keine Rolle und kann nicht verwendet werden. Es steht nur Unspecified Bit Rate (UBR) zur Verfügung.

93 LAN-Emulation User Network Interface
94 LAN-Emulation Network Network Interface
95 Anpassung – keine Integration der unterschiedlichen Funktionen

MPOA in der Version 1.1 bietet zum ersten Mal eine effiziente Möglichkeit im LAN, ATM-Vorzüge direkt auf IP zu übertragen. Dabei nimmt die Idee des virtuellen Routers konkretere Strukturen an. MPOA setzt auf LANEv2.0 auf und erweitert diesen Ansatz im Wesentlichen um NHRP und QoS. Letzteres ist allerdings nur bei direkter ATM-Verbindung an das Kernnetz möglich. Dadurch harmonisieren IP und ATM miteinander und können sich gegenseitig ergänzen. Beispielsweise ist das dynamische Routing-Verfahren P-NNI von ATM dem dynamischen IP-Routing OSPF sehr ähnlich und kann adaptiert werden. Da MPOA immer auf ATM aufsetzt, sind langfristig Mechanismen vorhanden, die in der Lage sind, eine effiziente Integration gewährleisten zu können. Hinzugekommen sind in der Version 1.1 ebenfalls Sicherheitsmechanismen, die vorher noch offen gelassen wurden. Nachteilig ist allerdings die Unflexibilität der Layer-2-Technologie bezüglich ATM, da die heutige Entwicklung oftmals in Richtung Ethernet und SDH bzw. PoS zeigt. Weiterhin ist MPOA schlecht zu skalieren und daher nur für das LAN und MAN geeignet. Im LAN allerdings wird MPOA keine Zukunft eingeräumt, da sich hier inzwischen der simplere Ansatz Gigabit-Ethernet durchsetzen konnte. Aus diesem Grund und wegen des relativ komplexen Ansatzes haben sich die Hersteller auch weitgehend von MPOA zurückgezogen, sodass diese Lösung nicht weiter vorangetrieben wird.

MPLS ist sofort in Hinblick auf den IP-Transport entwickelt worden. Die Vermittlung im MPLS basiert unter anderem auf konventionellen IP-Routing-Protokollen. Diese werden für den Aufbau der Label Switched Path (LSP) verwendet. Der Aufbau der LSP erfolgt automatisiert. Die Abbildung der IP-Adressen auf MPLS-Labels erfolgt im MPLS in den Label Edge-Routern (LER) und wird ohne zusätzliche Server durchgeführt. Hierbei bietet das MPLS selbst auch Multicast-Fähigkeiten. Aufgrund der Unabhängigkeit gegenüber der Schicht 2 ist MPLS auch sehr flexibel für unterschiedliche Netzwerktechnologien einsetzbar. Ein Integrationseinsatz von IP und ATM ist machbar und kann effektiv durchgeführt werden.

Nettobitrate

An dieser Stelle wird der Einfluss von ATM und MPLS bzw. von PoS und MPLS im nächsten Abschnitt auf die effektive Übertragungsrate (Nettobitrate) untersucht. Dieser Einfluss besteht in dem zu transportierenden Overhead. Hierbei sei ein Beispiel bei der Darstellung des Overheads in den Übertragungsrahmen der WAN-Protokolle für den Transport eines 1500 Byte großen IP-Pakets gegeben. Der Signalisierungsaufwand sowie der Overhead in den TCP-Rahmen und den SDH-Containern bleibt unberücksichtigt, da diese Vernachlässigung[96] keinen Einfluss auf den Vergleich zwischen ATM und MPLS bzw. später zwischen PoS und MPLS hat.

96 Mit Ausnahme der Signalisierung

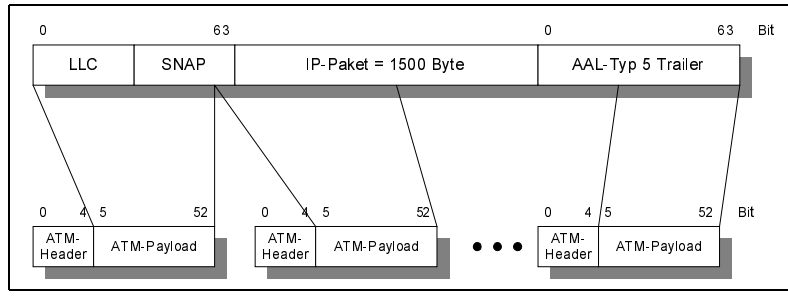


Abb. 4.40 ATM-Overhead

ATM-Netze zeichnen sich auf Grund der Verwendung von lediglich 53 Byte großen ATM-Zellen generell durch ein relativ schlechtes Overhead-Nutzdaten-Verhältnis aus. Abb. 4.40 verdeutlicht den ATM-Overheadanteil beim Transport eines 1500 Byte großen IP-Datenpakets.

Nach dem Einpacken in einen AAL-Typ-5-Rahmen wurde dem IP-Paket ein 16-Byte-Overhead angefügt. Dieser besteht im LLC-/SNAP-Header und dem AAL-Typ-5-Trailer. Für die Übertragung muss nun der gesamte Rahmen, welcher jetzt aus 1516 Byte besteht, in ATM-Zellen abgebildet werden. Eine ATM-Zelle nimmt hierbei 48 Byte über das Payload-Feld auf. Für die Übertragung von 1516 Byte werden 32 ATM-Zellen benötigt. Jede Zelle erhöht den Overhead zusätzlich um 5 Byte durch den ATM-Header. Bei 32 Zellen erhöht sich der Gesamt-Overhead auf diese Weise um weitere 160 Byte. Für den Transport eines 1500 Byte großen IP-Pakets werden somit insgesamt 176 Byte an Overhead benötigt. Hieraus ergibt sich ein relativer Overhead von 10,5%.

Der Overhead-Anteil im MPLS wird in Abb. 4.41 ebenfalls am Beispiel eines 1500-Byte-IP-Pakets dargestellt. Hierbei wird von zwei Einträgen im Label-Stack ausgegangen. Nach dem Anfügen der beiden Shim-Header und dem anschließenden Einpacken in einen AAL-Typ-5-Rahmen wurden dem IP-Paket 16 Byte Overhead angefügt. Auch hier werden für die Übertragung von 1516 Byte 32 ATM-Zellen benötigt. Der Gesamt-Overhead für den Transport eines 1500 Byte großen IP-Pakets mit zwei angefügten Shim-Headers beträgt somit ebenfalls 176 Byte, woraus sich ein identischer Overhead von 10,5% ergibt.

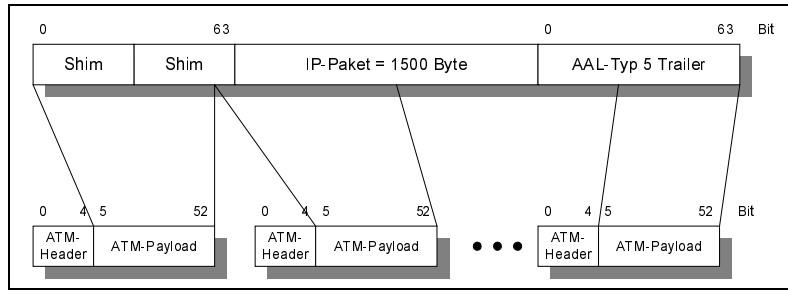


Abb. 4.41 MPLS-Overhead bei ATM-basierter Technologie

Performance

Die Unterstützung von Echtzeitanwendungen wurde bereits als eine wichtige Anforderung an die WAN-Technologie definiert. Die Realisierung einer hohen Performance, mit der eine geringe Verzögerungszeit und ein hoher Durchsatz gemeint ist, spielt hierbei eine entscheidende Rolle. Die Performance ist von verschiedenen Faktoren abhängig. Hierzu gehören beispielsweise das verwendete Übertragungsmedium, die Länge der Übertragungsstrecke und die Geschwindigkeit, mit der ein Paket in den Netzelementen verarbeitet wird. Diese Paketverarbeitung wird maßgebend durch das verwendete Forwarding-Konzept bestimmt.

ATM verwendet das ATM-Switching als Forwarding-Konzept. Die Forwarding-Entscheidung basiert auf den VCI/VPI-Feldern des ATM-Headers. Aus diesem Grund ist es für das Forwarding nicht notwendig, die ATM-Zellen zusammenzusetzen und die höheren Schichten auszuwerten. MPLS verwendet hingegen das Label-Switching als Forwarding-Konzept. Bei zugrunde liegender ATM-Technologie werden die Label-Werte in die hierfür umdefinierten VCI/VPI-Felder der ATM-Zellen geschrieben. Die Forwarding-Entscheidung basiert auf den umdefinierten VCI/VPI-Feldern des ATM-Headers. Daher ist es nicht notwendig, die ATM-Zellen wieder zusammenzusetzen, um den ebenfalls transportierten Shim-Header auszuwerten.

Traffic Engineering (TE)

Das TE erlaubt das gezielte Einrichten eines Datenstroms mit dem Ziel, das Netz gleichmäßig zu belasten. Des Weiteren besteht im TE eine wichtige Unterstützung zur Realisierung eines Quality-of-Service (QoS).

In konventionellen ATM-Netzen gibt es viele Funktionen, die ein Traffic Engineering ermöglichen. Ein wichtiges Element der TE-Fähigkeiten von ATM liegt im Private Network-to-Network Interface (P-NNI). Das P-NNI ist eine standardisierte Schnittstelle und regelt den Datenaustausch zwischen zwei ATM-Systemen. Hierfür wurde ein Protokoll festgelegt, dass den Verbindungsaufbau zwischen zwei Teilnehmern steuert. Es handelt sich um das P-NNI-Protokoll, zu dessen Aufgaben das Routing für den Verbindungsaufbau von ATM-VC gehört. Der Routing-Mechanismus des P-NNI-Protokolls gleicht dem OSPF, ist aber aufgrund zusätzlicher Funktionen komplexer. Das P-NNI-Routing ermöglicht die Definition bestimmter Verkehrsparameter, welche für die Berechnung des Pfades berücksichtigt und eingehalten werden müssen. Auf diese Weise kann die Überlastung einzelner Netzelemente vermieden werden, da das P-NNI-Protokoll in diesen Fällen einen alternativen Pfad wählt.

Um zu verhindern, dass ein Teilnehmer mehr Ressourcen in Anspruch nimmt, als er ursprünglich beim Verbindungsaufbau über das P-NNI-Routing gefordert hat, verfügt das ATM über eine Vielzahl weiterer Funktionen. Hierzu gehören beispielsweise das einfache Verwerfen von Zellen, das Herunterstufen von Prioritäten, das Traffic Shaping oder das Buffer Management in den Switches. Diese Funktionen stellen ebenfalls einen Teil des TE dar.

Im MPLS wird das TE über zwei verschiedene Ansätze realisiert. Zum einen ist das LDP mittels des Constraint-based Routing auf CR-LDP erweitert worden. Dieses berechnet den Pfad, über den ein Label Switched Path (LSP) aufgebaut werden soll. Hierfür wird der CSPF-Algorithmus verwendet. Dieser Algorithmus basiert auf dem Shortest-Path-First-Algorithmus, der beispielsweise von OSPF verwendet wird. Das CSPF erlaubt zusätzlich die Definition bestimmter Parameter (z.B. Bandbreite), welche für die Berechnung des Pfades berücksichtigt und eingehalten werden müssen. Auf der anderen Seite ist das RSVP-Protokoll auf Basis von RSVP-TE erweitert worden. Hierdurch lassen sich über den IntServ-Ansatz MPLS-Verbindungen bestimmte Prioritäten zuweisen, die über das gesamte MPLS-Netz erhalten bleiben. Beide Ansätze werden momentan sehr kontrovers diskutiert. Funktionen von MPLS, die sicherstellen, dass kein Teilnehmer mehr als die geforderten und reservierten Ressourcen in Anspruch nimmt, sind daher bislang noch nicht spezifiziert worden.

Quality-of-Service (QoS)

Die Realisierung von QoS ist nur dann sinnvoll, wenn sie auf der gesamten Übertragungsstrecke von der Quelle bis zur Senke[97] besteht. Die Anforderungen, die in diesem

Zusammenhang an die WAN-Technologien gestellt werden, beziehen sich aus diesem Grund nicht auf die Realisierung eines isolierten QoS in einem WAN, sondern auf die Unterstützung von bereits im LAN verwendeten QoS-Lösungen.

ATM bietet umfangreiche Möglichkeiten zur Realisierung von QoS. Um IP-Netze über ein ATM-Netz zu verbinden, ist es notwendig, den IP-QoS auf den ATM-QoS abzubilden. Nur so ist es möglich, einen QoS auf der gesamten Übertragungsstrecke zu realisieren. Das CLIP-Verfahren ist ausschließlich für Adressauflösung konzipiert worden und kann für diese Aufgabe nicht verwendet werden. Ebenso fällt LANE aus, da hier nur Anpassungen von Ethernet und Token Ring an ATM über den Overlay-Ansatz möglich sind.

Aus diesem Grund müssen zusätzliche Mechanismen eingesetzt werden, um die Abbildung des QoS durchzuführen. Bei entsprechender Router-Unterstützung ist es möglich, diese Abbildung über eine manuelle Konfiguration zu realisieren. Die Verwendung eines automatisierten Verfahrens ist ebenfalls denkbar. MPOA bietet diese Möglichkeit eines Ende-zu-Ende-QoS, wenn als Layer-2-Technologie ATM gewählt wurde und eine direkte Verbindung zum ATM-Netz besteht. Bei Einsatz von Ethernet- oder Token-Ring-Netzen am Netzwerkrand ist ebenfalls nur eine Anpassung ohne QoS machbar.

Im Vergleich zu ATM kann MPLS einen QoS Ende-zu-Ende durch die Unterstützung der IP-QoS-Modelle IntServ und DiffServ ermöglichen. Für das IntServ-Konzept ist die MPLS-Domäne transparent. Die Realisierung des QoS

97 Ende-zu-Ende

über das Signalisierungsprotokoll[98] erfolgt identisch zu anderen IP-Netzen. Wird das DiffServ-Konzept verwendet, so verhält sich die MPLS-Domäne ebenfalls wie ein IP-Netz. Die MPLS-Domäne bildet hierbei eine eigene DiffServ-Domäne, welche mit den anliegenden DiffServ-Domänen kooperiert. Beide Ansätze bzw. Modelle sind aber noch nicht abschließend standardisiert.

Bewertung

Die IP-Integration ist über das MPLS einfacher zu realisieren, als es über das ATM in Verbindung mit dem MPOA-Verfahren möglich ist. Werden im ATM PVC-Verbindungen verwendet, so erfordert dies einen hohen Konfigurationsaufwand. Insbesondere für große Netze besteht aus diesem Grund in der Verwendung von SVC-Verbindungen eine interessante Alternative, welche den Konfigurationsaufwand entscheidend reduzieren kann. Allerdings müssen hierbei zusätzliche Server für die Adressauflösung eingesetzt werden. Diese Server stellen einen Single-Point-of-Failure (SPOF) dar und müssen entsprechend ausgelegt werden. Dies ist mit weiteren Schwierigkeiten verbunden, da bei der Einführung von Ersatz-Servern beispielsweise die Synchronisation der Tabellen der einzelnen Server gewährleistet werden muss.

MPLS hingegen bietet den Vorteil eines geringen Konfigurationsaufwandes, da die LSPs unter anderem über die Verwendung von IP-Routing-Protokollen automatisch aufgebaut werden. Des Weiteren verwendet das MPLS keine Adressserver, wodurch die damit verbundenen Probleme vermieden werden. Der zu transportierende Overhead ist im MPLS im hier betrachteten Beispiel trotz Anfügen der beiden Shim-Header nicht höher als im ATM. Auch das Anfügen weiterer Shim-Header würde das ohnehin relativ schlechte Overhead-Nutzdaten-Verhältnis, welches durch die ATM-Zellen zustande kommt, nur geringfügig reduzieren. Da auf der anderen Seite das Overhead-Nutzdaten-Verhältnis im MPLS beim Transport eines einzelnen Shim-Headers sogar verbessert wird, ist an dieser generell betrachtet weder von einem Vor- noch einem Nachteil zu sprechen.

Tab. 4.5 Direkter Vergleich von ATM und MPLS

	WAN-Technologien bzgl. IP-Transport	
WAN-Funktionen	ATM	MPLS
IP-Integration	Zusätzliche Verfahren notwendig (z.B. MPOA)	Vorhanden
Nettobitrate und Overhead	Overhead = 10,5%	Overhead = 10,5%
Performance und Forwarding	ATM-2-Switching	Label-2-Switching

98 Üblicherweise RSVP

Tab. 4.5 Direkter Vergleich von ATM und MPLS (Forts.)

	WAN-Technologien bzgl. IP-Transport	
Traffic Engineering	P-NNI Routing	Verschiedene Verfahren: MPLS-LDP, RSVP-TE und Constraint-based Routing (CR-LDP)
IP QoS	Zusätzliche Verfahren notwendig (IntServ, DiffServ, MPOA)	Zusätzliche Verfahren notwendig (IntServ, DiffServ)

Eine höhere Performance ist im MPLS prinzipiell nicht gegeben. Beide Protokolle verwenden das Switching als Forwarding-Konzept. Hierbei wurde für die Verwendung von SVC-Verbindungen im ATM der Einsatz des NHRP vorausgesetzt. Ohne das NHRP ist es unter Umständen notwendig, Verkehr über Router zu transportieren, in denen das Forwarding-Konzept auf dem Schicht-3-Routing basiert. Der einzige Nachteil von ATM bezüglich der Performance besteht bei der Verwendung von SVC-Verbindungen, da diese bei Bedarf erst aufgebaut werden müssen. Hierbei kann es für den Aufbau unter Umständen sogar notwendig sein, eine Adressanfrage bei einem ATMARP- oder einem NHRP-Server durchzuführen, wodurch sich der Beginn des Datentransports noch stärker verzögert. Nachdem der SVC aufgebaut wurde, ist aber mit der gleichen Performance wie im MPLS zu rechnen. Aus diesem Grund besteht im MPLS in der Performance im Vergleich zu ATM bei verwendeten SVC-Verbindungen nur ein geringfügiger Vorteil, im Vergleich zu ATM bei verwendeten PVC-Verbindungen weder ein Vor- noch ein Nachteil.

Eine Realisierung des Traffic Engineering (TE) ist im MPLS ebenso möglich wie im ATM. In beiden stellt die Wegefindung einer Verbindung die Basis des TE dar. Es wurde hier nur die automatisierte Wegefindung über Routing-Protokolle betrachtet. ATM und MPLS verwenden dabei prinzipiell das gleiche Konzept. Allerdings können zur Realisierung des TE auch ergänzende Mechanismen wie das Traffic Shaping zum Einsatz kommen, die hier nicht explizit berücksichtigt wurden.

Das Bereitstellen eines IP QoS ist im MPLS einfacher und effektiver als im ATM. Obwohl das ATM über sehr gute QoS-Fähigkeiten verfügt, besteht das Problem darin, dass es sich hierbei um einen ATM-QoS handelt. Für die Realisierung eines IP-QoS muss eine Abbildung auf den ATM-QoS erfolgen, wodurch die Komplexität des Netzwerkes erhöht wird. Wird diese Abbildung über manuelle Konfiguration durchgeführt, so besteht an dieser Stelle ein zusätzlicher Konfigurationsaufwand. Des Weiteren bestehen in einem Abbildungsvorgang zusätzliche Operationen, die von den Netzelementen durchgeführt werden müssen. Hierdurch wird die Performance beeinträchtigt. Allerdings lassen sich dadurch feste Dienstgüten einstellen, die sich durch die Verkehrslast nicht verändern. Im MPLS kann über die IP-QoS-Modelle IntServ

und DiffServ ebenfalls ein IP-QoS realisiert werden. Ein komplexer Abbildungsmechanismus ist hierbei nicht notwendig. Dies ist vorteilhaft für MPLS. Wenn allerdings ATM als Layer-2-Technologie eingesetzt wird, kommt es ebenfalls zu einer höheren Komplexität beim MPLS-Einsatz.

Zusammenfassend gesagt bestehen die Hauptvorteile von MPLS im Vergleich zum ATM in der einfacheren IP-Integration und IP-QoS-Unterstützung, wenn man reine TE-Aspekte berücksichtigt.

4.5.2 MPLS contra PoS[99]

An dieser Stelle wird ein PoS-Netz mit einem auf PoS-Technologie aufsetzendem MPLS-Netzwerk für den IP-Transport verglichen. Auch hier werden zunächst die Funktionen beider Protokolle im Hinblick auf die Realisierung der WAN-Anforderungen gegenübergestellt. Die zu betrachtenden Funktionen, wie das Traffic Engineering (TE) oder die Unterstützung des IP-QoS und der VPN, sind vom zugrunde liegenden Schicht-2-Protokoll unabhängig. Aus diesem Grund unterscheiden sich die MPLS-over-PoS-Funktionen nicht von den MPLS-over-ATM-Funktionen. Auf eine erneute Darstellung wird an dieser Stelle verzichtet, sodass für MPLS-over-PoS lediglich die Overhead-Betrachtung durchgeführt werden muss. Anschließend folgt eine Bewertung, welche sich auf die Effizienz der betrachteten Funktionen stützt. Im Folgenden ist mit einem MPLS-Netz ein PoS-basiertes MPLS-Netz gemeint.

IP-Integration

PoS realisiert die Unterstützung eines Schicht-3-Protokolls über die Verwendung des entsprechenden Network Control Protocol (NCP). Das NCP ist ein Steuerungsprotokoll, dessen Aufgabe im Auf- und Abbau sowie der Konfiguration des zu verwendenden Schicht-3-Protokolls liegt. Für den Transport von IP-Daten wurde als NCP das Internet Protocol Control Protocol (IPCP) definiert. Über das IPCP realisiert PoS ein IP-Netz, das die entsprechende Adressierung und das Routing des Internetprotokolls verwendet. Da PoS selbst keine intelligenten Mechanismen besitzt und alle Schicht-3-Protokolle überträgt, musste auch keine Anpassung auf IP erfolgen. Auf diese Weise ist eine volle IP-Integration gegeben.

Nettobitrate

Der Overhead im PoS setzt sich für den Transport von IP-Paketen aus dem PPP- und dem HDLC-Rahmen zusammen. Abb. 4.42 verdeutlicht den Overheadanteil am Beispiel eines 1500 Byte großen IP-Pakets[100]. Der Overhead in den TCP-Rahmen und den SDH-Containern bleibt hier wie zuvor unberücksichtigt.

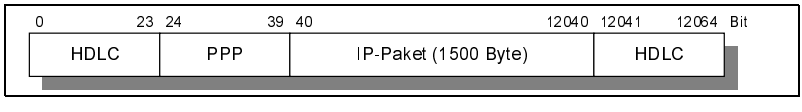
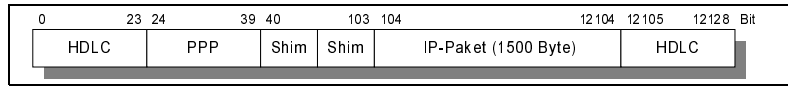


Abb. 4.42 PoS-Overhead

99 Packet-over-SONET
100 Typischer Ethernet-Rahmen

Durch die Abbildung kann man erkennen, dass für den Transport des IP-Pakets ein Overhead von acht Byte benötigt wird. Diese acht Byte setzen sich aus zwei Byte für den PPP-Rahmen und sechs Byte für den HDLC-Rahmen zusammen. Damit errechnet sich das Overhead-Nutzdaten-Verhältnis für ein 1500 Byte großes IP-Paket zu 0,5%. Es ist aber hierbei zu beachten, dass der Overhead im PPP in kritischen Fällen weiter reduziert werden kann. Für die Overhead-Betrachtungen wurde an dieser Stelle von dem maximalen Overhead ausgegangen.

Abb. 4.43 MPLS-Overhead bei PoS



Bei dem Overhead-Anteil von MPLS wird ebenfalls von der maximalen PPP-/HDLC-Einkapselung sowie von zwei Einträgen im Label-Stack ausgegangen. Abb. 4.43 verdeutlicht den Overhead im MPLS. Durch die zusätzlichen Bytes der Shim-Header erhöht sich der zu transportierende Overhead um 8 Byte. Hieraus ergibt sich ein relativer Overhead von 1%.

Performance Wird über PoS ein IP-Netz realisiert, so verwendet es das Schicht-3-Routing für die Paketvermittlung. Die Paketvermittlung basiert auf der Auswertung des IP-Headers eines Pakets und der Suche nach einem Eintrag in der Routing-Tabelle.

Traffic Engineering Das Traffic Engineering (TE) ist in PoS-basierten IP-Netzen über konventionelles Routing nicht zu realisieren. Load Sharing und Explicit Routing (ER) sind zusätzliche Verfahren, die ein TE ermöglichen. Da die Anwendung des Load Sharing auf kleine Netze beschränkt ist, wird hier nur das ER[101] betrachtet. Das ER ermöglicht es, IP-Pakete auf einen bestimmten Pfad zu zwingen. Dies wird realisiert, indem jedem IP-Paket eine ER-Information angefügt wird, anhand derer die Vermittlung in den Netzelementen durchgeführt wird. Die ER-Information legt den Pfad fest, dem ein IP-Paket folgen muss. Durch die Festlegung der ER-Information können bestimmte Netzbereiche entlastet werden.

Quality-of-Service (QoS) Wird über PoS ein IP-Netz realisiert, so bietet es die Möglichkeit, einen IP-QoS auf der Basis der IP-QoS-Modelle IntServ und DiffServ aufzusetzen. Diese Modelle wurden für den Einsatz in IP-Netzen konzipiert.

Bewertung Die IP-Integration ist sowohl im MPLS als auch im PoS gegeben. Aus diesem Grund besteht an dieser Stelle weder ein Vor- noch ein Nachteil von MPLS. Der zu transportierende Overhead ist im MPLS generell höher als im PoS, da die Shim-Header zusätzlich übertragen werden müssen. Dies reduziert die Nettobitrate. Allerdings erhöht sich der Overhead-Anteil auf nur 1%, was nur geringe Auswirkungen auf die Nettobitrate hat. Dennoch besteht an dieser Stelle ein

101 Welches auch als Source Routing bezeichnet wird.

Nachteil von MPLS. Es ist aber zu beachten, dass hierbei der zusätzliche Overhead im PoS, der für die Unterstützung des Traffic Engineering benötigt wird, nicht berücksichtigt wurde. Hier besteht bei MPLS sogar ein Vorteil, da es ein besseres Overhead-Nutzdaten-Verhältnis ermöglicht. Die Performance[102] der untersuchten Protokolle ist bei MPLS prinzipiell höher. Im Gegensatz zum PoS ist es bei MPLS für die Weiterleitung der Pakete nicht notwendig, diese in jedem Netzelement auf Schicht 3 zu verarbeiten. Auch muss MPLS in seiner Forwarding-Tabelle keine Einträge suchen. Diese werden über einen Index einfach ausgelesen. In der Performance liegt also ein Vorteil von MPLS. Allerdings ist zu beachten, dass dieser Vorteil mit der Entwicklung von Routern, die vollständig in Hardware realisiert sind, geringer geworden ist.

Die Realisierung des TE ist bei MPLS effektiver als im PoS. Die für die beiden Protokolle vorgestellten TE-Mechanismen basieren gleichermaßen auf dem Anfügen zusätzlicher Informationen. Diese Informationen enthalten eine Adressliste und bestimmen auf diese Weise den Pfad für ein Paket. Diese Informationen stellen zusätzlichen Overhead dar. Im PoS müssen diese Informationen jedem IP-Paket angefügt werden. Bei MPLS hingegen ist der Transport dieser Informationen nur für den Aufbau des LSP notwendig. Abhängig von der Anzahl der Hops und der damit verbundenen Größe dieser Adressliste wird das Overhead-Nutzdaten-Verhältnis bei MPLS wesentlich weniger belastet als im PoS. Hier liegt ein wichtiger Vorteil von MPLS.

Die Realisierung eines IP-QoS ist sowohl im MPLS als auch im PoS über die IP-QoS-Modelle IntServ und DiffServ vorgesehen. Aus diesem Grund besteht zunächst bei MPLS kein Vorteil. MPLS ermöglicht prinzipiell keine neuen QoS-Konzepte. Dennoch verbessert MPLS die QoS-Fähigkeiten eines IP-Netzes durch die Einführung des Constraint-based Routing, wodurch eine effektivere Nutzung von IntServ und DiffServ durch die aktive Rolle des Routing möglich wird.

Tab. 4.6 Vergleich zwischen PoS und MPLS

WAN-Funktionen	WAN-Technologien bzgl. IP-Transport	
	PoS	MPLS
IP-Integration	Vorhanden	Vorhanden
Nettobitrate und Overhead	Overhead = 0,5% (siehe Beispiel)	Overhead = 1% (siehe Beispiel)
Performance und Forwarding	Layer-3-Routing	Label-Switching
Traffic Engineering	Zusätzliche Verfahren	LDP, RSVP-TE, Contraint-based Routing (CR-LDP)
IP-QoS	IntServ, DiffServ	IntServ, DiffServ

102 Geringe Verzögerungszeiten und hoher Durchsatz

Die Hauptvorteile des MPLS bestehen im Vergleich zum PoS also im verbesserten TE. Die Performance-Vorteile sind abhängig von der verwendeten Hardware und in der Praxis ein nicht schlagkräftiges Argument für die Implementierung von MPLS. Der IP-QoS kann im MPLS optimiert werden. Dennoch besteht auch an dieser Stelle nur ein unbedeutendes Argument für die Einführung von MPLS.

5

Voice-over-IP (VoIP)

Um die Qualität der Sprachübertragung, gerade bezüglich der Kodierungsverfahren, abschätzen zu können, muss auf die Verarbeitung detailliert eingegangen werden. Der Schwerpunkt liegt dabei auf der Wandlung von analogen zu digitalen Signalen. Durch unterschiedliche Kodierungsverfahren ist es heute möglich, die Sprache stark zu komprimieren, wodurch bis zu 9/10 der Bandbreite eingespart werden kann. Dies ist in der Internetumgebung notwendig, um die teilweise schlechten Übertragungsverhältnisse auszugleichen. Nach der Digitalisierung und Komprimierung gelangen die Daten durch ein Übertragungssystem vom Sender (Quelle) zum Empfänger (Senke). Am Empfänger müssen die digitalen Signale wieder dekomprimiert werden, damit abschließend eine Umwandlung in Analogsignale erfolgen kann.

Bei dieser Bearbeitungsreihenfolge, die nicht veränderbar ist, gilt es verschiedene Probleme zu beachten. Dabei tritt bei der paketorientierten Sprachübertragung vor allem die Laufzeit als kritische Messgröße in den Vordergrund. Weitere negative Randbedingungen sind:

- Schwankende Verzögerungen
- Verbindungsunterbrechung
- Echoeffekte
- Bitfehler
- Paketverluste
- Fragmentierung
- Reihenfolgeänderungen der Pakete
- Paketverdopplung

5.1 Sprachqualität

Als wichtigste Eigenschaft bei der Sprachübertragung kann die Qualität der Sprache am Empfänger angesehen werden. Um diese definieren oder messen zu können, muss man bestimmte Parameter festlegen. Aber selbst dann ist die Bestimmung der Sprachqualität eine schwierige Aufgabe, da sie nur subjektiv erfasst werden kann. Die wichtigsten, aber durchaus nicht alle Aspekte, die für die Sprachqualität entscheidend sind, werden in der Abb. 5.1 dargestellt.

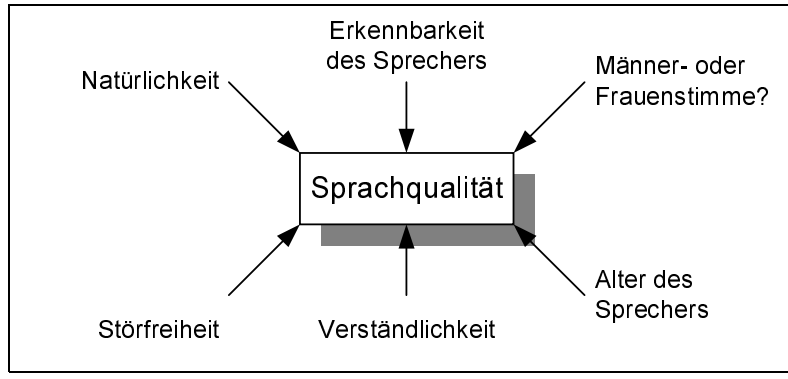


Abb. 5.1 Aspekte der Sprachqualität

Die meisten Messverfahren, die eine Qualität nachweisen sollen, sind subjektiv, da sie von der Beurteilung des Menschen abhängig und demnach Schwankungen unterworfen sind. Bei diesen Verfahren wird eine ausgewählte Sprachprobe von einer gewissen Anzahl von Menschen unter bestimmten Randbedingungen beurteilt. Diese Beurteilungen werden analysiert und statistisch erfasst, um die Ergebnisse einigermaßen reproduzierbar zu halten. Bei den objektiven Messverfahren werden physikalische Messungen und Berechnungen herangezogen, um die Sprachqualität zu ermitteln. Diese sind meistens allerdings nur auf spezielle Fälle anwendbar – man hat festgestellt, dass technische Messgrößen wie Latenzzeit, Jitter und Paketverluste nicht für eine Beurteilung ausreichen, weil der Zusammenhang zwischen Paketverlust und Einbußen bei der Sprachqualität nicht reproduzierbar ist. Aus diesem Grund kommen meistens die subjektiven Verfahren zum Einsatz.

Die wichtigste Messgröße der Sprachqualität ist die Sprachverständlichkeit, da das Hauptziel der Sprachübertragung darin besteht, Informationen zu übermitteln, die verständlich beim Empfänger ankommen müssen. Dabei kann die Sprache hervorragend zu verstehen sein, aber einen unnatürlichen roboterähnlichen Klang aufweisen. Dieser kann beim Hören ein unangenehmes Gefühl entstehen lassen. Also darf man die Sprachverständlichkeit auch nicht überbewerten.

Zur Beurteilung der Sprachqualität wird heute das subjektive Verfahren Mean Opinion Score (MOS) eingesetzt. Bei diesem Verfahren wird einer Reihe von Testpersonen (Größenordnung 25) mindestens ein Satz als Sprachprobe dargeboten. Die Testpersonen beurteilen die Sprachproben durch Vergabe von Noten. Tab. 5.1 zeigt die Testurteile für die Notenvergabe. [FELLB84]

Obwohl der MOS-Wert nicht sehr genau und nur bedingt reproduzierbar ist, hat er sich als Quasistandard durchgesetzt. Hinzu kommt, dass bei hochwertigen Sprachübertragungsverfahren andere Messverfahren oftmals keine großen Ergebnisunterschiede aufweisen. Um die MOS-Werte auch ohne auditive

Beurteilungen durch Menschen zu bekommen, sind Verfahren entwickelt worden, die den ursprünglichen Testmethoden sehr nahe kommen. Beispielsweise das Verfahren Perceptive Speech Quality Measurement (PSQM) nach dem Standard ITU-T P.861 wurde ursprünglich für Telefonnetze entwickelt, lässt sich aber nur bedingt für VoIP einsetzen. Dies liegt an den festen Paketlaufzeiten, von denen es ausgeht, die aber im Internet nicht verfügbar sind. Aus diesem Grund wird momentan bereits der Nachfolger Perceptual Evaluation of Speech Quality (PESQ) nach dem Standard ITU-T P.862 entwickelt, der die Eigenschaften von VoIP berücksichtigt.

Tab. 5.1 MOS-Werte für die Sprachqualität

MOS-Wert	Testurteil	Bedeutung
5	Excellent	Ausgezeichnet – wie das Original
4	Good	Gut – Aufmerksamkeit ist bereits notwendig, ohne Anstrengungen zur Sprachverständlichkeit
3	Fair	Ganz ordentlich – leichte Anstrengungen zur Sprachverständlichkeit notwendig
2	Poor	Mäßig – deutliche Anstrengungen zur Sprachverständlichkeit notwendig
1	Unsatisfactory	Mangelhaft – keine Verständigung möglich

Ein anderes Verfahren ist das Telecommunication Objective Speech Quality Assessment (TOSQA), welches von der Deutschen Telekom T-Nova entwickelt wurde. Dieses Verfahren vergleicht das Ursprungs- mit dem Empfangssignal. Die Vorgänge auditiver Tests werden hier mit den Mitteln der digitalen Signalverarbeitung nachgebildet. Einflussparameter wie unterschiedliche Laufzeiten und Jitter sowie Paketverluste werden berücksichtigt. [ROSS01]

5.2 Quellkodierung und Komprimierung

Dieser Abschnitt beschäftigt sich mit der Quellkodierung. Hier werden einige Verfahren beschrieben, die vor allem für Sprachkodierung angewendet werden. Die Sprachqualitätsbestimmung der einzelnen Quellkodierungsverfahren wird ebenfalls durch die MOS-Werte angegeben.

Die Grundproblematik aller Sprachkodierungsverfahren besteht darin, die Informationen, die im Sprachsignal stecken, so effizient wie möglich übertragen zu können. Zur effizienten Nutzung muss so viel Redundanz wie möglich aus dem Sprachsignal herausgenommen werden, ohne dass Einbußen in der Verständlichkeit und der Spracherkennung hingenommen werden müssen. Gleichzeitig muss das Kodierungsverfahren auch möglichst robust gegenüber den typi-

schen, auf der Übertragungsstrecke auftretenden Fehlern sein. In der Praxis treten sowohl Bitverfälschungen als auch Bitverschiebungen auf. Letztere sind besonders im Mobilfunkbereich durch reflexionsbedingte Mehrwegeausbreitungen vorhanden. Nicht zuletzt deswegen muss die Sprachkodierung auch schnell erfolgen, um eine effiziente Echtzeitübertragung zu ermöglichen. Verzögerungen von 200-500 ms im Halbduplex-Betrieb sind gerade noch akzeptabel. Für Vollduplex-Verbindungen darf der Versatz zwischen den beiden Kommunikationsrichtungen 200 ms nicht übersteigen, um angenehm und akzeptabel zu sein. [FRAN94]

Durch Datenkomprimierung ist es möglich, die zur Verfügung stehende Bandbreite besser auszunutzen. Dabei spielt der wirtschaftliche Aspekt eine wesentliche Rolle. Die eingesparte Bandbreite kann aber auch zur Verbesserung der Sprachqualität oder für die Verschlüsselung der Daten genutzt werden. Für die Kodierung und Komprimierung sollen hier drei verschiedene Ansätze aufgezeigt werden. Das sind als Erstes die Signalformkodierung und deren Komprimierung. Zum zweiten die parametrischen Verfahren zur Kodierung und als drittes die Verbindung aus den beiden erstgenannten Verfahren, die Hybridkodierer. Die Kodierung der drei Ansätze wird auch als Quellkodierung bezeichnet.

5.2.1 Signalformkodierung

Das Grundprinzip der Signalformkodierung ist die Puls Code Modulation (PCM), die wohl die älteste aber auch einfachste Form der Sprachdigitalisierung ist und in dem Standard ITU-T G.711 beschrieben wird. PCM ist ein Modulationsverfahren, das auf Zeitselektion beruht. Das zu digitalisierende Signal wird gewöhnlich in äquidistanten Zeitintervallen abgetastet und die Abtastprobe in einem Analog-/Digitalwandler (A/D) digitalisiert. Der sich ergebende digitalisierte Abtastwert wird dann in binärer Form übertragen. Das PCM-Verfahren erreicht einen MOS-Wert von 4,3. Beim PCM werden die digitalen Signale meistens zusammen mit Signalen anderer Kanäle in einem Zeitmultiplexverfahren digital übertragen. Das Prinzip der Kodierung und Dekodierung ist in Abb. 5.2 zu erkennen und wird in den nachfolgenden Abschnitten schrittweise erläutert.

Bei der Nachrichtenübermittlung zwischen der Quelle und der Senke durchlaufen die Daten unterschiedliche Abschnitte. Da in beide Richtungen kommuniziert wird, muss eine Umwandlung der Signale symmetrisch erfolgen. Aus diesem Grund sind Kodierer und Dekodierer in einer Baugruppe platziert, welche als Codec bezeichnet wird. PCM ist bedingt durch die Vielzahl geeigneter, preiswerter A/D- und D/A-Wandler vergleichsweise einfach zu erzeugen und zu handhaben. Zudem werden PCM- bzw. PCM-ähnliche Kodierungsverfahren in den meisten Sound-Karten in Personal-Computern verwendet.

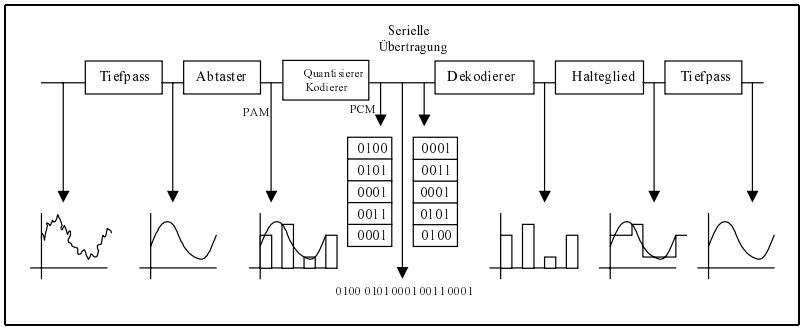


Abb. 5.2 VoIP-Übertragungsstrecke

Abtasttheorem

Das Prinzip von PCM beruht auf dem Abtasttheorem von Shannon. Um ein analoges Signal beim Empfänger exakt zurückzugewinnen, muss die Abtastung des Eingangssignals beim Sender in regelmäßigen Abständen und mit einer entsprechend großen Abtastrate erfolgen. Nach dem Abtasttheorem von Shannon kann man ein bandbegrenztes Signal mit der Grenzfrequenz f_g vollständig durch einzelne Signalwerte beschreiben, wenn die Abtastung in folgendem Abstand vorgenommen wird:

$$t \leq \frac{1}{2 \cdot f_g}$$

Das heißt, das analoge Eingangssignal kann am Empfänger durch mehr als die doppelte maximale Frequenz des Eingangssignals wieder vollständig rekonstruiert werden. Bei der Quantisierung ergibt sich allerdings ein systembedingter Fehler, da nicht alle Amplitudenwerte in digitaler Form vorliegen. Das heißt, dass bei der Zuweisung eines digitalen Wertes für eine abgetastete Amplitudenspannung nur diskrete Werte mit einer begrenzten Genauigkeit zur Verfügung stehen, da die digitale Darstellung begrenzt ist. Diesen Fehler bezeichnet man auch als Quantisierungsfehler.

Zu beachten ist dabei die Bandbegrenzung der Eingangssignale. Falls man ein Signal mit einer höheren Frequenz als der Grenzfrequenz abtastet, bildet der Empfänger ein niederfrequentes Signal zurück. Das heißt, Signale, die bei der Grenzfrequenz liegen, lassen unterschiedliche Interpretationen zu, da beide Signale sich nur in der Anfangsphase unterscheiden. Rekonstruktionen sind nur mehr oder weniger fehlerbehaftet möglich. Deutlich höhere Frequenzen als die der Grenzfrequenz können ebenfalls nicht eindeutig rekonstruiert werden, da beim Empfänger aufgrund der zu geringen Abtastrate ein niederfrequentes Signal erzeugt wird. Dies wird auch als Aliaseffekt bezeichnet.

Diesen Effekt will man in jedem Fall so klein wie möglich halten. Begrenzungen der Eingangssignale ermöglichen die Ausschaltung dieses Effekts. Ein Tiefpass wird normalerweise verwendet, um einen Anti-Aliasfilter einzusetzen.

In Fernsprechsystemen liegt die Bandbreite zwischen 300 Hz und 3,4 kHz. Deshalb wird in der Telefontechnik ein Bandpass eingesetzt, der das Band sowohl nach oben (3,4 kHz) als auch nach unten (300 Hz) begrenzt. [DETK99a]

Abtastung In der Fernsprechtechnik wurde von der ITU-T eine Bandbreite von 300 Hz bis 3400 kHz festgelegt. Weil weder der Abtastpuls noch der Bandpass in der Praxis ideal sind, muss eine entsprechende Toleranz berücksichtigt werden. Dadurch ergibt sich eine Signalfrequenz f_{S0} von 4000 Hz. Dadurch errechnet sich die Abtastfrequenz f_{ab} zu 8000 Hz:

$$f_{ab} = 2 \cdot f_{S0} = 2 \cdot 4000Hz = 8kHz$$

Dadurch ergibt sich der zeitliche Abstand zwischen zwei Proben von 125µs:

$$T_A = \frac{1}{f_{ab}} = \frac{1}{8000\,Hz} = 125\mu s$$

Durch die verwendete 8-Bit-Kodierung und der Abtastrate von 8 kHz ergibt sich eine Bitrate von 64 Kbit/s:

$$Bitrate = 8000\frac{1}{s} \cdot 8bit = 64\frac{Kbit}{s}$$

Weil alle 125 µs abgetastet wird und der Abtastpuls kürzer ist, können in dieser Zeitspanne auch mehrere Abtastwerte von verschiedenen Signalquellen verschachtelt werden. Diese Technik wird als Zeitmultiplexen bezeichnet.

Die Abtastung erfolgt über einen elektronischen Zeitschalter. Dieser wird von der Abtastfrequenz alle 125 µs angeregt, eine Probe des Signals (z.B. Sprache) zu nehmen. Das Ergebnis ist eine Amplitudenmodulation von Abtastfrequenz und Sprachsignal. Es treten somit im Linienspektrum neben den Spektrallinien $n \cdot f_{ab}$ Seitenbänder mit der Frequenz $n \cdot f_{ab} \pm f_S$:

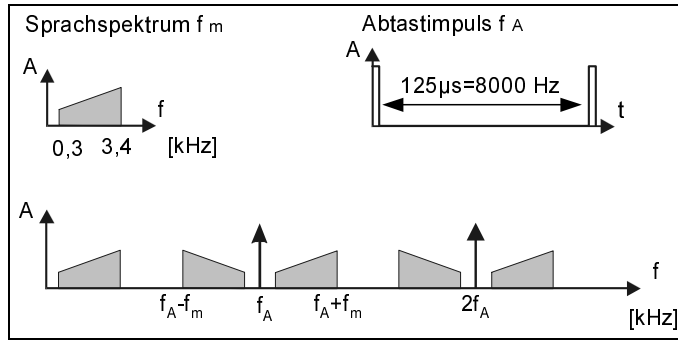


Abb. 5.3 Linienspektrum der Pulscodemodulation (PCM)

Quantisierung

Nach der Bandbegrenzung und der Abtastung ergibt sich ein immer noch analoges Sprachsignal nach der Pulse Amplitude Modulation (PAM). Die Nachricht steckt in den unterschiedlichen Amplituden der Abtastproben. Dieses PAM-Signal ist empfindlich gegen Amplituden- und Phasenstörungen. Um Fehler zu vermeiden, wird das PAM-Signal digitalisiert. Der erste Schritt dahin ist die Quantisierung.

In realen Systemen ist es unmöglich, unendlich viele Amplitudenstufen darzustellen. Deshalb ist man gezwungen, sich auf bestimmte zu beschränken. In PCM[1]-Systemen müssen zuerst die Grenzen der minimalen und maximalen Amplitudenspannung des Eingangssignals festgelegt werden. Innerhalb dieser Grenzen, auch Quantisierungsbereich genannt, müssen Entscheidungsschwellen bzw. Quantisierungsintervalle für die Zuweisung bestimmter digitaler Werte definiert werden. Je nach Größe des abgetasteten Amplitudenwertes wird diesem dann ein digitaler Wert zugewiesen. Allen Abtastwerten, die innerhalb eines Amplitudenintervalls liegen, wird dabei der gleiche Abtastwert zugeordnet. Auf der Empfängerseite wird für diese Abtastwerte die gleiche Amplitudenspannung nachgebildet, wodurch sich der bereits erwähnte Quantisierungsfehler ergibt.

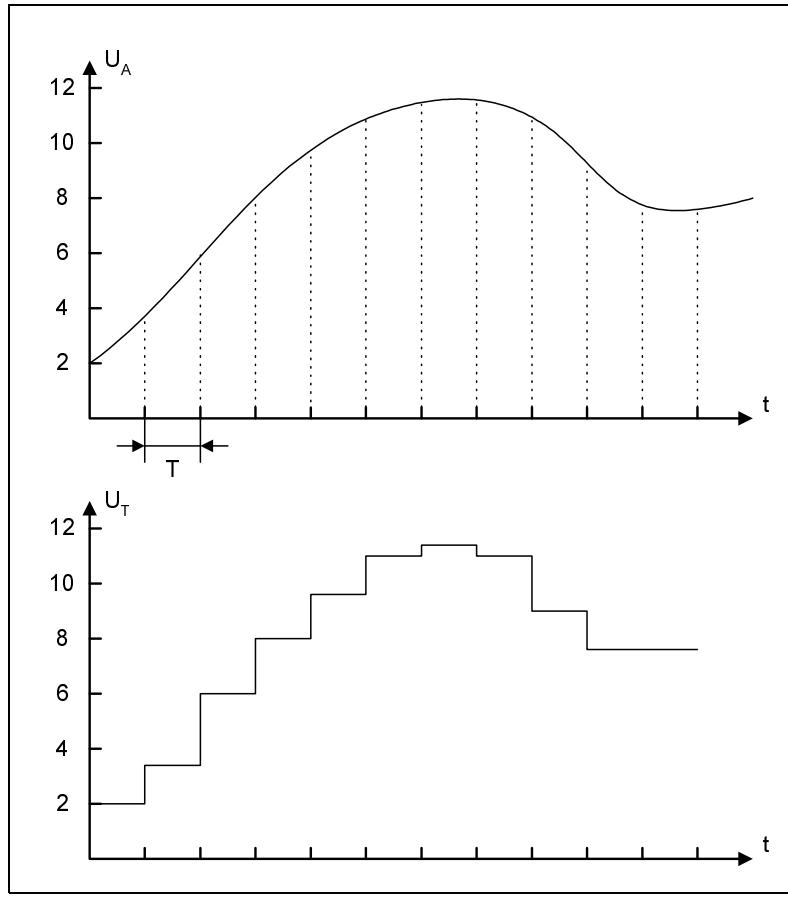
Die Quantisierung kann man unterteilen in die lineare und nichtlineare Quantisierung. Die lineare Quantisierung teilt den gesamten Abtastbereich in gleich große Intervalle ein. In der Mitte eines jeden Quantisierungsintervalls liegen Entscheidungsschwellen für die Ermittlung des Amplitudenwertes. Dabei wird jedem Quantisierungsintervall eine bestimmte Kodierung zugewiesen. Anstelle des exakten Amplitudenwertes wird bei der PCM der ermittelte, digitale Amplitudenwert übertragen, der dem Mittelwert des jeweiligen Quantisierungsintervalls entspricht. Die Amplitudenhäufigkeitsverteilung der Sprache zeigt dabei Folgendes: Kleinere Amplituden sind häufiger als große. Es ist demnach sinnvoll, die Quantisierung nicht linear vorzunehmen, um eine gleichmäßige nicht störende Quantisierungsverzerrung zu erhalten. Die nichtlineare Quantisierung wird nach der A-law-Kennlinie vorgenommen.

Abb. 5.4 zeigt die Abtastung eines analogen Signals mit einem definierten Quantisierungsintervall. U_A stellt dabei die analoge Spannung dar, während U_T die Treppenspannung zeigt, die das analoge Signal digital aufbereitet. Durch die lineare Quantisierung wird das Signal in 11 gleiche Intervalle eingeteilt. Auf der Empfängerseite wird aus dem Mittelwert jedes Quantisierungsintervalls der Signalwert zurückgewonnen. Dadurch ergeben sich Abweichungen vom Ursprungssignal *DU* von maximal einem halben Quantisierungsintervall. Diese Abweichungen können sich auf der Empfängerseite als Quantisierungsverzerrung bemerkbar machen. Wenn die Quantisierungsintervalle gleichmäßig ver-

1 Pulse Code Modulation

teilt werden (lineare Quantisierung), ist bei niedrigen Signalpegeln *DU* besonders groß. Hieran wird auch der Quantisierungsfehler verdeutlicht, da die Abtastung nur annähernd das analoge Signal abbilden kann. Der Quantisierungsfehler verursacht ein Quantisierungsgeräusch, das dem eigentlichen Signal überlagert ist und als Quantisierungsrauschen bezeichnet wird. [HELO90]

Abb. 5.4 Quantisierung



A-law-Kennlinie Die ITU-T empfiehlt in G.711 für die angewandte PCM[2] eine nicht lineare Quantisierung nach der so genannten A-law-Kennlinie. Diese A-law-Kennlinie setzt sich aus 13 linearen Teilstücken (Segmenten) zusammen und wird in Europa verwendet. In Japan und USA ist die so genannte µ-law-Kennlinie üblich. Beide Kennlinien unterscheiden sich geringfügig in der Kodierung, sind aber inkompatibel! Bei digitalen Verbindungen zwischen Europa und Amerika oder Japan muss daher eine Signalumsetzung erfolgen.

2 Pulse Code Modulation

Die Funktion der A-law-Kennline wird durch zwei Gleichungen beschrieben[3], wobei *Sgn* dem Vorzeichen der Eingangsspannung entspricht und der Wert *A* mit 87,6 von der ITU-T festgelegt wurde:

$$F(x) = Sgn(x) \cdot \left[\frac{1 + \lg(A \times |x|)}{1 + \lg A} \right] \text{ für } \frac{1}{A} \le |x| \le 1$$

$$F(x) = Sgn(x) \cdot \left[\frac{(A \times |x|)}{1 + \lg A} \right] \text{ für } 0 \le |x| \le \frac{1}{A}$$

Die ersten 64stel der positiven und negativen A-law-Kennlinie weisen die gleiche Steigung auf und werden daher als ein einzelnes Segment betrachtet. Dieses Segment wird in insgesamt 64 Quantisierungsintervalle unterteilt. Die weiteren sechs Segmente weisen 16 Quantisierungsintervalle auf. Alle 13 Segmente zusammen beinhalten 256 Intervalle. Durch die nichtlineare Quantisierung wird eine einwandfreie Übertragung der Sprache gewährleistet. Allerdings gilt dies nur im Quantisierungsbereich. Überschreitet die Signalamplitude den Quantisierungsbereich, wird sie praktisch abgeschnitten, sodass es auch zu Störungen kommen kann.

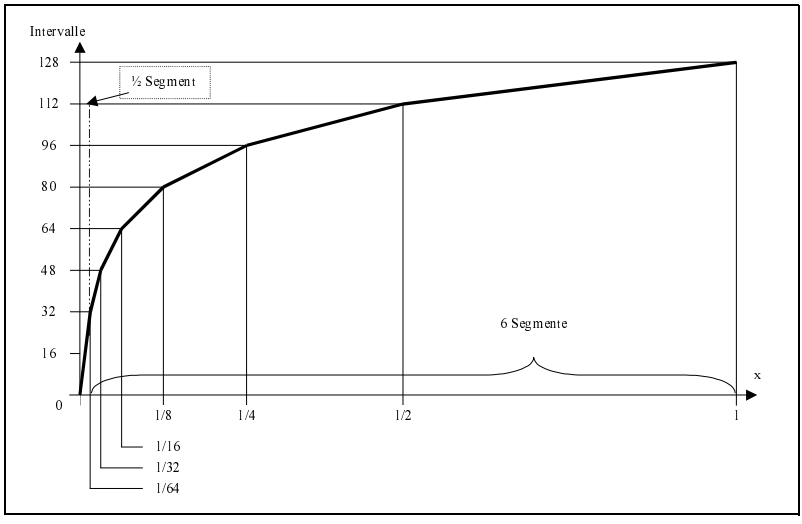


Abb. 5.5
x – Eingangssignal, normiert auf die halbe Maximalspannung
Der positive Ast der A-law-Kennlinie

Kodierung

Um Bits oder Bitsequenzen mit einer bestimmten Länge durch einen oder mehrere Grundimpulse auf dem Übertragungskanal abbilden zu können, ist eine bestimmte Vorschrift nötig. Diese Abbildungsvorschrift wird allgemein als Leitungscode bezeichnet. Unter dem Begriff Kodierung versteht man somit im

3 Symmetrisch für den positiven und negativen Kennlinien-Teil

Allgemeinen die Zuordnung einer Nachrichtenmenge zu einer Menge von Symbolen oder Zeichen. Dabei können unterschiedliche Ziele verfolgt werden. Die Kodierung zur Verschlüsselung wird Kryptographie genannt und wird verwendet, um Nachrichten gegen missbräuchliche Benutzung zu schützen. Hingegen versucht die Quellenkodierung eine Nachrichtenmenge zu komprimieren. Die Kanalkodierung versucht die Nachrichtenmenge gegen Störungen im Übertragungskanal unempfindlich zu machen. Genauer betrachtet enthält die Quellenkodierung eine gewisse Redundanz, während die Kanalkodierung Redundanz gezielt hinzufügt. Man unterscheidet bei der Kanalkodierung dabei deterministische und stochastische Codes.

Nach der Quantisierung wird somit jeder Amplitudenstufe eine eindeutige Bitkombination (Kodierung) zugewiesen. Nach der Definition der ITU-T wird jeder Amplitudenwert durch 8 Bit dargestellt. Das erste Bit gibt das Vorzeichen der abgetasteten Amplitude an, die nächsten 3 Bit definieren das Segment (lineares Teilstück der A-law-Kennlinie), und die letzten 4 Bit legen die Quantisierungsintervalle innerhalb des Segments fest. Diese Art der Kodierung wird symmetrischer Binärcode genannt.

Tab. 5.2 Quantisierungsintervalle mit Codewort

Quantisierungsintervalle	Codewort
+1	10000001
+0	10000000
-0	00000000
-1	00000001

Durch die häufig auftretenden kleinen Amplituden ergeben sich lange Nullfolgen. Diese langen Nullfolgen sind aus Gründen der Taktrückgewinnung in der Nachrichtenverarbeitung unerwünscht. Durch den symmetrischen Binärcode mit abwechselnder Zeichenfolge wird dieser Effekt verhindert. Hierbei werden die geraden Zahlenwerte (Bits) des Codewortes invertiert, wie es an dem +0-Intervall zu sehen ist.

Tab. 5.3 Symmetrischer Binärcode

Bitanzahl	1.	2.	3.	4.	5.	6.	7.	8.
Nicht invertiert	1	0	0	0	0	0	0	0
Gerade Zahlenwerte, invertiert	1	1	0	1	0	1	0	1

Die empfangenen Codewörter werden im Dekodierer des Empfängers mit der gleichen Taktfrequenz wie im Sender verarbeitet und als diskrete Spannungswerte ausgegeben. Diese werden in einer Abtast-Halteschaltung für eine Abtast-

periode gespeichert und durch einen Tiefpass vollständig demoduliert. Abb. 5.6 zeigt in einer Übersicht noch einmal die einzelnen Schritte von der Digitalisierung bis zur Rückwandlung des Signals.

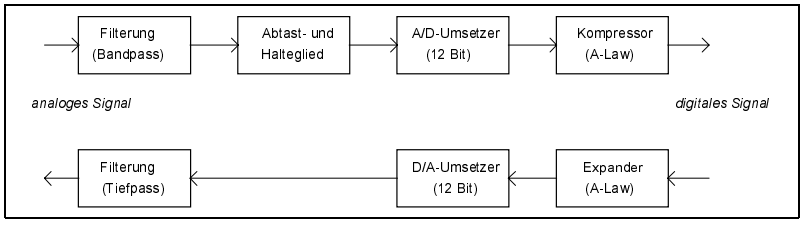


Abb. 5.6 Digitale Sprachkodierung

5.2.2 Komprimierung von PCM-Signalen

Die Kodierung von analogen Signalen nach PCM[4] hat eine breite Anwendbarkeit. Bei den analogen Signalen kann es sich, wie in der Telefonie, um Sprache handeln. Aber auch Musik oder Video kann mittels PCM kodiert werden. Auch ist die Abweichung zwischen empfangenen und gesendeten Signale sehr gering. Der Nachteil ist der große Bandbreitenbedarf von PCM. Da für die Internetkommunikation eine geringe Bandbreite entscheidend ist, um die Sprachqualität möglichst hochzuhalten, kann man verschiedene Komprimierungsverfahren einsetzen, die an dieser Stelle vorgestellt werden.

Difference Pulse Code Modulation (DPCM)

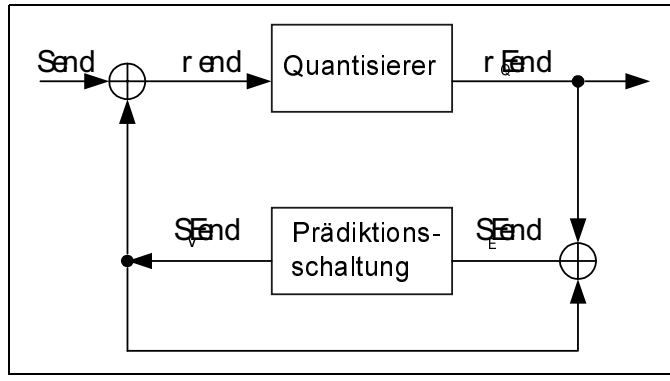
Bei der Übertragung von PCM wird ein beträchtlicher Teil an redundanten Informationen übertragen. Denn häufig ändern die aufeinander folgenden Signalproben ihre Werte nur minimal. Diese Tatsache macht sich die DPCM zu Nutze, um eine Vorhersage (Prediction) der zu erwartenden Werte zu bilden.

Mit Hilfe des momentanen Abtastwertes kann ein Vorhersagewert $s_V(n)$ für die nächste Signalprobe gebildet werden. Der Vorhersagewert wird von der tatsächlich anliegenden Signalprobe $s(n)$ abgezogen. Die bei der Subtraktion entstehende Differenz $d(n)$ wird quantisiert. Der Vorhersagewert $s_V(n)$ und die quantisierte Differenz $d_Q(n)$ bilden durch Addition das Eingangssignal $s_E(n)$ der Prädiktionsschaltung, welches den Vorhersagewert berechnet. Die Übertragungsfunktion der Prädiktionsschaltung ist der eines Integrators ähnlich.

Im Empfänger steht die Differenz $d(n)$ durch Übertragung zur Verfügung. Der Vorhersagewert wird auf die gleiche Weise gebildet wie im Sender. Dadurch ist es möglich, die Signalproben $s(n)$ im Empfänger durch Addition von der Differenz $d(n)$ und dem Vorhersagewert $s_V(n)$ wiederzugewinnen. Die Differenz $d(n)$ wird kodiert und übertragen. Da die Differenz $d(n)$ erheblich weniger Informationen aufweist als die ursprüngliche Signalprobe, kann ein kleineres Codewort verwendet werden und die Bitmenge wird komprimiert.

4 Pulse Code Modulation

Abb. 5.7
Prinzip der DPCM



Das Codewort kann umso kleiner sein, je gröber die Abstufungen des Quantisierers sind. Bei grober eingeteilten Quantisierungsstufen ergeben sich jedoch Probleme mit Eingangssignalen, die eine hohe Dynamik aufweisen. Dann kann der DPCM schnellen Spannungsänderungen nicht folgen und es kommt zu Verzerrungen des demolierten Signals.

Adaptive Difference Pulse Code Modulation (ADPCM)

Noch einen Schritt weiter als DPCM geht die Technik der Adaptive Difference Pulse Code Modulation (ADPCM). Ein System ist adaptiv, wenn es in der Lage ist, seine internen Parameter als Reaktion auf ein sich änderndes Eingangssignal anzupassen. Dies ermöglicht eine genauere Abgleichung des kodierten Signals an das ursprüngliche Sprachsignal.

Ein Möglichkeit besteht darin, die Quantisierungsstufen dem Eingangssignal anzupassen. Ein solcher adaptiver Quantisierer vergrößert die Quantisierungsstufen, wenn sich das Signal in der letzen Quantisierungsstufe bewegt (positiv oder negativ). Bei kleinen Signalen, die nur die erste Stufe ausfüllen, wird die Schrittweite verringert. Durch die adaptive Quantisierung kann die Kurvenform des Eingangssignals besser abgebildet werden. Die Verzerrungen der ADPCM gegenüber der DPCM und PCM sind deutlich geringer.

ADPCM wurde von der ITU-T mit 40, 32, 24 und 16 Kbit/s in G.726 standardisiert. Die Empfehlung G.721 und G.723, die im Zusammenhang mit ADPCM genannt werden, sind durch G.726 komplett ersetzt worden. ADPCM mit 32 Kbit/s benötigt als Eingangssignal ein PCM-Signal nach der A-law-Kennlinie. Die Ausgangssignale bilden einen Datenstrom. Dabei ist eine Differenz zwischen Probe und Vorhersagewert mit vier Bit kodiert. In dem Datenstrom sind in einem Byte zwei Differenzen enthalten. Der 32-Kbit/s-ADPCM-Strom erreicht dabei bereits einen MOS-Wert von 4,1.[5]

5 G.726: 40, 32, 24, 16 Kbit/s Adaptive Differential Pulse Code Modulation (ADPCM); ITU-T 12/1990

Linear Predictive Coder (LPC)

Das Verfahren LPC gehört zu den parametrischen Verfahren. Bei diesen Verfahren wird berücksichtigt, dass es sich bei dem zu übertragenden Signal um Sprache handelt, da Sprache typische Eigenschaften aufweist. Der Grundgedanke bei dem im Folgenden beschriebenen parametrischen Verfahren besteht nun darin, nur die Eigenschaften als reduziertes Sprachsignal zu übertragen. Mit Hilfe der Parameter und einem elektrischen Modell kann die Sprache synthetisch erzeugt werden.

Die wichtigsten parametrischen Systeme sind die Vocoder (Voice + Coder). Alle Vocoder analysieren senderseitig die Sprache nach Grundfrequenz und Lautbildung und übertragen diese durch Parameter. Auf der Empfangsseite wird mit den Parametern das elektrische Modell zur Spracherzeugung abgeglichen und so die Sprache wiedergewonnen. Die natürliche Wiedergabe der Sprache ist bei den Verfahren unterschiedlich. Beim Phonem-Synthesizer hat die Sprache einen sehr unnatürlichen Klang („Roboterstimme"). Ein wichtiger Vertreter der Vocoder ist der Linear Predictive Coder (LPC). Bei diesem Verfahren ist eine hohe „Natürlichkeit" der Sprachwiedergabe gewährleistet. Um die parametrischen Verfahren verstehen zu können, muss zunächst die Erzeugung von Sprache dargestellt werden.

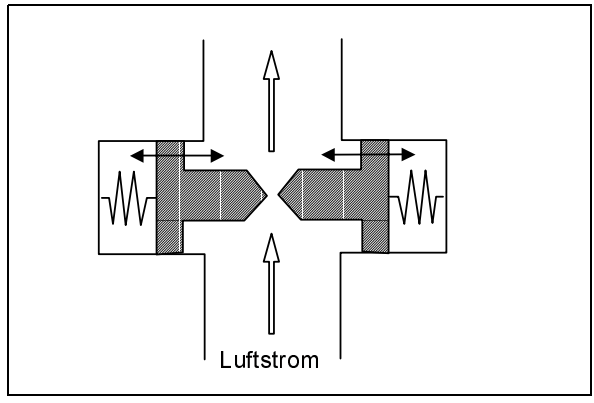


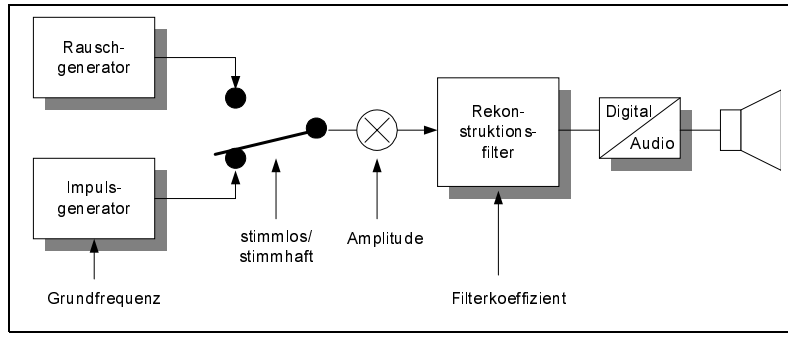
Abb. 5.8
Technisches Modell des Kehlkopfes

Um die parametrischen Verfahren besser nachvollziehen zu können, soll hier kurz die Erzeugung der Sprache beschrieben werden. Der Kehlkopf kann als eigentliches Stimmorgan bezeichnet werden. Ein Luftstrom regt die im Kehlkopf befindlichen Stimmbänder an, die beiden beweglichen Klappen zum Schwingen zu bringen, siehe Abb. 5.8. Diese Schwingung regt wiederum die an die Stimmbänder anschließende Luftsäule zum Schwingen an. Der Hohlraum über dem Kehlkopf (Mundhöhle, Rachenraum usw.) stellt also einen Hohlraumresonator da und wird Artikulationstrakt genannt. Dieser formt dann die verschiedenen Laute. Es wird bei der Lauterzeugung unterschieden zwischen „stimmhaftem Laut" (Vokale) und „stimmlosem Laut" (Konsonant). Der

stimmhafte Laut entsteht im Artikulationstrakt durch eine selbsterregte Schwingung der geschlossenen Stimmbänder. Die Frequenz des stimmhaften Lautes wird als Sprachgrundfrequenz oder kurz als Grundfrequenz bezeichnet. Sie liegt zwischen 80 Hz und 350 Hz.

Die Grundfrequenz ist nur beim Singen periodisch, beim Sprechen kann sie in Amplitude und Frequenz variieren. Es kann deshalb nur von einem quasi-periodischen Verlauf gesprochen werden und von diesem auch nur im eingeschwungenen Teil der stimmhaften Laute. Es sind also nur kurze Abschnitte von ca. 20 ms Dauer periodisch. Der stimmlose Laut entsteht, wenn die Stimmbänder auseinander stehen. Der hindurchströmende Luftstrom bricht sich an Kanten und Ritzen des Artikulationstraktes. Es entsteht ein rauschförmiges Anregungssignal, welches ein kontinuierliches Spektrum mit noch hohem Energieanteil bis 7 kHz aufweist. Der stimmlose Laut gibt der Stimme ihre spezifische Färbung. Durch gezieltes Mischen von stimmhaften und stimmlosen Lauten, wird die für einen Menschen charakteristische Sprache geformt. [FELLB84]

Abb. 5.9 Elektrisches Modell zur Spracherzeugung



Bei LPC[6] wird auf der Empfangsseite mittels eines elektrischen Modells Sprache erzeugt. Dieses Modell resultiert aus den Erkenntnissen der Spracherzeugung des Menschen. Durch eine Analyse auf der Senderseite werden Grundfrequenz, Entscheidung „stimmhaft/stimmlos“, Amplitude und ein Satz Resonanzfrequenzen gefiltert und als Parameter übertragen. Mit diesen Parametern wird das elektrische Modell abgeglichen. Die Parameter gelten für kurze Stücke der Sprache von 10 bis 30 ms.

Die Parameter können ähnlich wie bei der ADPCM durch vorangegangene Parameter vorhergesagt werden. Auf der Empfangsseite werden die so genannten Vorhersageparameter identisch mit der Sendeseite gebildet und mit der übermittelten Differenz zwischen tatsächlichem und vorhergesagtem Wert zusammengesetzt. LPC-Systeme sind seit geraumer Zeit im kontinuierlichen

6 Linear Predictive Coder

Einsatz. So ist z.B. das LPC-10-System seit 1984 standardisiert und erlaubt die Übertragung von Sprache bei ca. 2400 bit/s. Es werden Wandlungszeiten in der Größenordnung 20 ms erreicht. Die Sprachqualität erreicht einen MOS-Wert von 2,0.

Codebook Excited Linear Predictive Coding (CELP)

CELP ist ein hybrides Kodierungsverfahren, welches die Vorteile der Signalformkodierung und der parametrischen Verfahren ausnutzt, ohne die schlechten Eigenschaften beider Verfahren übernehmen zu müssen. Das Ergebnis ist eine gute Sprachqualität, die man mit der von PCM trotz einer hoher Kompression vergleichen kann. Die Grundlage für die meisten Hybridverfahren ist LPC. Unterschiede gibt es lediglich bei der Kodierung des verbleibenden Restsignals. Durch die Übermittlung sowohl des Fehlersignals als auch der LPC-Parameter kann eine niedrigere Datenrate (4-16 Kbit/s) bei guter Sprachqualität erzeugt werden. Die große Komplexität der hybriden Systeme, die durch die doppelte Kodierung entsteht, ist aufgrund der heute realisierbaren Verschmelzung aller Baugruppen auf einem Chip nicht mehr von Bedeutung. [FELLB84]

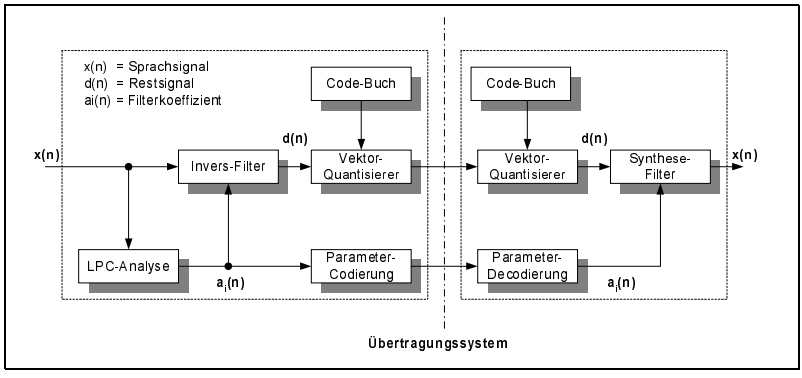


Abb. 5.10 Grundsystem nach dem CELP-Verfahren

Codebook Excited Linear Predictive Coding (CELP) ist ein typischer Vertreter der Hybridkodierer, der am Anfang eine LPC-Analyse durchführt. Diese Analyse liefert die Koeffizienten $a_i(n)$ für einen inversen Filter. Die Übertragungsfunktion des Filters ist durch die LPC-Koeffizienten näherungsweise invers zur Hüllkurve des Sprachsignals. So ergibt sich am Filterausgang ein Restsignal $d(n)$, welches aus der Differenz des Sprachsignals zur Übertragungsfunktion des Filters besteht. Dieses Restsignal $d(n)$ wird nun zusätzlich zu den LPC-Koeffizienten $a_i(n)$ übertragen und repräsentiert das gleiche Segment des Sprachsignals wie die LPC-Koeffizienten. Auf der Empfängerseite wird das Restsignal als Anregungssignal für ein Filter benutzt, welches wieder mit den LPC-Koeffizienten gespeist wird. Ein guter CELP-Codec erzeugt eine Sprachqualität, die man kaum von einer 64-Kbit/s-PCM-Übertragung unterscheiden kann.

Der Grund für die Namensgebung von CELP liegt in der Art und Weise der Kodierung des Restsignals. Die Kodierung arbeitet auf der Basis von Codebüchern. Das Muster des Restsignals eines Segmentes des Sprachsignals wird als Vektor V mit k-Elementen betrachtet. Dieser Vektor wird direkt als Zahlenwert mit R-Bits umgesetzt. Im Codebuch sind eine endliche Anzahl Vektoren Z, die voraussichtlich auftreten können, abgelegt. Die einzelnen im Codebuch gespeicherten Vektoren Z_i, sind durch Indizes gekennzeichnet. Durch Vergleichen der einzelnen Vektoren V_i mit den im Codebuch abgelegten Vektoren Z_1 bis Z_n wird der ähnlichste Vektor Z_i ausgewählt und durch seine Indizes übertragen. Weil die Kodierung durch Vektoren geschieht, wird dieses Verfahren als Vektor-Quantisierung bezeichnet. Allerdings kommt es zu Verzerrungen des ursprünglichen Signalverlaufs. V und Z können als Mengen aufgefasst werden:

$$V = \{V_1 \ldots V_n\} = \{V_i \mid V_i \in V\} \text{ und } Z = \{Z_1 \ldots Z_n\} = \{Z_i \mid Z_i \in Z\}$$

Für Z gilt weiter, dass es eine Teilmenge von V ist:

$$Z \subset V$$

Weil nicht alle Vektoren, die auftreten, im Codebuch enthalten sein können, muss ein gespeicherter Vektor Z_i für mehrere Elemente von V gelten. In allen Fällen, in denen der aus dem Codebuch gewählte Vektor Z_i ungleich zum tatsächlich auftretenden Vektor V_i ist, kommt es zu Verzerrungen.

Es wurden unterschiedliche Methoden entwickelt, die Leistungsfähigkeit der Kodierer zu steigern, um die Verzerrung zu minimieren. Einen großen Einfluss auf die Leistungsfähigkeit der Vektor-Quantisierung hat der Inhalt der Codebücher. Dieser muss sehr sorgfältig ausgewählt werden und den jeweiligen Erfordernissen angepasst sein. Daher gibt es Codebücher mit festem Inhalt und andere, bei denen der Inhalt während des Betriebs geändert werden kann. Ein weiterer wichtiger Punkt ist die Struktur, nach welcher die Vektoren in dem Codebuch abgelegt werden. Die Struktur bestimmt den Suchalgorithmus und die damit verbundene Rechenleistung und Speicherkapazität. [EPPI93]

Aktuelle Forschungstendenzen bestehen darin, neue Ansätze zu entwickeln und unter anderem auch die CELP-Kodierer zu verbessern. Die Modifikationen an den CELP-Kodierern betreffen vor allem die Anregungssignale, um die Probleme bei den Übergängen stimmhafter- und stimmloser Sprache zu verbessern. Aber auch Verbesserungen der Codebücher und der notwendigen Suchalgorithmen sind Gegenstand der Forschungen. In den nachfolgenden Abschnitten sind drei Verfahren aufgeführt, die auf CELP-Kodierern basieren und sich für die Sprach-Daten-Integration gut eignen. Sie werden ebenfalls von der ITU-T im Rahmen des Standards H.323 empfohlen: [PAKO99]

- Low-Delay-CELP (LD-CELP)
- Conjungate-Structur-Algebraic-CELP (CS-ACELP)
- Dual Rate Speech Coding

Low-Delay CELP (LD-CELP)

Der Standard LD-CELP, entsprechend der ITU-T-Spezifikation G.728, weist eine Bitrate von 16 Kbit/s auf und erreicht einen MOS-Wert von 4,0 für die Sprachqualität. Dabei liegt die Signalverzögerung, die durch das Kodieren und Dekodieren entsteht, nur bei 0,625 ms.[7]

Nach G.728 ist das Eingangssignal ein PCM-Signal nach der A-law-Kennlinie. Dabei werden fünf Abtastwerte des PCM-Signals als Vektor zusammengefasst. Dieser Vektor wird mit 1024 Vektoren in einem Codebuch verglichen. Der Vektor aus dem Kodebuch mit den geringsten Abweichungen zum Originalvektor wird durch seine Indizes übertragen. Die zu übertragenen Indizes sind 10 Bit groß.

Conjugate Structur Algebraic CELP (CS-ACELP)

CS-ACELP entspricht dem ITU-T-Standard G.729 und führt vor der Kodierung aufwendige Analysen beim Vergleich des Sprachsignals mit dem Modell durch. CS-ACELP erreicht eine identische Sprachqualität zu LD-CELP, benötigt jedoch nur die halbe Bitrate. Dafür ist eine wesentlich höhere Rechenleistung bei der Kodierung und Dekodierung der Signale erforderlich. Die verursachte Signalverzögerung durch den Kodierer beträgt mindestens 20 ms. [FRAN94]

In dem Standard G.729 wird als Eingangssignal eine lineare PCM, die eine Abtastfrequenz von 8 kHz aufweist und mit 16 Bit kodiert ist, vorgeschrieben. Im Kodierer werden immer zusammenhängende 10-ms-Sprachrahmen bearbeitet. Die Sprachrahmen enthalten 80 Proben mit 1280 Bit (160 Byte). Diese 1280 Bit werden im Kodierer in einige Parameter zerlegt (LPC-Parameter, Codebuch-Index, Verstärkung usw.). Somit wird aus den 10-ms-Sprachrahmen, die 1280 Bit lang waren, ein Rahmen mit der Längen von 80 Bit (10 Byte).[8]

Dual Rate Speech Coding

Die Empfehlung G.723.1 der ITU-T beschreibt das Dual Rate Speech Coding. Das bedeutet, es sind zwei Bitraten in dieses System fest integriert. Es ist möglich, jederzeit zwischen den Bitraten zu wechseln. Das System wurde für Sprache optimiert. Es ist aber auch möglich, Audiosignale anderer Multimediadienste zu übertragen, z.B. Musik. Dabei kommt es allerdings zu Qualitätseinbußen.

7 G.728: Coding of speech at 16 Kbit/s using low-delay code excited linear prediction; ITU-T 09/1992

8 G.729: Coding of speech at 8 Kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP); ITU-T 03/1996

G.723.1 beruht auch auf der LPC-Analyse und der Übertragung durch LPC-Parameter und Restsignal. Die zwei Bitraten sind durch unterschiedliche Anregungssignale möglich, die innerhalb des Systems erzeugt werden. Für die höhere Bitrate von 6,3 Kbit/s wird das Anregungssignal nach Multipulse Maximum Likelihood Quantization (MP-MLQ) verwendet. Die Sprachqualität liegt hier bei 3,9 MOS. Das Anregungssignal für die 5,3-Kbit/s-Bitrate wird nach Algebraic CELP (ACELP) gebildet, wobei die Sprachqualität ein bisschen schlechter ist. Das Eingangssignal ist das gleiche wie in G.729 (lineare PCM, 8 kHz, 16 Bit). Der Kodierer bearbeitet PCM-Rahmen mit 240 Sprachproben. Bei einer Bitrate der linearen PCM von 128 Kbit/s ist der Rahmen 30 ms lang. Bei der 5,3-Kbit/s-Bitrate enthalten diese Rahmen nach der Kodierung 20 Byte an Sprachinformation. 24 Byte Sprachinformation werden pro Frame bei der höheren Bitrate von 6,3 Kbit/s übertragen:

$$(1)\quad \frac{5300\,Bit}{1s} \Rightarrow \frac{5{,}3\,Bit}{1ms} \cdot 30ms = \frac{159\,Bit}{Frame} \cdot \frac{1}{8} Byte \approx \frac{20\,Byte}{Frame}$$

$$(2)\quad \frac{6300\,Bit}{1s} \Rightarrow \frac{6{,}3\,Bit}{1ms} \cdot 30ms = \frac{189\,Bit}{Frame} \cdot \frac{1}{8} Byte \approx \frac{24\,Byte}{Frame}$$

Es stellt sich eine Verzögerung durch die zeitliche Dauer der Rahmen von mindestens 30 ms ein. Zusätzlich entstehen Verzögerungen, die von den verwendeten Baugruppen abhängen.

5.2.3 Breitbandkodierung

Die Bandbreite der Breitbandkodierung wurde gegenüber dem klassischen Telefon mit einer Bandbreite von 300-3400 Hz auf 50-7000 Hz erweitert. Dabei bewegt sich die Sprachgrundfrequenz zwischen 80 und 350 Hz. Der Rauschanteil beinhaltet Frequenzen deutlich größer 3,4 als kHz. Deshalb wird durch den ausgedehnten Frequenzbereich die Sprachqualität wesentlich verbessert. Standardisiert ist ein Breitbandkodierer in G.722 von der ITU-T. Nach G.722 wird mit einer Abtastfrequenz von 16 kHz und Bitraten von 64, 56 und 48 Kbit/ s gearbeitet. Die MOS-Werte für die Bitraten ergeben sich zu:[9]

- 64 Kbit/s: MOS-Wert 4,1
- 56 Kbit/s: MOS-Wert 4,0
- 48 Kbit/s: MOS-Wert 3,7

Um die Bitraten bei gleicher Qualität weiter zu verringern, werden zusätzlich CELP-Kodierer eingesetzt. Diese Ansätze liefern bei Bitraten von 16 Kbit/s eine gute Sprachqualität.

9 G.722: 7 kHz audio-coding within 64 Kbit/s; ITU-T 1988

5.2.4 Moving Pictures Experts Group (MPEG)

Als Komprimierungsverfahren wird MPEG vorrangig für Video und Audio eingesetzt, also nicht speziell für Sprache. Allerdings ist das MPEG-Verfahren ebenfalls ein Bestandteil der ITU-T Empfehlung H.323 und muss daher auch berücksichtigt werden. Dabei beruhen alle unter MPEG standardisierten Kodierverfahren auf einer Trennung des Verfahrens in Frequenzanalyse und Quantisierung mit dynamischer Bitzuweisung. Zur Frequenzanalyse wird das Audiosignal in eine Anzahl von Teilbändern zerlegt. Die Quantisierung der Teilbänder wird im Wesentlichen von zwei Charakteristika eines zugrunde liegenden psychoakustischen Hörmodells beeinflusst. Diese sind simultane und zeitliche Maskierung. Die Quantisierung wird so vorgenommen, dass für das Gehör irrelevante Frequenzanteile nicht übertragen werden. Steht ausreichend Übertragungsbandbreite zur Verfügung, kann ein für das Gehör transparentes Audiosignal rekonstruiert werden.

Es wurden bis heute die MPEG-Verfahren 1, 2 und 4 standardisiert. Die Entwicklung von MPEG 7 soll im Jahr 2002 abgeschlossen sein. Innerhalb von MPEG-1 und MPEG-2 sind jeweils 3 Layer definiert. Dabei analysieren die einzelnen Layer unterschiedliche Bandbreiten. Die Präzision der Kodierer nimmt mit den Layern zu. MPEG-1 Layer 3 (kurz MPEG-3 genannt) arbeitet mit $32^2 18=576$ Teilbändern, um die Auflösung zu erhöhen, statt nur mit 32 Teilbändern wie MPEG-1 Layer 1. Das Prinzip der einzelnen MPEG-Standards ist immer das gleiche. Allerdings wurden die Verfahren erweitert und verfeinert. Es wurde beispielsweise bei MPEG-2 die Signalvorhersage integriert. Das Ergebnis der Erweiterung und Verfeinerung ist, dass immer größere Datenmengen bewältigt werden können.

In MPEG-7 werden zusätzlich zu den komprimierten Daten Informationen über deren Eigenschaften, Klassifizierung, Zugriffsrechte und Links abgespeichert. Dadurch wird eine datenbankartige Struktur für multimediale Daten ermöglicht. Durch die datenbankartige Struktur ist es möglich, verschiedene Dienste (Audio, Video usw.) zusammenzufassen, auch wenn diese Dienste nicht den gleichen Ursprungsort haben. Vorstellbar wäre ein großes intelligentes Archiv mit MPEG-7-komprimierten Daten. [PAKO99]

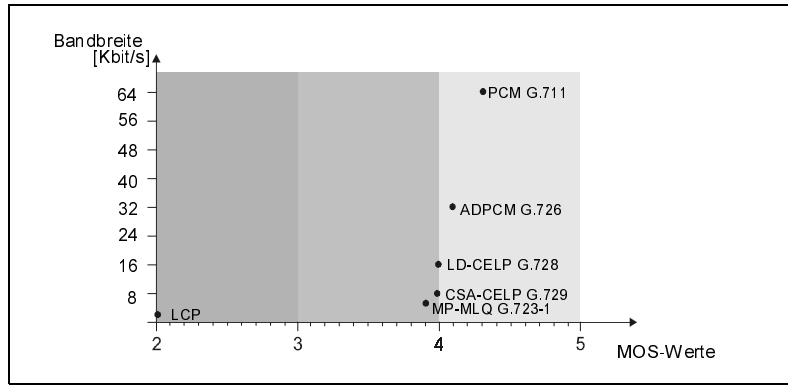
5.3 Störeffekte

Die heute immer mehr in den Vordergrund tretenden Multimediaanwendungen machen eine immer größere Komprimierung der Daten erforderlich. Es soll eine ständig wachsende Menge an Daten in Echtzeit übertragen werden. Allerdings sinkt mit der Bitrate auch die Qualität, wie anhand der Sprachkomprimierung in Abb. 5.11 zu erkennen ist.

Die Abbildung zeigt, dass durch neue Techniken wie Hybridkodierer (CELP) trotz stark reduzierter Bandbreite, die Qualität relativ gut ist. Das Forschungsziel ist es, durch neue Verfahren bei Bitraten von 4 Kbit/s die Sprache nahezu natürlich klingen zu lassen. Allerdings muss hier bedacht werden, dass ein solches Ziel nur durch aufwendige algebraische Vorgänge im Kodierer erreicht werden kann. Dadurch erhöht sich aber wiederum die Verzögerungszeit (Delay), die durch die Kodierung entsteht.

Wie in Abb. 5.12 zu sehen ist, steigt die Verarbeitungszeit mit sinkender Bitrate. Dadurch erhöht sich auch die Gesamtlaufzeit, was für Echtzeitdaten sehr kritisch ist. Ein weiteres Problem für die Gesamtlaufzeit ist die Komprimierung selbst, weil die Paketierzeit mit sinkender Bitrate zunimmt. Eine Lösung wäre es, die Leistungsfähigkeit der Netzwerke zu steigern. Über solche Netze können auch größere Datenmengen in Echtzeit übertragen werden. Die Komprimierung kann dann geringer ausfallen. Dadurch wäre der Aufwand im Kodierer geringer und die Paketierung erfolgte schneller. Ein Beispiel für ein solches Netzwerk wäre ein Netz auf Basis von ATM[10]. ATM bietet Gesamtbitraten von 2,45 GBit/s und es wird an noch höheren Bitraten gearbeitet. Aber auch die Integration von ATM im Backbone von z.B. traditionellen Ethernet-LANs würde die Leistungsfähigkeit des Netzwerks erhöhen.

Abb. 5.11
Sprachqualität und Kompression der Bitrate



Neben den Kodierverfahren sind weitere Störeffekte zu beachten, die die Sprachqualität entscheidend beeinflussen können. Folgende Störeffekte werden hier beschrieben, die nicht nur bei Sprachübertragung auftreten, sondern bei jeder Form der Datenübertragung:

- Bitfehler
- Jitter
- Echo

10 Asynchronous Transfer Mode

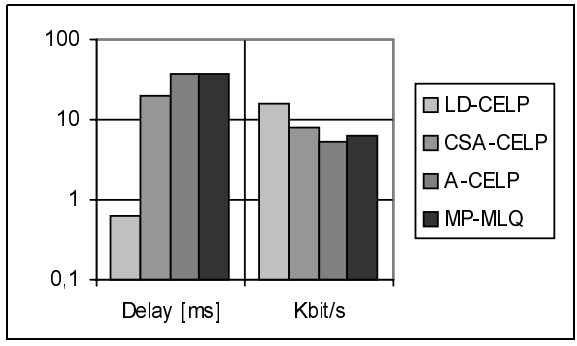


Abb. 5.12 Algebrahischer Delay und Bitrate einzelner Kodierverfahren

5.3.1 Bitfehler

Bitfehler sind durch Störeinflüsse umgewandelte Bits. Das heißt, aus einer beim Sender abgeschickten digitalen „1" kann auf der Übertragungsstrecke durch Störeinflüsse eine digitale „0" werden. Der Qualitätsmaßstab bei digitaler Übertragung ist die Bitfehlerquote oder Bit Error Rate (BER). Die BER gibt an, wie viele Bits einer übertragenen Bitmenge durch Störeinflüsse verfälscht wurden[11]:

$$BER = \frac{Bitfehler}{Bitmenge} = \frac{1bit}{128 \cdot 10^3 bit} = 7{,}8 \cdot 10^{-6} \approx 10^{-5}$$

Mit einer Bitrate von 64 Kbit/s ergibt sich für eine BER von 10^{-5}, dass alle zwei Sekunden ein Bit falsch übertragen wird. Es muss aber sichergestellt werden können, dass die in den Standards festgelegten Rahmenbedingungen und Störeinflüsse den zulässigen Wert nicht überschreiten. Bei Echtzeitanwendungen verschlechtert sich durch einen zu großen BER-Wert die subjektive Qualität des Dienstes.

Tab. 5.4 Subjektive Bewertung von PCM-Sprachübertragung

BER	Subjektive Wirkung
10^{-6}	Nicht wahrnehmbar
10^{-5}	Einzelne Knackgeräusche, die bei niedrigem Sprachpegel leicht wahrnehmbar sind.
10^{-4}	Einzelne Knackgeräusche, die bei niedrigem Sprachpegel störend wirken.
10^{-3}	Eine dichte Folge von Knackgeräuschen, die man bei jedem Signalpegel wahrnehmen kann.
10^{-2}	Starkes Prasseln, welches die Verständlichkeit merklich verringert.
$5 \bullet 10^{-2}$	Fast unverständlich

11 Beispiel: Bei 128 Kbit/s ist 1Bit falsch übertragen worden. Daraus würde sich eine BER von $7{,}8 \bullet 10^{-6}$ ergeben.

Durch die Einführung digitaler Dienste hat die BER-Messung an Bedeutung gewonnen. Die Messung erfolgt, indem ein geeignetes Bitmuster übertragen wird und dieses beim Empfänger (Ende-zu-Ende-Messung) oder wieder beim Sender (Schleifenmessung) mit dem Originalmuster verglichen wird. Das Bitmuster ist meist eine Pseudozufallsfolge. Diese entsteht durch ein rückgekoppeltes Schieberegister. Die Verteilung der Einsen und Nullen geschieht nach mathematischen Gesetzen, wodurch Rückschlüsse möglich sind.

Viele der entwickelten Kodierverfahren reagieren sehr empfindlich auf Kanalstörungen wie Bitfehler oder Paketverlust. Durch die Bitfehler werden einzelne Übertragungswerte gestört. Bei PCM, welches gegen Bitfehler sehr empfindlichen ist, sinkt bei einer Bitfehlerrate von 10^{-3} der MOS-Wert von 4,3 auf 2,2! Die ITU-T gibt die subjektive Wirkung für Sprachübertragung mittels PCM durch die BER an (siehe Tab. 5.4).

Generell sollten die BER-Werte laut ITU-T-Empfehlung G.713 immer kleiner als 10^{-6} sein, damit Sprachdienste nicht gestört werden. Um Bitfehler so gering wie möglich zu halten, könnte man verstärkt Glasfaserkabel einsetzen. Denn durch Glasfaser wird nicht nur die Geschwindigkeit der Datenübertragung erhöht, sondern auch die Bitfehlerrate verringert. Dies wird möglich, weil Glasfaser unempfindlich gegenüber elektromagnetischen Störeinflüssen ist. Allerdings darf der ökonomische Gesichtspunkt bei der Abwägung für oder gegen Glasfaser nicht außer Acht gelassen werden. Es würde beispielsweise keinen Sinn machen, in einem IEEE 802.3-LAN die Rechner mit Glasfaser zu verbinden, da sich durch Twisted-Pair-Kabel für kurze Entfernungen nur geringe Störungen ergeben, die sich nicht störend auswirken können.

5.3.2 Jitter

Der Jitter ist eines der entscheidendsten Probleme der Digitalenübertragung. Die ITU-T definiert Jitter so: *Jitter sind Kurzzeitabweichungen der Kennzeitpunkte*[12] *von Digitalsignalen gegenüber ihren idealen (äquidistanten) zeitlichen Positionen*. Die Einheit der Jittermessung ist Unit Interval (UI). Ein UI entspricht der Abweichung vom Sollzeitpunkt für die Dauer eines Bits. Die Abweichung der Flanken von ihren Kennzeitpunkten kann sich dabei ändern. Verläuft die Änderung periodisch, so wird sie als Jitterfrequenz bezeichnet. [HEIL92]

Die Entstehung von Jitter kann systemabhängig sein (Eigenjitter), was durch die Güte der frequenzbestimmenden Bauteile, Auswirkungen von Störungen der Versorgungsspannung und Einschwingen von Übertragern verursacht wird. Dazu kommen die systemunabhängigen Jitter (Grundjitter), die durch Impulsstörungen, Rauschen und Nebensprechen verursacht werden. Bei langen Regeneratorketten im Übertragungsweg wächst die Jitteramplitude J_A

12 Die Zeitpunkte, zu denen im Originalsignal die steigenden/fallenden Flanken liegen.

beim Grundjitter und Eigenjitter auf folgende Werte an, woraus ersichtlich wird, dass der Eigenjitter in der Praxis dominiert: [HERT94]

Grundjitter: $J_A \sim \sqrt[4]{N}$ Eigenjitter: $J_A \sim \sqrt{N}$

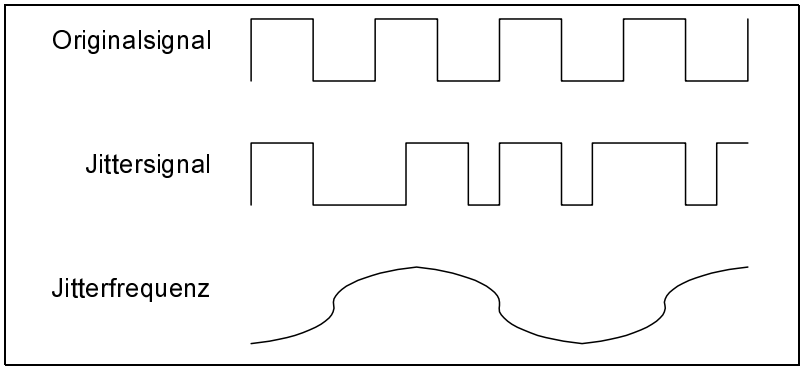


Abb. 5.13 Jittersignal und –frequenz

Eine äquidistante Abtastung im Empfänger sollte somit möglichst die Bitmitte abtasten, dort, wo das Fenster auch bei verschliffenen Signalen am weitesten offen ist, damit es zu keinen bzw. nur geringen Störungen kommt. Der Jitter ist eine Verschiebung der Signalflanken. Ist eine Flanke über den Abtastzeitpunkt verschoben, so findet eine Fehlabtastung statt.

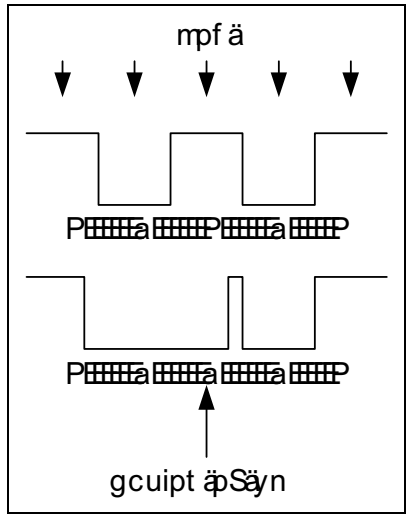


Abb. 5.14 Fehlabtastung durch Jitter

Schaltungen zur Taktrückgewinnung lassen sich bei zu großem Jitter nicht mehr anwenden. Dabei wird die Grenzgröße eines Jitters durch die Jitterverträglichkeit bestimmt. Sie stellt die Kombination aus Jitterfrequenz und Jit-

teramplitude dar, die ein System noch verträgt, um eine einwandfreie Funktion gewährleisten zu können. In der Praxis ist der Übergang zwischen dem Bereich, in dem ein Übertragungssystem Jitter verträgt, und dem Bereich, in dem die Jitterverträglichkeit überschritten wird, sehr schmal. Die Grenzen der Jitterverträglichkeit wird erreicht, wenn die aufgrund von Jitter verursachten Bitfehler ein bestimmtes Maß erreichen, wie beispielsweise eine Fehlerhäufigkeit von 10^{-6}. Daher wird die Jitterverträglichkeit eines Systems ermittelt und regelmäßig überprüft, ob der Jitter in den vertretbaren Grenzen bleibt. In der Literatur wird oft auch in einem anderen Zusammenhang von Jitter (Flattern) gesprochen. Die Abweichung der Verzögerung von einem Mittelwert wird hier auch als Jitter oder stochastische Laufzeit bezeichnet. [HEIL92]

5.3.3 Echoeffekt

Echoeffekte entstehen besondere in analogen Systemen am Übergang von der 4-Draht-Schaltung im Netz zur 2-Draht-Schaltung beim Teilnehmer, da diese Kopplungseinheiten keine ideale Dämpfung aufweisen. Auch akustische Rückkopplungen zwischen Hör- und Sprechkreis führen zu Echoeffekten. Die Störwirkung des Echos auf die Verständlichkeit der Sprache ist durch das Produkt aus der Echoverzögerung (Laufzeit des Echosignals) und der Lautstärke des Echos im Vergleich zum Originalsignal bestimmt. Bedingt durch die Physiologie der menschlichen Hörwahrnehmung lässt sich der Störeffekt nur subjektiv quantifizieren. Echoströme bewirken, dass die sprechende Person ihre eigenen Worte kurz nachschallen hört. Wie psychologische Studien ergeben haben, wird dieser Effekt als sehr störend empfunden und ruft im schlimmsten Fall Stottern und Verwirrung hervor. Der Echoeffekt tritt nur bei Signalverzögerungen von größer als 25 ms auf. Denn erst dann reichen die Sprachpausen, die durch die Verzögerung entstehen, aus, um ein Echo wahrnehmen zu können.

Früher wurden Echosperren eingesetzt, um das Problem zu beseitigen. Diese schalten die Übertragungsrichtung durch Sprachsignalerkennung in Richtung des Sprechens um. Es war also nur Halbduplex-Betrieb möglich. Weil die Echosperre nur für Sprachsignale, nicht aber für digitale Daten ausgelegt sind und die Umschaltung relativ lange dauert (ca. 5 ms) ist sie nicht für die moderne Sprach-Datenintegration geeignet. Eine Alternative zu den Echosperren stellen die Echoneutralisierer dar. Diese simulieren das erwartete Echo und subtrahieren die Simulation von dem Echo, um dann das Echo zu kompensieren. Durch den Einsatz von Echoneutralisierern ist nun auch Vollduplex-Betrieb möglich.

5.4 Paketorientierte Sprachübertragung

Die Sprachübertragung ist, seitdem die Telefonie eingeführt wurde, immer durch leitungsvermittelte Netze, also verbindungsorientiert, realisiert worden. Für ein Telefongespräch wird deshalb über spezifische Leitungen zwischen zwei Endpunkten eine physikalische Verbindung aufgebaut. Diese durchgeschaltete Leitung steht exklusiv für die Teilnehmer während einer Verbindung zur Verfügung und muss sich die Bandbreite nicht mit anderen Benutzern teilen. Nachdem das Gespräch zwischen den beiden Parteien beendet wurde, wird die Leitung wieder freigegeben.

Im Gegensatz dazu haben sich die paketvermittelten Netze für die reine Datenübertragung gebildet. Sie sind verbindungslos aufgebaut, verschicken Datenpakete zur Kommunikation und müssen unterschiedliche Datenmengen meistern. Die entstehenden Schwankungen im Verkehrsaufkommen, durch die sich unterschiedliche Belastungen ergeben, werden mit verschiedenen Signalarten (z.B. Daten und Sprache) und Teilnehmern kombiniert, um die Übertragungsbandbreiten und Vermittlungskapazitäten dynamisch zwischen diesen aufzuteilen. Die Datenpakete benötigen auf dem Weg durch das Netz einen Protokollkopf (Header), der u.a. die Zieladresse enthält. Dabei werden sie von Routern zwischen unterschiedlichen Netzen weitergeleitet. Router setzen zur Erkennung der optimalen Wegwahl Algorithmen ein. Erhöhte Belastungen des Routers werden durch Warteschlangen (Waiting Queue) ausgeglichen, indem die Pakete erst zwischengespeichert und nach der Entlastung des Routers weitergeleitet werden. Diese flexible heterogene Struktur beinhaltet aber höhere Verzögerungszeiten und schwankende Ankunftszeiten der Pakete. Dabei besitzen die Paketlaufzeiten, Paketverluste und Jitter einen erheblichen Einfluss auf die Übertragungsqualität eines Sprachsignals.

Die Übermittlung von Sprache auf Basis von Paketen gewinnt zunehmend an Bedeutung. Dabei spielt die Auslastung der Übertragungssysteme eine wichtige Rolle. Bei Sprachübertragung werden im Mittel beide Übertragungsrichtungen aufgrund der Sprachpausen nur zu 40% ausgelastet. Die Pausen können durch die paketorientierte Übertragung leicht für andere Dienste genutzt werden. Eine Möglichkeit, die Pausen für andere Dienste zu nutzen, ist das Verfahren der Sprachpausenunterdrückung[13].

5.4.1 Paketvermittlung

Als Erstes wird die Sprache von einem Sprachpaketsender (SPS) bearbeitet. In diesem wird mit Hilfe des Kodieres die Sprache mit den entsprechenden Verfahren digitalisiert und gegebenenfalls komprimiert. Anschließend unterteilt der Sprachdetektor die Sprache in Pausen- und Sprachblöcke. Als letzte Station

13 Silence Suppression

im SPS werden die ankommenden Sprachblöcke in Pakete aufgeteilt. Dazu werden ankommenden Sprachblöcke in die definierten Paketlängen abgetrennt und mit der Kopfinformation (Header) versehen. Als Pakete verpackt, werden die Sprachinformationen in das Netz gegeben. Dort werden sie zusammen mit Paketen anderer Dienste in Warteschlangen eingegliedert, bevor sie weitervermittelt werden.

Nach der Übertragung über das Netz kommen die Sprachpakete beim Sprachpaketempfänger (SPE) an. Im Depaketierer werden die Pakete um eine gewisse Zeit verzögert (Puffer), um Laufzeitschwankungen soweit wie möglich ausgleichen zu können. Die wiedergewonnenen Sprachblöcke können damit fast lückenfrei an den Dekodierer übergeben werden. Der Dekodierer wandelt die digitale Bitfolge in analoge Sprache zurück.

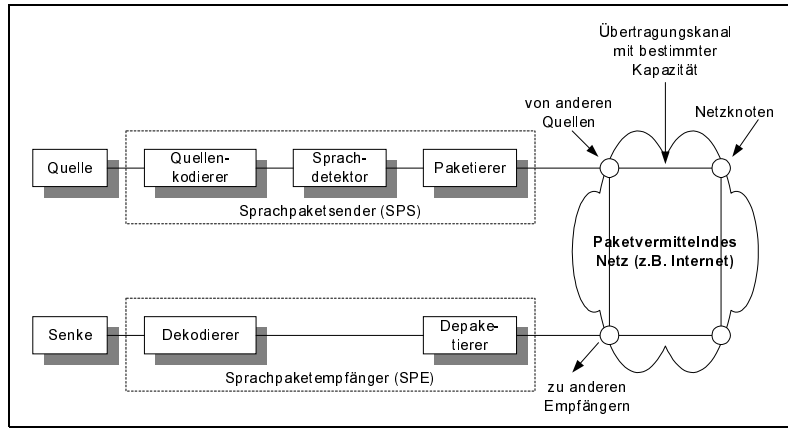


Abb. 5.15 Sprachübertragungssystem mit Paketübermittlung

5.4.2 Sprachpausenunterdrückung

Die Sprachpausenunterdrückung vermeidet, dass Audiodaten übertragen werden, obwohl einer der Gesprächsteilnehmer gar nichts von sich gibt. Die Telefonkommunikation erfolgt dabei im Halbduplex-Betrieb. Während eine Seite spricht, hört die andere zu und umgekehrt. Der Fall, dass beide Seiten gleichzeitig reden, tritt eher selten ein. Dennoch unterstützt das traditionelle Telefon den Vollduplex-Betrieb, überträgt also ständig Audiodaten in beide Richtungen. Indem man die Pausenzeiten auf einer Seite durch einen Standardwert kodiert, lässt sich die benötigte Bandbreite für die Telefonverbindung um ca. 40% senken. Dafür muss das System in der Lage sein, auf neue Geräusche, die ein Abweichen vom Ruhezustand darstellen, sofort zu reagieren. In Sprachpausen wird ein Rauschen als Hintergrundgeräusch eingespielt, welches eine permanente Verbindung vortäuscht[14]. Wenn gesprochen wird, hat die Silence Suppression keine Auswirkung auf die Sprachqualität. Die Sprachpausenunterdrückung kann zusätzlich zur Datenkompression eingesetzt werden.

Silence Suppression kann die Kapazität einer Datenleitung für Sprachübertragung verdoppeln, ohne allerdings die Anzahl der parallel übertragbaren Gespräche zu erhöhen. Dies liegt daran, dass der Vollduplex-Betrieb unterstützt werden muss, wenn beide Teilnehmer gleichzeitig sprechen. Auch wenn dieser selten auftritt, muss dennoch die notwendige Bandbreite bzw. Qualität zur Verfügung stehen. In der Praxis kann man von der gewonnenen Bandbreite profitieren, indem diese zur Übertragung nicht zeitkritischer Daten genutzt wird.

Nachteilig ist zu bemerken, dass je nach Stärke der Hintergrundgeräusche das Einsetzen der Silence Suppression zu hören ist. Das Aussetzen der Hintergrundgeräusche vermittelt den Eindruck einer „toten" Leitung. Weiterhin ist die Datenreduzierung abhängig von dem Gesprächsverhalten der Teilnehmer. Hinzu kommt, dass das Management schwieriger wird, da eine Datenreserve vorgesehen werden muss.

5.4.3 Laufzeiten

Die Beeinflussung der Sprachqualität ist in den vorangegangenen Kapitel bereits bezüglich Quantisierungsverzerrung durch den Kodierer, Bitfehler auf der Übertragungsstrecke usw. berücksichtigt worden. Allerdings stellen die größten Probleme bei paketorientierter Sprachübertragung die Laufzeiten dar. Denn nur bei konstanter Laufzeit sind kontinuierliche Übertragungen, wie sie für Sprache gegeben sein müssen, einzuhalten. Zusätzlich darf die konstante Laufzeit ein bestimmtes Maximum nicht überschreiten. Die Laufzeiten (Delay) müssen für die Betrachtung in einen konstanten und einen stochastischen Teil unterschieden werden. Stochastisch bedeutet, dass die Dauer, die Höhe und die Häufigkeit der Abweichung von der konstanten Laufzeit von zufälligen Ereignissen abhängt. Die stochastische Laufzeit wird dabei ebenfalls als Jitter bezeichnet.

Die konstanten Laufzeiten besitzen den großen Vorteil, dass sie vorhersehbar sind. Sie ergeben sich aus der physikalischen Laufzeit der Pakete durch das Netz und den Verarbeitungszeiten in den Netzknoten (einschließlich SPS und SPE). Die stochastischen Laufzeiten ergeben sich vor allem durch die Warteschlangen in den Netzknoten. Diese können je nach Netzauslastung unterschiedlich schnell abgearbeitet werden. Der Mittelwert der stochastischen Laufzeit ist als weiterer Anteil der konstanten Laufzeit zu sehen.

Durch den Kodierer entstehen typische Verzögerungen von 8-32 ms. Für Kodierer, die beispielsweise auf der LPC-Analyse beruhen, ergeben sich diese Verzögerungen aufgrund der Sprachsegmente. Diese sind 10-30 ms lang. Der Sprachdetektor arbeitet mit Verzögerungen, um den Sprachblockbeginn festlegen zu können. Es ergeben sich daraus typische Laufzeiten von 4-32 ms. Die

14 Dies wird sowohl in ISDN- als auch VoIP-Netzen gemacht, damit der Teilnehmer erkennt, wann die Verbindung vorhanden ist und wann nicht.

Zeit des Paketierers hängt von der Paketlänge[15] und den Kodierraten (wie z.B. 64 Kbit/s[16], 16Kbit/s[17]) des gewählten Verfahrens ab. Die typischen Laufzeiten liegen hier zwischen 2 und 128 ms. Berechnet werden die Laufzeiten eines Pakets durch den Paketierer *P* und aus dem Verhältnis der gesamten Paketlänge in Bits *B* zur Quellenkodierrate *R*. Unter Berücksichtigung der Paketköpfe *H* ist die Paketlaufzeit im Sender:

$$P = \frac{B - H}{R}$$

Der Paketnutzungsgrad *p* gibt Aufschluss über das Verhältnis der Nutzdaten zur Gesamtpaketlänge:

$$p = \frac{B - H}{B}$$

Durch gut konzipierte Protokolle können die drei Funktionen (Kodierer, Detektor und Paketierer) parallel auftreten. Deshalb summieren sich die drei Laufzeiten nicht. In der Regel wird nur die größte der drei Laufzeiten nach außen sichtbar.

Im Netzwerk tritt die physikalische Laufzeit auf, die von der zu überbrückenden Entfernung und dem benutzten Medium abhängt. In LANs ist diese Zeit zu vernachlässigen. Die Laufzeit auf terrestrischen Verbindungen liegt zwischen 20 und 40 ms. Für Satellitenverbindungen ergeben sich Zeiten von etwa 250 ms. Weitere Laufzeiten verursachen die im Netzwerk befindlichen Netzknoten. Diese benötigen Zeit zum Verarbeiten der Pakete (weiterleiten, vermitteln usw.). Durch die Warteschlangen, in der die Pakete eingereiht werden, entsteht eine Verzögerung, die sehr stark von der Auslastung des Netzwerkes abhängig ist. Die Zeit[18] *K*, bis ein Paket zur Weiterverarbeitung in den Netzknoten zur Verfügung steht, errechnet sich aus der verfügbaren Kanalkapazität *C* und der Paketlänge *B*:

$$K = \frac{B}{C}$$

Die Kanalkapazität muss größer sein als die Quellkodierrate. Denn im Netz müssen auch die Paketköpfe mit übertragen werden, die die Paketierzeit in den Netzknoten verlängern. Die für einen reibungslosen Datenfluss mindestens

15 Auf die Wahl der Paketlängen wird an späterer Stelle noch eingegangen.
16 Pulse Code Modulation (PCM)
17 Low Delay Codebook Excited Linear Predictive Coding (LD-CELP)
18 Paketlaufzeit der Netzknoten

bereitzustellende Kanalkapazität C_{min} berechnet sich aus der Paketlänge B und Paketierzeit Q im Sender:

$$C_{min} = \frac{B}{Q}$$

In LANs, die nach dem Kollisionsverfahren arbeiten (z.B. Ethernet), ergeben sich ebenfalls zusätzlich Laufzeiten, die stark von der Netzbelastung beeinflusst werden. Anzumerken ist, dass bei der Sprachübertragung keine Fehlerkorrekturmechanismen eingesetzt werden, die auf Paketwiederholung beruhen. Denn durch die Neuanforderung eines fehlerhaften Pakets würde man den Sprachfluss unterbrechen. Diese Verzögerungen treten bei Sprachübertragung nicht auf. Im Gegensatz dazu werden die Laufzeiten im Sprachpaketempfänger durch Zwischenspeicherung der Pakete ausgeglichen. Das verursacht den Hauptanteil an Verzögerung im SPE.

5.4.4 Laufzeitauswirkungen

Die Laufzeit kann die Qualität der Sprachübertragung maßgeblich beeinflussen. Dabei nehmen die beiden Anteile der Laufzeit (stochastisch und konstant) unterschiedlich Einfluss auf die Sprachübertragungsqualität. Beide Anteile der Laufzeit sind nicht zu vermeiden, sie können aber durch ein geeignetes Konzept auf ein Minimum reduziert werden.

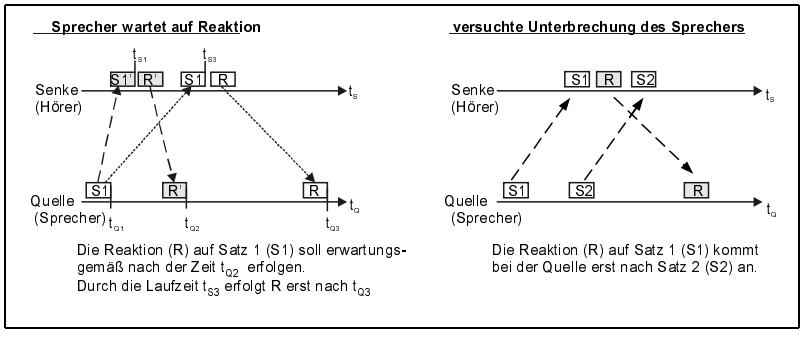


Abb. 5.16 Gesprächsqualitätsbeeinflussung durch lange Laufzeiten

Die konstante Laufzeit hat keinen Einfluss auf die Sprachqualität, sondern verschlechtert mit zunehmender Länge immer stärker die Gesprächsqualität. Die Gesprächsqualität soll an zwei Beispielen verdeutlicht werden. Ein Sprecher wartet länger als erwartet auf eine Reaktion von seinem Zuhörer. Dadurch bekommt er das Gefühl, sein Gegenüber hört nicht aufmerksam zu. Andersherum versucht ein Zuhörer während einer Pause zu reagieren, obwohl der Folgesatz vom Sprecher schon übermittelt wird. Ein Dialog wird durch lange Laufzeiten zunehmend unmöglich gemacht.

Die Länge der Laufzeiten über das Netz (Delays) ist von der Betriebsart abhängig. Wenn das Übertragungssystem nur in eine Richtung sendet und sein Gegenüber nur empfängt (Simplex-Betrieb), wäre die Verzögerung irrelevant, weil es nur ausschlaggebend ist[19], dass die Daten in richtiger Reihenfolge und vor allem kontinuierlich ankommen. Da für Sprachübermittlung wechselseitige Kommunikation über zwei Verbindungen (Vollduplex-Betrieb) oder wenigstens über eine Verbindung (Halbduplex-Betrieb) möglich sein muss, würden lange Verzögerungen unangenehme Sprachpausen erzeugen. Für Vollduplex-Betrieb ist daher eine maximale Verzögerung von 200 ms und für Halbduplex-Betrieb zwischen 200 ms und 500 ms erlaubt. Die ITU-T hat diese Zeiten ebenfalls in G.114[20] definiert und bis zu 250 ms als hohe, bis zu 400 ms als mittlere und bis zu 600 ms als noch tolerierbare Qualität eingestuft.

Ein weiteres Problem von langen Laufzeiten sind die Echos. Die Echolaufzeit entspricht der Laufzeit der Pakete. Je länger die Laufzeit ist, desto größer wird der Echoeffekt. Mit zunehmendem Echoeffekt erhöht sich auch der negative Einfluss auf die Gesprächsqualität. Denn der Sprecher wird zunehmend von dem Echo irritiert. Allerdings kann der Einfluss auf die Gesprächsqualität durch entsprechend hohe Dämpfung des Echosignals verringert werden. So muss bei einer Laufzeit von 50 ms das Echo um 30 dB, bei einer Laufzeit von 30 ms nur noch um 22 dB ab gesenkt werden, um nicht störend zu wirken. Die Dämpfung wird durch eine Echokompensation erreicht.

Subjektive Untersuchungen sollten klarstellen, ab wann die konstanten Laufzeiten Einfluss auf die Gesprächsqualität haben. Bei diesen Untersuchungen wurden geübte Sprecher angewiesen, viele Rückfragen zu stellen und bewusst Pausen zu machen. Durch die Untersuchung kann festgehalten werden, dass konstante Laufzeiten von bis zu 300 ms in einer Richtung keinen Einfluss auf die Gesprächsqualität haben, wenn sie keine Echos aufweisen. Deutliche Beeinträchtigungen ergeben sich zwischen 500 ms und 600 ms und Werte über 900 ms sind absolut indiskutabel.[21] Bei echobehafteter Übertragung reduzieren sich die akzeptablen Laufzeiten sehr stark. Werte von 50-80 ms beeinflussen die Gesprächsqualität in erheblichem Maße. Satellitenverbindungen stellen im Zusammenhang mit Laufzeiten ebenfalls ein großes Problem da. Hier empfiehlt der ITU-T-Standard G.114 Verbindungen über zwei Satelliten zu vermeiden.

Gerade bei Sprachübertragungen ist es wichtig, dass die Verzögerung nahezu konstant ist, da das Ohr sehr empfindlich auf Klangschwankungen oder Sprachunterbrechungen reagiert, auch wenn diese nur wenige Millisekunden andauern. Dadurch würde es zu einer erhebliche Verschlechterung der Sprach-

19 Bezogen auf die Zeit
20 Recommendation G.114 (02/96) – One-way transmission time
21 ITU-T G.114

qualität kommen. Durch stochastische Laufzeiten ergeben sich Abweichungen von der konstanten Laufzeit für jedes einzelne Paket. Wenn die Pakete später als vorhersehbar eintreffen, entstehen Lücken im Sprachfluss. Um die Gesamtlaufzeit nicht unnötig zu verlängern, ist es sinnvoll, auch das Verwerfen von Paketen[22] zuzulassen. Aber auch durch zu früh eintreffende Pakete kommt es ohne Gegenmaßnahme zu Paketverlusten. Die subjektiven Auswirkungen sind abhängig von:

- der Paketlänge *PL*
- dem Grad der statistischen Abhängigkeit der Paketverluste und der Paketverlustrate bzw. von der Häufigkeit der Paketverluste

Die Paketverlusthäufigkeit *PVH* wird über einen bestimmten Zeitraum festgehalten. Die Paketverlustrate *PVR* gibt Auskunft darüber, wie viele Pakete von einer bestimmten Menge verloren gehen. Bei einer Sprachhäufigkeit von *A* berechnet sich *PVH* nach folgender Gleichung:

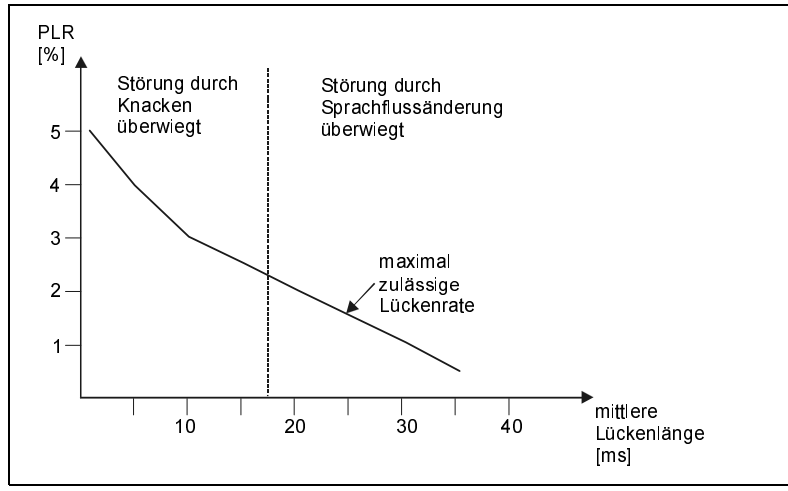
$$PVH = \frac{PVR \cdot A}{PL}$$

Die *PVR* kann durch geeignete Maßnahmen zwar nicht auf null gebracht, aber auf ein Minimum beschränkt werden. Da die *PVH* von zufälligen Ereignissen abhängt und somit nicht direkt beeinflusst werden kann, kann sie nur noch von der Paketlänge gesenkt werden. Kurze Paketlängen von weniger als 16 ms machen sich bereits mittels Knackgeräuschen bemerkbar. Bei langen Paketlängen von über 64 ms kommt es zu Silben- oder sogar Wortverlusten, die zu Beeinträchtigung der Sprachverständlichkeit führen. Die Gleichung zeigt auf, dass größer werdende Paketlängen die *PVH* verringern würden. Dies würde bedeuten, dass durch die schon geringe *PVR* selten, aber wenn viele Information verloren gingen. Zu klein dürfen die Pakete auch nicht sein, weil die *PVH* dann zunehmend größer wird. Es muss hier also ein Kompromiss gefunden werden, um einen möglichst guten, wenn auch subjektiven Eindruck der übertragenen Sprache zu bekommen. Untersuchungen, die nach dem MOS-Verfahren durchgeführt wurden, haben gezeigt, dass bei einer *PVR* von bereits 0,5[23]-1%[24] eine gute Sprachqualität um einen MOS-Wert von 4,0 erreicht werden kann. Eine *PVR* von 0,2% ist als erreichbarer Grenzwert zu sehen. Mit dieser Paketverlustrate und Paketlängen von 10-16 ms sind gute Sprachqualitäten gerade noch erreichbar. Durch Paketlängen von mehr als 50 ms Länge kommt es durch Paketverluste zu einer erheblichen Beeinflussung der Sprachverständlichkeit.

22 Packet Loss; Paketverlust
23 Bei 8 ms Paketlänge
24 Bei 5 ms Paketlänge

Die Zusammenhänge bei den Lücken sind ähnlich denen der Paketverluste. Der Unterschied besteht nur darin, dass bei der Lücke keine Informationen verloren gehen, sondern der Sprachfluss verzögert wird. Die subjektiven Auswirkungen hängen von der mittleren Lückenlänge[25], der Lückenrate[26] und der Häufigkeit auftretender Lücken in einer bestimmten Zeit[27] ab. Wenn die Paketlänge durch die mittlere Lückendauer ersetzt wird, ist der Zusammenhang der Abhängigkeiten der gleiche wie bei den Paketverlusten. Bei kurzen Lücken äußert sich die subjektive Wirkung als Knackgeräusch. Bei längeren Lücken ab ca. 17,5 ms ist die Wirkung in einer Sprachflussänderung zu bemerken.

Abb. 5.17 Maximale mittlere Lückenlänge für ausreichende Sprachqualität



Durch Hörversuche wurden die in der Abb. 5.17 dargestellten Lückenlängen und die dazugehörige Lückenrate ermittelt, mit denen eine befriedigende Sprachqualität erreicht werden kann. Dabei stellen, wie bereits erwähnt, die Lücken den Zeitraum dar, dessen Verzögerung notwendig ist, um ein Paket noch in den Sprachfluss einbauen zu können. Also wäre eine Lückenzeit von Null im Grunde optimal. Allerdings würden dann Pakete bei Verspätungen sofort verworfen und die Informationen gingen verloren. Demnach muss auch hier der subjektive Höreindruck entscheiden, wann eine Lücke besser als ein Paketverlust ist. Als Ergebnis der Hörversuche kann festgehalten werden, dass die mittlere Lückenlänge die Zeit von 17,5 ms nicht überschreiten darf. Denn bis zu diesen Längen treten vorrangig Knackgeräusche auf, die subjektiv weniger Bedeutung haben als Sprachflussänderungen.

25 Vergleichbar mit der Paketlänge
26 Vergleichbar mit der Paketverlustrate
27 Vergleichbar mit der Paketverlusthäufigkeit

5.4.5 Maßnahmen gegen Laufzeitauswirkungen

Kurze Paketlängen verringern die Paketierzeit, wie Abb. 5.18 zeigt. Außerdem wird die Wirkung bei Paketverlusten auf den Höreindruck verringert. So zeigte sich auch durch die Hörversuche, dass Paketlängen zwischen 8-16 ms am wenigsten auffallen. Hingegen verschlechtern wiederum zu kleine Paketlängen den Paketnutzungsgrad und erhöhen die Paketraten. Durch die höhere Paketrate müssen die Netzknoten pro Zeiteinheit mehr Pakete vermitteln[28]. Durch einen solchen Burst-Verkehr kann die stochastische Laufzeit beeinflusst werden, weil kurzzeitig die Netzlast stark ansteigt. Der Paketnutzungsgrad verschlechtert sich, weil die Paketköpfe eine bestimmte Länge haben müssen, da nur die Menge der zu übertragenen Nutzdaten pro Paket kleiner wird, während der Header die gleiche Größe beibehält. Also würde bei einem schlechten Paketnutzungsgrad eine höhere Übertragungsrate für den Overhead[29] benötigt als für die Nutzdaten.

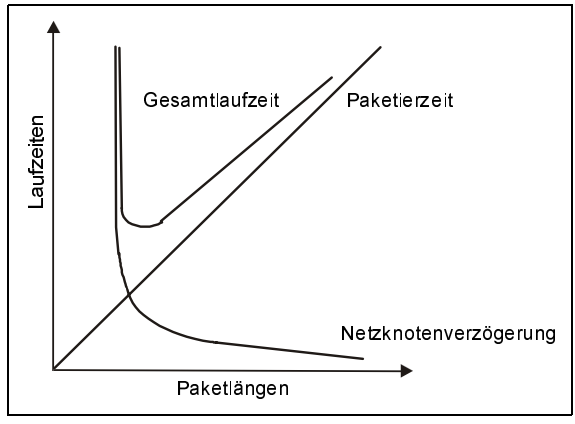


Abb. 5.18 Einfluss der Paketlängen auf unterschiedliche Laufzeiten

Bei einer Untersuchung für reine Sprachübertragung wurde die Bitzahl der Pakete soweit verringert, bis eine maximale mittlere Gesamtlaufzeit eingehalten wurde. Bei dieser maximalen Gesamtlaufzeit werden trotz eines gering auftretenden Burst-Verkehrs gute Gesprächswiedergaben erzielt. Die verschiedenen Paketlängen wirkten sich unterschiedlich auf die Paketierzeit, die Netzknotenverzögerung und die Gesamtlaufzeit aus. In den Netzknoten kommt es bei kleinen Paketen zu einem großen Paketaufkommen, weil zu den Nutzdaten immer mehr Overhead transportiert werden muss. Bei langen Paketen kehrt sich dieser Effekt um. Die Paketierzeit nimmt natürlich mit zunehmend langen Paketen linear zu, weil länger auf die entsprechende Anzahl von Bits gewartet werden muss. Die Gesamtlaufzeit nähert sich nun jeweils dem größeren Wert an. Paket-

28 Burst-artiges Verhalten
29 Paketkopf

längen von 300 Bit (ca. 38 Byte) bis 700 Bit (ca. 87 Byte) liefern optimale Ergebnisse für die Gesamtlaufzeit. Denn bei dieser Bitanzahl kann ein Minimum der Paketierzeit, unter Berücksichtigung kleiner Netzknotenverzögerungen durch weniger Paketaufkommen, erreicht werden.

5.4.6 Verringerung der stochastischen Laufzeitauswirkung

Die stochastischen Laufzeiten bzw. Jitter haben durch Lücken und Paketverlust Auswirkungen auf die Sprachqualität. Der Jitter sollte maximal ±0,5 ms um eine mittlere Übertragungszeit schwanken. Um die Auswirkungen so klein wie möglich zu halten, sollten folgende Punkte zur Kompensation eingeführt werden:

- Pufferung
- Flusssteuerung
- Wahl des Vermittlungssystems
- Kanalauslastung
- Zusammenfassung der Sprach- und Datenströme

Pufferspeicher stehen an erster Stelle der Gegenmaßnahmen, die in den Empfängern eingesetzt werden, um die auftretenden Laufzeitschwankungen auszugleichen. In den Puffern werden die Pakete gespeichert, sodass der Beginn der Sprachwiedergabe verzögert werden kann. Dabei muss beachtet werden, dass die Gesamtlaufzeit die Gesprächsqualität nicht beeinflusst.

Bei der Flusssteuerung gibt es, bei hoher Auslastung des Paketvermittlungsnetzes, die Möglichkeit, keine neuen virtuellen Verbindungen mehr zuzulassen und diese erst zu einem späteren Zeitpunkt aufzubauen. Es könnte auch eine Verbindung mit kleinerer Kanalkapazität angeboten werden. Eine weitere Möglichkeit wäre, durch Priorisierung der Pakete die Flusssteuerung einzusetzen. Dabei erhalten die Pakete entsprechend ihrer Priorität verschiedene Kennzeichnungen. Durch Vergabe der höchsten Priorität für Sprachpakete werden sie in den Warteschlangen der Netzknoten an die jeweils erste Abarbeitungsstelle geleitet, um so schneller weitertransportiert werden zu können.

Die Pakete können mit einem Paketdienst (z.B. IP) vermittelt werden. Sie enthalten im Paketkopf die vollständige Sender- und Empfängeradressen und werden als unabhängige Nachrichten in das Netzwerk geschickt. Für jedes Paket wird erneut der bestmögliche Weg gesucht. Weil bei Sprache immer wieder kurzzeitig viele Pakete übermittelt werden, wird das Netz kurzzeitig stark belastet (Burst). Die Wahrscheinlichkeit, dass die einzelnen Pakete andere Routen nehmen ist dabei sehr hoch. Dadurch stellen sich auch sehr unterschiedliche Laufzeiten für jedes Paket ein. Das kann dazu führen, dass die Pakete in der falschen Reihenfolge beim Empfänger ankommen. Die Zahl der Paketverluste wird sich vergrößern. Deshalb ist der Paketdienst für Sprachübertragung nicht sehr geeignet. Eine andere Möglichkeit, Pakete zu vermitteln, sind feste virtuelle

Verbindungen über X.25, Frame Relay oder ATM. Dabei wird zu Beginn der Weg ermittelt, über den alle Pakete der Verbindung transportiert werden sollen. Dabei sind die Pakete nicht in der Lage, sich gegenseitig zu überholen. Hinzu kommt, dass sie nicht die komplette Empfänger- und Senderadresse enthalten müssen, sondern nur eine Identifikationsnummer. Dadurch lässt sich der Header (Overhead) verkleinern. Die virtuellen Verbindungen sind daher für Sprachübertragungen sehr viel besser geeignet.

Die Kanalauslastung k ist das Verhältnis in Prozent von der vermittelten Verbindungsrate VR zur Kanalkapazität C:

$$k = \frac{VR}{C}$$

Simulationen mit reinem Sprachverkehr haben gezeigt, dass die Kanalauslastung k bei zunehmender Kanalkapazität C größer werden kann, wenn man gleichzeitig dabei die Laufzeiten auf den gleichen Werten belassen möchte. Bei der Simulation wurden Sprachpakete von 256 Bit mit 16 Kbit/s übertragen. Anschließend sind unterschiedlich viele Verbindungen bei verschiedenen Kanalkapazitäten übertragen worden. Die Datenrate der einzelnen Verbindungen wurden dabei übermittelt. Um eine Laufzeit von 20 ms erreichen zu können, kann eine Kanalkapazität von 48 Kbit/s nur bis zu 55%, eine Kanalkapazität von 1600 Kbit/s aber bereits bis zu 95% ausgelastet werden. Also ist durch geeignete Wahl von k und C die Laufzeit günstig beeinflussbar.

Durch die normale Gesprächsgestaltung[30] kommt es immer wieder zu Burst-artiger Belastung des Netzes. Dieser Effekt wird durch die günstigen Eigenschaften kleiner Paketlängen noch begünstigt. Dadurch kommt es zu starken Abweichungen von der mittleren Laufzeit. Hingegen sind die Belastungen bei reinem Datenverkehr kontinuierlicher und geringer. Durch die gleichzeitige Übermittlung von Sprach- und Datenverkehr, mit einem hohem Datenanteil, wird eine kontinuierliche Sprachübertragung eher ermöglicht. [PAKO99]

5.4.7 Gegenmaßnahmen zu Paketverlusten

Paketverluste lassen sich nicht immer ausschließend. In einem Corporate Network (CN) oder innerhalb eines Local Area Network (LAN) ist dies bei ausreichender Dimensionierung noch einigermaßen möglich, da man die eigene Infrastruktur beherrschen kann. Paketübermittlung über öffentliche Netze wie das Internet bietet heute noch keine Garantien und sie können dementsprechend auch Pakete verlieren. Das Vermittlungsprotokoll IP beinhaltet somit häufiger Paketverluste, die auf den Paketdienst zurückzuführen sind. Trotzdem können Gegenmaßnahmen ergriffen werden: Durch spezielle Korrekturverfah-

30 Eine Person spricht und der andere Teilnehmer hört zu.

ren können die Paketverluste minimiert werden. Ein Möglichkeit wäre das Verfahren Forward Error Correction (FEC). FEC bindet redundante Daten eines Sprachpakets in das nachfolgende ein. Die redundanten Daten erlauben die zufriedenstellende Rekonstruktion eines verspäteten oder verloren gegangenen Pakets, sodass der Benutzer den Verlust nicht bemerkt. Durch FEC können Paketverluste bis zu 4% verkraftet werden.

5.4.8 Fazit

Durch die paketweise Übertragung von Sprache ist die Integration mit anderen Diensten leichter zu erreichen. Auch die Gesprächspausen können anderen Diensten zur Verfügung gestellt werden. Allerdings treten bei paketorientierter Sprachübertragung auch einige Probleme auf. Durch Sprachdialoge sind immer wieder Burst-artige Auslastungen des Netzes vorhanden, welche zu Laufzeitschwankungen führen. Die Laufzeiten sind aber bei paketierter Sprache ein kritischer Faktor, weil die kontinuierliche Sprachübertragung gewährleistet sein muss. Die Verzögerung darf dabei für Sprachübertragung bei Vollduplex-Betrieb 250 ms nicht überschreiten.

Durch Paketlängen zwischen 38 und 87 Byte sind die Verzögerungen in den Netzknoten am geringsten und die Paketierzeit führt zu guten Ergebnissen. Durch zusätzliche Maßnamen wie Puffer, günstige Kanalauslastung, Flusssteuerung etc. kann die stochastische Laufzeit[31] auf einen vertretbaren Wert von maximal +/- 0,5 ms gebracht werden. Dadurch ist die paketorientierte Sprachübertragung ein sehr effizientes Übertragungssystem. Zusätzlich wirkt sich die heute geforderte Integration von Sprache und Daten in ein Netz auch noch positiv auf die Übertragungseigenschaften aus.

5.5 Standards und Protokolle

Um Sprache über Rechnernetze zu transportieren und Voice-over-IP-Anwendungen zu integrieren, werden hohe Anforderungen an eine Netzwerkumgebung gestellt. Die Laufzeiten sind dabei u.a. für eine störungsfreie Echtzeitübertragung entscheidend. Bisherige Netzprotokolle in der IP-Umgebung sind nicht für eine solche Übertragungsart entwickelt worden. Nimmt man das Protokoll TCP[32], so ist anhand der Quittierungsmechanismen nicht an eine Echtzeitübertragung zu denken, da dies bei reiner Sprachübertragung zu lange dauern würde. Hinzu kommt, dass Pakete nicht wiederholt angefordert werden dürfen, da es sonst zu Störungen im Sprachablauf kommen würde. Das Protokoll UDP[33] bietet hingegen Echtzeitfunktionalität an, da es keine Quittierungs-

31 Jitter
32 Transmission Control Protocol
33 User Datagram Protocol

mechanismen kennt. Allerdings sind wiederum nur sehr begrenzte Funktionen implementierbar, da UDP sehr einfach aufgebaut wurde. Es müssen also weitere Protokolle hinzukommen, um IP besser an isochrone Datenströme anpassen zu können. Seit Mitte der neunziger Jahre wurden daher an Echtzeitprotokollen gearbeitet, die die Eigenschaften von Sprachanwendungen besser abbilden können. Nachdem in den vorherigen Kapiteln die Sprachqualität sowie die negativen Effekte dargestellt wurden, wird nun auf die notwendigen Protokolle und Standards im Umfeld von VoIP eingegangen.

5.5.1 Real-Time Transport Protocol (RTP)

Das Real-Time Transport Protocol (RTP) arbeitet auf der Grundlage des Internetprotokolls und hat die Aufgabe, Datenströme in Echtzeit (z.B. Audio oder Video) zu transportieren. In erster Linie wurde RTP für Multicast-Pakete entwickelt, eine Verwendung für Unicast ist jedoch auch möglich. RTP definiert keine Mechanismen für eine garantierte Dienstgüte und für die Verwaltung von Bandbreite. Zusätzliche Kontrollinformationen während des Datenaustauschs liefert das Real-Time Transport Control Protocol (RTCP).

RTP ist in der Arbeitsgruppe Audio-Video Transport der IETF nach den Spezifikationen RFC-1889 und RFC-1890 entwickelt worden. Es ist 1996 entstanden, um Paketfolgen auf Seiten des Empfängers besser synchronisieren zu können, wodurch Jitterbuffer und Sequenzfolgen abgestimmt werden konnten. RTP erlaubt das Versenden mehrerer Datenformate, die durch ein RTP-Profil genau bezeichnet sind. Durch die Formatangabe wird zwischen verschiedenen Medien (Audio oder Video) und dem Kodierungsformat unterschieden, sodass die übertragenen Daten anwendungsunabhängig verwendet werden können. Weiterhin ist RTP weitgehend unabhängig von den darunter liegenden Transportsystemen und kann daher eingesetzt werden, ohne den Kernel des Betriebssystems zu ändern. RTP besitzt zusätzlich RTCP, das Statusinformationen der Teilnehmer überträgt. Die Übertragung der RTCP-Nachrichten ist nicht gesichert, die Teilnehmer übertragen die Statusmeldungen periodisch. Das Überwachungsprotokoll RTCP sorgt für die Kontrolle der Signallaufzeiten und misst die Paketverlustraten. Dadurch kann die Übertragung an die jeweiligen Verbindungseigenschaften angepasst werden.

Funktionalität von RTP

RTP verwendet wie erwähnt das User Datagram Protocol (UDP) für den verbindungslosen und unzuverlässigen Transport von Daten über paketbasierte Netzwerke. UDP eignet sich gut für Multicast und überträgt Daten auf effiziente Weise ohne Rücksicht auf verloren gegangene Pakete zu den Zieladressen. Diese Eigenschaft ist sehr wichtig für die Übertragung in Echtzeit, bei der es mehr auf Schnelligkeit und weniger auf den kompletten Erhalt aller Pakete und deren Inhalte ankommt. Damit multimediale Datenströme bei den Empfän-

gern in Echtzeit verarbeitet werden können, stellt RTP verschiedene Mechanismen zur Verfügung:

- **Zeitmarkierung:** Im Moment der Auswahl des ersten Bytes eines Pakets wird vom Sender eine Zeitmarke für dieses Paket festgelegt. Nach dem Eintreffen von Paketen beim Empfänger können diese aufgrund der Zeitmarken in die richtige zeitliche Reihenfolge gebracht und entsprechend der gesendeten Sequenz verarbeitet werden. Die Zeitmarken können auch zur Synchronisation verschiedener Datenströme (Audio und Video) dienen, wobei die eigentliche Synchronisation nicht von RTP übernommen wird.
- **Sequenznummerierung:** Datenpakete können unterschiedliche Wege zum Empfänger zurücklegen, sodass später abgeschickte Pakete einer Sequenz trotzdem vor früher gesendeten Paketen eintreffen können. Jedes Paket erhält daher eine Sequenznummer, mit deren Hilfe die Pakete beim Empfänger wieder in die richtige Reihenfolge gebracht werden können. Die Sequenznummerierung ist zusätzlich zu den Zeitmarken notwendig, da beispielsweise ein Videoausschnitt auf mehrere Pakete aufgeteilt werden kann, die alle die gleiche Zeitmarke aufweisen. Mit Hilfe von Sequenznummern können diese beim Empfänger dann wieder richtig zugeordnet werden. Durch die Sequenznummerierung lassen sich auch auf dem Weg verloren gegangene Pakete ermitteln.
- **Typidentifikation der Nutzlast:** Zur Identifikation des Formats der Nutzlast und der damit angewandten Kodierungs- und Kompressionsverfahren dient der Payload Type Identifier (PTI). Beim Empfänger angekommen kann die entsprechende Anwendung anhand des PTI die Nutzlast interpretieren und entsprechend verarbeiten. Als Standard-Nutzlasttypen spezifiziert RFC-1890 Verfahren wie PCM, MPEG1/2, JPEG Video sowie H.261 Video.
- **Senderidentifikation:** Diese Funktion erlaubt es einem Empfänger, den Absender von Daten zu identifizieren.

Die beschriebenen Mechanismen werden durch den RTP-Header umgesetzt, der in einem UDP/IP-Paket eingebettet ist und die folgenden Felder enthält: [SCFJ96]

- **V – Version (2 Bit)**: Gibt die Version von RTP an. Die aktuelle Version ist 2.
- **P – Padding (1 Bit):** Dieses Bit wird gesetzt, wenn am Ende der Nutzlast zusätzliche Bytes als Puffer angehängt wurden. Das letzte Byte des Puffers enthält die Anzahl der zu ignorierenden Bytes des Puffers.
- **X – Extension (1 Bit)**: Wird gesetzt, wenn dem festgelegten Header eine Erweiterung folgt.
- **CC – CSRC Count (4 Bit)**: Enthält die Anzahl der CSRC-Identifier, die dem festgelegten Header folgen. Diese Zahl ist größer als 1, wenn das RTP-Paket Daten verschiedener Quellen beinhaltet.

- **M – Marker (1 Bit)**: Dieses Bit kann gesetzt werden, um bestimmte Fälle wie z.B. Rahmengrenzen in einem Paketstrom zu markieren.
- **PT – Payload Type (7 Bit)**: Dient der Identifikation des Formats der Nutzlast.
- **SN – Sequence Number (16 Bit)**: Enthält die Nummer eines RTP-Pakets. Diese wird ausgehend von einem Zufallswert je Paket um den Wert 1 erhöht.
- **Timestamp (32 Bit)**: Entspricht dem Zeitpunkt der Auswahl des ersten Bytes eines Pakets und dient der Synchronisation oder Berechnung von Verzögerungen. Der Startwert wird zufällig gewählt.
- **SSRC – Synchronization Source (32 Bit)**: Zufällig ausgewählte Zahl, um Synchronisationsquellen innerhalb der gleichen RTP-Sitzung unterscheiden zu können.
- **CSRC – Contribution Source (Liste mit bis zu 15 Punkten mit jeweils 32 Bit)**: Zusätzliche Quellen der Nutzlast dieses Pakets. Die Anzahl ergibt sich aus dem Wert von CC.

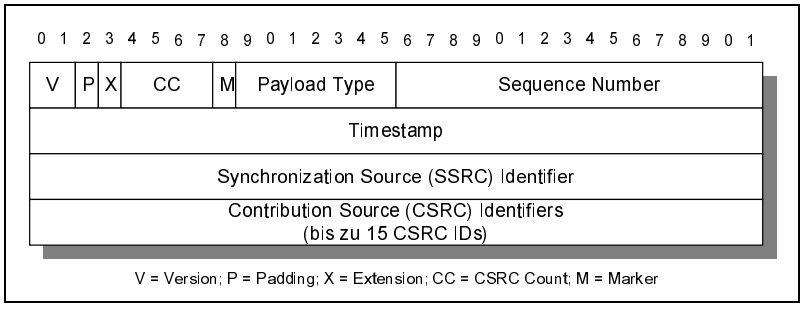


Abb. 5.19 Aufbau eines RTP-Headers

Funktionalität von RTCP

RTCP ist ebenso wie RTP in RFC-1889 und RFC-1890 spezifiziert. Es basiert auf der periodischen Übermittlung von Kontrolldatenpaketen an alle Teilnehmer einer Session[34], wobei der gleiche Verteilungsmechanismus wie für die Datenpakete benutzt wird. Das zugrunde liegende Protokoll muss das Multiplexen der Daten zur Verfügung stellen und die Datenpakete steuern, beispielsweise getrennte Port-Nummern für UDP. Das RTCP führt dabei nach RFC-1889 vier Funktionen aus: [SCFJ96]

1. Die Hauptfunktion ist, eine Rückmeldung über die Qualität der Datenverteilung zu erhalten. Dies gibt Aufschluss über einen reibungslosen Datenfluss oder von Datenstaus.
2. RTCP beinhaltet einen Transport-Level-Identifizierer für eine RTP-Quelle[35]. Da sich das SSRC-ID verändern kann, wenn ein Konflikt auftritt oder ein Pro-

34 Bei Punkt-zu-Punkt nur an ein Teilnehmer
35 CNAME = Canonical Name

gramm wieder in Betrieb gesetzt wird, benötigt der Empfänger die CNAME, um den Überblick über die Teilnehmer zu behalten. Der CNAME wird auch am Empfänger benötigt, um eine Synchronisation zwischen Audio- und Videoströmen zu gewährleisten.

3. Die ersten zwei Funktionen verlangen, dass alle Teilnehmer RTCP-Datenpakete senden. Darum muss die Datenrate kontrolliert werden, in Abhängigkeit von der Anzahl der Teilnehmern. Daraus lässt sich ein RTCP Übermittlungsabstand bestimmen.
4. Eine vierte optionale Funktion ist die Beförderung minimaler Sitzungssteuerinformationen, z.B. Teilnehmeridentifizierung.

Die Funktionen von 1 bis 3 sind vorgeschrieben, wenn RTP in IP-Multicast-Umgebung benutzt wird. Während einer RTP-Sitzung liefert RTCP Informationen über die Teilnehmer und die Qualität der ankommenden Daten. Zu diesem Zweck unterscheidet RTCP fünf unterschiedliche Pakettypen, die Kontrollinformationen übertragen:[36]

1. **Receiver Report** (**RR**): Es werden Berichte von Teilnehmern erstellt, die keine aktiven Sender sind. Diese beinhalten ein Feedback über die Datenzustellung, die durch die höchste empfangene Paketnummer gemessen werden, die Anzahl verlorener Pakete und der Verzögerung zwischen dem Empfang verschiedener Pakete und Zeitmarken.
2. **Sender Report** (**SR**): Berichte werden auch von den aktiven Sendern erstellt. Zusätzlich zu den Informationen der Receiver Reports beinhalten Sender Reports Daten zur Synchronisation und Angaben über die insgesamt verschickten Pakete und Bytes.
3. **Source Description Items** (**SDES**): Informationen zur Beschreibung der Quellen
4. **BYE**: Dieser Typ leitet das Ende einer Sitzung ein.
5. **Application Specific Functions** (**APP**): Dienen experimentellen Zwecken für neue Anwendungen.

RTP ist ein offenes Protokoll, das zukünftige Veränderungen zulässt. Es können neue Formate der Nutzlast genauso wie neue Multimedia-Programme einbezogen werden. Die Implementierung von RTP/RTCP erfolgt in der Anwendungsschicht und muss von den Entwicklern selbst vorgenommen werden. Zur Erleichterung der Implementierung könnte man allerdings die Source Codes verschiedener Module aus dem Internet beziehen.[37] [SCHU96]

36 Der Aufbau der einzelnen Pakete ist in [SCFJ96] detailliert beschrieben.

37 Siehe z.B. ftp://ftp.cs.columbia.edu/pub/schulzrinne/rtptools/

5.5.2 Session Initiation Protocol (SIP)

Das Session Initiation Protocol (SIP) nach RFC-2543 ist ein Signalisierungsprotokoll zum Einrichten, Modifizieren und Beenden von Multimedia-Konferenzen, IP-Telefonie-Verbindungen und ähnlichen Anwendungen. Die Veröffentlichung von Multicast-Adressen zur Teilnahme an Multimedia-Konferenzen geschieht durch das Session Announcement Protocol (SAP). Die Beschreibung der Nutzlastformate innerhalb der Multimedia-Sitzungen erfolgt durch das Session Description Protocol (SDP). Die nachfolgende Tabelle zeigt die möglichen Kombinationen für Unicast- und Multicast-Signalisierung. Möchte beispielsweise ein Konferenzteilnehmer einen anderen Anwender zu einer Multicast-Konferenz einladen, erfolgt die Signalisierung in Unicast. In einem anderen Fall kann ein Anwender in Multicast signalisieren, dass er mit dem nächstmöglich erreichbaren Gesprächspartner eine Punkt-zu-Punkt-Verbindung in Unicast aufbauen möchte[38].

Signalisierung	Unicast-Konferenzeinladung	Multicast-Konferenzeinladung
Unicast	IP-Telefonie	Konferenzeinladung
Multicast	Automated Call Distribution	Gruppeneinladung

Tab. 5.5 Signalisierung mit SIP [SR98a]

SIP wird von der Arbeitsgruppe Multiparty Multimedia Session Control [HCBO97] der IETF als Teil einer umfassenden Multimedia Daten- und Kontrollarchitektur entwickelt. Diese Architektur stellt als Light Weight Session Model eine Alternative zur ITU H.323-Protokollfamilie dar und setzt sich hauptsächlich aus folgenden Protokollen zusammen, wobei SIP nicht von diesen Protokollen abhängt:

- **Session Announcement Protocol (SAP)**: Zur Bekanntmachung von Multimedia-Sitzungen über Multicast-Pakete nach RFC-2974
- **Session Description Protocol (SDP)**: Zur Beschreibung von Multimedia-Sitzungen nach RFC-2327
- **Resource Reservation Protocol (RSVP)**: Signalisierungsprotokolle zur Ressourcenreservierung nach RFC-2205
- **Real-Time Protocol (RTP)**: Echtzeitprotokoll zum Transport von isochronen Datenströmen und für QoS-Rückmeldungen nach RFC-1889
- **Real-Time Streaming Protocol (RTSP)**: Zur Kontrolle der Leistungserbringung für Streaming Media nach RFC-2326

38 Automated Call Distribution

Funktionsweise SIP ist ein Protokoll auf der Anwendungsebene, welches Multimedia-Sitzungen oder Telefonanrufe etablieren und kontrollieren kann. Diese Multimedia-Sitzungen beinhalten Multimedia-Konferenzen, Distance Learning, Internet-Telefonie und ähnliche Anwendungen. SIP ist in der Lage, Personen zu Unicast- oder Multicast-Sitzungen einzuladen. Dabei muss der Initiator eines Calls nicht notwendigerweise ein Mitglied einer Sitzung sein, zu der er einladen möchte. Medien und Teilnehmer können zu einer existierenden Sitzung hinzugefügt werden. Somit kann SIP verwendet werden, um Personen und „Robots" zu kontaktieren, um beispielsweise einen Datenspeicher[39] für die Aufnahme einer laufenden Konferenz hinzuzufügen oder einen Video-on-Demand (VoD) Server zu starten, der ein Video in einer Konferenz abspielt. Dabei kontrolliert SIP nicht direkt diese Dienste, sondern überlässt das dem Protokoll RTSP. [SRL98]

Weiterhin kann SIP verwendet werden, um Sitzungen zu initiieren oder Personen zu einer Sitzung einzuladen, die durch andere Personen etabliert wurden. Sitzungen können über das Protokoll SAP durch die Verwendung von Multicast-Paketen bekannt gemacht werden. Andere Möglichkeiten sind E-Mails, entsprechende Newsgroups, Webseiten oder Verzeichnisdienste wie LDAP[40]. [HPW00]

SIP unterstützt ebenfalls transparent die Anpassung von Namen und Umleitungsdiensten[41], die das Implementieren von ISDN- und IN[42]-Telefon-Teilnehmerdiensten ermöglicht. [MOCK87] diskutiert diese Dienste im Detail. Ebenfalls ermöglicht SIP eine personalisierte Mobilität. Das heißt, Teilnehmer können Calls empfangen oder aufsetzen sowie Zugriff zu Telekommunikationsdiensten von jedem Terminal von jedem beliebigen Standort erlangen. Das Netzwerk hat dabei die Möglichkeit, die Teilnehmer zu identifizieren, auch wenn sie sich in Bewegung befinden. Personalisierte Mobilität basiert auf der Verwendung einer eindeutigen Personalidentität (z.B. Personalnummer). Es ergänzt somit die Terminalmobilität, die die Fähigkeit besitzt, die Kommunikation aufrechtzuerhalten, wenn ein einzelnes Endsystem von einem Netzwerk zu einem anderen wechselt[43]. [PAND95]

SIP unterstützt einige oder alle fünf Facetten um Multimedia-Kommunikationen zu etablieren und zu beenden:

1. **User Location:** Bestimmung des zu benutzenden Endsystems, über welches die Kommunikation stattfindet.
2. **User Capabilities:** Bestimmung des Mediums und dessen Parameters, welche verwendet werden.

39 Media Storage Device
40 Lightweight Directory Access Protocol
41 Redirection Service
42 Intelligent Network
43 Roaming

3. **User Availability:** Festlegung der Bereitschaft einer rufenden Partei, an einer Kommunikation teilzunehmen.
4. **Call Setup:** Etablieren von Anrufparametern für die gerufene und rufende Partei
5. **Call Handling:** Beinhaltet den Ruftransfer und die Ruferkennung .

SIP wird ebenso in Verbindung mit anderen Rufkonfigurationen und Signalisierungsprotokollen verwendet. In diesem Modus verwendet ein Endsystem das SIP-Protokoll als so genannte Vermittlungsstelle, um zugewiesene Endsystemadresse und -protokolle von einer gegebenen Adresse, unabhängig vom Protokoll, bestimmen zu können. Beispielsweise kann SIP verwendet werden, um festzustellen, dass die Gegenseite versucht, über H.323 zu kommunizieren. Es wird die H.245-Gateway- und Benutzeradresse ausgewertet und anschließend H.255.0 genutzt, um einen Call aufzubauen. In einem anderen Beispiel kann SIP benutzt werden, um zu erkennen, ob der rufende Gesprächsteilnehmer über das Public Switched Telephone Network (PSTN) erreicht werden kann oder nicht. Es wird die Telefonnummer angezeigt, die anruft, und falls möglich ein Gateway (Internet-zu-PSTN) vorgeschlagen, um den Anruf zu erwidern.

Weiterhin ist SIP in der Lage, Multi-Partie-Anrufe zu initiieren. Um dies tun zu können, wird eine Multipoint Control Unit (MCU) mittels Unicast angesprochen oder eine voll vermaschte Kommunikation mittels Multicast zwischen den Teilnehmern etabliert. Internet-Telefon-Gateways, die PSTN-Teilnehmer miteinander verbinden, können ebenfalls SIP zur Kommunikation zwischen ihnen nutzen. Dabei bietet SIP selbst keinen Konferenzkontrolldienste an und gibt nicht vor, wie eine Konferenz verwaltet werden soll. Allerdings ist SIP in der Lage, die Protokolle, die für eine Kontrolle und Verwaltung notwendig sind, einzuführen. SIP vergibt ebenfalls keine Multicast-Adressen. Diese Funktionalität muss von anderen Protokollen wie SAP angeboten werden.

SIP kann Teilnehmer zu vorhandenen Sitzungen mit oder ohne Ressourcenreservierung einladen. Das Protokoll selbst reserviert dabei keinerlei Ressourcen, sondern übermittelt dem eingeladenen System die notwendigen Informationen, damit es dies selbst tun kann. Garantien für Quality-of-Service (QoS) können von den Mitgliedern einer Sitzung abhängen, was durch den Agenten bemerkt werden kann, der die Sitzungseinladung ausführt.

SIP-Operationen

Als Client-Server-Protokoll ist SIP bezüglich Client-Anfragen und Server-Antworten mit dem HTTP-Protokoll verwandt. Anfragen bzw. Antworten sind textbasiert und beinhalten Header-Felder, die Informationen für die Verbindungssignalisierung transportieren. Anrufer und Angerufene werden über SIP-URLs identifiziert,[44] die der mailto- oder telnet-URL mit der Syntax anwen-

44 Zum Beispiel sip: info@tu-darmstadt.de

der@host ähnlich sind. Der Anwender kann ein Benutzername, ein Vor- bzw. Nachname oder eine Telefonnummer sein. Der Host wiederum kann entweder ein Domainname oder eine numerische IP-Adresse sein.

Bei einem Anruf lokalisiert der Anrufer zunächst den passenden SIP-Server und schickt diesem eine SIP-Anfrage. Diese kann unter Umständen über mehrere Proxy-Server bis zum gewünschten Gesprächspartner weitergeleitet werden. Ein Gesprächspartner kann einem SIP-Server die Kontaktadressen seiner unterschiedlichen Aufenthaltsorte dynamisch mitteilen. [HSSR99]

Ein SIP-Client muss den folgenden Schritten folgen, um den Host-Teil einer rufenden Adresse auflösen zu können. Wenn ein Client nur TCP oder UDP unterstützt, also nicht beide, wird die jeweilige Adresse weggelassen. Falls die SIP-Adresse eine Portnummer enthält, die bereits verwendet wird, ist dies ebenfalls der Fall. Die Standard-Portnummern für UDP und TCP sind dieselben. Folgende Schritte sind nun notwendig, um einen SIP-Server zu lokalisieren:

1. Wenn die SIP-Adresse eine numerische IP-Adresse ist, kann ein SIP-Server diese Adresse kontaktieren.
2. Falls die SIP-Adresse keine Portnummer enthält und ein SRV DNS[45] Resource Record [MOCK87] des Typs sip.udp vorhanden ist, kontaktierten die aufgeführten SIP-Server in der Reihenfolge der bevorzugten Werte[46] jene Resource Records mittels der Verwendung des Transportprotokolls UDP oder TCP mit der Portnummer, die in den DNS Resource Records aufgeführt ist.
3. Wenn ein DNS MX Record vorhanden ist, kontaktiert der Host in der Reihenfolge der bevorzugten Werte mit der Standard-Portnummer den SIP-Server. Dies erfolgt zuerst mit UDP, anschließend mit TCP.
4. Letztendlich wird überprüft, ob der DNS CNAME oder A Record für einen gegebenen Host vorhanden ist, und es wird versucht, den SIP-Server durch eine oder mehrere Adressen zu kontaktieren. Ebenfalls wird versucht, die Kommunikation hier zuerst mittels UDP und anschließend über TCP aufzubauen.
5. Wenn alle Methoden fehlschlagen, kontaktiert der Anrufer wahrscheinlich einen SMTP-Server auf der Benutzerseite und verwendet das SMTP-EXPN-Kommando, um eine alternative Adresse zu erhalten und die vorherigen Schritte zu wiederholen.
6. Als letzten Ausweg wählt der Client die Möglichkeit aus, die Sitzungsbeschreibung an alle Anrufer über E-Mail weiterzuverteilen.

45 Domain Name Service

46 Sind in in den Resource Records enthalten.

Zusammenfassend lässt sich SIP wie folgt charakterisieren:

SIP-Charakteristika

- Transportsmöglichkeiten über UDP, TCP, IPX, Frame Relay, ATM AAL5 oder X.25
- Signalisierung von Gesprächen mit und ohne Ressourcenreservierung
- Multicast-Signalisierung von Gesprächen und Konferenzen
- Authentifizierung von Anrufern und Lokalisierung von Gesprächspartnern
- Mobilität durch mehrere Kontaktadressen und Gesprächsweiterleitung
- Fähigkeitsaustausch zwischen verschiedenen Terminals
- Erweiterte Telefondienste wie z.B. Blind Transfer, Operator-Assisted Call Transfer, Auto-Dialer und Click-to-Call-Verknüpfungen in Webseiten
- Zusammenarbeit mit H.323, PSTN oder ISDN Terminals

5.5.3 H.323-Rahmenwerk

Trotz der Protokollentwicklung besitzen IP-Protokolle grundsätzlich einen verbindungslosen Charakter. Die Übertragung findet weiterhin Hop-by-hop statt. Eine ähnliche Qualität, Verfügbarkeit und Stabilität wie in verbindungsorientierten Netzen ist daher schwer zu realisieren. Auch vorhandene Echtzeitprotokolle wie RTP basieren auf der Netzqualität, die ihnen von den darunter liegenden Schichten bereitgestellt wird. Deshalb spielt QoS ein wichtige Rolle bei der Realisierung von VoIP[47] und anderen Echtzeitapplikationen. Tab. 5.6 zeigt die unterschiedlichen ITU-Standards, die für die Realisierung einer VoIP-Lösung beachtet werden müssen. Hinzu kommen hierbei die Ansätze der IETF.

Im Jahr 1996 wurde von der ITU-T ebenfalls die Empfehlung H.323 mit dem Ziel verabschiedet, multimediale Kommunikation über lokale Netzwerke, die in den meisten Fällen keine Dienstgüte gewährleisten, zu ermöglichen. Die Kommunikationsdienste beinhalten Übertragungen von Audio in Echtzeit, Video (optional) und Daten (optional) für Punkt-zu-Punkt- und Multipoint-Konferenzen. Zwei Jahre später erfolgte mit der zweiten Version von H.323 eine Erweiterung auf alle IP-basierten Netzwerke. Dazu gehören lokale Netzwerke (LAN), firmeneigene Netze, Großstadtnetze (MAN), Intranetzwerke und Internetzwerke wie das Internet. Dazu zählen auch Wählverbindungen oder Punkt-zu-Punkt-Verbindungen über GSTN[48], die auf einem paketvermittelten Transport wie z.B. PPP[49] basieren. Unterstützte LAN-Netzsysteme sind u.a. Ethernet (IEEE 802.3), Fast Ethernet (IEEE 802.3u), FDDI[50], Token Ring (IEEE 802.5) und ATM[51].

Der H.323-Standard beschreibt die Komponenten und Mechanismen, die zum Telefonieren über paketbasierte Netzwerke notwendig sind. Zu den Komponenten gehören Terminals, Gateways, Gatekeeper, Multipoint Controller,

47 Voice-over-IP
48 General Switched Telecommunication Network
49 Point-to-Point Protocol
50 Fiber Distributed Data Interface
51 Asynchronous Transfer Mode

Multipoint-Prozessoren und Multipoint Control Units. Die Mechanismen werden durch die Einbindung weiterer ITU-Protokolle realisiert und ermöglichen den Verbindungsaufbau, den Austausch der technischen Möglichkeiten der Teilnehmer, die Kontrolle der laufenden Kommunikation und die Überwachung stattfindender Konferenzen. Da zu diesem Zweck auch andere Protokolle wie das IETF-Protokoll RTP, weitere ITU-Protokolle wie H.225.0, H.245 und Kodierungsprotokolle einbezogen sind, wird H.323 auch als Protokollfamilie bezeichnet. Der H.323-Rahmen beinhaltet dabei die Standardisierung von System-, Verbindungs- und Sitzungs-Kontrollinformationen, Paketmultiplexing sowie Video- und Sprach-Codecs und deren Übertragungsparameter. Außerhalb des Rahmens von H.323 sind Definitionen von Netzwerk-Schnittstellen, physikalischen Netzwerken, garantierte Dienstgüte oder verwendete Netzwerkprotokolle.[52] [H.323(99)]

Tab. 5.6 Überblick über ITU-Videokonferenzstandards [DOMM98]

	H.320	**H.321**	**H.322**	**H.323**	**H.324**
Anerkennungsdatum	1990	1995	1995	1998	1996
Netzwerk	ISDN	ATM oder LAN	paketbasierte Netze mit QoS	paketbasierte Netze ohne QoS	PSTN
Videokodierung	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263
Audiokodierung	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728 G.723 G.729	G.723
Multiplex	H.221	H.221	H.221	H.225.0	H.223
Kontrolle	H.230 H.243	H.242	H.242 H.230	H.245	H.245
Mehrpunktverbindungen	H.231 H.243	H.231 H.243	H.231 H.243	H.323	
Datenübertragung	T.120	T.120	T.120	T.120	T.120
Schnittstelle	I.400	AAL I.363 AJM I.361 PHY I.400	I.400 TCP/IP	TCP/IP	V.34 Modem

52 Die Ausführungen über das H.323-Kommunikationssystem basieren hauptsächlich auf den ITU-T-Empfehlungen [H.323(99)], [H.225.0(99)] und [H.245(99)].

Systembeschreibung

Die Kommunikation der H.323-Komponenten untereinander erfolgt durch den Austausch von Informationsströmen. Dabei kann man Informationsströme wie folgt einteilen:

1. **Audio:** Audiosignale bestehen aus digitalisierter und kodierter Sprache. Die Audiosignale werden begleitet von Audiokontrollsignalen.
2. **Video:** Videosignale beinhalten digitalisierte und kodierte Bewegtbilder. Die Bildwiederholrate der Videoübertragung ergibt sich aus dem Austausch der technischen Fähigkeiten der Teilnehmer. Die Videosignale werden von Videokontrollsignalen begleitet.
3. **Daten:** Datensignale transportieren Inhalte von Bildern, Faxen, Dokumenten oder Dateien.
4. **Kommunikationskontrolle:** *Kommunikationskontrollsignale* werden für den Austausch der technischen Fähigkeiten der Teilnehmer, das Öffnen und Schließen logischer Kanäle sowie für die Moduskontrolle zwischen den entfernten Kommunikationspartnern verwendet.
5. **Verbindungskontrolle:** Verbindungskontrollsignale dienen sowohl der Einrichtung als auch der Beendigung von Gesprächen und bieten darüber hinaus weitere Verbindungskontrollfunktionen.

Ein H.323-Terminal muss über Audio-Codec, eine Systemkontrolleinheit, eine H.225.0-Schicht und eine Netzwerkschnittstelle verfügen. Video-Codec und Datenanwendungen können optional hinzugefügt werden.

Die Empfehlung H.225.0 [H.225.0(99)] spezifiziert den Verbindungsaufbau zwischen zwei H.323-Endpunkten durch Kontrollpakete des Protokolls Q.931.[53] Nach der Einrichtung einer TCP-Verbindung zwischen den Kommunikationspartnern werden über den H.225.0-Gesprächssignalisierungskanal die für den Gesprächsaufbau nötigen Q.931-Nachrichten ausgetauscht. Für eine einfache Verbindung sind vier Nachrichtentypen zu unterscheiden: Setup, Alerting, Connect und Release Complete.

Der H.225.0-Gesprächssignalisierungskanal ist unabhängig vom RAS[54]-Kanal und H.245-Kontrollkanal und wird vor der Einrichtung eines H.245-Kontrollkanals oder anderer logischer Kanäle geöffnet. In Systemen ohne Gatekeeper wird der Gesprächssignalisierungskanal unmittelbar zwischen den Endpunkten eingerichtet, in Systemen mit Gatekeeper hingegen erfolgt die Einrichtung zwischen den Endpunkten und den zugehörigen Gatekeepern.

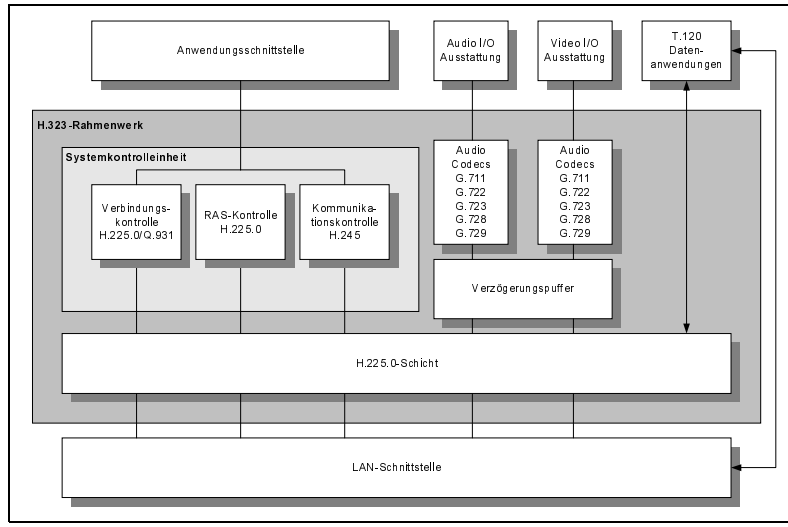
Die RAS-Kontrolle erfolgt mit H.225.0-Nachrichten für die Registrierung und Zulassung von Teilnehmern, die Veränderung der Bandbreite, den Status und die Auflösung von Verbindungen zwischen Endpunkten und Gatekeepern.

53 Ein Endpunkt kann ein H.323 Terminal, ein Gateway oder eine MCU sein; er kann Verbindungen auf- und abbauen und Informationsströme erzeugen und schließen.

54 Registration Admission Status

Der RAS-Kanal ist unabhängig vom H.225.0-Gesprächssignalisierungskanal und dem H.245-Kontrollkanal und kommt ausschließlich in Netzwerken zum Einsatz, in denen sich ein Gatekeeper befindet. In einem Netzwerk mit Gatekeeper wird der RAS-Kanal zwischen dem Gatekeeper und den zugehörigen Endpunkten vor der Öffnung weiterer Kanäle zwischen den H.323-Endpunkten eingerichtet und bleibt so lange geöffnet, bis ein Endpunkt seine Registrierung beim Gatekeeper auflöst. Der Transport von RAS-Nachrichten erfolgt über UDP[55].

Abb. 5.20
Aufbau eines H.323 Terminals [H.323(99)]



Über den H.245-Kontrollkanal werden Ende-zu-Ende-Kontrollnachrichten ausgetauscht, um die Arbeitsweise von H.323-Einheiten zu steuern. Dazu gehören der Austausch der technischen Fähigkeiten, das Öffnen und Schließen logischer Kanäle, Anfragen nach Moduspräferenzen, Flusskontrollnachrichten und weitere Befehle oder Verbindungsmerkmale.

Der Austausch von H.245-Nachrichten findet entweder zwischen zwei Endpunkten, einem Endpunkt und einem Multipoint Controller (MC) oder aber einem Endpunkt und einem Gatekeeper statt. Jeder Endpunkt richtet nur einen H.245-Kontrollkanal für eine Verbindung ein. Ein Terminal, eine MCU, ein Gateway oder ein Gatekeeper können viele Verbindungen und daher auch viele H.245-Kontrollkanäle unterstützen. Ein H.245-Kontrollkanal ist von der Verbindungseinrichtung an bis zum Verbindungsende geöffnet.

Zur Kommunikationskontrolle spezifiziert H.245 unterschiedliche und voneinander unabhängige Protokolleinheiten. Diese müssen von H.323-Endpunkten, bezogen auf die Syntax, die Semantik und die Prozeduren, unterstützt werden: [H.245(99)]

55 User Datagram Protocol

- **Master/Slave-Bestimmung**: Diese Protokolleinheit dient der Auflösung von Konflikten zwischen zwei Endpunkten, die beide die Funktion eines MCs für eine Konferenz übernehmen können oder die beide einen bidirektionalen Kanal öffnen möchten.
- **Austausch der technischen Fähigkeiten**: Mit den entsprechenden H.245-Prozeduren werden Sende- und Empfangsfähigkeiten zwischen den Endpunkten ausgetauscht. Ferner kann ein Terminal mitteilen, in welchen verschiedenen Modi es gleichzeitig arbeiten kann.
- **Einrichtung von unidirektionalen logischen Kanal**: Jeder logische Kanal überträgt Informationen von einem Sender zu einem oder mehreren Empfängern und wird durch eine logische Kanalnummer für jede Übertragungsrichtung identifiziert. Die meisten logischen Kanäle sind unidirektional. Es besteht jedoch auch die Möglichkeit, bidirektionale Kanäle aufzubauen (z.B. für die Übertragung von Daten unter T.120 mit einem Paar von unidirektionalen Kanälen). Logische Kanäle werden geöffnet und geschlossen durch die H.245-Prozeduren *open_Logical_Channel* und close_Logical_Channel. Eine Nachricht open_Logical_Channel beschreibt den Inhalt des logischen Kanals, d.h. den Medientyp, den verwendeten Algorithmus sowie alle weiteren Informationen, welche für die inhaltliche Interpretation der Daten für den Empfänger erforderlich sind. Bei der Eröffnungsprozedur eines logischen Kanals zur Übertragung von Audio oder Video werden die RTP- und RTCP-Transportadressen zwischen Sender und Empfänger ausgetauscht.
- **Einrichtung von bidirektionalen logischen Kanälen:** Es werden Prozeduren von H.245 zum Öffnen und Schließen bidirektionaler logischer Kanäle angeboten.
- **Schließen eines logischen Kanals**: Die entsprechenden H.245-Prozeduren werden von Terminals verwendet, um logische Kanäle zu schließen.
- **Anfrage nach Modus:** Empfänger können Anfragen an Sender schicken, in denen sie einen bevorzugten Übertragungsmodus angeben.
- **Bestimmung der Round-Trip-Time (RTT):** es werden H.245-Prozeduren zur Verfügung gestellt, mit deren Hilfe die Round-Trip-Time (RTT) zwischen zwei Terminals bestimmt werden kann. Weiterhin kann festgestellt werden, ob eine bestimmte H.245-Protokolleinheit noch existiert.
- **Aufrechterhaltungsschleifen:** H.245 verfügt über Prozeduren für die Einrichtung von Aufrechterhaltungsschleifen zwischen Endpunkten.

Alle H.323-Terminals müssen über Audio-Codecs verfügen. Mit Audio-Codecs werden Audiosignale eines Mikrofons kodiert und umgekehrt eingehende kodierte Audiosignale für die Ausgabe am Lautsprecher dekodiert. Für die Kodierung und Dekodierung gibt es verschiedene ITU-Empfehlungen. Gemäß H.323 müssen Terminals die Empfehlung G.711 unterstützen. Optional kön-

nen auch die Codecs der Empfehlungen G.722, G.728, G.729, MPEG 1 Audio und G.723.1 implementiert werden. Der verwendete Audio-Algorithmus wird während des Austauschs der Fähigkeiten unter H.245 ermittelt.

Ein Terminal kann auch asymmetrisch arbeiten; durch die Verwendung von G.711 für die Kodierung zu sendender Sprachsignale und von G.728 für die Dekodierung zu empfangender Sprachsignale. Ferner kann ein H.323-Terminal mehr als ein Audiokanal senden und empfangen (z.B. für zweisprachige Übertragung). Die Formatierung des Audio-Datenstroms geschieht entsprechend der H.225.0-Empfehlung. Zusätzliche Parameter wie z.B. Audio Delay Jitter oder Skew Indication werden unter Verwendung von H.245 ausgetauscht.

Video-Codecs sind optional. Falls jedoch H.323-Terminals Videokommunikation unterstützen, sollte die Kodierung und Dekodierung von Video gemäß H.261 CIF implementiert sein. Optional können zusätzlich andere Modi von H.261 oder von H.263 integriert werden. Andere Video-Codecs und Bildformate können über den H.245-Kontrollkanal ebenso wie mehrere Videokanäle zum gleichzeitigen Senden oder Empfangen vereinbart werden. Die Video-Bitrate, das Bildformat sowie Algorithmusoptionen werden während des Austauschs der technischen Fähigkeiten unter H.245 festgelegt. Der Kodierer kann alles senden, was den Fähigkeiten des Dekodierers entspricht. Der Dekodierer kann über H.245 einen bestimmten Modus beantragen, wobei der Kodierer sich darüber hinwegsetzen kann.

H.323-Terminals sollten mit asymmetrischen Video-Bitraten, Rahmenraten und Bildauflösungen arbeiten können.[56] Dadurch kann ein Terminal z.B. im QCIF-Format senden, während es Bilder im CIF-Format empfängt. Nach der Öffnung der Videokanäle wird dem Empfänger durch die jeweiligen H.245-Nachrichten[57] der Modus jedes einzelnen Videokanals mitgeteilt.

Die Qualität von Multimedia-Anwendungen in Netzwerken wird von zwei Einflussgrößen maßgeblich beeinträchtigt, wie in diesem Kapitel verdeutlicht wurde:

- **Verzögerung (Latency):** Summe der gesamten Verzögerungen beim Transport eines Pakets vom Sender zum Empfänger
- **Verzögerungsschwankungen (Jitter):** ist der Umstand, dass die Verzögerungszeiten eintreffender Pakete schwanken.

Latency führt dazu, dass verschiedene Mediaströme wie beispielsweise Audio oder Video mit unterschiedlichen Verzögerungen beim Empfänger eintreffen. Das kann dazu führen, dass Sprach- und Videosignale nicht mehr aufeinander abgestimmt sind. Um diesen Effekt zu vermeiden, können zur Synchronisation optional Verzögerungen hinzugefügt werden. Das Gleiche kann geschehen, um

56 Falls mehrere Bildauflösungen unterstützt werden.
57 Open_Logical_Channel

schwankende Verzögerungszeiten eintreffender Pakete (Jitter) eines Mediastroms auszugleichen und somit ruckartige Bilder oder verzerrte Sprache zu verhindern. Verzögerungen dürfen dabei nicht von Sendern, sondern ausschließlich von Empfängern hinzugefügt werden.

Die Empfehlung T.120 ist die Basis für die Austauschbarkeit von Daten zwischen einem H.323-Terminal und anderen H.323-, H.324-, H.320- oder H.310-Terminals. Es können ein oder mehrere Datenkanäle für den Austausch von Daten optional geöffnet werden. Entsprechend der jeweils verwendeten Datenanwendung[58] kann der Datenkanal uni- oder bidirektional sein. Die Öffnung eines Datenkanals erfolgt durch das Senden einer Nachricht open_Logical_Channel über den H.245-Kontrollkanal, in welcher weitere Parameter zur Spezifikation des Datenkanals mitgeteilt werden. T.120 kann die H.225.0-Schicht zum Senden und Verpacken von Datenpaketen nutzen oder mit Hilfe eigener Mechanismen Daten direkt ins Netzwerk versenden.

H.225.0 definiert eine Schicht, welche die zu übertragenden Audio-, Video-, Daten- und Kontrollströme zum Versenden formatiert bzw. zum Empfangen wiederherstellt. Für die Übertragung dieser Ströme werden logische Kanäle entsprechend den Prozeduren von H.245 eingerichtet, denen vom Sender willkürlich eine Kanalnummer zwischen 0 und 65535 zugeordnet wird, wobei 0 für den H.245-Kontrollkanal reserviert ist.[59] H.225.0 verwendet für Audio- und Videoströme das Paketformat von RTP und RTCP für Framing, Sequenznummerierung, Fehlererkennung und Fehlerkorrektur.[60] [H.225.0(99)]

Die Schnittstelle des paketbasierten Netzwerks ist implementierungsabhängig und liegt außerhalb des Rahmens des H.323-Standards. In jedem Fall sollte dabei die Schnittstelle die Dienste der H.225.0-Empfehlung unterstützen, das heißt, einen zuverlässigen Ende-zu-Ende Dienst (z.B. über TCP oder SPX) für den H.245-Kontrollkanal und die Datenkanäle und den Gesprächssignalisierungskanal bereitstellen. Ferner ist die Unterstützung eines unzuverlässigen Ende-zu-Ende-Dienstes (z.B. über UDP oder IPX) notwendig für die Audiokanäle, die Videokanäle und den RAS-Kanal. Diese Dienste können entweder im Simplex- oder Duplex-Betrieb, in Unicast oder auch in Multicast arbeiten. [H323(99)]

H.323-Funktionseinheiten

Ein H.323-Gateway ist ein Endpunkt im Netzwerk, der Zwei-Wege-Kommunikation zwischen H.323-Terminals in paketbasierten Netzen und anderen ITU-Terminals in leitungsvermittelten Netzen oder zu einem anderen H.323-Gateway ermöglicht. Dabei bietet das Gateway unterschiedliche Dienste an, die in der Empfehlung H.246 spezifiziert sind: [H.246(98)]

58 Z.B. File Transfer, Application Sharing oder Electronic Whiteboard

59 Siehe die Kommunikationskontrolle durch H.245.

60 H.225.0 beinhaltet die Spezifikation von RTP/RTCP im Anhang und ist bis auf wenige Unterschiede identisch mit RTP/RTCP der IETF.

- Übersetzung zwischen Übertragungsformaten (z.B. H.225.0 zu/von H.221) und Kommunikationsprozeduren (z.B. H.245 zu/von H.242)
- Verbindungsaufbau und -abbau sowohl auf Netzwerk- als auch auf PSTN-Seite
- Übersetzung zwischen unterschiedlichen Video-, Audio- und Datenformaten

Verbindungen zu reinen Sprachterminals auf PSTN-Seite können Gateways durch die Erzeugung und Erkennung von DTMF[61]-Signalen von entsprechenden H.245-Nachrichten user_input_indications für 0-9, * und # aufbauen. [H.245(99)]

Der Gatekeeper ermöglicht es hingegen, zusätzliche Dienste für die Verbindungskontrolle zur Unterstützung von H.323-Endpunkten zur Verfügung zu stellen. Die Dienste sind über die RAS[62]-Funktionen in H.225.0 definiert und lassen sich wie folgt einteilen:

- **Adressübersetzung:** Der Gatekeeper übersetzt zwischen Alias- und Transport-Adressen unter Zuhilfenahme einer Übersetzungstabelle, die regelmäßig aktualisiert wird.
- **Zugangskontrolle:** Die Zulassung von Endpunkten zu einem Netzwerk läuft über den Gatekeeper durch die H.225.0-Nachrichten Admission Request, Admission Confirm und Admission Reject. Die Zugangskontrolle kann von der gewünschten Bandbreite oder anderen Faktoren abhängen.
- **Bandbreitenkontrolle:** Ein Endpunkt kann während eines Gesprächs beim Gatekeeper beantragen, dass für ein bestehendes Gespräch mehr oder weniger Bandbreite innerhalb der Zone zur Verfügung gestellt werden soll. Ebenso kann ein Gatekeeper die zugeteilte Bandbreite vergrößern oder verkleinern. Die entsprechenden H.225.0-Nachrichten sind Bandwidth Request, Bandwidth Confirm und Bandwidth Reject.
- **Verbindungskontrolle:** Der Gatekeeper kann wählen, ob die Gesprächssignalisierung unter seiner Mitwirkung oder zwischen den beteiligten Endpunkten direkt abläuft.
- **Gesprächsautorisierung:** Der Gatekeeper kann den Anruf eines Terminals aufgrund des Scheiterns der Autorisierung ablehnen. Die Gründe dafür können vielfältig sein (z.B. eingeschränkter Zugang bestimmter Terminals oder Gateways) und werden von H.323 nicht spezifiziert.
- **Bandbreitenmanagement:** Kontrolle der Anzahl der H.323-Terminals, welche zu einer Zone gleichzeitigen Zugang haben. Mit Hilfe von H.225.0-Nachrichten kann der Gatekeeper Terminals wegen begrenzter Bandbreite zurückweisen. [DOMM98]

61 Dual Tone Multi-Frequency
62 Remote Access Service

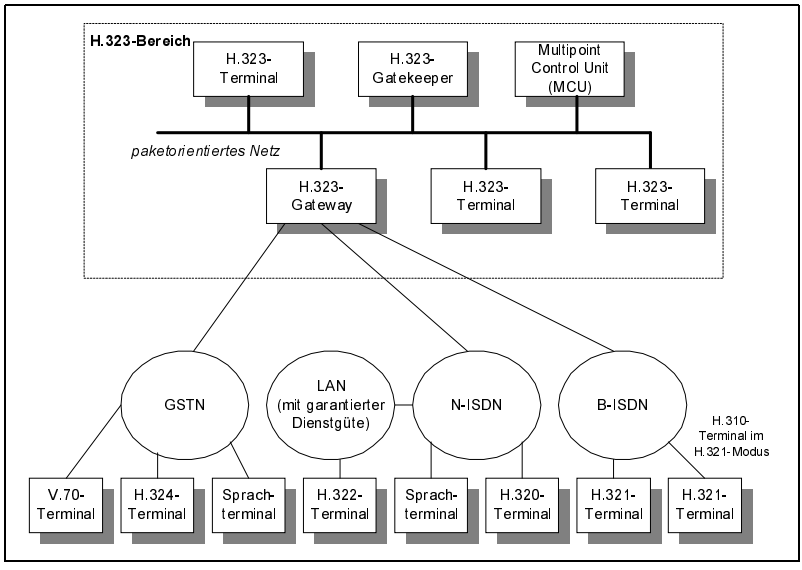


Abb. 5.21 Kommunikation zwischen H.323 und anderen Terminals [H.323(99)]

Im Gegensatz zum Gatekeeper bietet ein Multipoint Controller (MC) Kontrollfunktionen zur Unterstützung von Konferenzen zwischen drei oder mehr Endpunkten. Er führt den Austausch der technischen Möglichkeiten mit jedem Endpunkt durch und sendet jedem Endpunkt die möglichen Übertragungsmodi innerhalb einer Konferenz. Während einer Konferenz kann der MC die Übertragungsmodi ändern, falls Terminals mit bestimmten Fähigkeiten hinzukommen oder sich abmelden. Auf diese Weise bestimmt der MC den Übertragungsmodus für eine Konferenz. Befinden sich beispielsweise zwei reine Sprachterminals ohne Videounterstützung in einer Konferenz und ein weiteres Terminal mit Audio- und Videofähigkeiten möchte dieser Konferenz beitreten, teilt es dem MC seine Fähigkeiten mit. Der MC wiederum legt den Übertragungsmodus auf Audio fest und teilt diese Entscheidung dem hinzugekommenen Terminal mit, woraufhin dieses nur Audio- und keine Videodaten sendet, obwohl es dazu in der Lage wäre.

Die Aufnahme eines Terminals zu einer Konferenz geschieht durch die Verbindung zum MC über den H.245-Kontrollkanal. Dies kann auf folgende Arten geschehen:

- Direkte Verbindung mit einer MCU[63]
- Indirekte Verbindung zu einem MC in einem Gatekeeper
- Indirekte Verbindung zu einem MC in einem anderen Terminal oder Gateway, die an einer Multipoint-Konferenz teilnehmen.
- Indirekte Verbindung über einen Gatekeeper zu einer MCU

63 Multipoint Control Unit

Die Wahl des Konferenzmodus (zentral, dezentral oder hybrid) geschieht nach der Kontaktaufnahme mit dem MC über den H.245-Kontrollkanal. Die Auswahlentscheidung wird von den Fähigkeiten des MC und der teilnehmenden Endpunkte beeinflusst. Ein MC kann dabei in einem Terminal, einem Gateway, einem Gatekeeper oder einer MCU angesiedelt sein:

- **MC in Terminal:** Zum MC kann keine Verbindung aufgebaut werden. Er kann in einer bestehenden Verbindung zur Einrichtung einer Ad-hoc-Konferenz eingeschalten werden, um die Kontrolle der H.245-Nachrichten zu übernehmen.
- **MC in Gateway:** Ein Gateway kann die Funktion eines Terminals oder einer MCU übernehmen. Als Terminal kann dabei ein Gateway ein MC beinhalten, welcher wie zuerst beschrieben charakterisiert ist.
- **MC in Gatekeeper:** Zum MC kann keine Verbindung aufgebaut werden. Er kann nur zur Unterstützung von Konferenzen aktiviert werden, falls Endpunkte H.245-Kontrollkanäle zum Gatekeeper aufbauen. Auf diese Weise kann der Gatekeeper die H.245-Kontrollkanäle bei Verbindungsbeginn oder bei Beginn einer Multipoint-Konferenz zum MC umleiten.
- **MC in MCU:** Eine MCU beinhaltet immer einen MC. Zu einer MCU kann eine Verbindung aufgebaut werden, wobei der MC die H.245-Kontrollkanäle aller Endpunkte verwaltet.

Nehmen an einer Konferenz mehr als zwei Teilnehmer bzw. Endpunkte teil, erfolgt die Bestimmung des für die Konferenzkontrolle verantwortlichen MC mit Hilfe der Master-/Slave-Prozedur der H.245-Empfehlung. [H.245(99)]

Ein Multipoint Prozessor (MP) ist hingegen Teil einer MCU und empfängt Audio-, Video- und Datenströme von Endpunkten in zentralen und hybriden Multipoint-Konferenzen und sendet diese nach erfolgter Verarbeitung zu den Endpunkten zurück. Zur Unterstützung von Terminals, die an einer Konferenz mit unterschiedlich ausgewählten Konferenzmodi teilnehmen, kann ein MP verschiedene Algorithmen und Formatkonvertierungen anbieten.

Ein MP verarbeitet m eingehende Audioströme zu n ausgehenden Audioströmen durch Switching, Mixing oder deren Kombination. Beim Audio Mixing werden die eingehenden Audiosignale in lineare Signale dekodiert (PCM oder analog), miteinander linear kombiniert und anschließend in das entsprechende Audioformat für das Weiterleiten zu den anderen Endpunkten kodiert. Ein MP kann eingehende Audiosignale abschwächen oder eliminieren, um Rauschen und andere unbrauchbare Signale zu unterdrücken.

Bei der Verarbeitung von Video sollte ein MP entweder Video Switching oder Video Mixing leisten können. Video Switching bedeutet, dass der MP die Videoströme einzelner Quellen zu den empfangenden Terminals durchschaltet. Unter Video Mixing versteht man das Formatieren mehrerer Videoquellen in den Videostrom, der vom MP an die Terminals geschickt wird (z.B. Kombina-

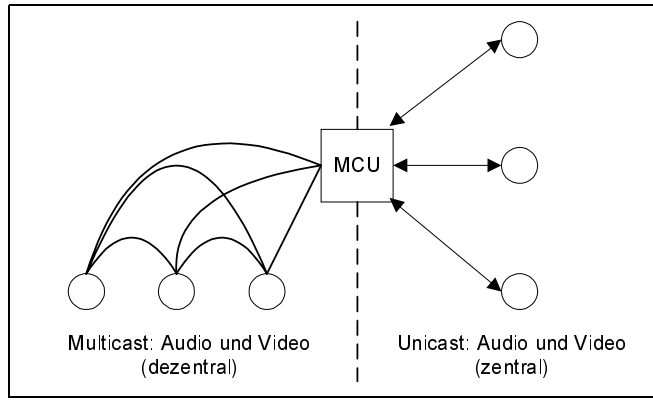
tion von vier Videoquellen in ein zweidimensionales Bildfeld). Der MC entscheidet, welche und wie viel Videoquellen durch Video Mixing verarbeitet werden.

Eine Multipoint Control Unit (MCU) nimmt einen zentralen Stellenwert in der Mehrpunktkommunikation ein, da sie die Unterstützung für Multipoint-Konferenzen bietet. Sie sollte aus einem MC und entweder keinem oder mehreren MPs bestehen. Die klassischen Bestandteile einer MCU in einer zentralen Multipoint-Konferenz sind ein MC und jeweils ein MP für Audio, Video und Daten. In einer dezentralen Multipoint-Konferenz besteht eine typische MCU aus einem MC und einem Daten-MP gemäss der T.120-Empfehlung. Die Audio- und Videoverarbeitung erfolgt dezentral. Mit den entsprechenden Prozeduren können H.323-Endpunkte Verbindungen zur MCU aufbauen.

Unter H.323 können zentrale, dezentrale und hybride Multipoint-Konferenzen stattfinden. In Abb. 5.22 symbolisieren die Kreise Endpunkte von H.323, die sich im linken Teil der Abbildung in einer dezentralen und im rechten Teil in einer zentralen Multipoint-Konferenz befinden, wie folgt:

- **Zentrale Multipoint-Konferenzen:** In dieser Konfiguration laufen alle Informationsströme über die MCU, indem die beteiligten Terminals ihre Audio-, Video-, Daten- und Kontrollströme mit der MCU in Punkt-zu-Punkt-Verbindungen austauschen. Der MC überwacht die Anzahl der Audio- und Videoströme jedes Terminals, entsprechend deren technischen Fähigkeiten. Der MP ist verantwortlich für Audio-Mixing, Video-Switching/-Mixing sowie die Datenverteilung und sendet die entsprechenden verarbeiteten Ströme an die jeweiligen Terminals.
- **Dezentrale Multipoint-Konferenzen:** In dezentralen Multipoint-Konferenzen senden H.323-Terminals in Multicast Audio- und Videoströme direkt zu den anderen teilnehmenden Konferenzterminals. Die Kontrolle der Konferenz erfolgt jedoch zentral durch die MCU über H.245-Kontrollkanäle, die zwischen den Terminals und der MCU bestehen. Die Verarbeitung der unterschiedlichen Audio- und Videoströme wird von den Terminals vorgenommen, wobei ein Terminal dem MC über den H.245-Kontrollkanal mitteilt, wie viele simultane Audio- und Videoströme von diesem empfangen werden können. Davon unabhängig ist die Anzahl der gesamten Audio- und Videoströme, die in Multicast innerhalb einer Konferenz gesendet werden.
- **Hybride Multipoint-Konferenzen:** Hierbei handelt es sich um eine Kombination von zentralen und dezentralen Multipoint-Konferenzen. Die H.245-Kontrollsignale und die Audio- oder die Videoströme tauschen die Terminals zentral mit der MCU aus. Der verbleibende Audio- oder Videostrom wird direkt in Multicast an die anderen Terminals gesendet.

Abb. 5.22 H.323-Multipoint Konferenzen



Unter H.323 können auch Konferenzen stattfinden, in welchen sich einige Terminals entweder in einer zentralen oder in einer dezentralen Konferenz befinden, wobei die MCU die Brücke zwischen den zwei Konferenztypen bildet. In den dargestellten Konferenzformen können die Teilnehmer Medienströme sowohl senden als auch empfangen. Im Unterschied dazu ermöglicht die Einbeziehung der H.332-Empfehlung zwei neue Konferenztypen, die das Abhalten sehr großer Konferenzen gestatten. In Broadcast-Konferenzen schickt ein Sender Medienströme an eine Vielzahl von Empfängern. Die Übertragung zwischen Sender und Empfänger erfolgt dabei ohne bidirektionale Kontroll- und Medienströme. In Broadcast-Panel-Konferenzen empfangen viele Terminals die Medienströme von einigen Terminals, die sich in einer Multipoint-Konferenz befinden. Zwischen den Terminals in der Multipoint-Konferenz sind die Kontroll- und Medienströme im Gegensatz zu den nur empfangenden Terminals bidirektional. [H.332(98)]

Schwächen Die H.323-Empfehlung ist eine sehr umfangreiche Spezifikation. Dazu kommen weitere einbezogene Empfehlungen wie beispielsweise die Kodierungen für Audio, Video und Daten, erweiterte Dienste durch den optionalen Bestandteil H.450.x des H.323-Standards oder Sicherheit durch H.235. Die Einarbeitung und Umsetzung des H.323-Standards durch Entwickler und Hersteller ist daher sehr aufwendig und unüberschaubar. Hinzu kommt, dass einbezogene Empfehlungen und Standards nicht deutlich voneinander abgegrenzt sind und dadurch verschiedene Aktionen mehrere Protokolle betreffen oder ähnliche Funktionalitäten in unterschiedlichen Protokollen spezifiziert sind.[64] [H.235(98)]

Für die Gesprächseinrichtung zwischen zwei oder mehreren Endpunkten können unterschiedliche H.323-Prozeduren wie die Standardprozedur, Fast

64 Z.B. RTCP und H.245 mit Kontrollfunktionen und Feedback für Multipoint-Konferenzen

Start oder Tunneling ablaufen. Zur Unterstützung aller Verfahren sind komplexe Implementierungen in H.323-Terminals, Gateways, Gatekeepern und Firewalls notwendig. Zusätzlicher Implementierungsaufwand entsteht durch die Unterstützung dynamischer Ports.

Der Aufbau einer einfachen Punkt-zu-Punkt-Verbindung zwischen zwei Endpunkten ohne Beteiligung eines Gatekeepers benötigt eine große Anzahl von Round-Trips, die bei komplexeren Verbindungen über Gatekeeper und Gateways in Multipoint-Konferenzen ansteigen und zusätzliche Verzögerungen verursachen. Weitere Anforderungen, die es zu beachten gilt, sind:

- **Interoperabilität:** Durch die Vielzahl der in H.323 einbezogenen Protokolle können Entwickler von H.323-Komponenten eine Vielzahl von Funktionalitäten hinzufügen, was dazu führt, dass Produkte zwar entsprechend der H.323-Empfehlung protokollkonform realisiert werden, jedoch nicht zwingend mit anderen H.323-Produkten zusammenarbeiten. Ferner bestehen teilweise unterschiedliche Auffassungen bei der Interpretation von umzusetzenden Elementen verschiedener Protokolle, sodass Paketformate und Kodierungen unterschiedlich sein können.
- **Erweiterbarkeit:** Anwendungsentwickler und Hersteller können die Funktionen von H.323 durch Eigenentwicklungen erweitern, die in none_ Standard_Parametern der H.245-Empfehlung mit den entsprechenden Herstellercodes angezeigt werden. Nachteilig ist, dass H.323 keine Mechanismen bereithält, mit denen jedes Terminal seine einzelnen Erweiterungen mitteilen kann. Durch Abwärtskompatibilitäten wird die Erweiterbarkeit zusätzlich eingeschränkt bzw. sehr komplex.
- **Skalierbarkeit:** Es treten Skalierbarkeitsprobleme aufgrund der komplexen Domainstruktur auf. Ein Gatekeeper z.B. mit einem bestimmten Domainnamen muss nicht zwingend über eine physische Transportschnittstelle mit einer eigenen IP-Adresse verfügen, die im DNS registriert ist. Dies erschwert die Lokalisierung eines Gatekeepers über DNS.
- **Mehrpunktkommunikation:** Die MC ist ein optionales Element innerhalb der H.323-Spezifikation, wodurch entweder bestehende Verbindungen zwischen zwei Endpunkten bei fehlendem MC nicht zu Multipoint-Konferenzen erweitert werden oder Multipoint-Konferenzen werden plötzlich beendet, falls das einzige Terminal, das über einen MC verfügt, die Konferenz verlässt.
- **Performance:** Bestehende Multipoint-Konferenzen können zu hohen Serverbelastungen führen, da für jeden Gesprächsteilnehmer eine TCP-Verbindung aufrechterhalten werden muss und die Signalisierungsnachrichten jeder Verbindung verarbeitet werden müssen. Mit steigender Teilnehmerzahl wächst daher die Notwendigkeit, das Routing und die Signalverarbeitung entsprechend der Belastung anzupassen und zu verteilen.

- **Adressierung:** Ursprünglich wurde H.323 für die Verwendung in lokalen Netzwerken konzipiert. Dieser Mechanismus stößt allerdings schnell an seine Grenzen, wenn man das zugrunde liegende LAN-Konzept auf das gesamte Internet erweitert. Darüber hinaus umfasst H.323 durch die Verwendung von Gateways auch die Zusammenarbeit mit öffentlichen Telefonnetzen. Diese Erweiterungen werfen eine Reihe von Fragen hinsichtlich der Adressierung von Endpunkten, Gatekeepern und Gateways auf.

Die derzeitigen Schwachpunkte bei der Adressierung von H.323-Komponenten könnten durch Einbeziehung von Verzeichnisdiensten wie z.B. LDAP[65] verbessert werden. Dazu sind zentrale und allgemein bekannte Datenbanken notwendig, in denen Adressinformationen von Anwendern, Gatekeepern und Gateways gespeichert sind. Ferner sollten die Datenbankinhalte ständig aktualisiert werden, um auch Informationen über H.323-Komponenten mit dynamischen Adressen wie z.B. Terminals mobiler Benutzer oder Gatekeeper ohne feste IP-Adresse bereitzustellen. Als Suchkriterien könnten Attribute wie Vor- und Nachname, Ländercode, E-Mail-Adressen, E.164 Nummern etc. dienen.

5.5.4 Bewertung des Standards H.323

Abschließend erfolgt in diesem Kapitel eine kurze Bewertung des Standards H.323, um Schwachstellen und Standardisierungslücken aufzuzeigen. Dafür werden die folgenden Bereiche verglichen:

- Kompatibilität
- Interoperabilität
- Erweiterbarkeit
- Skalierbarkeit
- Standardisierung

Komplexität Die H.323-Empfehlung ist eine sehr umfangreiche Spezifikation, die zusammen mit der H.245- und H.225.0-Empfehlung über 500 Seiten umfasst. Dazu kommen weitere einbezogene Empfehlungen wie beispielsweise die Kodierungen für Audio, Video und Daten sowie erweiterte Dienste durch H.450.x oder Security durch H.235. Die Einarbeitung und Umsetzung des H.323-Standards durch Entwickler und Hersteller ist daher sehr aufwendig und unüberschaubar. Hinzu kommt, dass einbezogene Empfehlungen und Standards nicht deutlich voneinander abgegrenzt sind und dadurch verschiedene Aktionen mehrere Protokolle betreffen oder ähnliche Funktionalitäten in unterschiedlichen Protokollen spezifiziert sind.[66]

65 Lightweight Directory Access Protocol

66 Z.B. RTCP und H.245 mit Kontrollfunktionen und Feedback für Multipoint-Konferenzen

Die Nachrichten in H.323 werden mit ASN.1[67] in binäre Darstellung transformiert. Zum Dekodieren und Lesen der Nachrichten sind im Gegensatz zur textbasierten Darstellung von Nachrichten wie beispielsweise in HTTP[68] zusätzliche Werkzeuge notwendig. Für die Gesprächseinrichtung zwischen zwei oder mehreren Endpunkten können unterschiedliche H.323-Prozeduren wie die Standardprozedur, Fast-Start oder Tunneling ablaufen. Zur Unterstützung aller Verfahren sind komplexe Implementierungen in H.323-Terminals, -Gateways, -Gatekeepern und Firewall-Systemen notwendig. Zusätzlicher Implementierungsaufwand entsteht durch die Unterstützung dynamischer Ports.

Der Aufbau einer einfachen Punkt-zu-Punkt-Verbindung zwischen zwei Endpunkten ohne Beteiligung eines Gatekeepers benötigt eine große Anzahl von Round-Trips, die bei komplexeren Verbindungen über Gatekeeper und Gateways in Multipoint-Konferenzen ansteigen und zusätzliche Verzögerungen verursachen. Diese Verzögerungen nehmen dann wieder erheblichen Einfluss auf die Sprachqualität.

Interoperabilität

Durch die Vielzahl der in H.323 einbezogenen Protokolle können Entwickler von H.323-Komponenten eine Vielzahl von Funktionalitäten hinzufügen, was dazu führt, dass Produkte zwar entsprechend der H.323-Empfehlung protokollkonform realisiert werden, jedoch nicht zwingend mit anderen H.323-Produkten zusammenarbeiten. Ferner bestehen teilweise unterschiedliche Auffassungen bei der Interpretation von umzusetzenden Elementen verschiedener Protokolle, sodass Paketformate und Kodierungen unterschiedlich sein können.

Erweiterbarkeit

Aufgrund der schnellen Entwicklung im Bereich des Internets ergeben sich neue Anwendungsmöglichkeiten, die von entsprechenden Protokollen unterstützt werden müssen. Deshalb ist ein entscheidendes Kriterium für die Beurteilung eines VoIP-Protokolls dessen Erweiterbarkeit und Anpassungsfähigkeit.

Anwendungsentwickler und Hersteller können die Funktionen von H.323 durch Eigenentwicklungen erweitern, die in non_Standard_Parametern der H.245-Empfehlung mit den entsprechenden Herstellercodes angezeigt werden. Nachteilig ist, dass H.323 keine Mechanismen bereithält, mit denen jedes Terminal seine einzelnen Erweiterungen mitteilen kann. Aufgrund der Abwärtskompatibilität von H.323 wird die Anzahl der zu unterstützenden und einzubeziehenden Empfehlungen immer größer, da im Laufe der Zeit neue Verfahren wie beispielsweise für Kodierung von Audio und Video hinzukommen, während alte und überholte Verfahren weiterhin unterstützt werden müssen. Dadurch wird die Implementierung von H.323 immer aufwendiger.

67 Abstract Syntax Notation
68 Hyper-Text Transport Protocol

H.323 ist weniger eine modulare als vielmehr eine vertikal angelegte Protokollfamilie, in welcher verschiedene Protokolle miteinander verflochten sind. Diese Eigenschaft erschwert das Ersetzen überholter Funktionen und funktionale Erweiterungen wie z.B. garantierte Dienstgüte oder Verzeichnisdienste.

Skalierbarkeit Ursprünglich wurde H.323 für den Einsatz in lokalen Netzwerken konzipiert. Erst in der zweiten Version erfolgte eine Erweiterung für allgemein paketbasierte Netze. Durch das Zonenkonzept können E-Mail-Adressen zur Adressierung verwendet werden. Dennoch können Skalierbarkeitsprobleme aufgrund der komplexen Domainstruktur auftreten. Ein Gatekeeper beispielsweise mit einem bestimmten Domainnamen muss nicht zwingend über eine eigene IP-Adresse verfügen, die im DNS[69] registriert ist. Dies erschwert die Lokalisierung eines Gatekeepers über DNS.

Für die Kommunikation zwischen drei und mehr Terminals innerhalb einer Multipoint-Konferenz ist ein MC[70] erforderlich. Da dieser jedoch ein optionales Element innerhalb der H.323-Spezifikation ist, können entweder bestehende Verbindungen zwischen zwei Endpunkten bei fehlendem MC nicht zu Multipoint-Konferenzen erweitert werden oder Multipoint-Konferenzen werden plötzlich beendet, falls das einzige Terminal, das über einen MC verfügt, die Konferenz verlässt.

Weiterhin können bestehende Multipoint-Konferenzen zu hohen Serverbelastungen führen, da man für jeden Gesprächsteilnehmer eine bidirektionale TCP-Verbindung aufrechterhalten muss und die Signalisierungsnachrichten jeder Verbindung verarbeitet werden müssen. Mit steigender Teilnehmerzahl wächst daher die Notwendigkeit, das Routing und die Signalverarbeitung entsprechend der Belastung anzupassen und zu verteilen. H.323 bietet die Möglichkeit, die Verwaltung von H.245-Kontrollkanälen durch Kaskadierung auf mehrere MCs aufzuteilen. Zunächst wird eine Verbindung zwischen H.323-Einheiten aufgebaut, die über einen oder mehrere MCs verfügen. Ist ein H.245-Kontrollkanal eingerichtet und ein aktiver MC gemäß der Master-/Slave-Prozedur bestimmt worden, kann dieser je nach Bedarf weitere in den H.323-Einheiten vorhandene MCs zur Kontrolle bestimmter H.245-Kanäle aktivieren. Offen bleibt, wie die Signalverarbeitung von Audio-, Video- und Datenströmen innerhalb von Multipoint-Konferenzen unterschiedlichen Belastungen angepasst werden kann.

Standardisierung Die Empfehlung H.323 beschreibt die technischen Erfordernisse für Multimedia-Kommunikationssysteme über paketbasierte Netze. Die aktuelle Version von H.323 lässt jedoch noch einige Punkte offen, die in einer Nachfolgeversion spezifiziert werden sollten, um den neuen Anforderungen gerecht zu werden.

69 Domain Name Service
70 Mulipoint Controller

Ursprünglich wurde H.323 für die Verwendung in lokalen Netzwerken konzipiert, was sich in den aktuellen H.323-Adressierungsmechanismen widerspiegelt. Ein Endpunkt kann entweder einen anderen Endpunkt direkt anrufen, falls er dessen IP-Adresse kennt, oder er muss die Anfrage Location_Reject an den Gatekeeper schicken, die eine Endpunktbezeichnung des gewünschten Endpunkts beinhaltet, bei dem sich der anfragende Endpunkt registriert hat. Der Gatekeeper versucht daraufhin, anhand einer Tabelle die der Endpunktbezeichnung entsprechende IP-Adresse herauszufinden. Dies ist jedoch nur dann möglich, falls sich der gesuchte Endpunkt zuvor bei demselben Gatekeeper mit seiner Endpunktbezeichnung und eigener IP-Adresse registriert hat. War die Adressauflösung erfolgreich, teilt der Gatekeeper in der Nachricht Location_Confirm die Gesprächssignalisierungsadresse des gewünschten Endpunkts mit. Daraufhin kann der anrufende Endpunkt eine Setup-Nachricht an die erhaltene Adresse schicken.

Dieser Mechanismus stößt allerdings schnell an seine Grenzen, wenn man das zugrunde liegende LAN-Konzept auf das gesamte Internet erweitert, auf das sich die zweite Version von H.323 erstreckt. Darüber hinaus umfasst H.323 durch die Verwendung von Gateways auch die Zusammenarbeit mit öffentlichen Telefonnetzen. Diese Erweiterungen werfen eine Reihe von Fragen hinsichtlich der Adressierung von Endpunkten, Gatekeepern und Gateways auf. Als Beispiel sei ein Endpunkt gegeben, der einen Gatekeeper sucht. Also schickt er die Nachricht Gatekeeper_Request an die Multicast-Adresse 224.0.1.41 mit der Portnummer 1718 und erwartet die Nachricht Gatekeeper_Reject. Folgende Punkte bleiben dabei unberücksichtigt:

- Der gefundene Gatekeeper ist nicht in der Lage, aus dem Gatekeeper-Discovery-Prozess heraus optimal auf die Entfernung und Bandbreite Rücksicht zu nehmen, da gemäß H.323 eine Zone unabhängig von der Netztopologie ist.
- Es können bei größeren Entfernungen zwischen Endpunkten und Gatekeeper zusätzlich unnötige Verzögerungen entstehen.
- Es kann ein Endpunkt oder Gatekeeper nicht die Adresse eines anderen Endpunkts herausfinden, der sich bei keinem Gatekeeper registriert hat.
- Ein Endpunkt oder Gatekeeper kann ebenfalls nicht die Adresse eines anderen Gatekeepers herausfinden, bei dem ein gesuchter Endpunkt registriert ist.
- Ein Endpunkt oder ein Gatekeeper ist nicht in der Lage, ein Gateway für einen bestimmten Terminaltyp (z.B. H.320 oder H.324) zu lokalisieren.
- Ein Gateway kann bei einem eingehenden Anruf aus dem Internet keinen passenden Gatekeeper ermitteln, der eine E.164-Nummer in die entsprechende IP-Adresse eines Endpunkts übersetzen kann.

Die derzeitigen Schwachpunkte bei der Adressierung von H.323-Komponenten könnten durch Einbeziehung von Verzeichnisdiensten wie beispielsweise LDAP[71] verbessert werden. Dazu sind zentrale und allgemein bekannte Datenbanken notwendig, in denen Adressinformationen von Anwendern, Gatekeepern und Gateways gespeichert sind. Ferner sollten die Datenbankinhalte ständig aktualisiert werden, um auch Informationen über H.323-Komponenten mit dynamischen Adressen wie beispielsweise Terminals mobiler Benutzer oder Gatekeeper ohne feste IP-Adresse bereitzustellen. Als Suchkriterien könnten Attribute wie Vor- und Nachname, Ländercode, E-Mail-Adressen, E.164-Nummern etc. dienen.

Die Signalisierung von H.323-Anrufen ist auf Unicast beschränkt. Das heißt, ein Anrufer kann andere Gesprächsteilnehmer nur einzeln zu einem Gespräch einladen. Mit Multicast-Signalisierung könnte ein Anwender gleichzeitig eine Vielzahl von Personen anrufen, indem er Signalisierungsnachrichten an eine bekannte Multicast-Adresse schickt, die von mehreren Personen abgehört werden kann. In einer hybriden Multipoint-Konferenz mit zentraler Verarbeitung von Audio können bei H.323 nur die Endpunkte die dezentral verarbeiteten Videoströme über Multicast untereinander austauschen. Es besteht keine Möglichkeit, dass auch ein MP[72] die verarbeiteten Audioströme über Multicast an die Endpunkte weiterleitet.

Bevor ein MP die zentral verarbeiteten Audioströme aller Konferenzteilnehmer an die beteiligten Terminals weiterleitet, werden aus dem ausgehenden Audiostrom, den jeweils ein Terminal empfängt, die von diesem Terminal gesendeten Audiosignale herausgefiltert, um Echoeffekte und unnötige Übertragung von Daten zu vermeiden. Nachteilig ist dabei, dass ein MP bei großen Konferenzen an seine Kapazitätsgrenzen stoßen kann oder zusätzliche Verzögerungen entstehen. Man könnte diese Nachteile dadurch vermeiden, dass die Terminals selbst ihre gesendeten Audiosignale aus dem empfangenen Audiostrom des MP herausfiltern und somit zur Entlastung des MP beitragen. Erfolgt der Verbindungsaufbau zwischen zwei Endpunkten mit Beteiligung eines Gatekeepers, wird der H.245-Kontrollkanal über den Gatekeeper geleitet. H.323 beschreibt zwar den Fall, dass der H.245-Kontrollkanal auch direkt zwischen den Endpunkten eingerichtet werden kann, das wurde aber bislang nicht realisiert.

Soll die Kommunikation zwischen H.323-Komponenten mit verschlüsselten Informationsströmen erfolgen, müssen sich Endpunkte zuvor authentifizieren. Dabei wird davon ausgegangen, dass Gatekeeper oder MCU[73] tatsächlich vertrauenswürdige H.323-Einheiten darstellen. Momentan gibt es noch keine Mechanismen zur Authentifizierung von Gatekeepern oder MCUs, um

71 Lightweight Directory Access Protocol
72 Multipoint Prozessor
73 Multipoint Control Unit

unerlaubtes Abhören der Informationsströme zu verhindern. Auch die Kommunikation zwischen den Teilnehmern wird nicht verschlüsselt, sodass sämtlicher Telefonverkehr eines VoIP[74]-Netzes abgehört werden könnte.

Ein Netzwerkendpunkt kann über mehrere gebündelte Kanäle im PSTN[75] eine Verbindung aufbauen. Zu diesem Zweck sendet er in der Setup-Nachricht mehrere E.164- oder Party_Number-Adressen an das entsprechende Gateway. Bei einem Anruf ins ISDN-Netz mit 2x64 Kbit/s-Leitungen beispielsweise gibt der Endpunkt zwei E.164-oder Party_Number-Adressen in der Setup-Nachricht an. Das Gateway versucht anschließend, die gewünschte Verbindung mit der entsprechenden Anzahl von Kanälen herzustellen. Falls allerdings der Verbindungsaufbau zu einer einzigen Adresse nicht gelingt, bricht das Gateway den gesamten Verbindungsaufbau ab und schickt die Nachricht Release Complete an den Endpunkt zurück. Das liegt daran, dass für den gesamten Verbindungsaufbau nur ein einziger Call Reference Value (CRV) zur Verfügung steht und somit die Bezugnahme auf jeden einzelnen Kanal unmöglich ist. Benötigt werden daher mehrere CRVs, damit jeder Kanal individuell ausgehandelt werden kann und ein Gespräch auch dann zustande kommt, wenn der Verbindungsaufbau zu einer oder mehreren Adressen nicht gelingt. Darüber hinaus fehlt zur Zeit noch ein Mechanismus, um während eines bestehenden Gesprächs weitere Kanäle im PSTN hinzuzufügen. Hinzu kommt, dass Gatekeeper momentan noch nicht die Reservierung von Bandbreite für Terminals übernehmen können, welche diese Funktion nicht unterstützen.

H.323-Einheiten sind derzeit auf die Verwendung festgelegter und standardisierter Codecs beschränkt, sodass viele existierende Codecs nicht verwendet werden können. Im Sinne einer offenen Architektur sind Mechanismen erforderlich, welche die Aushandlung beliebiger Codecs zwischen H.323-Endpunkten erlauben, um zu einer Verbesserung der Interoperabilität beizutragen.

Ebenfalls ist man aufgrund der wachsenden Bedeutung der Mobiltelefonie bestrebt, für eine umfassende Zusammenarbeit von VoIP mit bestehenden Telefonnetzen, entsprechende Gateways zur Konvertierung der Informationsströme zu entwickeln und in H.323 einzubeziehen. Mit H.323 kann ein Anwender mehrere Kontaktadressen besitzen. Wird er unter einer bestimmten Adresse angerufen und hält sich an einem anderen Ort auf, sendet er eine Facility-Nachricht mit alternativen Kontaktadressen weiterer Terminals an den anrufenden Endpunkt zurück. Darüber hinaus wäre ein Mechanismus zur automatischen Rufweiterleitung wünschenswert. Das würde zu geringeren Verzögerungen führen, da nicht erst der Verbindungsaufbau zur ersten Adresse mit der Nachricht Release Complete abgebrochen und anschließend eine neue Prozedur für den Verbindungsaufbau zu anderen Adressen gestartet werden müsste.

74 Voice-over-IP

75 Public Switched Telecommunication Network

Ebenfalls noch nicht realisiert sind Dienste, mit denen ein Anrufer während des Verbindungsaufbaus Präferenzen angeben kann, ob er an ein bestimmtes Terminal, einen Anrufbeantworter, einen Festnetzanschluss oder ein Mobiltelefon weitergeleitet werden möchte. Als Präferenz könnte der Anrufer ferner angeben, welche Priorität das Gespräch hat, in welcher Sprache das Gespräch stattfinden soll und ob es sich um ein privates oder geschäftliches Telefonat handelt. [DOMM98]

5.5.5 Erweiterungen von H.323

Im Anhang der H.323-Empfehlung werden Mechanismen zur Reservierung von Ressourcen für die Erfüllung von Dienstgüteanforderungen von Audio- und Videoströmen in Echtzeit spezifiziert. Obwohl verschiedene Protokolle für diese Aufgabe unter Einhaltung bestimmter Grundoperationen herangezogen werden können, werden in H.323 die Prozeduren von RSVP[76] nach RFC-2205 empfohlen und beschrieben.

Wenn ein Endpunkt eine Zugangsanfrage an einen Gatekeeper schickt, teilt er in der ARQ-Nachricht mit, ob er in der Lage ist, Ressourcen zu reservieren. Der Gatekeeper entscheidet daraufhin aufgrund der Informationen über den Endpunkt und das zugrunde liegende Netz, ob entweder der Endpunkt die Reservierung oder der Gatekeeper die Reservierung für den Endpunkt übernimmt bzw. überhaupt eine Reservierung notwendig ist. Die Entscheidung wird dem Endpunkt in der Nachricht Admissions_Confirm mitgeteilt. Ein Gatekeeper kann einen Netzzugang mit Admissions_Reject ablehnen, falls eine Reservierung notwendig ist und der Endpunkt über keine Reservierungsmechanismen verfügt. Signalisiert werden die auszutauschenden Informationen im Feld Transport_QOS, das in die entsprechenden H.225.0-Nachrichten des RAS-Kanals eingefügt wird. Dabei tauschen Endpunkt und Gatekeeper auch Informationen über die gewünschte und mögliche Bandbreite aus.

Alternativ zur Beteiligung eines Gatekeepers kann der Austausch von RSVP-Nachrichten auch direkt zwischen den Endpunkten ablaufen. Zu unterscheiden sind Reservierungen in Punkt-zu-Punkt- und Multicast-Verbindungen. In beiden Fällen erfolgt der Austausch der RSVP-Parameter im Feld QOS_Capability der Nachricht Open_Logical_Channel über den H.245-Kanal.

Bei Punkt-zu-Punkt-Verbindungen teilt der empfangende Endpunkt dem Sender seine Port-ID in der Antwort Open_Logical_Channel_ACK mit. Anschließend schickt der Sender die PATH-Nachricht von RSVP an die Port-ID des Empfängers, der daraufhin mit der RESV-Nachricht antwortet. Wenn der Empfänger die Bestätigung der Reservierung mit der RSVP-Nachricht RESV_CONF des Senders erhalten hat, ist die Reservierung eingerichtet. Optional kann der Empfänger mit den Nachrichten Flow_Control_To_Zero und Flow_Control_Command dem Sender anzeigen, dass dieser seine Pakete erst

76 Resource Reservation Protocol

nach vollständiger Reservierung verschicken soll. Bevor ein Endpunkt das Schließen eines logischen Kanals mit bestehender Reservierung durch die Nachricht Close_Logical_Channel einleitet, sendet er zuvor die Nachricht Path_Tear an den Empfänger. Nachdem dieser die Mitteilung Close_Logical_Channel erhalten hat, schickt er dem Sender eine RESV_TEAR-Nachricht zurück, bevor er mit Close_Logical_Channel_ACK den logischen Kanal schließt.

Bei der Reservierung mittels Multicast tritt ein empfangender Endpunkt einer Multicast-Gruppe hinzu und verbindet sich mit der Quelle des Multicast-Baumes durch das Versenden der IGMP-Reportnachricht, nachdem er die Mitteilung Open_Logical_Channel des Senders erhalten hat. Anschließend schickt er die Bestätigung Open_Logical_Channel_ACK an den Sender zurück. Im Gegensatz zur Punkt-zu-Punkt-Verbindung legt der Sender mit Open_Logical_Channel die Port-ID des Empfängers fest. Der Austausch der sich anschließenden RSVP-Nachrichten erfolgt identisch zur Punkt-zu-Punkt-Verbindung. Zum Schließen eines logischen Kanals versendet der Sender vor der Nachricht Close_Logical_Channel die RSVP-Mitteilung PATH_TEAR, vorausgesetzt, es handelt sich um den letzten offenen logischen Kanal innerhalb des Multicast-Stroms. Nachdem der Empfänger die Nachricht Close_Logical_Channel erhalten hat, sendet er einen RESV_TEAR und die IGMP-Nachricht Leave, bevor er mit Close_Logical_Channel_ACK den logischen Kanal schließt.

Die ITU-Empfehlung H.235 beschreibt Sicherheitsmechanismen wie Authentifizierung, Datenintegrität und Datenverschlüsselung, die in H.323 zum Schutz von Punkt-zu-Punkt- und Multipoint-Konferenzen implementiert werden können:

- **Authentifizierung** ist ein Mechanismus zur Identifizierung von Endpunkten, die an einer Konferenz teilnehmen.
- Unter der **Datenintegrität** ist der Schutz gegen Veränderung von Datenpaketen zu verstehen, die während einer Verbindung zwischen zwei H.323-Endpunkten transportiert werden.
- **Datenverschlüsselung** verhindert das Abhören von VoIP-Paketen mit Hilfe unterschiedlicher Ver- und Entschlüsselungsalgorithmen.

Diese Sicherheitsmechanismen basieren auf der Verwendung von Schlüsseln, mit deren Hilfe Anwender identifiziert, Pakete vor Veränderungen geschützt und Datenströme ver- bzw. entschlüsselt werden können. Der Austausch von Schlüsseln zwischen H.323-Endpunkten kann durch unterschiedliche Methoden stattfinden, die jedoch nicht Gegenstand von H.235 sind. Unter Out-of-band versteht man den Schlüsselaustausch über E-Mail, verbalen Kontakt oder andere Methoden außerhalb von H.323. Da E-Mails und leitungsvermittelte Telefonverbindungen abgehört werden können, kann diese Art der Schlüsselübermittlung unter Umständen unsicher sein. Aus diesem Grund müssen Verfahren zum Einsatz kommen, die ein geteiltes Geheimnis zwischen den

Kommunikationspartnern unterstützen, wie die Methode nach Diffie-Hellman. Eine Verteilung der erzeugten Schlüssel muss über Key-Management-Verfahren vorgenommen werden, wenn VoIP in größeren Umgebungen eingesetzt wird, da es sonst zu Verwaltungs- und Sicherheitsproblemen kommen kann. Da die Kommunikation über H.323 in mehreren Phasen mit verschiedenen Protokollen abläuft, kann sich die Sicherung von Datenpaketen nicht allein auf RTP[77]-Medienströme beziehen, sondern muss sich auch auf H.225.0-, Q.931- und H.245-Datenpakete erstrecken. Aus Sicherheitsgründen ist ebenfalls die uni- oder bidirektionale Authentifizierung von Gatekeepern und Endpunkten sowie der Schutz vor Veränderung der RAS-Nachrichten spezifiziert worden. Maßnahmen zum Schutz vor Veränderung der H.225.0-RAS-Nachrichten sind in H.235 ebenfalls vorhanden sowie Ver- bzw. Entschlüsselungstechniken für RAS-Nachrichten. [H.235(98)]

Für den Verbindungsaufbau ist die Authentifizierung von Endpunkten Inhalt von H.235. Die Gewährleistung der Integrität und Vertraulichkeit muss über Transport Level Security (TLS) oder IPsec erfolgen. [Q.931(98)] H.245-Kontrollnachrichten können mit den gleichen Mechanismen wie Q.931-Nachrichten gesichert werden. Bereits in der Phase des Verbindungsaufbaus können im Element H245_Security_Capability der Q.931-Setup-Nachricht mögliche Sicherheitsmechanismen für den H.245-Kontrollkanal mitgeteilt werden. Auch der Austausch der Public Keys für die Verschlüsselung der Medienströme kann in Q.931-Nachrichten erfolgen.

Die einzelnen RTP-Pakete (Audio oder Video) werden zunächst für den Transport segmentiert und vor der Übertragung verschlüsselt. Die Entschlüsselung erfolgt entsprechend der verwendeten und über den H.245-Kanal ausgetauschten Schlüssel. Je anspruchsvoller der verwendete Sicherheitsalgorithmus ist, um so höher werden dabei natürlich die durch Ver- bzw. Entschlüsselungsprozesse verursachten Verzögerungen. [H.245(99)]

Die Empfehlungen der Serie H.450.x spezifizieren außerdem erweiterte Dienste, die von H.323-Komponenten implementiert werden können. H.450.1 beschreibt die Prozeduren und das Signalisierungsprotokoll zwischen H.323-Einheiten für die Kontrolle von erweiterten Diensten und Prozeduren, die in H.450.2 und H.450.3 detailliert dargestellt werden. Zusätzlich können proprietäre Lösungen von Herstellern hinzugefügt werden. [H.450.1(98)]

5.5.6 SIP contra H.323

Das Signalisierungsprotokoll SIP[78] stellt im Vergleich zu H.323 eine Alternative für den Verbindungsaufbau zum Telefonieren über paketbasierte Netze dar. Es ist im Gegensatz zu H.323 leichter zu implementieren, da aufgrund der Text-

77 Real-Time Protocol
78 Session Initiation Protocol

kodierung keine aufwendigen Codegeneratoren nötig sind. Für eine einfache Implementierung genügen im Grunde vier Header-Felder (To, From, Call-ID und C_{seq}) und drei Request-Typen (INVITE, ACK und BYE). Die Spezifikationen von SIP und SDP[79] sind wesentlich kompakter als in H.323, was nicht zuletzt auch die Einarbeitung und Implementierung durch Entwickler vereinfacht.

Folgende Merkmale unterscheiden weiterhin beide Ansätze voneinander:

1. **Aufbau:** Das Protokoll SIP ist im Gegensatz zum vertikalen Aufbau von H.323 modular aufgebaut. SIP ist verantwortlich für die Gesprächssignalisierung und die Lokalisierung von Anwendern und deren Registrierung. Für die Dienstgüte, Verzeichniszugriffe, inhaltliche Beschreibung von Sitzungen und Konferenzkontrolle sind zusätzliche Protokolle im Einsatz. Durch den modularen Aufbau von SIP ist eine Zusammenarbeit mit H.323 möglich. Ein Anrufer kann einen gewünschten Anwender über SIP lokalisieren und diesem anschließend signalisieren, dass ein Gespräch unter H.323 stattfinden soll. Mit der Zusammenarbeit von SIP/SDP/SAP[80] und H.323 beschäftigt sich die Arbeitsgruppe Multiparty Multimedia Session Control der IETF.
2. **Codecs:** In H.323 werden zentral registrierte Audio und Video Codecs verwendet. Diese Art von Beschränkung besteht bei SIP nicht. Mit SDP teilen sich Endpunkte die unterstützten und zu verwendenden Codecs mit, die über Strings identifiziert werden und von Personen und Gruppen registriert werden können. Dadurch kann SIP mit allen Codecs zusammenarbeiten.
3. **Transport:** SIP kann sowohl über TCP[81] als auch über UDP[82] transportiert werden. Im Fall von UDP können Server stark entlastet werden, da nach dem Verbindungsaufbau keine TCP-Verbindungen aufrechterhalten werden müssen, die die Ressourcen stark belasten können. Ein SIP-Server empfängt Anfragen, führt Operationen durch, leitet Anfragen entsprechend weiter und erfüllt anschließend andere Aufgaben. Im Gegensatz dazu müssen H.323-Gatekeeper, welche das Routing von Gesprächen übernehmen, für die Dauer des Gesprächs die Informationsströme verarbeiten und bei mehreren Teilnehmern eine Vielzahl von TCP-Verbindungen aufrechterhalten, was zu einer starken Belastung führen kann.
4. **Multicast:** Ebenfalls bietet H.323 keine Möglichkeit, Anrufe über Multicast-Adressen zu signalisieren. Mit SIP kann jeder Anwender Anfragen zu Multicast-Adressen verschicken. Dadurch ergeben sich unterschiedliche Anwendungen für Multicast-Einladungen. Ein Anwender kann beispiels-

79 Session Description Protocol
80 Session Announcement Protocol
81 Transmission Transport Protocol
82 User Datagram Protocol

weise eine Gruppe von Freunden zu einem Gespräch einladen. Die Gruppe wird durch eine Bezeichnung wie beispielsweise intellect-group@decoit.de beschrieben, der eine bestimmte Multicast-Adresse zugeordnet ist. Jeder Angehörige der Gruppe hält an dieser Adresse nach eingehende Einladungen Ausschau und kann diese entgegennehmen.

5. **Verbindungsaufbau:** Der Verbindungsaufbau mit H.323 erfolgt mit den Protokollen H.225.0, Q.931 und H.245, was zu einer großen Anzahl von Round-Trips mit entsprechenden Verzögerungen führt. Im Vergleich dazu sind die Verzögerungen beim Verbindungsaufbau mit SIP geringer, da eine SIP-Anfrage alle notwendigen Informationen enthält und daher entsprechend weniger Round-Trips nötig sind.
6. **Verzeichnisdienste:** Während die Zusammenarbeit von H.323 mit Verzeichnisdiensten noch nicht detailliert genug ist, arbeitet SIP mit DNS, LDAP und programmierbaren Verzeichnisdiensten bereits zusammen. DNS ist nicht geeignet für individuelle Anfragen nach Anwendern. SIP kann DNS jedoch dazu verwenden, um Zusammenhänge zwischen Domain- und Servernamen für bestimmte Anfragen zu ermitteln. Mit LDAP kann ein SIP-Server in Datenbanken anhand bestimmter Attribute suchen und als Antwort mehrere Adressen erhalten, an welche er Anfragen weiterleitet. Bei SIP-Suchanfragen an programmierbare Verzeichnisdienste kann die Antwort von der Identität des Anrufers, dem zu sendenden oder zu empfangenen Medientyp, der Tageszeit oder anderen Faktoren abhängen.
7. **Telefondienste:** SIP ermöglicht im Vergleich zu H.323 zusätzliche Telefondienste durch die Verwendung von einfachen und standardisierten Mechanismen. Einer dieser Dienste ist Blind Transfer. In diesem Szenario befindet sich ein Anwender *A* im Gespräch mit einem anderen Anwender *B*. *A* übergibt nun das Gespräch Anwender *C*, ohne zu wissen, ob die Übergabe erfolgreich war. Ein anderer Dienst ist Operator-Assisted Call Transfer. Mit diesem Dienst kann eine Sekretärin ein Gespräch von einem Anwender entgegennehmen, bei ihrem Chef das Gespräch anmelden und anschließend übergeben. Die Sekretärin kann sich am Gespräch beteiligen oder die Konferenz verlassen. Eine Variation zu diesem Dienst ist Autodialer. Der Autodialer übernimmt die Rolle der Sekretärin und ruft einen potenziellen Kunden an. Hat der Kunde geantwortet, ruft der Autodialer einen dritten Anwender an, den Telemarketer. Danach verbindet der Autodialer den Kunden mit dem Telemarketer und verlässt das Gespräch. Ein weiterer Dienst ist der Click to Call Service, der es dem Besucher einer Webseite erlaubt, auf einen Button zu klicken, um ein Telefonat ins öffentliche Telefonnetz, z.B. zu einem Support Service, zu führen. Das Kontrollprotokoll für diesen Dienst ist SIP.

8. **Mobilität:** Ein Anwender ist möglicherweise nicht ständig in der Nähe eines bestimmten Terminals, um Anrufe entgegenzunehmen, sondern hält sich an verschiedenen Orten auf. In H.323 gibt es die Möglichkeit, mit der Facility-Nachricht Anrufern alternative Kontaktadressen anderer Terminals mitzuteilen. Nachteilig dabei ist, dass keine zusätzlichen Informationen und Präferenzen für die Kontaktaufnahme angegeben werden können. Bei Verwendung von SIP kann ein Anwender wie bei H.323 über mehrere Kontaktadressen verfügen, unter denen er erreichbar ist. Ein Anrufer kann an verschiedene Adressen weitergeleitet werden, wobei im Gegensatz zu H.323 für jede Kontaktadresse weitere Informationen übermittelt werden können, z.B. Adressen für Geschäfts- oder Privatgespräche und Weiterleitung an einen Mobil- oder Festanschluss.

5.6 Zusammenfassung

Als Arbeitsgruppe des International Multimedia Teleconferencing Consortium (IMTC)[83] beschäftigt sich das VoIP-Forum mit offenen Fragen, die durch die H.323-Spezifikation nicht geklärt worden sind. Es wird jedoch nicht die Absicht verfolgt, als Konkurrenz zu H.323 aufzutreten. Vielmehr geht es um die Kombination und Ergänzung der bestehenden Standards. Das IMTC hat es sich als nichtkommerzielle Organisation zum Ziel gesetzt, ein Forum für alle Organisationen zu bieten, die sich mit der Entwicklung von Telekommunikationsstandards für multimediale Anwendungen und Dienste beschäftigen. Ferner wird eine einheitliche industrieweite Umsetzung wichtiger Standards vorangetrieben. Dies ist auch die Zielsetzung des Voice-over-IP-Forums, welches sich, aufbauend auf dem H.323-Standard, mit weiteren noch nicht gelösten Problemstellungen auseinandersetzt.

Die Aktivitäten lassen sich dabei in folgende Gebiete einteilen:

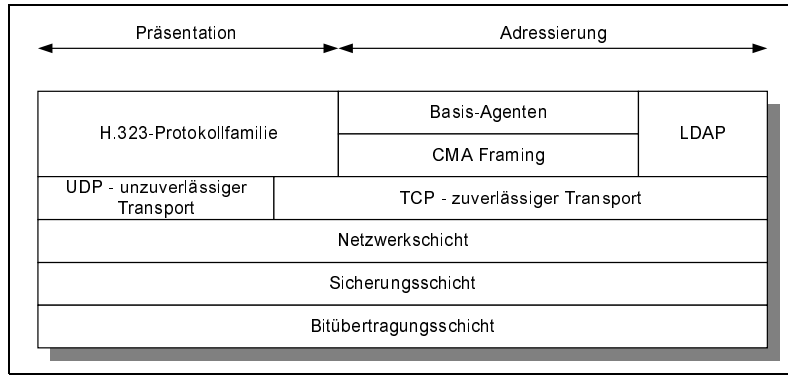
- Definition und Aufstellung von offenen und konsistenten Richtlinien für Implementierungen im Bereich der IP-Telefonie
- Ermöglichung von Dienstgüte und Interoperabilität zwischen Produkten unterschiedlicher Hersteller
- Definition eines Adressierungsmechanismus zur Lokalisierung von Gateways und Anwendern

Unter anderem erarbeitet das VoIP-Forum das Referenzmodell Service Interoperability Implementation Agreement. Basis dieses Modells bilden die Standardprozeduren von H.323 wie Gesprächssignalisierung, Verbindungsaufbau und Fähigkeitsaustausch. Stärker berücksichtigt als in H.323 wird die Übertragung von DTMF[84]-Signalen, um die Zusammenarbeit mit dem traditionellen

83 http://www.imtc.org

Telefonsystem zu verbessern. Ferner werden Aspekte wie Kompatibilität zwischen H.323-Version 1 und 2, Sprachkodierung und Erkennung von Ruhephasen behandelt. Darüber hinaus spezifiziert das Modell ein Call Management Agent System, das Verbindungsdienste zur Unterstützung von Gesprächen zwischen paket- und leitungsvermittelten Netzen bietet. Zu diesen Diensten gehören das Management verschiedener Terminaladressen von Personen oder Organisationen, die dynamische Adressauflösung von IP-Telefonie-Adressen durch Einbeziehung bestehender Verzeichnisdienste wie LDAP und schließlich das intelligente Weiterleiten von Gesprächen entsprechend bestimmter Informationen von Agenten.

Abb. 5.23
VoIP-Schichtenmodell



Das European Telecommunications Standards Institute (ETSI) hat das Projekt Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)[85] initiiert, da die Technologie der IP-Telefonie noch nicht ausgereift genug ist und gemeinsame Lösungen für die Industrie benötigt werden. Das allgemeine Ziel ist die Zusammenarbeit von IP-Telefonie mit den leitungsvermittelten Telefonnetzen wie PSTN[86] und GSM[87]. Die Philosophie hinter TIPHON besteht in der Schaffung einer offenen Konzeption mit der Konzentration auf Spezifikationen, um eine weltweite Akzeptanz und Anwendung der IP-Telefonie zu erreichen. Eine breite Anerkennung soll unter anderem auch durch die Zusammenarbeit mit anderen Organisationen wie ITU[88], IETF[89], IMTC/VoIP und EURESCOM[90] geschaffen werden.

84 Dual Tone Multi-Frequency
85 http://webapp.etsi.org/tbhomepage/
86 Public Switched Network
87 Global System for Mobile communication
88 International Telecommunication Union
89 Internet Engineering Task Force
90 European Institute for Research and Strategic Studies in Communications

Tab. 5.7 Arbeitsgruppen der IETF bzgl. VoIP

Arbeitsgruppe	Schwerpunkte
Audio/Video Transport (avt)	Spezifikation von Protokollen (z.B. RTP) zur Echtzeit-Übertragung von Audio und Video über UDP und IP Multicast
Integrated Services (intserv)	Entwicklung eines erweiterten Service-Modells, das Komponenten und Schnittstellen zur Nutzung multimedialer Dienste bereitstellt
IP-Telephony (iptel)	Verhalten von Servern und Terminals bei ein- und ausgehenden Anrufen; Entwicklung eines Gateway Location Protocol
Multicast Address Allocation (malloc)	Spezifikation von Protokollen für einen globalen Verteilungsmechanismus von Multicast-Adressen
Multiparty Multimedia Session Control (mmusic)	Kontrolle von kleinen und großen Telekonferenz-Sitzungen; Gesprächssignalisierung (SIP)
PSTN and Internetworking (pint)	Entwicklung von Schnittstellen zwischen Internet und leitungsvermittelten Netzen
Resource Reservation Setup Protocol (rsvp)	Reservierung von Bandbreite in paketvermittelten Netzen; besonderer Bedarf bei Datenübertragung in Echtzeit

Ausgehend von den Ergebnissen der H.323-Empfehlung sind die folgende Arbeitsgruppen mit unterschiedlichen Themenschwerpunkten aktiv:

- **TIPHON 01:** Anforderungen an die Interoperabilität von Diensten, Verrechnungs- und Sicherheitsvorgänge
- **THIPHON 02:** Architekturmodell und Schnittstellen
- **THIPHON 03:** Rufkontrollvorgänge
- **THIPHON 04:** Bennennungs- und Adressübersetzungsfragen
- **THIPHON 05:** Aspekte für Quality-of-Service
- **THIPHON 06:** Verifizierungs-, Demonstrations- und Implementierungsvorgänge
- **THIPHON 07:** Drahtlose Kommunikation und Mobilitätsvorgänge
- **THIPHON 08:** Security

Die IETF besitzt ebenfalls eine Arbeitsgruppe IPTEL[91], die sich mit der Entwicklung von VoIP-Protokollen, Interoperabilität, Signalisierung, Security und Dienstübergaben auseinandersetzt. Das Verhalten von Servern und Terminals bei ein- und ausgehenden Anrufen [SCRO00] sowie die Entwicklung eines Gateway Location Protocol [LESC00] bilden die Schwerpunkte. Weitere wichtige Arbeitsgruppen der IETF sind in Tab. 5.7 zusammengefasst.

91 http://www.ietf.org/html.charters/iptel-charter.html

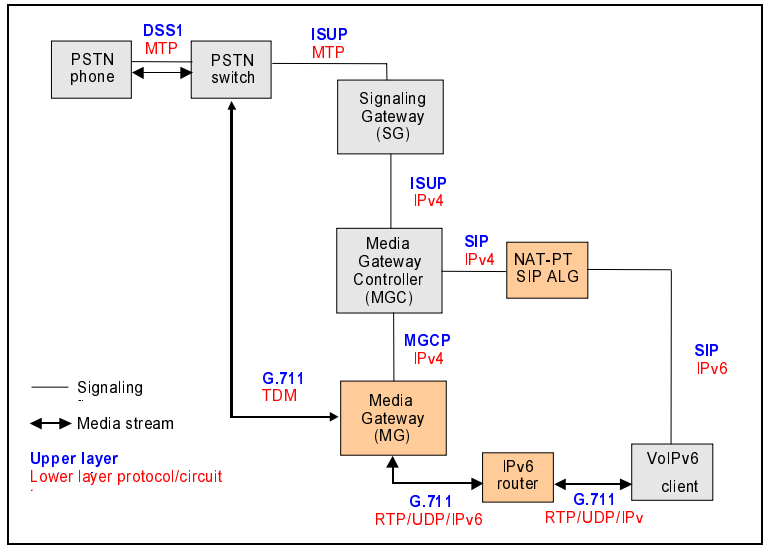


Abb. 5.24 INIT-Implementierung für VoIP

Weitere Vorschläge wurden bereits eingereicht:

- Call Processing Language (CPL): A Language for User Control of Internet Telephony Services
- Telephony Routing over IP (TRIP)
- Authentication Attribute for TRIP

Mit der Verabschiedung wichtiger Standards wie H.225.0, H.245, Q.931 und Q.932 sowie der Serie H.3xx hat die ITU-T eine führende Rolle im Bereich der Übertragung von audiovisuellen Daten eingenommen. Der für die IP-Telefonie bedeutsame Standard H.323 steht heute im Mittelpunkt von VoIP-Produkten, kann aber durch SIP Konkurrenz bekommen. Forschungsprojekte wie das 6init-Projekt[92] setzen beide Spezifikationen in ihren Netzen ein, sodass man noch nicht von einer abschließenden Standardisierung sprechen kann. Im Gegenteil, hier werden neue Ansätze gesucht, um die bestehenden Signalisierungsmöglichkeiten zu vereinfachen und die Interoperabilität heraufzusetzen. Weitere Schwerpunkte des Projekts sind:

- **Voice-over-IPv6:** Interworking zwischen PSTN, IPv4 und IPv6
- **Gateways:** Signaling Gateway, Media Gateway Controller / Media Gateway
- **NAT-PT:** Interworking zwischen IPv4 und IPv6

Alle Standardisierungsgremien sind bestrebt, VoIP zu etablieren und interoperabel zu gestalten. Dabei sind noch nicht alle Spezifikationen und Drafts in gemeinsame Standards überführt worden und bedürfen noch der Weiterentwicklung.

92 http://www.6init.com

6

Echtzeitplattform

In den letzten Kapiteln bin ich auf die Grundlagen und Möglichkeiten im Bereich Security, Quality-of-Service (QoS), Traffic Engineering (TE) und Voice-over-IP (VoIP) eingegangen. In diesem Kapitel soll nun der Aufbau einer Echtzeitplattform anhand dieser Rahmenbedingungen vorgenommen werden. Zuerst wird eine Sicherheitsplattform geschaffen, die anschließend um eine garantierte Dienstgüte erweitert wird. Abschließend werden als Echtzeitanwendung VoIP-Applikationen eingeführt.

6.1 Sicherheitsinfrastruktur

Nachdem ausführlich auf Kryptographie, Key Management und den sicheren Übertragungskanal eingegangen wurde, geht es hier ausführlich um die notwendigen Protokolle, die für die Datenintegrität und Vertraulichkeit von Nutzinformationen stehen und für den Aufbau einer sicheren Infrastruktur notwendig sind. Abb. 6.1 zeigt mögliche Sicherheitsprotokolle in Bezug auf die IP-Schichten, um die Anordnung der Protokolle auf die Schichten zu verdeutlichen.

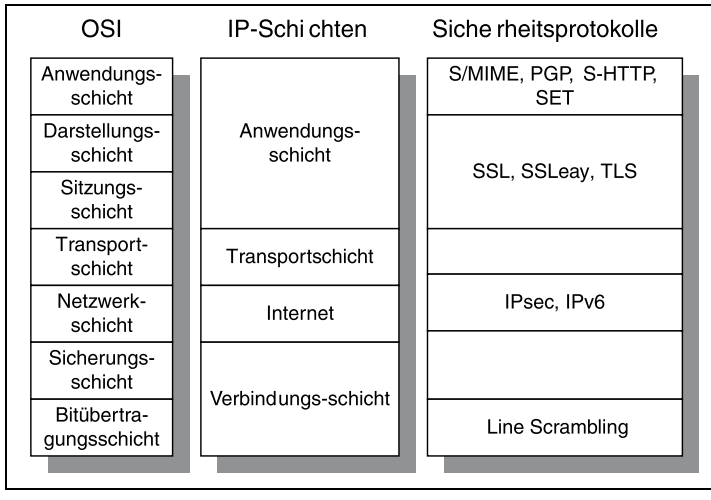


Abb. 6.1
IP-Sicherheitsschichten

Man unterscheidet zwischen der theoretischen und der praktischen Sicherheit eines Systems oder eines Algorithmus. Als theoretisch sicher gilt ein System, wenn es einem Angreifer, dem unbegrenzt viel Zeit und Hilfsmittel zur Verfügung stehen, nicht möglich ist, das System zu brechen. Als praktisch sicher gilt ein System, wenn es dem Angreifer innerhalb einer begrenzten Zeit mit seinen Hilfsmitteln nicht gelingt, das System zu brechen. Dabei kann Sicherheit auf verschiedenen Schichten des OSI-Referenzmodells angesiedelt werden, wie Abb. 6.1 verdeutlicht. Das Line Scrambling setzt ganz unten auf der Bitübertragungsschicht an. Hier findet eine Link-to-Link-Verschlüsselung statt. Alle durchlaufenden Daten werden dabei unabhängig von den verwendeten Applikationen verschlüsselt. Die Geschwindigkeit ist sehr hoch, da keinerlei Informationen über die verschlüsselten Daten enthalten sind und die Implementierungen in der Hardware erfolgen. Allerdings müssen alle intelligenten Switching- oder Speicherknoten auf dem Übertragungsweg den Datenstrom dechiffrieren, bevor sie ihn weiterleiten können. Die Daten müssen also in den dazwischen liegenden Knoten aufgedeckt werden.

Das IPsec-Protokoll und die Spezifikationen von IPv6 sind auf der Schicht 3 angeordnet. Sie ermöglichen es ebenfalls, eine anwendungsunabhängige Verschlüsselung durchzuführen. Dabei wird allerdings noch die Authentifizierung hinzugenommen. Ebenfalls ist es möglich, Komponenten zu tunneln, die nicht die genannten Protokolle unterstützen. Das Secure Socket Layer (SSL) und Secure Shell Protocol (SSH) kann man als Sicherheitsprotokolle auf Transportebene bezeichnen, die den sicheren Datenaustausch von Client zu Server ermöglichen. Beide Protokolle sind in der Arbeitsgruppe Transport Layer Security (TLS) der IETF übernommen worden. Im Gegensatz dazu ist das Secure Hypertext Transfer Protocol (S-HTTP) ein Übertragungsmechanismus, der speziell für WWW-Transaktionen einen sicheren Datenaustausch gewährleistet. Die Sicherheit des Sockets soll Socket Security (SOCKS) hingegen garantieren, welches ein Rahmenwerk für Client-/Server-Anwendungen für UDP und TCP definiert, sodass alle Dienste einer Firewall schnell und sicher genutzt werden können. Hinzu kommen weitere Protokolle wie S/MIME, PGP, SET, SSLeay und PCT, die direkt auf die Anwendungen Einfluss nehmen und deshalb auf der Applikationsebene angesiedelt sind.

Hier werden die wichtigsten Protokolle zum Aufbau einer Sicherheitsinfrastruktur erläutert. Der Schwerpunkt liegt dabei auf den Protokollen SSL für die sichere Kommunikation beliebiger Nutzer, S-HTTP als Anwendungsprotokoll für SSL, SSH zum sicheren Einloggen und Konfigurieren und IPsec zum Aufbau von Virtual Private Networks (VPN).

6.1.1 Secure Socket Layer (SSL)

Das Protokoll SSL wurde von Netscape entwickelt und ist als Draft-Spezifikation bei der IETF eingereicht worden. Bei der IETF wird SSL in der gleichnamigen Arbeitsgruppe als Transport Layer Security (TLS) bezeichnet, da es auf der Version 3 von SSL aufbaut. Eine Abwandlung von SSL ist SSLeay. Dabei handelt es sich um eine SSL-Implementierung des Australiers Eric Young. Aufgrund der amerikanischen Exportbeschränkungen, die nur die Ausfuhr amerikanischer Software mit einer Schlüssellänge von 40 Bit gestattete, wurde diese Version geschaffen, der ansonsten keinerlei Bedeutung beigemessen wird.

Funktionsweise

SSL bzw. TLS ist ein Protokoll, das zur sicheren Ende-zu-Ende-Kommunikation entwickelt wurde. Es ist für den Einsatz zwischen der Anwendungsschicht (HTTP, NNTP, FTP etc.) und der Internetschicht (TCP/IP) einsetzbar. Für die Applikation verhält sich SSL wie TCP. Allerdings sind zusätzliche Steuerungsmöglichkeiten für kryptographische Sicherung des TCP-Datenstroms vorhanden, wobei eine vertrauliche und integritätsgeschützte Kommunikation ermöglicht wird. Ebenfalls kann ein Informationsserver authentifiziert werden, während optional eine Authentifizierung des Client möglich ist. Bezogen auf die ITU-Standards H.323 und H.235 wird SSL/TLS zur Sicherung der H.245- und H.225.0-Kontrollkanäle verwendet. TLS setzt dabei eine zuverlässige Transportschicht voraus und kann aus diesem Grund nicht zur Sicherung eines echtzeitfähigen Protokolls wie UDP verwendet werden.

Weiterhin bietet SSL Datenverschlüsselung, Server-Authentifizierung und Nachrichtenintegrität. Optional ist ebenfalls eine Authentifizierung des Clients für TCP/IP vorgesehen. Das W3C (WWW Consortium) hat SSL als Quasistandard für Webbrowser und Webserver vorgesehen und favorisiert diesen Ansatz. Hauptanforderung an SSL ist es, die Vertraulichkeit und Zuverlässigkeit zwischen zwei kommunizierenden Anwendern sicherstellen zu können. [KAEO98]

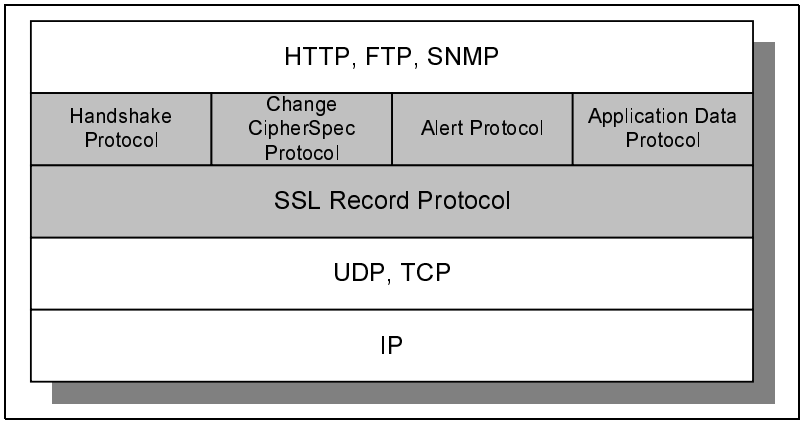


Abb. 6.2 SSL-Protokollschichten

SSL besteht aus verschiedenen Schichtprotokollen, die in Abb. 6.2 dargestellt sind. Sie besitzen folgende Aufgaben:

- **Handshake Protocol:** Durch dieses Protokoll werden kryptographische Parameter für den Sitzungsstatus erzeugt.
- **Change CipherSpec Protocol:** Dieses Protokoll informiert den Record Layer über die verwendeten kryptographischen Algorithmen.
- **Alert Protocol:** Probleme während der Datenübertragung werden durch Fehler- und Warnmitteilungen angezeigt.
- **Application Data Protocol:** Dieses Protokoll transportiert die Daten von der Anwendungsschicht zum SSL Record Protocol.
- **SSL Record Protocol:** Die gesamte gesicherte Kommunikation wird über dieses Protokoll eingekapselt, inklusive der Informationen-Fragmente, übertragenen Daten, Komprimierung und Datenverschlüsselung, die von einer aktiven Sitzung festgelegt wurden.

Wenn SSL-Client und -Server zum ersten Mal die Kommunikation starten, stimmen sie im ersten Schritt die folgenden Parameter miteinander ab:

- Protokollversion
- Kryptographische Algorithmen (z.B. DES, RC4, Triple-DES)
- Optional: gegenseitige Authentifizierung (z.B. RSA)
- Verwendung eines Public Key zur Erzeugung eines Sitzungsschlüssels

Diese Parameter werden durch das Handshake Protocol ausgehandelt. Nach der Verhandlung der Sicherheitsparameter einer Sitzung informiert das Change CipherSpec Protocol die Record Layer über die verwendeten Verschlüsselungsparameter. Anschließend wird die gesamte Kommunikation über das SSL Record Protocol mit den jeweiligen Parametern sicher eingekapselt und verschlüsselt, wie Abb. 6.3 zeigt. Das Application Data Protocol überträgt die Daten von der Applikationsschicht zum SSL Record Protocol. Die Datenübertragung umfasst dabei die Überprüfung der Nachrichtenintegrität mit Hilfe eines mit Kennwort versehenen Authentifizierungscodes. Für die Berechnung werden sichere Hash-Funktionen wie Secure Hash Algorithm (SHA), MD4[1] und MD5[2] eingesetzt. Wenn dabei Probleme auftreten, werden Fehlermeldungen und Warnungen generiert, die durch das Alert Protocol erzeugt werden. Beim Empfänger wird der umgekehrte Weg eingeschlagen, um die verschlüsselten und komprimierten Daten wieder lesbar zu machen.

Durch die Einkapselung der Daten werden andere Ports verwendet, als sie die verwendeten Dienste normalerweise besitzen. Es sind dabei folgende Portnummern von der Internet Assigned Numbers Authority (IANA) für die Verwendung mit SSL reserviert worden:

1 Message Digest, Version 4
2 Message Digest, Version 5

- **Port 443:** Hypertext Transfer Protocol mit SSL (https)
- **Port 465:** Simple Mail Transfer Protocol mit SSL (ssmtp)
- **Port 563:** Network News Transfer Protocol mit SSL (snntp)
- **Port 636:** Light Directory Access Protocol mit SSL (ssl-ldap)
- **Port 990:** File Transfer Protocol mit SSL (ftps)
- **Port 995:** Post Office Protocol mit SSL (spop3)

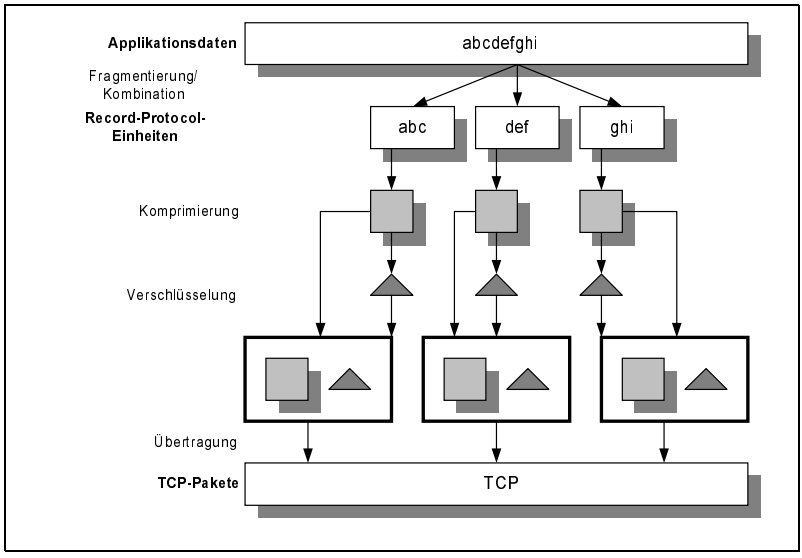


Abb. 6.3
Funktionalität des SSL Record Protocol

SSL unterstützt eine große Zahl von kryptographischen Algorithmen. Während des Handshake-Prozesses wird entweder RSA zum Schlüsselaustausch mit Zertifikat genutzt oder Diffie-Hellman ohne Zertifikat. Nach dem Schlüsselaustausch werden symmetrische und asymmetrische Verschlüsselungen angewendet, die folgende Algorithmen beinhalten können:

- RC2 / RC4 (40 Bit)
- IDEA[3] (128 Bit)
- DES[4] / Triple-DES (56 / 168 Bit)
- MD5 (128 Bit)
- SHA-1 (160 Bit)
- X.509

Der wesentliche Vorteil von SSL ist die Unabhängigkeit von den Anwendungen. Protokolle höherer Schichten können dadurch transparent auf SSL aufsetzen. Die Verbindungssicherheit von SSL fasst somit folgende Eigenschaften zusammen:

3 International Data Encryption Algorithm
4 Data Encryption Standard

- Vertraulichkeit der Verbindung durch Verschlüsselung
- Authentifizierung der jeweiligen Kommunikationspartner
- Integritätsprüfung der versendeten Nachricht mittels Message Authentication Code (MAC)
- Zuverlässigkeit der Verbindung

Wie dargestellt ist SSL ein geschichtetes Protokoll. Jede Schicht kann Nachrichten enthalten wie Länge, Beschreibung und Inhalte. Angewendet wurde SSL häufig mit HTTP. Dabei wurde nur ein Teil der möglichen Funktionalität ein- und umgesetzt. Zukünftig muss SSL deshalb auch seine Gesamtfunktionalität unter Beweis stellen.

SSL Handshake Protocol

In Abb. 6.4 ist ein SSL-Verbindungsaufbau dargestellt, der mit der Hello-Phase startet, in der der Client eine so genannte Client_Hello-Nachricht sendet, die folgende Einträge enthält:

- Protokollversionsnummer
- Struktur aus einer 28 Byte langen Zufallszahl und einem Zeitstempel (Client_Hello_Random)
- Session-ID, wenn der Client eine alte Verbindung mit dieser Session-ID wiederaufnehmen will.
- Auswahl von kryptographischen Verfahren und Hash-Funktionen (Auswahl so genannter Cipher Suites), die der Client unterstützt.
- Auswahl von Kompressionsverfahren, die der Client unterstützt.

Nach dem Senden der Client_Hello-Nachricht wartet der Client auf die Antwort des Servers. Der Server beantwortet eine Client_Hello-Nachricht entweder mit der Server_Hello-Nachricht oder mit der Fehlermeldung Handshake Failure und einem anschließenden Verbindungsabbruch. Wenn keine Session-ID in der Client_Hello-Nachricht enthalten ist, bestimmt der Server eine neue ID, die die neue Verbindung eindeutig kennzeichnen soll. Anschließend wählt der Server eine der angebotenen Cipher Suites sowie ein Kompressionsverfahren aus. Er bestimmt auch eine 28 Byte lange Zufallszahl und einen Zeitstempel (Server_Hello_Random). Die Zufallszahl muss sich von der des Client unterscheiden. Diese Informationen sind Bestandteil der Server_Hello-Nachricht.

Im Anschluss an die Server_Hello-Nachricht sendet der Server entweder sein Zertifikat oder, wenn er kein entsprechendes besitzt, eine Server_Key_Exchange-Nachricht, die für den Schlüsselaustausch gedacht ist. Im Folgenden soll davon ausgegangen werden, dass der Server ein gültiges Zertifikat besitzt. Dabei handelt es sich meist um X.509. Wenn sich auch der Client gegenüber dem Server authentifizieren soll, sendet der Server eine Certificate_Request-Nachricht. Abschließend folgt eine Server_Hello_Done-Nachricht, die das Ende der Server_Hello-Nachricht und anschließender Nachrichten anzeigt.

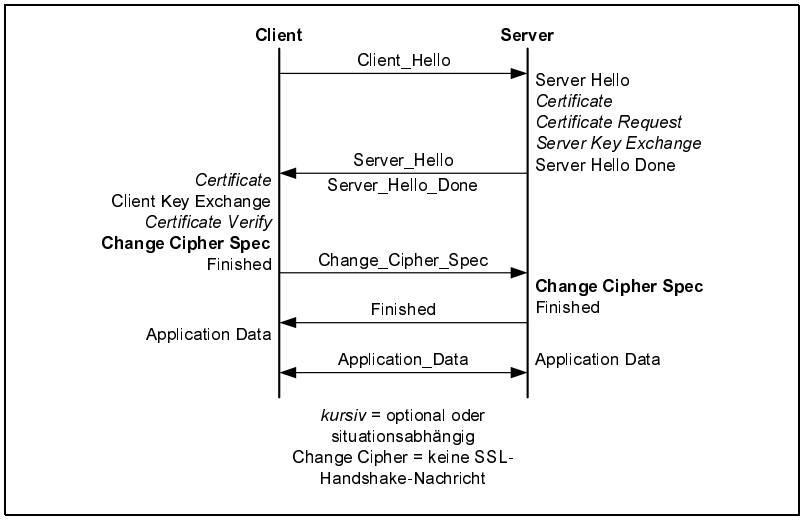


Abb. 6.4 SSL-Verbindungsaufbau

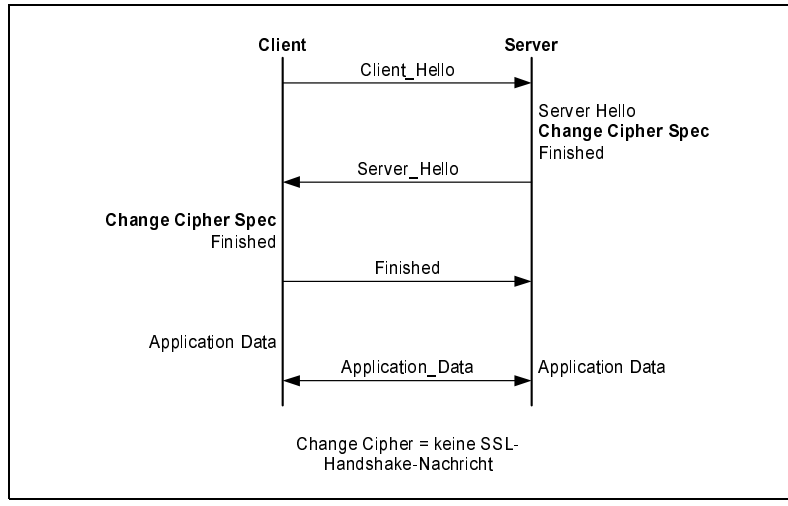
Nach Empfang der Server_Hello_Done-Nachricht sendet der Client, wenn vom Server gefordert, sein Zertifikat. Dieser Fall soll zunächst nicht behandelt werden. In Abhängigkeit von dem vom Server ausgewählten Public-Key-Verfahren sendet der Client eine Client_Key_Exchange-Nachricht. Der weitere Verlauf soll unter der Vorgabe beschrieben werden, dass der Server das RSA-Verfahren ausgewählt hat. Für diesen Fall generiert der Client eine 46 Byte lange Zufallszahl, die zusammen mit der Versionsnummer des SSL-Protokolls ein so genanntes Premaster Secret bildet. Dieses wird mit dem öffentlichen Schlüssel des Servers verschlüsselt und als Client_Key_Exchange-Nachricht versendet.

Das Premaster Secret dient dazu, dass eine für Client und Server gemeinsame geheime Information, das so genannte Master Secret, generiert wird, aus dem sich dann unter anderem Schlüssel für die Verschlüsselungsalgorithmen und MAC-Berechnungen erstellen lassen.

Nach der Change_Key_Exchange-Nachricht erfolgt die Certificate_Verify-Nachricht, wenn die Authentifizierung des Client vom Server gefordert wurde. Dann sendet der Client eine Change_Cipher_Spec-Nachricht, die mit den vereinbarten Verfahren verschlüsselt und komprimiert wird. Sie ist dabei nicht eigentlicher Bestandteil des SSL Handshake Protocol. Die Change_Cipher_Spec-Nachricht soll anzeigen, dass die folgenden Nachrichtenblöcke mit den vereinbarten Verfahren bearbeitet werden. Der Server sendet seine Change_Cipher_Spec-Nachricht, nachdem er die Client_Key_Exchange-Nachricht erfolgreich verarbeitet und die nötigen Schlüssel generiert hat. Eine unerwartete Change_Cipher_Spec-Nachricht führt immer zu einer Fehlermeldung und zum Abbruch der Verbindung.

Den Abschluss des Handshake bilden die Finished-Nachrichten. Sie werden sofort nach den Change_Cipher_Spec-Nachrichten gesendet und sollen den erfolgreichen Schlüsselaustausch und die erfolgreiche Authentifizierung verifizieren. Die Finished-Nachrichten werden durch die vereinbarten Verfahren, Schlüssel und Geheimnisse gesichert. Bei Empfang einer solchen Nachricht muss überprüft werden, ob der Inhalt korrekt ist. Nach der Finished-Nachricht können bereits vertrauliche Daten versendet werden, ohne dass auf eine Bestätigung gewartet werden muss.

Abb. 6.5 Wiederherstellung einer alten Verbindung



Die Finished-Nachricht besteht aus folgenden Einträgen:

- **MD5** (Master Secret + PAD2 + MD5 (Handshake Messages + Sender + Master Secret + PAD1))
- **SHA** (Master Secret + PAD2 + SHA (Handshake Messages + Sender + Master Secret + PAD1))

Das Plus-Zeichen steht für Verknüpfungen. Die Handshake Message enthält hingegen alle Nachrichten, die während des Handshake gesendet wurden. Der Wert Sender ist für Server (0x53525652) und Client (0x434C4E54) unterschiedlich. Die Paddings PAD1 und PAD2 werden auf folgende Weise gebildet:

- **PAD1:** 0x36 – 48 mal für MD5 oder 40 mal für SHA
- **PAD2:** 0x5C – 48 mal für MD5 oder 40 mal für SHA

In Abb. 6.5 ist der Mechanismus gezeigt, der bei Wiederaufnahme einer alten SSL-Verbindung durch den Client abläuft. Dabei verwendet der Client bei der Client_Hello-Nachricht die Session ID von der Verbindung, die er wieder aufnehmen möchte. Wenn der Server diese spezielle Session ID in seinem Cache findet und die Verbindung wieder unter den alten Vereinbarungen aufsetzen

möchte, sendet er seine Server_Hello-Nachricht mit der gleichen Session ID an den Client zurück. Es müssen nur noch die Change_Cipher_Spec-Nachrichten, gefolgt von den Finished-Nachrichten gesendet werden. Danach kann der vertrauliche Datenaustausch erfolgen. Wenn die Session ID nicht vom Server gefunden wurde, sendet er die Server_Hello-Nachricht mit einer neuen Session ID zum Client zurück und der gesamte Handshake muss durchgeführt werden. Die Möglichkeit der Wiederaufnahme alter Verbindungen soll die Zeit für den Handshake verkürzen. Die Session-IDs werden jedoch nur für eine bestimmte Zeit zwischengespeichert.

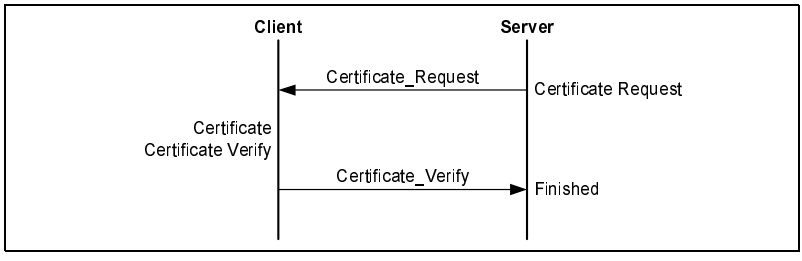


Abb. 6.6 Authentifizierung des Client

Die Authentifizierung des Client erfolgt, wie in Abb. 6.6 dargestellt. Mit der Certificate_Request-Nachricht fordert der Server die Authentifizierung des Client. Der Client kommt dieser Forderung nach, indem er sein Zertifikat sendet. Doch erst mit der Client_Verify-Nachricht erfolgt die tatsächliche Authentifizierung des Client. Denn anhand dieser Nachricht lässt sich vom Server überprüfen, ob der Client auch in Besitz des zum Zertifikat gehörenden privaten Schlüssels ist. Die Client_Verify-Nachricht ist ähnlich wie die Finished-Nachricht aufgebaut und enthält vom Client signierte Hash-Werte.

Die Hash-Werte ergeben sich folgendermaßen:

- **MD5** (Master Secret + PAD2 + MD5 (Handshake Messages + Master Secret + PAD1))
- **SHA** (Master Secret + PAD2 + SHA (Handshake Messages + Master Secret + PAD1))

Schlüsselgenerierung

An dieser Stelle wird noch kurz auf die Schlüsselgenerierung eingegangen. Der Schlüssel (Master_Secret) wird sowohl durch den Client als auch durch den Server aus dem vom Client erzeugten Premaster_Secret gebildet. Aus diesem Master_Secret werden die Schlüssel für die MAC-Berechnungen, die Schlüssel für die Verschlüsselungsverfahren und die Initialisierungsvektoren (IV) für die Verschlüsselungsalgorithmen generiert.

Das Master_Secret wird mit den Hash-Funktionen MD5 und SHA berechnet. Aus diesem 384 Bit langen Master_Secret wird dann ein so genannter Key_Block erzeugt, aus dem sich die einzelnen Schlüssel und Informationen für

die Verschlüsselung gewinnen lassen. Das Verfahren wird so lange durchgeführt, bis der Key_Block eine ausreichende Länge hat. Der Key_Block wird nach den ausgehandelten Vereinbarungen folgendermaßen aufgeteilt:

- **Client_Write_MAC_Secret:** Geheimnis für die MAC-Berechnungen aus Daten des Server
- **Server_Write_MAC_Secret:** Geheimnis für die MAC-Berechnungen aus Daten des Client
- **Client_Write_Key:** Geheimer Schlüssel für das Chiffrieren beim Client und Dechiffrieren beim Server
- **Server_Write_Key:** Geheimer Schlüssel für das Chiffrieren beim Server und Dechiffrieren beim Client
- **Client_Write_IV:** Initialisierungsvektor für Blockchiffren im CBC[5]-Betriebsmodus, zugehörig zum Client_Write_Key
- **Server_Write_IV:** Initialisierungsvektor für Blockchiffren im CBC-Betriebsmodus, zugehörig zum Server_Write_Key

Die Länge der einzelnen Schlüssel bzw. IV ergibt sich aus der ausgewählten Cipher_Suite. Sollten Bits vom Key_Block übrig bleiben, werden diese verworfen. Eine genauere Definition der Verschlüsselungsalgorithmen IDEA[6] und RC2 findet sich in Tab. 6.1.

Für die Hash-Funktionen MD5 und SHA gilt, dass die Hash_Size bei MD5 16 Byte beträgt, während die Padding_Size 48 Byte beinhaltet. SHA besitzt hingegen eine Hash_Size von 20 Byte und eine Padding_Size von 40 Byte.

Tab. 6.1 Verschlüsselungsalgorithmen IDEA und RC2

Cipher	Cipher Type	Export-tabelle	Schlüssel-material [Byte]	Erweiter-tes Schlüs-selmate-rial [Byte]	Effektive Schlüs-selbits [Bit]	IV-Größe [Byte]	Block größe [Byte]
IDEA_CBC	Block	Nein	16	16	128	8	8
RC2_ CBC_40	Block	Ja	5	16	40	8	8

SSL Record Layer Die Aufgaben des SSL Record Layer wird in Abb. 6.7 angedeutet. Die einzelnen Verarbeitungsstufen eines Datenpakets im SSL Record Layer beinhalten konkret die Klartextstruktur, komprimierte Struktur und den Chiffretext.

5 Ciper Block Chaining
6 International Data Encryption Algorithm

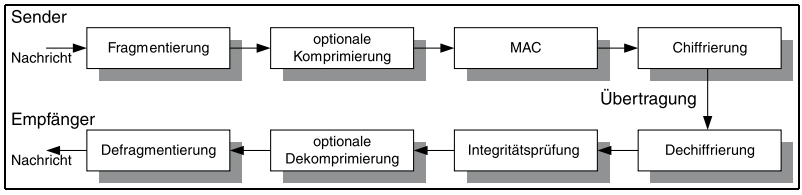


Abb. 6.7 SSL-Verarbeitungsschema

Der SSL Record Layer fragmentiert die Datenpakete, die er von den höheren Schichten empfängt, in Klartextstrukturen, so genannte SSL_Plaintext_Records, die nicht größer als 214 Byte (16 Kbyte) sind. Eine solche Klartextstruktur umfasst die folgenden Punkte:

- Verweis auf höher gelegene Protokolle, die die Daten verarbeiten
- Protokollversionsnummer
- Länge des Datenfragments
- Datenfragment

Diese Klartextsstrukturen werden dann mit dem im Handshake ausgehandelten Kompressionsverfahren komprimiert. Dabei werden die Klartextstrukturen in komprimierte Strukturen überführt. Diese Struktur enthält neben dem Verweis auf das höhere Protokoll und der Protokollversionsnummer noch die Länge des komprimierten Datenfragments und das komprimierte Datenfragment. Danach folgt die MAC-Berechnung und die Verschlüsselung für jede komprimierte Struktur, wobei die MAC-Berechnung vor der Verschlüsselung erfolgt. In diesem Schritt wird aus der komprimierten Struktur der SSL-Chiffretext gebildet.

Die MAC-Berechnung erfolgt durch Hash (MAC Write Secret + PAD2 + Hash (MAC Write Secret + PAD1 + Sequenznummer + Länge + Inhalt)). Mit Hash ist hier die vereinbarte Hash-Funktion gemeint. Die Länge steht für die Länge des komprimierten Datenfragments, das in Inhalt enthalten ist. Danach erfolgt die Verschlüsselung mit dem vereinbarten Verschlüsselungsverfahren und dem dafür berechneten Schlüssel. Der Message Authentication Code wird ebenfalls verschlüsselt. So bildet sich der SSL-Chiffretext aus:

- Verweis auf das höhere Protokoll
- Protokollversionsnummer
- Länge des Chiffretexts
- Chiffretext bestehend aus dem komprimierten Datenfragment und MAC

Damit ist die Arbeit des SSL Record Layer für die Sendeseite abgeschlossen. Auf der Empfangsseite werden die Arbeitsschritte in umgekehrter Reihenfolge durchlaufen. [DEER01]

6.1.2 Secure Hypertext Transfer Protocol (S-HTTP)

S-HTTP, welches in der Spezifikation RFC-2660 beschrieben ist, ist ein Kommunikationsprotokoll, das für den sicheren Einsatz in Verbindung mit HTTP entwickelt wurde. S-HTTP verwendet SSL für die Kommunikation zwischen dem Browser und dem Webserver. Es wird im Browser durch https und einem anderen Server-Port 443 gegenüber HTTP mit Port 80 kenntlich gemacht. Die Entwicklung nimmt daher sehr stark Bezug auf das Messaging-Modell von HTTP, wodurch S-HTTP sehr einfach in bestehende HTTP-Umgebung integriert werden kann. Anforderungen und Antworten werden sofort behandelt, weshalb das Protokoll symmetrische Funktionalität für Client und Server bereitstellt. Dabei bleiben allerdings das Transaktionsmodell und die Implementierungsmerkmale von HTTP erhalten.

S-HTTP Client und Server können mehrere Standardformate für verschlüsselte Nachrichten integrieren. Ebenfalls ist es möglich, dass sich Clients mit S-HTTP mit Servern ohne S-HTTP verstehen können. Allerdings fallen dann natürlich die Sicherheitsmechanismen weg. S-HTTP verwendet keine Zertifikate für öffentliche Schlüssel bei den Clients, da es symmetrische Betriebsmodi verwendet. Trotzdem kann auch S-HTTP für Zertifizierungsinfrastrukturen genutzt werden.

S-HTTP unterstützt somit sichere Ende-zu-Ende-Transaktionen, anders als Authentifizierungsmechanismen, bei denen der Client den Zugriff versucht, bevor Sicherheitsmechanismen verwendet werden. Clients können angewiesen werden, so genannte sichere Transaktionen einzuleiten. Dies wird meistens durch die Verwendung von Informationen aus dem Nachrichtenkopf vorgenommen. Das bedeutet, dass bei S-HTTP keine sensitiven Daten über das Netzwerk gesendet werden. Ausnutzen kann man dieses Merkmal, indem beispielsweise Webformulare verschlüsselt werden.

Ebenfalls bietet S-HTTP eine hohe Flexibilität gegenüber kryptographischen Algorithmen, Modi und Parametern. Die Optionen werden dabei ausgehandelt zwischen Client und Server und beinhalten die Transaktionsmodi (soll signiert oder verschlüsselt werden oder beides), kryptographische Algorithmen (u.a. RSA oder DSA für die Signatur, Triple-DES oder RC4 für die Verschlüsselung) und Zertifikatwahl. Da S-HTTP ebenfalls die Verschlüsselung mit öffentlichen Schlüsseln unterstützt, kann man auch digitale Signaturen einsetzen und Vertraulichkeit gewährleisten. Das einzige Manko von S-HTTP bleibt letztendlich der geringe Verbreitungsgrad.

Ablauf einer HTTP-Verbindung

HTTP ist ein Request-/Response-Protokoll. Es bildet die Basis für das World Wide Web (WWW) als textorientiertes Protokoll zwischen WWW-Client und WWW-Server. Im Allgemeinen setzt eine HTTP-Kommunikation auf einer TCP/IP-Verbindung auf. Dabei ist HTTP jedoch nicht auf TCP/IP beschränkt,

sondern setzt lediglich ein zuverlässiges Transportprotokoll voraus. HTTP bietet abgesehen von zwei Verfahren zur Authentifizierung keine weiteren Sicherheitsmechanismen. Da es aber als Grundlage für Secure HTTP wichtig ist, soll es an dieser Stelle kurz behandelt werden. Der Ablauf einer Kommunikation über HTTP besteht aus folgenden Schritten:

- Aufbau einer TCP/IP-Verbindung
- Request: HTTP-Nachricht vom Client zum Server
- Response: HTTP-Nachricht vom Server zum Client
- Abbau der TCP/IP-Verbindung

Der Client baut eine TCP/IP-Verbindung zum Server auf. Dabei ist Port 80 der Standard-TCP-Port. Nach erfolgreichem Verbindungsaufbau sendet der Client seine Antwort (Request). Diese besteht oft aus der Anforderung eines bestimmten Dokuments. Der Server beantwortet die Anfrage (Request) mit einer Antwort (Response). Die Antwort enthält z.B. das angeforderte Dokument oder eine entsprechende Fehlermeldung. Der Verbindungsabbau kann sowohl vom Server als auch vom Client erfolgen. Bei HTTP Version 1.0 wird in der Regel für jedes neue Request/Response-Paar eine Verbindung aufgebaut. Bei HTTP Version 1.1 kann eine bestehende Verbindung für die weitere Kommunikation genutzt werden. Diese Verbindung wird Persistent Connection genannt.

Ein Request besteht aus den folgenden Parametern:

- Version des Protokolls
- Methode, die auf das entsprechende Dokument angewendet werden soll (z.B. GET oder PUT).
- Uniform Resource Identifier (URI), der das gewünschte Dokument bestimmt.
- einem Anteil, der ein ähnliches Format wie Multipurpose Internet Mail Extensions (MIME) hat.

Die Response des Server beinhaltet hingegen:

- Version des Protokolls
- Zustandscode, der über Erfolg oder Fehler Auskunft gibt
- Anteil in MIME-ähnlichem Format

Authentifizierung

Der einzige Sicherheitsmechanismus, der von HTTP unterstützt wird, ist die Authentifizierung des Clients. Dafür stehen in HTTP zwei Verfahren zur Verfügung:

- Basic Authentication (BA)[7]
- Digest Access Authentication (DAA)

7 Wird auch als Basic Access Authentication bezeichnet.

Bei der Basic Authentication muss sich der Client gegenüber dem Server über eine User-ID und ein zugehöriges Passwort authentifizieren. Da sowohl Client als auch Server in Besitz des zur User-ID gehörigen Passworts sein müssen, muss im Vorfeld ein gemeinsames Passwort vereinbart werden. Wie die Vereinbarung aussehen soll, ist nicht Bestandteil der Spezifikation. Welche Dokumente bzw. welche Verzeichnisse durch BA geschützt werden sollen, wird vom Administrator des Server durch entsprechende Einträge in den Konfigurationsdateien des Server festgelegt.

Die DAA arbeitet nach dem gleichen Mechanismus wie die Basic Authentication. Der Unterschied besteht jedoch darin, dass bei der DAA das Passwort nicht im Klartext übermittelt wird. Da sich die Konzepte ähnlich sind, soll nur speziell auf die Unterschiede eingegangen werden. Der WWW Authentication Header ist bei DAA um einige Parameter erweitert worden. Neben der Authentifizierungsmethode und dem Authentifizierungsbereich enthält er noch einen Einmalwert (Nonce) und weitere optionale Zusätze, wie z.B. die zu verwendende Hash-Funktion. Ist keine Hash-Funktion explizit angegeben, wird MD5 verwendet.

Sowohl BA als auch DAA sollen vor unberechtigtem Zugriff schützen. Jeder Hacker, der den Übertragungskanal abhört, kann das jeweils übermittelte Dokument mitlesen. Der Vorteil von DAA gegenüber Basic Authentication besteht darin, dass ein Angreifer nicht in Besitz des Passworts gelangen kann. Somit kann er nicht eigenständig zugriffsgeschützte Dokumente abfragen. [DEER01]

Umsetzung von S-HTTP

S-HTTP bietet Sicherheit durch Signieren, Authentifizieren und Verschlüsseln. Die Spezifikation schreibt vor, dass die einzelnen auf ein Dokument angewendeten Sicherheitsmechanismen dem Benutzer angezeigt werden. Die folgenden sicherheitsrelevanten Operationsmodi sind von Bedeutung:

- Signatur
- Schlüsselaustausch und Verschlüsselung
- Integrität der Nachricht und Authentifizierung des Senders

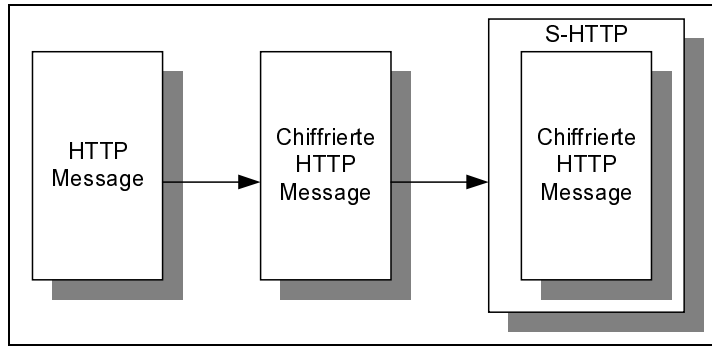


Abb. 6.8 Einkapselung einer HTTP-Nachricht

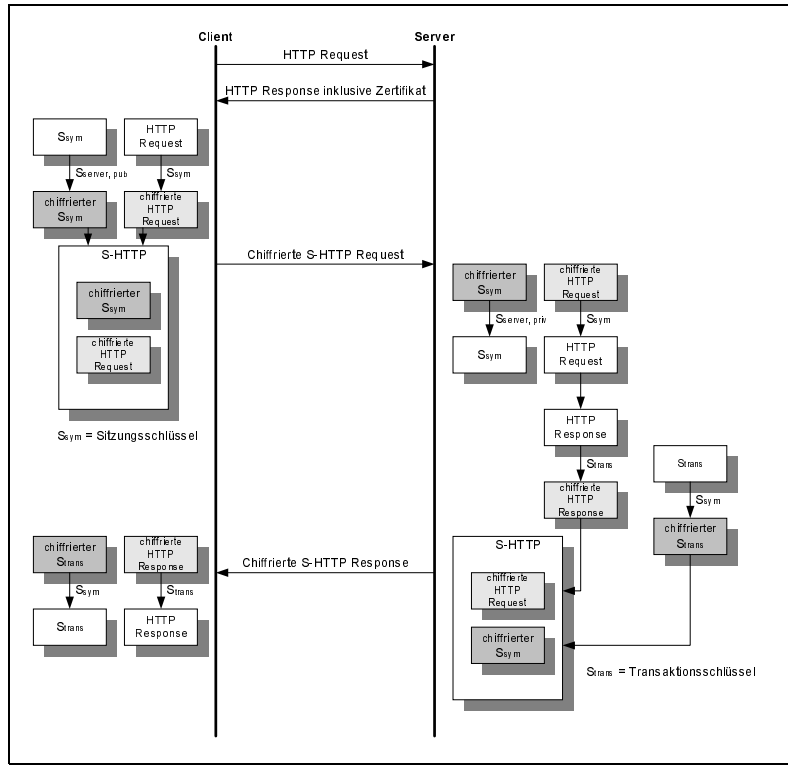
Die Signaturvergabe ist besonders für kommerzielle Anwendungen wie E-Commerce interessant. Der Benutzer hat damit nicht nur die Möglichkeit, signierte Nachrichten bzw. Dokumente über S-HTTP zu versenden, sondern es bietet sich ihm auch die Möglichkeit, signierte Nachrichten zu empfangen und auszuwerten. Ebenfalls kann ein Dokument auf Wunsch des Benutzers signiert werden. Um eine Signatur zu erstellen, benötigt der Sender ein gültiges Zertifikat. Für den Schlüsselaustausch stehen dabei zwei unterschiedliche Transfermechanismen zur Verfügung: der Einsatz eines Public-Key-Verfahrens oder eines extern vereinbarten Schlüssels. Beim Public-Key-Verfahren wird der Sitzungsschlüssel mit einem öffentlichen Schlüssel chiffriert. Der zweite Weg ermöglicht die Verwendung von kryptographischen Verfahren, ohne auf eine Public Key Infrastructure (PKI) zurückgreifen zu müssen. Die Integrität der Nachricht und die Authentizität des Senders lässt sich durch Message Authentication Codes (MAC) überprüfen. Das gemeinsame Geheimnis, das für den MAC benötigt wird, kann dabei manuell ausgetauscht werden.

S-HTTP kapselt die zu schützenden Daten ein, bei denen es sich in der Regel um HTTP-Nachrichten handelt. Es kann sich dabei aber auch um andere Nachrichtentypen oder Daten handeln. Dabei werden vorrangig die kryptographischen Nachrichtenformate PKCS#7 und MIME Object Security Services (MOSS) verwendet. Der S-HTTP-Header enthält Informationen, die für die Auswertung der gekapselten Daten notwendig sind. Sie geben dem Empfänger beispielsweise Auskunft über das verwendete Nachrichtenformat oder über den verwendeten Schlüssel.

Um S-HTTP-Nachrichten zu erstellen, sind ein Klartext, eine HTTP-Nachricht oder andere Daten notwendig. Weiterhin ist eine Liste der vom Empfänger und Sender unterstützten kryptographischen Verfahren und Schlüssel anzufordern. Der Sender muss die vom Empfänger unterstützten Verfahren kennen. Die Informationen erhält er aus einem Zusatz in einem HTTP- oder S-HTTP-Header bzw. aus einem Zusatz im Anchor 1, dem Verweis auf ein neues Dokument in einem Hypertext-Dokument. Das eigentliche Aushandeln der Verfahren und Schlüssel verläuft folgendermaßen: Aus den Vorgaben des Empfängers und des Senders ergibt sich eine Auswahl von kryptographischen Verfahren, die von beiden unterstützt werden. Aus dieser Liste wird dann gegebenenfalls durch Interaktion mit dem Benutzer eine Auswahl getroffen. Die ausgewählten Verfahren werden auf den Klartext angewendet. Das Ergebnis wird als gekapselte Nachricht versendet. Der Empfänger muss dem S-HTTP-Header entnehmen, welche kryptographischen Verfahren und Schlüssel eingesetzt wurden. So lässt sich der Klartext wiederherstellen. Dabei kann der Empfänger auch überprüfen, ob die von ihm gestellten Vorgaben an die kryptographischen Verfahren vom Sender erfüllt wurden.

Abb. 6.9 zeigt den Ablauf einer sicheren Kommunikation über S-HTTP. Dabei wird in diesem Beispiel davon ausgegangen, dass die Informationen über die vom Server unterstützten Verfahren durch eine HTTP-Nachricht übermittelt werden und der Client nicht in Besitz eines gültigen Schlüsselpaars ist. In einem Schritt sendet der Client eine Anfrage (Request) an den Server. Auf die Anfrage des Client sendet der Server seine Antwort, die sein Zertifikat, Vorgaben für die folgende Anfrage des Client und einen neuen Verweis (Link) auf das zu schützende Dokument enthält. Der Client erzeugt einen HTTP-Request, mit dem er das Dokument abrufen möchte. Dann generiert er einen zufälligen Sitzungsschlüssel *Ssym* und chiffriert damit seinen HTTP-Request. Der Sitzungsschlüssel *Ssym* wird nun mit dem öffentlichen Schlüssel des Server *Sserver,pub* chiffriert. Beides wird von einer S-HTTP Nachricht gekapselt und an den Server gesendet.

Abb. 6.9 S-HTTP-Verbindungsablauf



Der Server kann mit seinem privaten Schlüssel *Sserver,priv* den Sitzungsschlüssel *Ssym* dechiffrieren und dann mit *Ssym* den chiffrierten Request entschlüsseln. Auf die Anfrage antwortet der Server mit einer entsprechenden Antwort (Response), die das geforderte Dokument enthält. Der Server erzeugt dabei

einen zufälligen Transaktionsschlüssel *Strans*, mit dem er die Antwort verschlüsselt. Der mit *Ssym* chiffrierte *Strans* wird im Header der S-HTTP-Nachricht, die den Chiffretext kapselt, an den Client gesendet. Im S-HTTP-Header verweist der Server darauf, dass er den vereinbarten Sitzungsschlüssel *Ssym* zum Chiffrieren des Transaktionsschlüssels *Strans* verwendet hat. Der Client muss den chiffrierten Transaktionsschlüssel dechiffrieren und kann damit die gekapselte chiffrierte HTTP-Nachricht entschlüsseln. Jetzt kann das angeforderte Dokument im Klartext gelesen werden.

Dieser Ablauf stellt das Prinzip einer S-HTTP-Verbindung dar. S-HTTP stellt eine sichere Ende-zu-Ende-Verbindung im Gegensatz zu HTTP her, bei dem die Authentifizierungsmechanismen erst einsetzen, wenn der Zugriff durch den Server verweigert wurde. Eine S-HTTP-Nachricht lässt sich in zwei Blöcke aufteilen. Der erste beinhaltet die so genannten S-HTTP-Header und der zweite die gekapselten Daten, die den so genannten Message Body der S-HTTP-Nachricht bilden. In den S-HTTP-Headern werden die folgenden, für den Inhalt der S-HTTP-Nachricht wichtigen Parameter festgelegt:

- Verwendetes Nachrichtenformat (Content-Privacy-Domain)
- Typ der gekapselten Nachricht (Content-Type)
- Verwendete Transferkodierung (Content-Transfer-Encoding)
- Hinweise auf vereinbarte Schlüssel (Prearranged-Key-Info)
- Message Authentication Code (MAC-Info)

Dabei sind nur die ersten beide Punkte für jede S-HTTP-Nachricht vorgeschrieben, die anderen sind optional. In der Spezifikation von S-HTTP sind auch Erweiterungen für HTTP enthalten. So werden durch S-HTTP neue Header für HTTP spezifiziert. Über diese Header lassen sich alle für die sichere Übertragung geforderten Optionen aushandeln. Dies beinhaltet z.B. folgende Punkte:

- Verschlüsselungsalgorithmus
- Algorithmus für MAC
- Sicherheitsmechanismen (Signieren, Authentifizieren und Verschlüsseln)
- Zu verwendenden Schlüssel

Es lassen sich aber auch symmetrische Schlüssel über diese Header austauschen. Das setzt natürlich voraus, dass die entsprechende HTTP-Nachricht chiffriert übermittelt wird. Beim Aushandeln kann eine bestimmte Option bzw. ein bestimmtes Verfahren gefordert (Required), immer abgewiesen (Refused) oder, wenn es vom Verbindungspartner gewünscht wird, akzeptiert (Optional) werden. Dies lässt sich sowohl für Sende- und Empfangsseite durchführen. Damit kann dem Verbindungspartner genau mitgeteilt werden, welche Verfahren und Optionen er verwenden und unterstützen kann bzw. muss. [RESC99]

6.1.3 Secure Shell (SSH)

SSH ist für Remote Access spezifiziert worden, um den sicheren Zugriff über ein unsicheres Netzwerk gewährleisten zu können. SSH bietet damit sicheren Remote Access, sichere Datenübertragung und TCP/IP-/X.11-Weiterleitungen an. Übertragene Daten können automatisch verschlüsselt, authentifiziert und komprimiert werden. SSH kann daher gegen Kryptoanalyse und Protokollangriffe schützen. Dabei besteht SSH aus den folgenden Komponenten:

- Das Transportschichtprotokoll beinhaltet Server-Authentifizierung, Vertraulichkeit und Integrität mit Geheimhaltung in der Vorwärtsrichtung. Optional kann ebenfalls Komprimierung genutzt werden, um auch Verbindungen mit geringer Datenrate zu unterstützen.
- Das Benutzer-Authentifizierungsprotokoll authentifiziert den Client gegenüber dem Server.
- Das Verbindungsprotokoll multiplext den verschlüsselten Tunnel in mehrere logische Kanäle.
- Die Kommunikation von SSH wird immer verschlüsselt, wobei unterschiedliche Verschlüsselungsverfahren wie IEDA, RC4-128, Triple-DES eingesetzt werden. Die Schlüssel werden mit RSA ausgetauscht. Die Daten, die man bei einem Schlüsselaustausch generiert, werden stündlich gelöscht. Hinzu kommt, dass die Schlüssel nirgends gespeichert werden. Jeder Host besitzt seinen eigenen RSA-Schlüssel, der zur Authentifizierung des Host verwendet wird. Die Verschlüsselung ermöglicht einen Schutz gegenüber dem IP-Spoofing[8], während die Authentifizierung gegen DNS[9]- und Routing-Spoofing verwendet wird.
- Auf der anderen Seite besitzt SSH eine mögliche Abhängigkeit der Datenreihenfolge auf der Sitzungsschicht. Das heißt, bei Einsatz von SSH kann ein Angriff auf TCP den Abbruch einer Verbindung zur Folge haben. Anschließend müsste eine neuen Sitzung etabliert werden, obwohl aus der Sicht von TCP keine Probleme bestehen. [KAEO98]
- Der Verbindungsablauf einer SSH-Verbindung, die durch Abb. 6.10 verdeutlicht wird, besitzt die folgende Struktur: Zuerst baut der Client eine Verbindung zum Server über TCP auf. Es folgt ein Austausch der Protokoll- und Programmversion (Version_Identification_String). Falls die Versionen nicht unterstützt werden, kommt es zum Abbruch der Verbindung. Nach der Protokoll-Identifizierungsphase wird auf ein paketbasiertes Binärprotokoll umgeschaltet. Der Server sendet dem Client den öffentlichen Host-Key. Dieser ist ein RSA-Schlüssel, der zur Authentifizierung des Host dient. Ebenfalls sendet der Server dem Client den öffentlichen Server-

8 Fälschen der Absenderadresse in einem IP-Paket
9 Domain Name Service

Key. Hierbei handelt es sich auch um einen RSA-Schlüssel, der jede Stunde neu generiert wird. Eine Liste der unterstützten Verschlüsselungsverfahren sowie weitere Informationen werden zusätzlich ausgetauscht. Der Client generiert anschließend einen zufälligen 256-Bit-Sitzungsschlüssel und verschlüsselt diesen mit beiden RSA-Schlüsseln des Server. Danach wird ein Verschlüsselungsverfahren aus den vorhandenen Möglichkeiten ausgewählt. Der Client sendet nun den verschlüsselten Sitzungsschlüssel mit dem ausgewählten Verschlüsselungsverfahren an den Server. Jede weitere Kommunikation erfolgt verschlüsselt. Der Server entschlüsselt den Sitzungsschlüssel mit den beiden privaten Schlüsseln und sendet eine mit dem vereinbarten Verfahren verschlüsselte Quittung an den Client. Anschließend authentifiziert sich der Client. Wenn die Authentifizierung erfolgreich vorgenommen wurde, wird eine Arbeitsumgebung für den Benutzer geschaffen, indem Umgebungsvariablen gesetzt werden. Abschließend erfolgt der eigentliche Austausch der Nutzdaten. [DEER01]

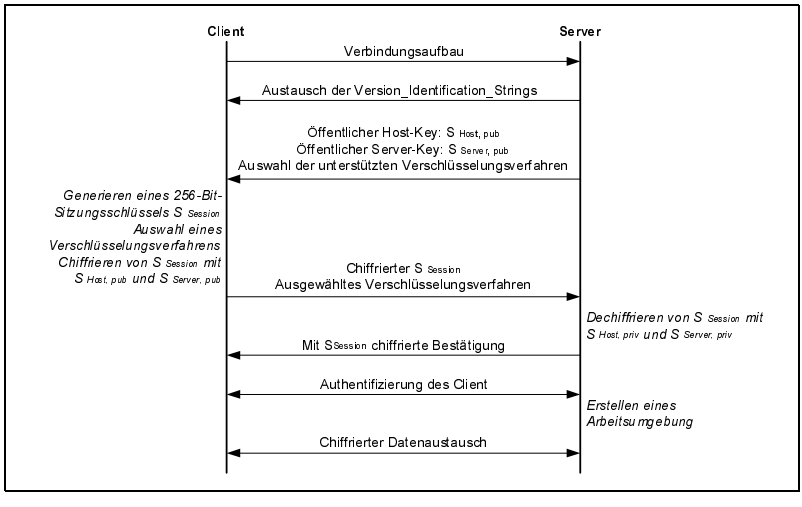


Abb. 6.10 SSH-Verbindungsaufbau

Mit SSH sind keine spontanen Verbindungen möglich, da vor dem ersten Verbindungsaufbau die entsprechenden Schlüssel generiert und auf den beteiligten Rechnern eingetragen werden müssen. Für die Authentifizierung des Client stehen vier Verfahren zur Verfügung. Dabei unterteilen sich die Verfahren in rechner- oder benutzerbezogene Authentifizierung:

1. **Rechnerbezogene Authentifizierung**
   - Authentifizierung über die richtige IP-Adresse des Client-Rechners
   - Authentifizierung über die richtige IP-Adresse des Client-Rechners, jedoch mit RSA-basierter Host-Authentifizierung: Der Client-Rechner benötigt dazu ein eigenes Schlüsselpaar. Der öffentliche Schlüssel wird

beim Server gespeichert. Der Client-Rechner muss bei der Authentifizierung nachweisen, dass er in Besitz des zugehörigen privaten Schlüssels ist.

2. **Benutzerbezogene Authentifizierung**
   - RSA-basierte Authentifizierung des Benutzers: Der Benutzer benötigt dazu ein eigenes Schlüsselpaar. Der öffentliche Schlüssel wird beim Server gespeichert. Der Benutzer muss bei der Authentifizierung nachweisen, dass er in Besitz des zugehörigen privaten Schlüssels ist.
   - Authentifizierung über Unix-Passwort

Bei der RSA-basierten Authentifizierung muss im Vorfeld die Echtheit des öffentlichen Schlüssels sowohl des Server als auch des Client bzw. des Benutzers überprüft werden. Die für eine Verbindung ausgewählten Authentifizierungsverfahren werden der Reihe nach bis zum ersten Erfolg oder bis zum definitiven Misserfolg abgearbeitet. Die Vorteile der Sicherheit über die Transportschicht (z.B. SSL oder SSH) lassen sich wie folgt zusammenfassen

- Alle vorhandenen TCP/IP-Protokollstapel und Application Programming Interfaces (API) wie WinSock, BSD[10]-Socket, Streams werden unterstützt.
- Effiziente Kontrolle bei langsamen Verbindungen ist vorhanden. Dies wird u.a. erreicht durch Header-Komprimierung nach van Jacobson und unterschiedlichen Verfahren zur Sättigungssteuerung, welche TCP/IP-Header untersuchen.
- Keine Probleme durch Fragmentierung, Pfad-MTU-Erkennung etc.
- Effektivere Kombination aus Komprimierung und Verschlüsselung als auf Paketebene

6.1.4 IP Security (IPsec)

Die Arbeitsgruppe der IETF hat 12 Basisspezifikationen von Request-for-Comments (RFC) veröffentlicht, welche die Arbeitsweise, Architektur, Schlüsselverwaltung, Protokolle und Transformationen beschreiben. Zu IPsec gehören die folgenden Basisprotokolle:

- Authentication Header (AH)
- Encapsulating Security Payload (ESP)
- Internet Key Exchange (IKE)
- Internet Security Association and Key Management Protocol (ISAKMP)

IPsec ist demnach eine Protokollfamilie, die unterschiedliche Möglichkeiten der Zusammenarbeit bietet. Die IPsec-Architektur definiert, wie die verschiedenen Komponenten zusammenarbeiten, siehe Abb. 6.11.

10 Berkeley Software Distribution

Die IPsec-Architektur, wie sie im Security-Kapitel behandelt wurde, definiert die Möglichkeit, Hosts und Gateways zu unterstützen. Sie verlangt explizit, dass die Hosts Vertraulichkeit mittels ESP genauso unterstützen wie Datenintegrität[11] sowie den Schutz vor wiederholter Sendung[12]. Die Architektur stellt die Semantik der IPsec-Protokolle und die Probleme dar, die bei der Interaktion der verschiedenen Protokolle untereinander auftauchen können. Die Dokumente für ESP[13] und AH[14] definieren das Protokoll, den Header für die Nutzdaten und die Leistungen, die sie erbringen. Zusätzlich werden Regeln für die Verarbeitung der Pakete festgelegt. Sie definieren allerdings nicht die Transformationen, die dafür notwendig sind. Dadurch besteht die Möglichkeit neue Transformationen zu definieren, was nicht zu einer Änderung der Basisprotokolle führen muss.

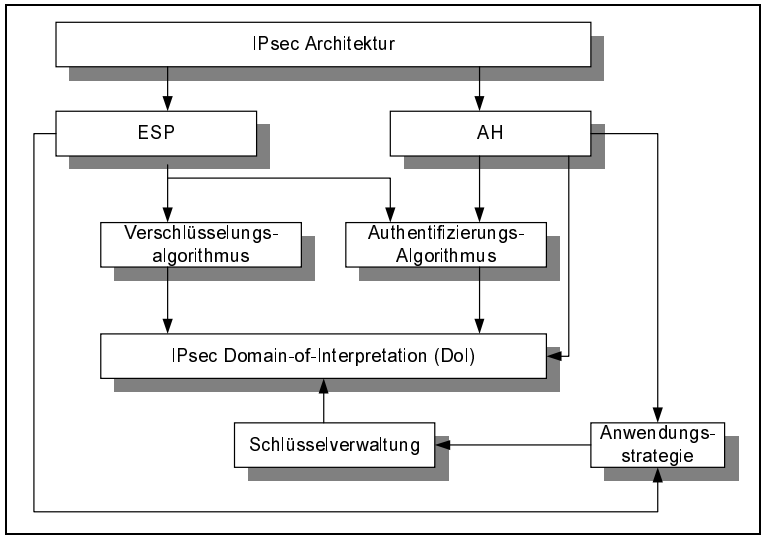


Abb. 6.11 IPsec-Architektur [PIPE98]

Die Transformationen definieren die Verknüpfungen, die auf die Daten angewendet werden, um sie zu schützen. Dazu gehören die Algorithmen, die Schlüsselgrößen und ihre Ableitung, der Transformationsprozess und alle algorithmischen Informationen. Dabei muss beachtet werden, dass die notwendigen Informationen so detailliert wie möglich sind, damit unterschiedliche Implementierungen zusammenarbeiten können. Wenn beispielsweise nicht festgelegt wurde, wie der Initialisierungsvektor abgeleitet wird, so können hier Unterschiede auftauchen, die zur Inkompatibilität führen.

11 Entweder mit AH oder ESP
12 Anti-Replay
13 RFC-2406
14 RFC-2402

Die Schlüssel werden bei IPsec mittels IKE erzeugt. IKE wird außerdem verwendet, um Schlüssel für andere Protokolle zu vereinbaren. Das IKE-Format ist dabei sehr allgemein gehalten worden, um es universell einsetzen zu können. Diese Unabhängigkeit wird dadurch geschaffen, dass die Parameter getrennt vom eigentlichen Protokoll selbst vereinbart werden. Diese zu vereinbarenden Parameter sind in einem separaten Dokument definiert, welches IPsec Domain-of-Interpretation (IPsec-DOI) genannt wird. [PIPE98]

Eine weitere wichtige Komponente, die noch nicht standardisiert ist, ist die Policy[15]. Die Policy ist eine Kernfrage, da zwei Entitäten nur dann miteinander kommunizieren können, wenn eine gemeinsame Transformation genutzt wird. Die Probleme der Policy sind Repräsentation und Implementierung. Die Repräsentation behandelt die Definition der Policy sowie deren Speicherung und Wiederherstellung, während die Implementierung die Policy für die tatsächliche Kommunikation verwendet. Dabei ist darauf zu achten, dass korrekte Filter und Regeln angewandt werden. [DOHA00]

Implementierung IPsec kann in den Endpunkten, den Hosts einer Kommunikation oder in den Gateways bzw. Routern implementiert und verwendet werden.

Tab. 6.2 Vor- und Nachteile möglicher IPsec-Implementierungen

Implementierungen	Betriebssystem	BITS	Router-Betriebssystem	BITW
Vorteile	Netzwerkdienste wie Fragmentierung sind integriert Sicherheit kann per Datenfluss angeboten werden Alle Modi werden unterstützt	Unabhängigkeit von dem verwendeten Betriebssystem Komplettlösung durch Integration in Firewall-Systeme	Sicherung eines kompletten Netzwerks Unterstützen von Hardware für verbesserte Performance Schnelle Integration neuer Merkmale	Unabhängig von den verwendeten Routern /Gateways Sehr gute Performance, da Hardware-Implementierung
Nachteile	Betriebssysteme müssen IPsec unterstützen Neue Lösungen können schlechter implementiert werden	Duplizierung der Eigenschaften der Netzwerkschicht wie Fragmentierung Unerwünschte Komplikationen durch Duplizierung	schlechte Performance bei reiner Software-Implementierung kein direkter Benutzerkontext herstellbar	sehr aufwendig, da jede Verbindung einzeln gesichert werden muss Update kann nicht ohne Hardware-Austausch durchgeführt werden

15 Anwendungsstrategie

Dabei wird bei der Implementierung im Host zwischen der Integration in das Betriebssystem und als Middleware zwischen der Netzwerk- und Datenübertragungsschicht im Protokollstapel[16] unterschieden. Vorteilhaft ist die bereitgestellte Ende-zu-Ende-Sicherheit, die Unterstützung aller IPsec-Modi und der Benutzerkontext, der jederzeit für die Authentifizierung aufrechterhalten werden kann. Bei der Router-/Gateway-Implementierung können Netzwerkbereiche geschützt werden. Hier unterscheidet man die native Implementierung direkt in das Betriebssystem der jeweiligen Geräte oder das Hinzufügen von zusätzlichen Geräten zwischen die Kommunikationsendpunkte[17]. Grundsätzliche Vorteile bestehen durch die Sicherung der Datenströme zwischen zwei Netzwerken und die Möglichkeit, Benutzer zu authentifizieren und zu autorisieren, wenn sie Virtual Private Networks (VPN) vertreten. Tab. 6.2 verdeutlicht die Vor- und Nachteile der verschiedenen Lösungen.

Bei allen Implementierungen muss über die Effizienz nachgedacht werde, was im nächsten Kapitel auch messtechnisch getan wird, da zusätzliche Operationen durch IPsec notwendig geworden sind:

- Öffentliche Schlüsseloperationen
- Generierung von Zufallszahlen
- Ver- und Entschlüsselung des Datenstroms
- Berechnung von Zugriffsschlüsseln

Die meisten Implementierungen legen die folgenden Komponenten fest:

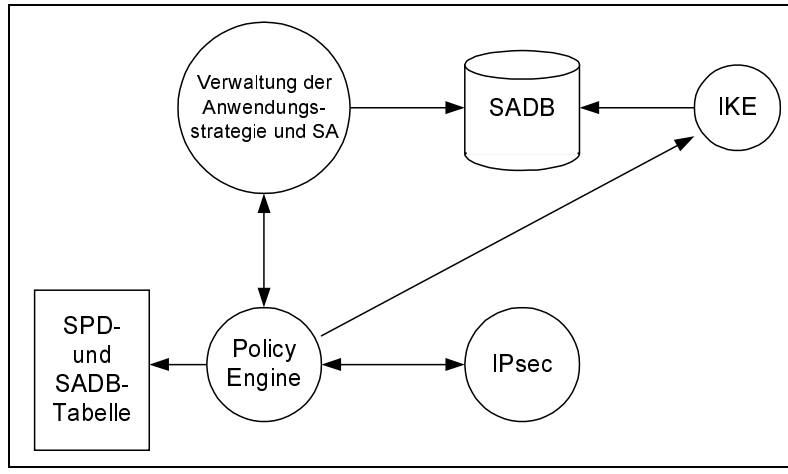
- **IPsec-Basisprotokolle:** Diese Komponente implementiert AH[18] und ESP[19]. Das Basisprotokoll verarbeitet die Header und interagiert mit der Datenbank für Sicherheitsstrategien und der Security Association Database (SADB), um den Sicherheitsgrad festzulegen, der für das Paket angewandt wird. Es werden dabei alle Problempunkte der Netzwerkschicht gehandelt, wie z.B. Fragmentierung und Path-MTU.
- **SADB für Sicherheitsstrategien:** Die Datenbank für die Sicherheitsstrategie ist eine wichtige Komponente, da sie den Sicherheitsgrad festlegt. Sie wird für ein- und ausgehende Pakete abgefragt. Bei ausgehenden Paketen wird die Sicherheitsstrategie der Basisprotokolle abgefragt, während bei eingehenden Paketen überprüft wird, ob der vereinbarte Sicherheitsgrad in der Anwendungsstrategie eingehalten wurde. Die SADB enthält eine Liste aller aktiven Sicherheitsassoziationen.
- **Internet Key Exchange (IKE):** Der IKE ist normalerweise ein Prozess auf der Benutzerebene, außer in einem eingebetteten Betriebssystem. Das Letztere ist beispielsweise in einem Router der Fall, der ein eigenes Betriebssys-

16 Bump-in-the-Stack (BITS)
17 Bump-in-the-Wire (BITW)
18 Authentication Header
19 Encapsulating Security Payload

tem hat, aber nicht zwischen Benutzer- und Kernel-Adressraum unterscheidet. IKE wird von der Verwaltung der Anwendungsstrategie[20] aufgerufen, kann aber ebenfalls von Teilnehmern direkt aufgerufen werden.

- **Verwaltung der Anwendungsstrategien und SA:** Diese Anwendungen verwalten Anwendungsstrategien und SA.

Abb. 6.12 Interaktion von IPsec-Komponenten



Die Wahl der Datenstruktur zur Speicherung der Datenbank für Sicherheitsstrategien und der SADB ist entscheidend für die Leistungsfähigkeit der Verarbeitung für IPsec. Dabei hängt die Implementierung von den Anforderungen an die Geschwindigkeit und von den Systemmöglichkeiten ab. Der IKE ist dafür verantwortlich, die SADB dynamisch mit Einträgen zu füllen. Dabei können die Dienste entweder durch den Aufbau einer Sicherheitsassoziation (SA) von der SADB oder einem Teilnehmer abgerufen werden. IKE unterstützt einerseits eine Schnittstelle, die die Datenbank für die Sicherheitsstrategie bei der Erzeugung einer neuen SA informiert, und andererseits eine Schnittstelle für entfernte IKE-Teilnehmer, die dadurch den Aufbau einer neuen SA anfordern können. IKE wird von der Policy Engine aufgerufen, wenn die Anwendungsstrategie nach einer SA oder einem Bündel von SA verlangt, um zwischen zwei Knoten sicher kommunizieren zu können. Das Modul zur Verwaltung der Anwendungsstrategien[21] wird auf der Benutzerebene implementiert. Der Benutzer interagiert mit diesem Modul für alle Belange, die mit der Anwendungsstrategie zusammenhängen. Dieses Modul arbeitet mit dem Kernel zusammen, um die Datenbank

20 Policy Engine
21 Policy Management

für Sicherheitsstrategien auf den neuesten Stand zu bringen. Außerdem können darüber Schlüssel manuell erstellt werden. [KEAT98a]

Sicherheitsassoziation (SA)

Die Sicherheitsassoziation oder Security Association (SA) bildet die Basis von IPsec. Sie stellt den Vertrag zwischen zwei kommunizierenden Entitäten dar und bestimmt unter anderem Transformation, Schlüssel, Zeitdauer der Schlüsselgültigkeit für das IPsec-Protokoll. Eine SADB hält dabei immer die vorhandene SA bereit. Dabei verläuft die SA immer nur in eine Richtung. Wenn zwei Hosts sicher mittels ESP kommunizieren wollen, dann haben beide jeweils zwei SAs:

- **SAausgehende:** Verarbeitung von ausgehenden Paketen
- **SAeingehende:** Verarbeitung von eingehenden Paketen

Die *SAausgehende* von Host *A* und die *SAeingehende* von Host *B* werden dabei dieselben kryptographischen Parameter benutzen. Dies gilt genauso für die umgekehrte Reihenfolge. Zusätzlich sind die SAs protokollspezifisch, wodurch bei der Verwendung von AH und ESP separate SAs erzeugt werden.

Eine weitere Komponente ist die Security Policy Database (SPD), welche die Sicherheitsstrategien verwaltet. Die Datenbank für die Sicherheitsstrategien arbeitet mit der SADB eng zusammen, um Pakete weiterzuverarbeiten. Dabei legt die Strategie fest, welche Charakteristika der Kommunikation zwischen zwei Hosts verwendet wird. Sie definiert zusätzlich, wie IP-Pakete behandelt werden.

Der Security Payload Index (SPI) beinhaltet ein 32-Bit-Feld bei ESP- und AH-Headern. Um eine Identifizierung von SA beim Empfänger zu ermöglichen, wird jeder SA ein eindeutiger SPI zugewiesen. Der Empfänger verwendet diesen Index, um auf die SADB zuzugreifen. Dabei ist der Empfänger dafür zuständig, die Eindeutigkeit für jedes Protokoll global, pro Datenquelle und pro Adresse auf dem Host zu garantieren. Es ist eine notwendige Bedingung, dass eine separate Menge von SPI pro Protokoll aufgebaut werden muss. Dabei legt IPsec nur fest, dass im Paket eine SA für das Paar <SPI, Zieladresse> eindeutig bestimmt wird. Wenn am Empfänger keine Eindeutigkeit garantiert werden kann, werden die Pakete bei der Sicherheitsüberprüfung durchfallen und es müssen Transformationen verwendet werden.

Der sendende Host benutzt Selektoren[22], um eindeutig auf einen Index in der sendenden SADB zu verweisen. Die Selektoren bilden dabei den IP-Verkehr auf eine IPsec-Anwendungsstrategie ab. Sie identifizieren bestimmte Anteile des Verkehrs als fein- oder grobkörnig. Der Rückgabewert ist eine SA, die alle Sicherheitsparameter bereits beinhaltet. Der Host, der den SPI festlegt, garan-

22 Selektoren sind: IP-Zieladresse, IP-Ursprungsadresse, Namen, übergeordnete Protokollschichten, Ports von Ziel oder Ursprung, eine Stufe von Datensensitivität (wenn IPsec-Systeme auch Sicherheit für Datenfluss unterstützen)

tiert dessen Einzigartigkeit. Der SPI wird wieder verwendet, wenn die SA nicht weiter benutzt wird. Solange die Zuordnung <SPI, Zieladresse> gültig ist und die SA 1:1 ist, bleibt ein SPI immer garantiert. Die Ursprungsadresse wird verwendet, wenn der Ziel-Host mehrere Adressen aufweist[23] Der empfangene Host nutzt den Tupel <SPI, Empfänger, Protokoll>[24], um die SA eindeutig zu identifizieren. Es kann zusätzlich die Senderadresse verwendet werden, was aber nicht Teil des Standards ist.

Die SA kann manuell oder durch ein Schlüsselmanagementprotokoll wie IKE gelöscht werden. Es ist bei sicheren Verbindungen notwendig, dass die Schlüssel regelmäßig erneuert werden, um Angriffsversuche zu unterbinden. Dabei wird eine bestehende SA komplett gelöscht und wieder neu erzeugt. Anschließend kann in diesem Fall der benutzte SPI wieder verwendet werden. Damit die Kommunikation nicht unterbrochen wird, wird eine neue SA vor dem Ablauf der Lebensdauer[25] definiert. In der Übergangsphase stehen somit kurz mehrere SAs zwischen den beiden Entitäten zur Verfügung. Zusammenfassend gibt es folgende Gründe, eine SA zu löschen:

- Die Lebensdauer ist abgelaufen.
- Die Schlüssel wurden kompromittiert.
- Die Anzahl der durch die SA verschlüsselten bzw. entschlüsselten oder authentifizierten Bytes hat einen bestimmten, durch die Strategie festgelegten Grenzwert überschritten.
- Die andere Endstelle verlangt, dass die SA gelöscht wird.

Die SA hält also einen Kontext aufrecht, um die Kommunikation zwischen zwei Entitäten zu sichern. Dafür muss sie protokollspezifische und generische Felder abspeichern. Das SPI-Feld des ESP- und AH-Headers enthält dafür die folgenden Felder, die mit den beschriebenen Funktionen ausgestattet sind:

1. **Sequence Number:** Die Seriennummer ist ein 32-Bit-Feld, welches für die ausgehende Verarbeitung genutzt wird. Sie wird jeweils um „1“ erhöht, wenn die SA benutzt wird, um ein Paket zu sichern. Dadurch können Replay-Attacken erkannt werden. Bei Aufbau einer SA wird das Feld auf „0“ gesetzt.
2. **Sequence Number Overflow:** Dieses Feld wird ebenfalls bei ausgehender Verarbeitung benutzt. Es wird eingesetzt, wenn die Seriennummern überlaufen. Die Strategie legt dabei fest, ob die SA weiterhin benutzt werden kann, um Pakete zu verarbeiten.

23 Z.B. durch mehrere Netzwerkkarten
24 Wobei mit Empfänger die Zieladresse des IP-Header gemeint ist.
25 Time-to-Live (TTL)

3. **Anti-Replay Window:** Eingehende Verbindungen werden auf wiederholtes Senden von Paketen untersucht. Dadurch werden Replay-Attacken verhindert und der sendende Host kann zusätzlich identifiziert werden.
4. **Lifetime:** Jede SA wird mit einer bestimmten Lebensdauer versehen, nach deren Ablauf die SA nicht mehr verwendet werden kann. Dabei kann die Lebensdauer in Byte und/oder Sekunden angegeben werden. Es gibt eine Soft Lifetime und eine Hard Lifetime, um die Kommunikation nicht unterbrechen zu müssen. Dabei wird die Soft Lifetime genutzt, um den Kernel zu warnen, dass die Lebensdauer der SA demnächst abläuft. Dadurch kann der Kernel eine neue SA erzeugen, bevor die Hard Lifetime abläuft.
5. **Mode:** Die Nutzdaten werden im Tunnel- und Transportmodus unterschiedlich behandelt. Das Feld kann entweder für den Tunnelmodus, den Transportmodus oder frei[26] eingesetzt werden. Wenn das Feld frei eingesetzt wird, dann muss die Information über den Modus von einer anderen Stelle geholt werden, wie z.B. von den Sockets.
6. **Tunnel Destination:** An dieser Stelle wird der Zielort des Tunnels angegeben, was der IP-Zieladresse des äußeren Headers entspricht.
7. **PMTU Parameters:** Wenn der Tunnelmodus eingesetzt wird, dann muss die Path Maximum Transmission Unit (PMTU) festgelegt werden, die zum Fragmentieren der Pakete benutzt wird. Als Teil des Feldes speichert die SA zwei Werte: die PMTU und das Aging-Field (AF).

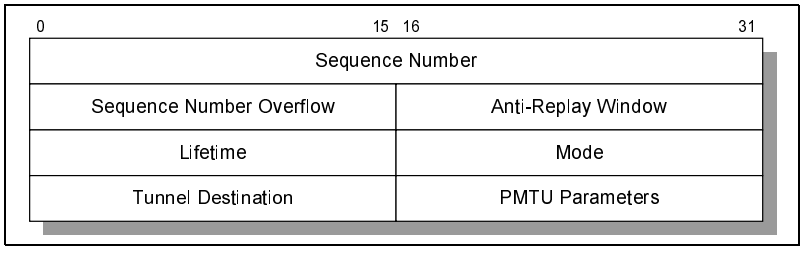


Abb. 6.13 SPI-Header

Abschließend legt die Sicherheitsstrategie fest, welche Sicherheitsdienste auf ein Paket angewendet werden. Diese Informationen werden alle in einer Datenbank[27] gespeichert. Diese Datenbank wird durch die Selektoren mit einem Index versehen. Die Informationen werden sowohl bei eingehenden als auch bei ausgehenden Paketen zu Rate gezogen. Es können separate SPDs aufgebaut werden, um asymmetrische Strategien zu ermöglichen, wie beispielsweise verschiedene Sicherheitsdienste für eingehende und ausgehende Pakete zwischen zwei Hosts. Das Key Management Protocol (KMP) stellt trotzdem eine bidirek-

26 Wildcard
27 Security Policy Database (SPD)

tionale SA her. Das Tunneln und Verschachteln der Pakete wird allerdings meist ebenfalls symmetrisch vorgenommen.

Für den ausgehenden Verkehr ist der Rückgabewert einer Abfrage der SAs in der SADB ein Zeiger auf eine SA oder ein SA-Bündel, wenn man davon ausgeht, dass die Sicherheitsassoziationen bereits aufgebaut wurden. Wenn die SAs noch nicht aufgebaut wurden, wird das KMP aufgerufen, um eine Verbindung aufzubauen. Die Sicherheitsstrategie fordert von einer Strategieverwaltung, dass die Strategien hinzugefügt, gelöscht und modifiziert werden können. Die Datenbank für Sicherheitsstrategien wird im Kernel gespeichert. Die jeweilige IPsec-Implementierung sollte dabei eine Schnittstelle zur Verfügung stellen, um die Datenbank manipulieren zu können. [DOHA00]

Eingehender und ausgehender Datenverkehr

Es wird hier hauptsächlich zwischen eingehendem und ausgehendem Datenverkehr unterschieden. Bei der Verarbeitung von ausgehendem Verkehr legen die Pakete ihren Weg von der Transportschicht in die IP-Schicht zurück. Die IP-Schicht fragt die Datenbank für Sicherheitsstrategien ab, um die Sicherheitsdienste zu bestimmen, die auf dieses Paket angewendet werden sollen. Die Eingabe in die Datenbank für Sicherheitsstrategien sind die Selektoren. Die Ausgabe beinhaltet eine der folgenden Möglichkeiten:

- **Discard:** Die Pakete werden fallen gelassen und das Paket wird nicht verarbeitet.
- **Bypass:** Die Sicherheit wird umgangen und die IP-Schicht fügt hierfür den Nutzdaten den IP-Header hinzu und verschickt das Paket.
- **Apply:** Die Sicherheit wird angewendet und der Verweis auf die SA, falls diese schon existiert, zurückgeliefert. Andernfalls wird KMP aufgerufen, um eine SA zu erzeugen. Wenn die SA bereits existiert, enthält die Datenbank je nach Strategie einen Verweis auf die SA oder das SA-Bündel. Wenn die Ausgabe der Strategie verlangt, dass IPsec auf die Pakete angewandt wird, dann werden die Pakete so lange nicht übertragen, bis die SA erzeugt wird.

Falls die SA noch nicht vorhanden sind, wartet die IPsec-Implementierung, bis die SAs für dieses Paket etabliert sind. Anschließend werden die entsprechenden AH- und ESP-Header hinzugefügt. Die SAs enthalten dabei alle notwendigen Informationen. Abb. 6.14 zeigt ein Verarbeitungsbeispiel. In diesem Fall tunnelt der Host *A* mittels ESP-Header ein Paket zum Router *B*, authentifiziert sich aber bis zum Host *B*. In diesem Fall werden vier SAs etabliert: zwei zum Senden und zwei zum Empfangen. Bei den ausgehenden SAs betrifft dies SA_1 und SA_2, wobei SA_1 zwischen Host *A* und dem Router *B* und SA_2 zwischen dem Host *A* und dem Host *B* besteht. Dabei ist die Reihenfolge bei IPsec sehr wichtig. Wenn SA_2 nach SA_1 angewandt würde, dann würde das Paket inkorrekt erzeugt werden.

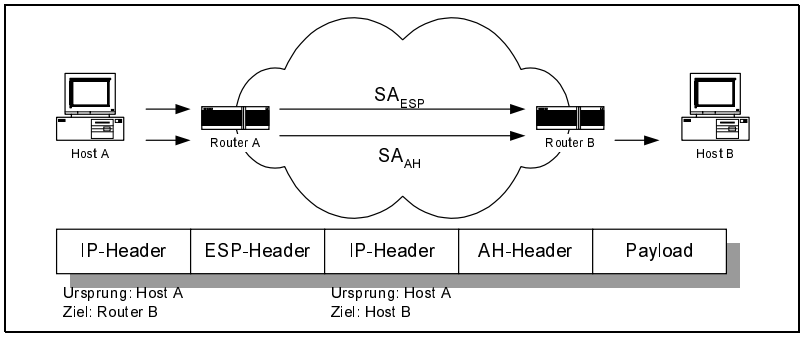


Abb. 6.14 Verarbeitung von ausgehendem Verkehr

Die Verarbeitung von eingehendem Verkehr unterscheidet sich von der des ausgehenden Verkehrs. Wenn das IP-Paket irgendwelche IPsec-Header beinhaltet, dann wird es von der IPsec-Schicht verarbeitet. Die IPsec-Schicht extrahiert den SPI[28], die Senderadresse und die Zieladresse des IP-Pakets. Sie versieht die SADB mittels des Tupel <SPI, Zieladresse, Protokoll>[29] mit einem Index. Das Protokoll ist entweder AH oder ESP. Nachdem die Protokollnutzlast verarbeitet wurde, wird die Strategie nach der Gültigkeit gefragt. Die Selektoren werden verwendet, um die Strategie abzurufen. Der Prozess, mit dem die Gültigkeit überprüft wird, beinhaltet die Kontrolle der ordnungsgemäßen Nutzung der SA. Hier wird überprüft, ob die Ursprungsadresse und die Zieladresse in der SA mit dem, was vorgeschrieben wird, übereinstimmt bzw. ob die SA das Transportschichtprotokoll schützt. Bei getunnelten Paketen befinden sich die Selektorfelder im inneren Header und nicht im äußeren. Würde man Indizes in die Datenbank für Sicherheitsstrategien setzen, würde das zu falschen Ergebnissen führen, da die Einträge aufgrund der echten Ursprungs- und Zieladresse vorgenommen wurden und nicht aufgrund der Tunneleinträge.

An dem vorherigen Beispiel kann man die Funktionsweise gut wiedergeben. Hier werden die Pakete von Host *B* nach Host *A* getunnelt. Auf dem Host *A* bestimmt die angewendete Strategie, dass von *B* ankommende Pakete mittels ESP getunnelt werden und dass der Ursprung des Tunnels der Router *B* ist. Es wäre nicht korrekt, Indizes in die Datenbank für Sicherheitsstrategien zu setzen, bei denen der Router *B* als Ursprungsadresse benutzt wird und nicht der Host *B*. Wenn die IPsec-Schichten die Strategie bestätigt haben, dann werden die IPsec-Header entfernt und die Pakete an die nächste Schicht weitergeleitet. Die nächste Schicht ist entweder die Transport- oder Netzwerkschicht. [KEAT98a]

28 Security Parameters Index

29 Je nach Implementierung wird zusätzlich die Ursprungsadresse verwendet.

Internet Key Exchange (IKE)

IKE ist ein hybrides Protokoll und wird nach RFC-2409 definiert. Es basiert auf dem Rahmenwerk, welches durch Internet Security Association and Key Management Protocol (ISAKMP) in der Spezifikation RFC-2408 beschrieben ist und implementiert Teile von zwei Protokollen zum Austausch von Schlüsseln: Oakley und SKEME. Zusätzlich werden durch IKE auch eigene Protokolle definiert.

Oakley wurde von Hilarie Orman, Universität Arizona, entwickelt, welches jedem Teilnehmer ermöglicht, in den Zuständen des Protokolls in einer eigenen Geschwindigkeit fortzuschreiten. Von Oakley verwendete IKE die Idee der verschiedenen Modi, von denen jeder gleichwertige Ergebnisse erzielt, obwohl der Austausch der Informationen in unterschiedlichen Geschwindigkeiten vorgenommen wird. Bei Oakley existiert keine Definition der Information, die in jeder Nachricht ausgetauscht werden. Durch die Einschränkung der Flexibilität von Oakley limitiert IKE die vielen Möglichkeiten, die Oakley bereitstellt, erlaubt aber trotzdem unterschiedliche Modi in einer eng definierten Art.

SKEME ist ein weiteres Protokoll zum Schlüsselaustausch, welches von Hugo Krawcyk entwickelt wurde. Es definiert einen Typ authentifizierten Schlüsselaustauschs, bei dem die Teilnehmer eine Verschlüsselung mittels öffentlicher Schlüssel für ihre gegenseitige Authentifizierung sowie Komponenten des Austauschs gemeinsam nutzen. Jeder Teilnehmer verschlüsselt eine Zufallszahl mit dem öffentlichen Schlüssel des Teilnehmers, sodass beide Zufallszahlen nach ihrer Entschlüsselung zum endgültigen Schlüssel beitragen. Dabei kann optional ein Austausch nach Diffie-Hellman zusammen mit SKEME vorgenommen werden, um so ein Perfect Forwarding Secrety (PFS) zu ermöglichen. IKE verwendet diese Techniken von SKEME für die eigenen Authentifizierungsmethoden und verwendet ebenfalls die Notation für das schnelle Erneuern von Schlüsseln ohne PFS.

ISAKMP legt fest, wie zwei Teilnehmer miteinander kommunizieren, wie die Nachrichten, die sie zur Kommunikation nutzen, aufgebaut sind und die Zustandsübergänge, die die Teilnehmer durchlaufen. Es stellt zusätzlich eine Authentifizierung der Teilnehmer, den Austauschmodus der Informationen für einen Schlüsselaustausch und die Form der Vereinbarung der Sicherheitsdienste zur Verfügung. Allerdings wird nicht definiert, wie ein bestimmter Schlüsselaustausch vorgenommen wird und wie die Eigenschaften sind, die für das Aufbauen von SA notwendig sind. Für IPsec wird der Schlüsselaustausch durch das Protokoll Internet Key Exchange (IKE) definiert. IKE benutzt dabei die Sprache von ISAKMP und definiert die Anzahl von Austauschvorgängen und Optionen, die auf diese angewandt werden können. Das Endergebnis eines IKE-Austauschvorgangs ist ein authentifizierter Schlüssel und aufeinander abgestimmte Sicherheitsdienste, die eine IPsec-SA zur Folge haben.

Nachrichten, die in einer auf ISAKMP-basierten Schlüsselverwaltung ausgetauscht werden, beinhalten ISAKMP-Nutzdaten und -Header. Der Initiator-Cookie und der Responder-Cookie werden von jedem Teilnehmer erzeugt und zusammen mit der Nachrichten-ID verwendet, um den Zustand zu definieren, der einen Austausch nach ISAKMP auszeichnet. Das Feld Next Payload indiziert, welche der verschiedenen ISAKMP-Nutzdaten dem Header sich direkt anschließen. Die Version von ISAKMP ist durch die Versionsfelder gekennzeichnet. Der spezifische Typ des ISAKMP-Austauschs wird im Feld für den Exchange Type festgelegt. Die gesamte Länge der Nachricht, inklusive des Headers, wird im Feld Message Length wiedergegeben. Die Flags geben dem Empfänger weitere, zur Nachricht gehörige Informationen an. In einem für Erweiterungen offenen 8-Bit-Feld sind drei unterschiedliche Flags definiert:

- **Encryption:** Das Verschlüsselungsflag zeigt an, dass die folgenden Nutzdaten verschlüsselt sind.
- **Commit:** Das Bestätigungsflag zeigt an, dass der Teilnehmer eine Benachrichtigung haben möchte, wenn der Austausch abgeschlossen ist.
- **Authentification Only:** Das Bit für die Authentifizierung von Nachrichten ermöglicht zusätzlich die Wiederherstellung des Schlüssels.

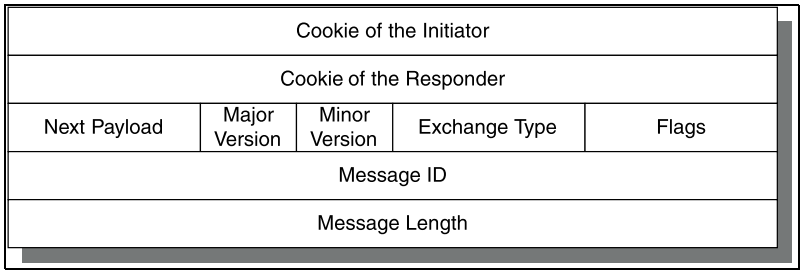


Abb. 6.15 ISAKMP-Header

Es sind 13 verschiedene Nutzdaten in ISAKMP definiert. Dabei sind einige Nutzdaten voneinander abhängig, wie beispielsweise der Typ Transformation vom Typ Proposal umschlossen wird, die wiederum vom Typ SA umschlossen werden. Einige der Nutzdaten definieren ebenfalls Attribute, die spezifisch für den jeweiligen Typ sind. Beispielsweise definieren Nutzdaten vom Typ Certification, welche Zertifikatart sie beinhalten. ISAKMP definiert somit Nutzdaten, um bestimmte Konstrukte während eines Austauschs zu beschreiben. Dabei sind einige generisch[30], andere spezifisch[31]. Die generischen Nutzdaten besitzen alle einen identischen Aufbau und unterscheiden sich nur durch das Feld, welches die Nutzdaten beschreibt. [MSST98]

30 Z.B. Hash-Werte oder Nonce-Nutzdaten, die für den Austausch notwendige Pseudo-Zufallszahlen enthalten.

31 Z.B. Zertifikate oder SA

IKE benutzt zwei Phasen von ISAKMP, um die SA für IPsec festzulegen. Dabei unterscheiden sich IKE von ISAKMP durch die Definition seiner Attribute. IKE definiert zwei Austauschvorgänge in Phase I und einen in Phase II. Dazu kommen noch zwei zusätzliche Austauschvorgänge, um die SA ordnungsgemäß aufrechtzuerhalten. Für die Austauschvorgänge in Phase I benutzt IKE den Identity Protection Exchange (IPE) und den Aggressive Exchange (AE) aus ISAKMP und nennt sie den Haupt- und Aggressive-Modus. Für Phase II definiert IKE den Quick Mode Exchange (QME), welcher die Sicherheitsdienste für andere Protokolle als IKE[32] verhandelt. Die beiden anderen Austauschvorgänge, die IKE definiert, sind informelle Austauschvorgänge, in denen sich Teilnehmer am IKE-Protokoll über Fehler und Statusinformationen benachrichtigen können, sowie eine neue Gruppe von Austauschvorgängen, die es den Teilnehmern erlaubt, die Benutzung einer neuen Diffie-Hellman-Gruppe untereinander zu vereinbaren.

Die Modi der ersten Phase, Haupt- und Aggressive-Modus, erreichen beide dasselbe Ziel: sie sichern und authentifizieren den Kommunikationskanal[33] und die Erzeugung authentifizierter Schlüssel, welche für die Vertraulichkeit, Nachrichtenintegrität und Authentifizierung der Nachrichtenquelle für die IKE-Kommunikation zwischen den beiden Teilnehmern bereitzustellen sind. Alle anderen definierten Austauschvorgänge in IKE haben anschließend eine authentifizierte IKE-SA als Vorbedingung. Daher muss ein Austauschvorgang der Phase I entweder im Haupt- oder Aggressiv-Modus ausgeführt werden, bevor ein weiterer Austausch stattfinden darf.

Der Haupt-Modus benutzt sechs Nachrichten in drei Durchgängen, um eine IKE-SA aufzubauen. Diese drei Schritte sind die SA-Verhandlungen, welche aus dem Diffie-Hellman-Austausch, Nonce-Austausch und der Authentifizierung der Teilnehmer bestehen. Die wichtigsten Eigenschaften des Haupt-Modus sind der Schutz der Identität und die volle Nutzung der Möglichkeiten zur Verhandlung von ISAKMP. Der Aggressive-Modus unterscheidet sich nur hinsichtlich der Anzahl der verwendeten Nachrichten, die sich auf drei halbiert. Durch die Begrenzung der Anzahl sind die Verhandlungsmöglichkeiten eingeschränkt. Zusätzlich wird der Identitätsschutz nicht unterstützt, da keine Verschlüsselung von eventuell übertragenden IDs und Zertifikaten erfolgt.

Die IKE-SA hat verschiedene Parameter, auf die sich die Teilnehmer während der Verhandlung geeinigt haben. Da einige IKE-Nachrichten verschlüsselt und authentifiziert sind, müssen sich die Teilnehmer auf eine Art der Verschlüsselung und Authentifizierung für Nachrichten einigen. Da jeder Teilnehmer die Identität des anderen authentifizieren muss, müssen sie sich auch darüber verständigen. Für alle diese Parameter definiert IKE Attribute und die möglichen

32 Hauptsächlich IPsec
33 IKE-SA

Werte, die diese Attribute haben können. Diese Parameter[34] werden auch Protection Suite (PS) genannt. Die PS wird als Einheit durch den Austausch von ISAKMP-Nachrichten verhandelt. Jedes der Attribute einer PS ist dabei in Nutzdaten vom Typ Transformation enthalten. Zusätzlich zu den notwendigen Attributen können auch optionale Attribute als Teil der PS vereinbart werden[35]. Es wird zwar ein Hash-Algorithmus ebenfalls verhandelt, aber standardmäßig eine HMAC[36]-Version des Hash-Algorithmus als Pseudo-Zufallsfunktion verwendet, um einen scheinbar zufälligen Bitstrom zu erzeugen.

HMAC ist eine besondere Form des Keyed Hashing. Key Hashing kann benutzt werden, um einen Datenfluss zu authentifizieren. Dies wird gemacht, indem der Fluss in leicht verwertbare Stücke zerlegt und ein MAC[37] für jedes einzelne Stück berechnet wird. Die MAC werden dann Teil des Datenflusses und dazu verwendet, um die Integrität des empfangenen Flusses zu kontrollieren. Ein weiterer Vorteil besteht darin, dass die Erzeugung eines Hash-Digest sehr viel schneller ist als mittels einer digitalen Signatur. HMAC ist in der Spezifikation RFC-2104 definiert und kann mit jeder beliebigen Hash-Funktion verwendet werden. SHA[38] wird dann zu HMAC-SHA und MD5[39] zu HMAC-MD5. Die Konstruktion von HMAC ist kryptographisch stärker als die darunter liegende Hash-Funktion, sodass mögliche Lücken in MD5 oder SHA sich nicht unmittelbar auswirken würden. Ein HMAC ist ein verschachtelter Keyed Hash. Er benutzt zwei Werte, einen inneren und einen äußeren, um den Status des HMAC aufrechtzuerhalten. Der HMAC der Nachricht *M*, welcher auf dem Hash-Algorithmus *H* basiert und den Schlüssel *K* benutzt, verwendet ein Array mit 64 Elementen mit dem Wert 0x36, welches als *ipad* bezeichnet wird sowie das *opad* als 64-elementiges Array mit dem Wert 0x5c. Der HMAC, welcher immer für die Authentifizierung von Nachrichten in IPsec verwendet wird, ist wie folgt definiert: [KBC97]

$$HMAC(K,M) = H(K \quad XOR \quad opad, H(K \quad XOR \quad ipad, M))$$

Bei Austausch mittels Diffie-Hellman legt die Diffie-Hellman-Gruppe die Parameter fest, die verwendet werden müssen. Dafür definiert IKE fünf Gruppen, um eindeutige Werte zuweisen zu können. Dabei gibt es drei Typen einer traditionelle Exponentiation über einen Primzahl-Modulus (MODP) sowie zwei Typen von elliptischen Kurven (EC2N)[40]. Die folgenden Gruppen sind in IKE definiert:

34 Z.B. Hash-Algorithmus, Authentifizierungsmethode, Diffie-Hellman-Gruppe
35 Am meisten wird die Lebensdauer als optionales Attribut verwendet.
36 Ist ein verschachtelter Key Hash, welcher in RFC-2104 beschrieben ist.
37 Message Authentication Code
38 Secure Hash Algorithm
39 Message Digest Version 5
40 Eine über G[P](ECP) und die andere über G[2^N](EC2N)

1. MODP-Gruppe mit 768-Bit-Modulus
2. MODP-Gruppe mit 1024-Bit-Modulus
3. EC2N-Gruppe mit 155-Bit-Feldgröße
4. EC2N-Gruppe mit 185-Bit-Feldgröße
5. MODP-Gruppe mit 1680-Bit-Modulus

Neue Gruppen können ebenfalls einfach definiert werden. Es ist lediglich festgelegt, dass die Gruppe 1 implementiert sein muss. Gruppe 1 und 3 haben dabei ungefähr die gleiche Sicherheit für einen Austausch. Ähnlich verhält es sich mit 2 und 4. Der Unterschied liegt hauptsächlich in der größeren Geschwindigkeit, die mit elliptischen Kurven über traditionelle Exponentiation in einem finiten Feld (Gruppe 1 und 2) erreicht werden kann. Heute gibt es allerdings noch keine elliptische Kurve, die sich analog zu Gruppe 5 verhält.

Das wichtigste Attribut für den IKE-Austausch ist die Authentifizierung. Dabei kann sich ein IKE-Austausch aufgrund der Authentifizierungsmethode ändern. Folgende Methoden sind möglich:

- Preshared Keys (PK)
- Digitale Signaturen, die Digital Signature Algorithm (DAS) verwenden.
- Digitale Signaturen, die Rivest Shamir Adelman (RSA) Algorithmus verwenden.
- Zwei ähnliche Methoden der Authentifizierung durch den Austausch von verschlüsselten Nonces[41]

Diese Attribute werden zwischen den Teilnehmern als Erstes verhandelt. Dies sind die extern sichtbaren Eigenschaften einer IKE-SA. Jede Seite enthält jedoch auch geheime Informationen, die nicht sichtbar sind. Es werden vom Teilnehmer vier Geheimnisse erzeugt:

- **SKEYID:** auf diesem Geheimnis basieren alle anderen Geheimnisse.
- **SKEYID_d:** wird zur Ableitung von Schlüsselmaterial von IPsec-SA verwendet.
- **SKEYID_a:** ermöglicht die Datenintegrität sowie die Authentifizierung der Datenquelle.
- **SKEYID_e:** verschlüsselt die IKE-Nachrichten.

Die Erzeugung von SKEYID hängt von der verhandelten Methode der Authentifizierung ab. Alle anderen SKEYID-basierten Geheimnisse werden auf identische Weise generiert, unabhängig von der Authentifizierungsmethode. Dabei steuert jede Teilnehmerseite einen Cookie und eine Nonce zur Generierung eines SKEYID-Zustands bei: Der Initiator gibt seinen Cookie CKY-I und seine Nonce N_i, wie der Responder mit CKY-R und N_r. Diese werden als Beweis für die Aktivität[42] verwendet, indem jede Seite der anderen ihren Cookie und

41 Pseudo-Zufallszahlen, die nur ein einziges Mal benutzt werden.
42 Proof-of-liveliness

Nonce zeigt. Beide Seiten versorgen ihr Geheimnis mit neuen Informationen. Als Ergebnis de Diffie-Hellman-Austauschs teilen die Teilnehmer ein gemeinsames Diffie-Hellman-Geheimnis g^{xy}, welches auch für die Erzeugung von SKEYID verwendet wird. Die Generierung eines SKEYID-Zustands kann nun folgendermaßen dargestellt werden, wobei PRF die Pseudo-Zufallsfunktion darstellt, die normalerweise die HMAC-Version der vereinbarten Hash-Funktion ist:

$$SKEYID = PRF(preshared - key, N_i | N_r)$$
$$SKEYID = PRF(N_i | N_r, g^{xy})$$
$$SKEYID = PRF(hash(N_i | N_r), CKY - I | CKY - R)$$

Bei zu kleinen Ausgaben der PRF für die Verschlüsselungsschlüssel muss SKEYID_e über Feedback- und Verkettungsoperationen erweitert werden[43]. Austauschvorgänge in der Phase I werden durch das Berechnen eines Hashes authentifiziert, den nur die beteiligten Seiten kennen können. Da eine Hash-Funktion nicht umkehrbar ist, kann ein Hash-Digest im Klartext übertragen werden. Die korrekte Erzeugung der Digest identifiziert somit die Teilnehmer. Diese Berechnung ist von der vereinbarten Methode der Authentifizierung unabhängig. Der Initiator und der Responder berechnen ihre Werte dabei auf unterschiedliche Arten:

$$HASH - I = PRF(SKEYID, g^i | g^r | CKY - I | CKY - R | SA - Offer | ID - I)$$
$$HASH - R = PRF(SKEYID, g^r | g^i | CKY - R | CKY - I | SA - Offer | ID - R)$$

Hierbei sind *gi* und *gr* die öffentlichen Zahlen nach Diffie-Hellman des Initiators bzw. des Responders, während das SA-Angebot die Nutzdaten der SA mit allen PS enthält, die dem Responder vom Initiator angeboten werden. ID-I und ID-R sind die Identitäten des Initiators und Responders. Es gilt zu beachten, dass alle SA-Nutzdaten eingeschlossen werden, um Angriffe entlang des Übertragungswegs zu verhindern. Die IKE-SA ist nun bidirektional etabliert und kann sowohl eingehenden als auch ausgehenden Verkehr schützen. Beide Teilnehmer können unabhängig voneinander die Phase II initiieren. Die Cookies im ISAKMP-Header werden nicht ausgetauscht, wenn der Responder zum Initiator wird, da das Cookie-Paar benutzt wird, um die entsprechende SA in der IKE-SADB zu identifizieren. Alle Verschlüsselungen in IKE müssen im CBC[44]-Modus durchgeführt werden, weshalb zusätzlich ein Initialisierungsvektor (IV)

43 Z.B. HMAC-MD5 erzeugt eine 128-Bit-Ausgabe; Blowfish kann aber 448 Bit verlangen.

44 Ciper Block Chaining

benötigt wird. Dieser wird zu Beginn durch das Aneinanderfügen zweier öffentlicher Werte nach Diffie-Hellman berechnet. Nach jeder Verschlüsselung und Entschlüsselung wird der IV des letzten verschlüsselten Blocks verarbeitet. Dadurch werden die Verschlüsselungsoperationen blockweise miteinander verkettet.

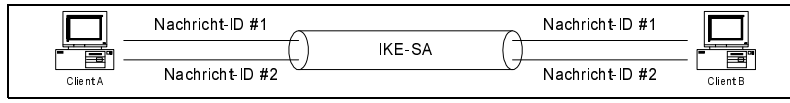


Abb. 6.16 IKE-SA mit Phase I und Phase II

Nach der Etablierung der IKE-SA wird diese benutzt, um SA für andere Sicherheitsprotokolle wie IPsec zu etablieren. Diese SA werden dann mittels eines Austauschvorgangs im Quick Mode erzeugt. Dabei können mehrere Austauschvorgänge unter dem Schutz einer einzelnen IKE-SA durchgeführt werden. Es verhandeln beim QME zwei Teilnehmer die Charakteristika einer IPsec-SA und generieren den Schlüssel dafür. Die Nachrichten werden dabei mittels PRF-Funktionen authentifiziert, was durch die HMAC-Version der vereinbarten Hash-Funktion geschieht. Der SKEYID_a-Wert mit der IKE-SA wird als Schlüssel genutzt, um die gesamte Nachricht im Quick-Mode zu authentifizieren. Dies stellt sowohl die Datenintegrität als auch die Authentifizierung der Datenquelle zur Verfügung, da man nach dem Empfang weiß, dass die Nachricht nur vom authentifizierten Teilnehmer stammen kann und die Daten nicht zwischendurch geändert wurden. Die Entschlüsselung mittels SKEYID_a sorgt außerdem für die Vertraulichkeit des Austauschvorgangs.

Um mehrere Austauschvorgänge gleichzeitig ausführen zu können, werden mittels der Nachrichtenidentifizierung von ISAKMP die IKE-Nachrichten vervielfacht, wie Abb. 6.16 verdeutlicht. Dabei erhält jeder einzelne Austausch im Quick-Mode eine einmalige Nachricht-ID, die von den Cookies im ISAKMP-Header verwendet wird, um den Status zu identifizieren, auf den sich eine bestimmte Nachricht bezieht. Da alle Nachrichten von derselben IKE-SA geschützt werden, muss der Initialisierungsvektor (IV) bezogen auf die Ver- und Entschlüsselung koordiniert werden können, um zu verhindern, dass zwei IKE-Teilnehmer ihren Gleichtakt verlieren. Die Nachricht-ID wird ebenfalls zu diesem Zwecke benutzt. Jeder Austausch im Quick-Mode beginnt mit einem einmaligen IV, der von dem IV der Phase I und der Nachricht-ID der Austauschvorgänge der Phase II abgeleitet wird. Der IV für einen bestimmten Quick-Mode ist ein laufender IV, vergleichbar mit dem der Phase I. Der ursprüngliche IV ist von einem deterministischen Algorithmus abgeleitet worden. Nachfolgende IVs werden jeweils so definiert, dass sie der letzte Block der letzten Nachricht sind.

Nach Beendigung eines Austauschs im Quick-Mode und der Bildung von IPsec-SA kann der Status, welcher aus dem laufenden IV, den Noncen und den Diffie-Hellman-Werten besteht und der zur Durchführung des Austauschs erzeugt wurde, gelöscht werden. Der Status, der für die IKE-SA geschaffen wurde, wird aber nicht gelöscht, da dieser dazu dient, die folgenden Nachrichten im Quick-Mode zu schützen.

Der Quick-Mode leitet die Schlüssel für die IPsec-SA aus einem SKEYID_d-Status ab. Dieser Schlüssel wird in einer Pseudo-Zufallszahl[45] mit den ausgetauschten Noncen, dem Protokoll und dem SPI aus dem Angebot der IPsec-SA verwendet. Dies garantiert einmalige Schlüssel für jede SA. Alle IPsec-Schlüssel werden von derselben Quelle abgeleitet und stehen deshalb in Zusammenhang miteinander. Damit ein Angreifer nicht alle Schlüssel der IPsec-SA ableiten kann, wenn er die SKEYID_d der IKE-SA bestimmt hat, wird ein zusätzlicher Diffie-Hellman-Austausch durchgeführt. Das daraus resultierende Geheimnis wird für die Erzeugung von IPsec-Schlüsseln verwendet.

Zusätzlich können noch Informationen zur Identität der Teilnehmer in dieser Phase II ausgetauscht werden. Dies ist zwar bereits in der Phase I geschehen, wird jedoch in der Phase II genutzt, um Informationen zur Selektion auszutauschen, die den Zweck der Nutzdaten der SA beschreiben. Die IPsec-DOI[46] definiert Typen der Identität, Port und Protokollfelder für die Nutzdaten vom Typ ID im ISAKMP. Da sich die DOI mit Austauschvorgängen der Phase II beschäftigt, werden diese Informationen auch verwendet. Die Identität kann dabei eine einfache Adresse, ein Adressintervall oder ein Subnetz sein. Auswahlinformationen in der Form von Nutzdaten vom Typ ID werden vom Initiator zusammen mit den Angeboten zur SA an den Responder geschickt. Dabei bezieht sich die Auswahlinformation auf alle SAs, die in einem einzelnen Austausch im Quick-Mode ausgehandelt werden. Der Responder kann diese Informationen nutzen, um seine eigene Anwendungsstrategie zu überprüfen und festzustellen, ob es dem Initiator erlaubt wird, für den jeweiligen Zweck eine SA aufzubauen. Wenn die Überprüfung zu einem positiven Ergebnis kommt, muss diese Auswahlinformation zusammen mit den Eigenschaften der SA an die IPsec-SADB weitergeleitet werden. Die SA kann dabei nur für den Verkehr benutzt werden, der durch die Nutzdaten vom Typ ID identifiziert ist.

Der Quick-Mode wurde hier als einfacher Anfrage-/Antwort-Austausch behandelt. Er beinhaltet allerdings noch einiges mehr. Der Initiator benötigt einen Aktivitätsbeweis vom Responder, weshalb der Responder die Nonce des Initiators und die Nachricht-ID des Austauschs in die Nutzdaten der authentifizierenden Hash-Funktion einbezieht. Dieser Hash unterstützt damit nicht nur die Nachrichtenintegrität und Authentizität der Datenquelle, sondern schafft

45 Dieselbe wie in Phase I

46 Domain-of-Interpretation

ebenfalls einen Beweis für die aktuelle Aktivität. Der Responder benötigt ebenfalls diesen Beweis vom Initiator, den er durch das Hinzufügen einer dritten Nachricht bekommt. In dieser schließt der Initiator die beiden Noncen und die Nachricht-ID des Austauschs in die authentifizierenden Nutzdaten der Hash-Funktion mit ein. Zu diesem Zeitpunkt kann der Responder die SA zur SADB von IPsec hinzufügen.

Es gilt somit:

$$HASH1 = PRF\left(SKEYID\_a, M-ID \middle| SA \middle| N_i [\,[\,KE\,]\,[\,[\,IDci \middle| IDcr\,]\right)$$

$$HASH2 = PRF\left(SKEYID\_a, M-ID \middle| N_i \middle| SA \middle| N_r [\,[\,KE\,]\,[\,[\,IDci \middle| IDcr\,]\right)$$

$$HASH3 = PRF\left(SKEYID\_a, 0 \middle| M-ID \middle| N_i \middle| N_r\right)$$

Der Initiator besitzt bereits vor dem Responder alle nötigen Informationen, um die IPsec-SA der SADB hinzuzufügen. Nach Empfang der Nachricht des Responders ist ihm bekannt, welches Angebot dieser ausgewählt hat. Wenn der Austausch PFS[47] besitzt, dann kennt er ebenfalls den öffentlichen Diffie-Hellman-Wert des Responders und sieht, ob dieser aktiv ist. Der Responder seinerseits muss erst warten, bis er die letzte Nachricht vom Initiator erhalten hat, bevor er fertig ist. Daher ist es dem Initiator bereits möglich, geschützte Pakete durch IPsec zu senden, bevor der Responder seine letzte Nachricht empfangen hat. Der Responder wird dabei so lange alle IPsec-Pakete fallen lassen, bis er in der Lage ist, die letzte IKE-Nachricht zu verarbeiten.

Diese Verarbeitung könnte bei UDP-Paketen zu Schwierigkeiten führen, da hier keine sichere Übertragung durchgeführt wird. Um Pakete nicht zu verlieren, benutzt IKE das Commit-Bit im ISAKMP-Header, um den Quick-Mode um eine Nachricht zu erweitern. Jeder Teilnehmer ist in der Lage dieses Commit-Bit zu setzen. Wenn dies geschieht, muss der Responder eine letzte Nachricht senden, die die Nutzdaten für die authentifizierende Hash-Funktion enthält und der direkt Nutzdaten vom Typ Notify folgen. Diese enthalten die Nachricht, dass die Verbindung hergestellt wurde. An dieser Stelle fügt der Initiator die SA nicht eher zur SADB hinzu, bis er diese Verbindungsnachricht erhält. Dadurch kann der Responder zuerst seine SA hinzufügen und anschließend IPsec-Pakete annehmen. Da IKE-SA bidirektional arbeitet, kehren sich die Rollen der Phase II um, nicht aber die rollenspezifischen Informationen der Phase I. Die Cookies des Initiators und des Responders identifizieren in dieser Reihenfolge weiterhin die IKE-SA, die verwendet wird, um die SA der Phase II zu schützen. [DOHA00]

47 Perfect Forwarding Secrety

6.1.5 Fazit und Zusammenfassung

Jedes Sicherheitsschema, welches eine Authentifizierung beinhaltet, muss eine Public Key Infrastructure (PKI) berücksichtigen, damit die Sicherheitsplattform skalierbar bleiben kann. Beispielsweise können Preshared Keys mittels IKE verwendet werden, wodurch aber die Verteilungslast sehr schnell anwächst. Ähnlich ergeht es den Mechanismen für die sichere Verteilung von öffentlichen Schlüsseln, die mit Authentifizierungsmethoden von IKE benutzt werden. Sie würden ihrer eigenen Last erliegen. Zwar benötigt IPsec mit IKE eine PKI, aber es sind noch Problempunkte offen, die nicht in der Rahmenspezifikationen von IPsec enthalten sind. Diese Punkte werden durch die Arbeitsgruppe PKIX bearbeitet und gelöst. Im Rahmen der Arbeiten wurden eine Vielzahl von Spezifikationen bereits erarbeitet. Die PKIX-Gruppe hat dabei ein Architekturmodell geschaffen, welches eine PKI definiert. Zusätzlich fließt die Arbeit anderer Standardisierungsgremien in die Arbeiten der IETF-Gruppe ein. Es müssen beispielsweise umfangreiche Beschreibungen spezifizierter Protokolle auf die Belange des Internets heruntergebrochen werden. Hinzu kommt, dass zurücklaufende Erfahrungswerte in den Entwicklungsprozess wieder zurückfließen und für eine stetige Verbesserung sorgen. Trotzdem ist zu beachten, dass eine PKI keine Technologielösung darstellt, sondern eine individuelle Lösung für unterschiedliche Netze und Anforderungen beinhaltet. Sie stellt einen nicht zu unterschätzenden Investitionskostenfaktor dar, der pro Unternehmen abgewogen werden muss. Deshalb kann an dieser Stelle auch keine Patentlösung beschrieben werden.

Ziel ist es, eine sichere Kommunikationsplattform zu schaffen, die mittels der Internet-Technologie Informationen austauscht, dabei aber keine Informationen nach außen dringen lässt. Eine Möglichkeit ist der Aufsatz eines Virtual Private Networks (VPN) bzw. eines Extranets, die ein sicheres Overlay-Netz innerhalb des Internet darstellen. Extranets tauschen ihre Daten über das Internet aus und können als Erweiterung des unternehmensinternen Netzes angesehen werden. Somit entsteht eine unternehmensübergreifende Kommunikationsbasis, die den Informationsfluss zwischen einer Unternehmung und beispielsweise seinen Kunden und Lieferanten beschleunigt. Ein Extranet ist im Grunde nur ein Intranet, welches nach außen[48] hin operiert bzw. geöffnet ist.

Bei der Umsetzung eines VPN/Extranet wird jedes Datenpaket mit einem IP-Header versehen, der die eigene Adresse und die Zieladresse beinhaltet. Netzwerkprotokolle wie IP sind auf Ebene 3 in dem OSI-Schichtenmodell definiert. Sie benötigen deshalb noch eine Schicht 1 und 2, um Datenpakete bis zum nächsten Vermittlungsrechner (Router) übertragen zu können. Die ersten beiden Schichten können verbindungslos oder verbindungsorientiert arbeiten. Im letzteren Fall erfolgt die Einwahl beispielsweise über ISDN über das DFÜ-

48 Über die eigenen Unternehmensgrenzen hinweg

Netz zum Network Access Server (NAS) eines Providers. Nach Aufbau der ISDN-Verbindung über einen B-Kanal wird das Layer-2-Protokoll PPP[49] verhandelt. Anschließend kann der Rechner IP-Pakete zum Provider übermitteln. Während einer PPP-Session werden bestimmte Attribute wie die öffentliche IP-Adresse, DNS-Server und WINS-Server zwischen NAS und dem entfernten Rechner ausgetauscht. Die IP-Adresse behält der Remote Client nur so lange, wie die PPP-Verbindung steht. Eine Layer-2-PPP-Verbindung ist multiprotokollfähig, das heißt, es können neben IP auch andere Protokolle wie beispielsweise IPX und AppleTalk übermittelt werden. Des Weiteren beinhaltet eine PPP-Verhandlung sicherheitsspezifische Abläufe wie Authentifizierung mit Benutzername/Passwort (PAP[50]/CHAP[51]) und einfache Verschlüsselung[52] sowie weitere Merkmale wie Datenkompression)[53] und Rückruf[54].

Ein weiteres Beispielszenario für den Einsatz von PPP ist die Anbindung eines Local Area Network (LAN) an das Internet und der Zugriff von beliebigen Arbeitsplatzrechnern auf das zentrale Datennetz. Hierfür baut der in der Filiale installierte IP-Router für den Versand seiner IP-Pakete eine ISDN-Verbindung zum nächsten NAS auf, über die anschließend alle PPP-Verhandlungen erfolgen. Der Router erhält vom NAS eine öffentliche IP-Adresse[55]. Das Problem ist allerdings, dass aufgrund der Adressenknappheit von IPv4 die Rechner im LAN jeweils eine abweichende private IP-Adresse besitzen. Bei jedem von einem LAN-Rechner erzeugten Datenpaket wird vom Router die private Quell-IP-Adresse gegen die öffentliche IP-Adresse ausgetauscht. Das Ergebnis: Alle LAN-Rechner erscheinen im Internet ausschließlich mit der einen öffentlichen IP-Adresse. Dieser Vorgang wird als Network Address Translation (NAT) oder auch als Masquerading bezeichnet. Das Verfahren bietet allerdings auch gleichzeitig Schutz gegen unerwünschte Verbindungen vom Internet in das LAN.

Ein Remote-Access-VPN ist die Abbildung eines traditionellen Direkteinwahl-Remote-Access-Netzes über ein öffentliches IP-Netz. Unternehmen, die ein solches VPN planen, stellen an dieses die gleichen Anforderungen wie beim direkten Zugriff auf die Unternehmenszentrale. Im Wesentlichen geht es um die Erfüllung folgender Anforderungen:

- Starke und persönliche Authentisierung jedes einzelnen Benutzers
- Unterstützung verschiedener Authentisierungsmethoden wie PKI, X.509.v3-Zertifikate, Smartcards, User-ID/Passwort (RADIUS[56]), SecureID oder OTP (One Time Password)

49 Point-to-Point Protocol
50 Password Authentication Protocol
51 Challenge Handshake Authentication Protocol
52 Encryption Control Protocol (ECP)
53 Compression Control Protocol (CCP)
54 Link Control Protocol (LCP)
55 Nur diese Adressen sind einmalig weltweit vorhanden und werden im Internet vermittelt.

- Starke Verschlüsselung
- Sicherer Schlüsselaustausch
- Überschaubare Konfiguration
- Administration der VPN-Clients und der dazu gehörenden Security Policy
- keine Topologie-Einschränkungen
- Multiprotokollfähigkeit beim Einsatz unterschiedlicher Netzwerkprotokolle

Grundlage von Layer-2- und Layer-3-VPNs ist das Tunneling. Dieses Verfahren ist notwendig für den Transport der Daten zwischen einem zentrale VPN-Gateway und Remote VPN-Client über ein öffentliches, unsicheres Netz. Etabliert wird ein Tunnel dadurch, dass jedem erzeugten Datenpaket ein extra IP-Header und ein oder mehrere spezifische Header[57] vorangestellt werden. Der Tunnel beginnt da, wo der extra IP-Header hinzugefügt wird, und endet, wo der IP-Header wieder entfernt wird. Da Authentisierung und Verschlüsselung komplett innerhalb des Tunnels ablaufen, spielen die Tunnelendpunkte eine sehr wichtige Rolle.

Abhängig vom Tunnelendpunkt spricht man von Site-to-Site-VPN, End-to-Site-VPN und End-to-End-VPN. Ein Beispiel für Site-to-Site-VPN ist ein Tunnel zwischen einem Filial-Router und einem VPN-Gateway in der Zentrale. Bei einem End-to-Site-VPN wird entweder ein Tunnel zwischen einem Einzelplatzrechner und einem VPN-Gateway oder einem Client in einem Filialnetz und dem zentralen VPN-Gateway aufgebaut. Dabei ist es nicht erforderlich, dass der Filial-Router VPN-fähig ist. Das End-to-End-VPN funktioniert im Grunde genauso wie ein End-to-Site-VPN, nur dass das VPN-Gateway im Applikations-Server integriert ist. Dadurch ist die gesamte Strecke vom Client-PC bis zur zentralen Applikation gesichert. Unternehmen gehen zunehmend dazu über, die Tunnel nicht im Filial-Router enden zu lassen, sondern in den einzelnen LAN-Workstations. Die Vorteile sind, dass eine persönliche Authentisierung der Teilnehmer und eine Verschlüsselung der Unternehmensdaten im Filialnetz erfolgt. Am Remote-Access-Markt zeichnet sich daher zunehmend eine Entwicklung weg von Site-to-Site- hin zu End-to-Site-VPN-Lösungen ab.

Zu den bekanntesten Layer-2-Verfahren gehören L2F (Layer-2- Forwarding), L2TP (Layer-2-Tunneling-Protocol) und PPTP (Point-to-Point-Tunneling-Protocol). Ein Layer-2-Tunnel stellt einen virtuellen Pfad über jede IP-Plattform dar und lässt sich über jede IP-Struktur hinweg aufbauen. Dabei ist es unerheblich, ob die installierten Router zwischen Client und VPN-Gateway IP-NAT einsetzen. Ein Layer-2-Tunnel ist multiprotokollfähig, wodurch die Möglichkeit besteht, über ein öffentliches Netz, das nur IP vermittelt, mit der Unternehmenszentrale über beliebige Netzwerkprotokolle zu kommunizieren. Der

56 Remote Access Dial-In User Service
57 Je nach eingesetztem Tunneling-Verfahren

Tunnelaufbau bei einem Layer-2-VPN wird realisiert durch einen extra IP-Header, einen UDP- oder TCP-Header und einen zusätzlichen Header, der vom Tunnelverfahren abhängig ist. Der Overhead pro Paket ist dabei abhängig vom eingesetzten Verfahren und liegt bei ungefähr 40 Byte. Trotz guter Eigenschaften für den Bereich Remote Access sind die Nachteile, dass sie nur zum Teil Datenintegrität und eine relativ schwache Authentisierung (CHAP/PAP) gewährleisten. Weiterhin fehlen wesentliche Sicherheitsfunktionen wie Verschlüsselung, Unterstützung von digitalen Zertifikaten und ein leistungsfähiges Schlüsselmanagement.

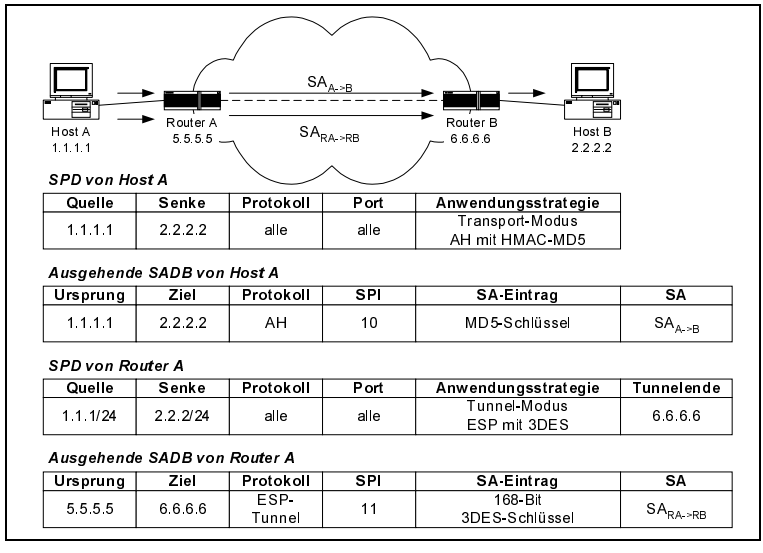
Eine Möglichkeit, Sicherheitsmechanismen und Layer-2-VPN-Tunneling zu vereinen, bietet sich mit der Erweiterung des PPP an. Man unterscheidet hier zwei Ansätze: Erweiterung der PPP Authentication Phase (PAP/CHAP) und Erweiterung des PPP Encryption Control Protocols (ECP). In beiden Fällen wird für die Erweiterung das SSLv3.0-Handshake-Protokoll eingesetzt. Dieses Verfahren ist in der Spezifikation RFC-2716 beschrieben. Es unterstützt die Anwendung von Zertifikaten, das heißt, starke Authentisierung und sicherer Schlüsselaustausch sind möglich, wodurch das VPN PKI-enabled ist. Nach Ablauf der SSL-Verhandlung wird sämtlicher Datenverkehr verschlüsselt, der über die PPP-Verbindung übertragen wird: die Netzwerkprotokolle IP/IPX und auch der letzte Teil der PPP-Verhandlung, in der die Zuweisung der privaten IP-Adresse, DNS- und/oder WINS-Server an den Client erfolgt. [ABSI99]

Neben der globalen Lösung einer PKI und eines Extranets sind aber auch noch andere Probleme der vorhandenen Protokollspezifikationen zu beachten. Während SSL mit der Anwendung S-HTTP bereits zum Standard heutiger Web-Server-Plattformen zählt, lassen sich kontinuierlich neue Schwächen in den verwendeten Verschlüsselungsalgorithmen bzw. unsichere oder schlechte Implementierungen finden. Aus diesem Grund müssen vorhandene Lösungen und Implementierungen auch immer hinterfragt werden. Neben der Sicherheit spielt dabei auch die Performance eine entscheidende Rolle, da die zusätzlichen Sicherheitsmechanismen möglichst unbemerkt greifen sollten. Die Performance wird in dem nächsten Kapitel noch eingehend betrachtet.

Das Rahmenwerk IPsec ist jedoch gegenüber SSL und S-HTTP sehr komplex, da es eine Möglichkeit beschreibt, wie ganze Subnetze gegenüber dem Internet abgesichert werden können. Außerdem müssen sich alle neuen Spezifikationen nach IPsec richten und gegebenenfalls Erweiterungen anbieten. Mittels des Transportmodus wäre es sogar möglich, das gesamte Internet abzusichern. Dies würde aber ein Upgrade sämtlicher aktiven Komponenten bzw. der Router beinhalten. Hinzu kommt, dass jeder ISP unterschiedliche Dienste anbietet und verschiedene Anforderungen gegenüber seinen Kunden zu erfüllen hat.

Layer-2-Tunneling mit den beschriebenen Sicherheitsmechanismen ermöglicht den Aufbau eines Remote Access VPN, das jede Infrastruktur unterstützt. Bereits vorhandene Netzwerkkomponenten können problemlos genutzt werden. Es ist unerheblich, ob ein Client den Tunnel über einen Provider, einen Branch-Office-Router, der sich zum Provider einwählt, einen NAS in der Zentrale oder direkt zum VPN-Gateway in der Firmenzentrale aufbaut. Der Remote-User erhält für die PPP-Session immer dieselben Attribute: IP-Adresse, Datenkompression, DNS-Adresse usw.

Ein Layer-3-VPN ist hingegen nicht multiprotokollfähig, sondern bezieht sich immer auf ein bestimmtes Netzwerkprotokoll, wie im Falle von IPsec auf IP. Mit den IPsec-Spezifikationen lässt sich ein VPN mit vorgegebener Security für das IP-Protokoll realisieren. Es steht ein komplettes Rahmenwerk zur Verfügung, das sowohl Layer-3-Tunneling als auch alle notwendigen Sicherheitsmechanismen wie starke Authentisierung, Schlüsselaustausch und Verschlüsselung umfasst. Dabei wird auf Herstellerunabhängigkeit großen Wert gelegt.



SPD von Host A

Quelle	Senke	Protokoll	Port	Anwendungsstrategie
1.1.1.1	2.2.2.2	alle	alle	Transport-Modus AH mit HMAC-MD5

Ausgehende SADB von Host A

Ursprung	Ziel	Protokoll	SPI	SA-Eintrag	SA
1.1.1.1	2.2.2.2	AH	10	MD5-Schlüssel	$SA_{A->B}$

SPD von Router A

Quelle	Senke	Protokoll	Port	Anwendungsstrategie	Tunnelende
1.1.1/24	2.2.2/24	alle	alle	Tunnel-Modus ESP mit 3DES	6.6.6.6

Ausgehende SADB von Router A

Ursprung	Ziel	Protokoll	SPI	SA-Eintrag	SA
5.5.5.5	6.6.6.6	ESP-Tunnel	11	168-Bit 3DES-Schlüssel	$SA_{RA->RB}$

Abb. 6.17 IPsec-Kommunikationsverarbeitung

Jede Kommunikationskomponente, die IPsec unterstützt, verfügt über ein IPsec-Modul. Mit Hilfe dieses Moduls wird jedes Datenpaket gegenüber einer Security Policy Database (SPD) entsprechend überprüft. Die SPD besteht aus Einträgen (SPD-Entries), in denen u.a. die Sicherheitsmerkmale beschrieben sind. Die SPD-Entries beinhalten einen zusätzlichen Filterteil, den Selektor. Der Selektor setzt sich aus IP-Adressen, UDP- und TCP-Ports sowie spezifischen IP-Header-Einträgen zusammen. Stimmt nun ein Datenpaket mit dem Selektor eines SPD-Eintrags überein, wird die weitere Vorgehensweise überprüft. Je nach Ergebnis wird das Paket entweder durchgelassen, verworfen oder es wer-

den bestimmte Sicherheitsmerkmale angewandt. Nach Übereinstimmung eines Datenpaketes mit einem SPD-Eintrag wird überprüft, ob bereits eine Security Association (SA) für diesen SPD-Eintrag existiert. Die SA bestimmt, ob das IPsec-Protokoll ESP oder AH verwendet werden soll. ESP unterstützt die Verschlüsselung und Authentisierung des IP-Pakets, AH ermöglicht nur die Authentisierung. Eine weitere Aufgabe der SA besteht darin, den Modus festzulegen: Tunnel Mode oder Transport Mode. Im Tunnel Mode wird der gesamte Rahmen verschlüsselt. Das Datenpaket bekommt einen neuen Header. Quelle und Ziel sind damit versteckt und nur die Tunnelendpunkte sind erkennbar. Im Transport Mode wird der Rahmeninhalt verschlüsselt, der ursprüngliche IP-Header wird beibehalten. Quell- und Zieladresse bleiben ungeschützt.

Darüber hinaus legt die SA den Algorithmus für die Authentisierung, die Verschlüsselungsmethode[58] und den verwendeten Schlüssel fest. Funktionsvoraussetzung ist, dass die Gegenstelle über dieselbe SA verfügt. Der Transportmodus ist für Rechner-zu-Rechner-Kommunikation geeignet, während der Tunnelmodus für den Betrieb über ein VPN-Gateway eingesetzt wird. Der IP-Header ermöglicht den Datentransport vom Client über das Internet zu einem Gateway. Das VPN-Gateway entfernt den IP-Header, entschlüsselt und sendet das Paket weiter zum Unternehmensnetz. Für Remote Access und End-to-Site-VPNs kommt nur der ESP-Tunnelmodus in Frage. Wenn eine IPsec-SA für einen bestimmten SPD-Eintrag nicht vorliegt, muss sie mit der Gegenstelle ausgehandelt werden. In einem solchen Fall kommt die zweite große Komponente einer IPsec-Implementation zum Einsatz: das IKE-Protokoll. Wie wichtig IKE in diesem Zusammenhang ist, verdeutlicht das Beispiel einer Internetverbindung zwischen Remote Client und dem VPN-Gateway in der Firmenzentrale. Der Client hat einen SPD-Eintrag mit einem Selektor, der ESP im Tunnelmodus für IP-Pakete an die Firmenzentrale vorgibt. Vom Client wird zunächst eine Layer-2-Verbindung mittels PPP zum Provider hergestellt. Anschließend erhält das IPsec-Modul im Client ein IP-Paket mit der Zieladresse „Firmenzentrale". Es existiert zwar ein SPD-Eintrag für dieses IP-Paket, jedoch keine SA. Nun stellt das IPsec-Modul an das IKE-Modul die Anforderung, eine IPsec-SA mit der Gegenstelle[59] in der Phase II auszuhandeln. Dabei bekommt das IKE-Modul auch die im SPD-Eintrag vorhandenen Sicherheitsmerkmale. Voraussetzung für den Ablauf dieses Prozesses ist eine Art Kontrollverbindung zwischen Client und VPN-Gateway. Diese Kontrollverbindung beinhaltet die Phase I und das Resultat die IKE-SA. Phase I übernimmt den gesamten Authentisierungsvorgang vom Client zum VPN-Gateway und erzeugt gleichzeitig eine verschlüsselte Kontrollverbindung, über die anschließend die Verhandlungen der Phase II für die IPsec-SA abläuft.

58 Nur bei ESP
59 Dem VPN-Gateway

Die Verhandlung der Phase I ist ein Handshake, das den Austausch von digitalen Zertifikaten erlaubt und einen Schlüsselaustausch für die Kontrollverbindung beinhaltet. Dieses Verfahren kann in zwei verschiedenen Hauptmodi erfolgen: dem Main Mode und dem Aggressive Mode. Im Main Mode erfolgt ein Austausch von insgesamt sechs Nachrichten, wobei übertragene IDs oder Zertifikate verschlüsselt werden. Im Aggressive Mode hingegen werden nur drei Nachrichten ausgetauscht und es wird keine Verschlüsselung eventuell übertragener IDs und Zertifikaten vorgenommen.

Unabhängig vom Verhandlungsmodus gibt es verschiedene bidirektionale Authentisierungsmethoden. Die zwei am häufigsten implementierten Methoden sind Preshared Keys bei denen Client und VPN-Gateway jeweils dieselben vorkonfigurierten Schlüssel besitzen und RSA-Signaturen, welche die Anwendung von digitalen Zertifikaten unterstützt. Während einer IKE-Verhandlung mit Preshared Key und Main Mode muss der Client vom VPN-Gateway anhand seiner IP-Adresse eindeutig identifizierbar sein. Denn der Preshared Key wird in die symmetrische Schlüsselberechnung mit einbezogen, das heißt, die Verschlüsselung beginnt bereits vor der Übertragung von Merkmalen, die den Client identifizieren. Erfolgt nun die Einwahl eines Client über einen Provider mit dynamischer Adressvergabe, bleibt seine IP-Adresse aufgrund der unklaren Adresszuweisung unerkannt. Eine Möglichkeit, dieses Problem zu umgehen, ist der so genannte Aggressive Mode. Der Nachteil bei dieser Methode: Die Übertragung der IDs und/oder digitalen Zertifikate erfolgt im Klartext. Eine andere Methode ist, dass alle Clients denselben Preshared Key erhalten. Diese Vorgehensweise ist allerdings mit einer Schwächung der Authentisierung verbunden.

Durch die Komplexität der Spezifikationen und durch die noch recht junge Standardisierung haben sich aber bis heute folgende Mängel herausgestellt:

1. Wenn Preshared Key, Main Mode und dynamische IP-Adressen benutzt werden, muss der Preshared Key für alle IPsec-Clients identisch sein.
2. IPsec unterstützt nicht die traditionell im Remote Access eingesetzten unidirektionalen Authentisierungsmethoden RADIUS (PAP/CHAP), SecureID oder OTP.
3. Die IP-, DNS- und WINS-Adresszuweisung vom VPN-Gateway zum Client ist innerhalb des IKE-Protokolls nicht spezifiziert. Die privaten IP-, DNS- und WINS-Adressen müssten danach in jedem IPsec-Client fest konfiguriert werden.
4. Bei großen Remote-Access-Netzen (>300 Teilnehmer) können Ressourceprobleme im VPN-Gateway auftreten. Die IPsec-Clients unterbrechen ihre Layer-2 PPP-Verbindung zum Provider immer wieder und löschen evtl. ihre eigenen SA, die nach wie vor im VPN-Gateway existieren.
5. Zwischen IPsec-Client und VPN-Gateway darf kein IP-NAT-Verfahren eingesetzt werden, da der IPsec-ESP-Header nicht über genügend Informa-

tionen verfügt. Setzt beispielsweise ein Filial-Router IP-NAT bei der Einwahl zum Provider ein, muss sich der Tunnelendpunkt im Router befinden. Das heißt, es findet nur eine Authentisierung des LAN statt, jedoch keine persönliche Authentisierung der Teilnehmer. Insbesondere beim Einsatz von digitalen Zertifikaten (PKI) muss der Tunnelendpunkt im Filial-PC liegen.

6. Bei großen Remote-Access-Installationen ist die Konfiguration und Administration der SPD-Einträge sowohl auf Client-Seite als auch im Zentralsystem sehr aufwendig.

Es gibt bereits verschiedene Ansätze, um diese Schwächen zu beseitigen und IPsec für Remote Access zu optimieren. Als Lösung für Punkt 1 und 2 existiert der von Cisco Systems eingereichte Draft XAUTH (Extended Authentication). Er setzt jedoch eine Veränderung und Erweiterung des IKE-Protokolls voraus. Für Punkt 3 sind derzeit keine Verbesserungsvorschläge bekannt. Um das in Punkt 4 dargestellte Problem zu lösen, gibt es einen Vorschlag, der eine Art Polling beschreibt. Mit Hilfe dieses Verfahrens soll signalisiert werden, wenn eine SA nicht mehr aktiv ist. Polling wird allerdings zu einem Problem, wenn der Client im Shorthold-Mode arbeitet, da die Layer-2 PPP-Verbindung zum Provider immer bestehen bleiben muss. Für Punkt 5 gibt es den neu eingereichten Draft NAT Traversal 28/2-2001 der Firma SSH. Hier geht es darum, dass jedes erzeugte IPsec-Paket zusätzlich in einen IP- und UDP-Header verpackt wird, um so eine Kommunikation über Geräte, die IP-NAT einsetzen, zu ermöglichen. Punkt 6 ist davon abhängig, wie ein Hersteller die Verwaltung der SPD implementiert hat.

Bei den genannten Lösungsansätzen handelt es um Drafts und Vorschläge, wobei nicht alle von der IPSRA und IPSEC Working Group akzeptiert sind. So sind beispielsweise Ansätze, die eine Erweiterung des IKE-Protokolls vorsehen, nicht akzeptabel. Der interessanteste Ansatz, um die in den Punkten 1 bis 5 aufgezeigten Probleme zu lösen, ist im Informational RFC-2888 IPSec-over-L2TP[60] beschrieben. Hier werden keinerlei Veränderungen oder Erweiterungen an der RFC-Reihe von IPsec vorgenommen, womit der Akzeptanz seitens der IETF nichts im Wege stehen dürfte. Funktionsseitig wird zunächst ein Layer-2-Tunnel zwischen Remote Client und VPN-Gateway aufgebaut. Anschließend läuft eine Standard-PPP-Verbindung[61], über die die IKE-Verhandlung und die IPsec-Datenpakete übertragen werden (Punkt 1-3). Trennt der Client seine Layer-2-Verbindung zum Provider, erfolgt automatisch der Tunnelabbau und die IPsec-SAs können gelöscht werden (Punkt 4). Mit Punkt 5 (IP-NAT) gibt es seitens eines L2TP-Tunnels wie beschrieben keine Probleme.

60 Secure Remote Access with L2TP

61 Ohne zusätzliche Sicherheit wie PPP-EAP-TLS

Die in der Spezifikation RFC-2888 beschriebenen Nachteile sind der Overhead[62], die Komplexität der Implementierung[63] sowie das unverschlüsselte Versenden der privaten IP-Adressen, da IPsec erst nach dem Tunnelaufbau und anschließender PPP-Verhandlung aktiv wird. Die praktische Erfahrung mit IPsec-over-L2TP zeigt jedoch, dass sich der Overhead entweder durch PPP- oder IP-Kompression auf IPsec-Ebene begrenzen lässt. Die Komplexität der Implementierung eines Herstellers hängt von dessen Kenntnissen über die Layer-2-Technologie ab. Außerdem ist das Problem bezüglich des unverschlüsselten Versands von privaten IP-Adressen letztlich abhängig vom Grad des Sicherheitsbedürfnisses des VPN-Betreibers. [SRIS00]

IPsec ist ein Standard mit ausgezeichneten Sicherheitsmechanismen, der in VPN-Szenarien sehr effizient und sicher funktioniert, in denen mit festen IP-Adressen gearbeitet wird (z.B. B2B[64], Extranet). In diesen Fällen lassen sich heute auch VPN-Gateways verschiedener Hersteller ohne Probleme einsetzen, wenn man einheitliche digitale Zertifikate verwendet. Im Bereich Remote Access weist IPsec allerdings noch Mängel auf, deren Kenntnis wichtig für Unternehmen sind, die ein Remote-Access-VPN aufbauen wollen. Ohne Zweifel hat IPsec für Remote Access seine Berechtigung, wenn es allen Herstellern gelingt, sich auf ein einheitliches Verfahren zu einigen (z.B. RFC-2888). Solange aber immer wieder neue unterschiedliche Drafts eingereicht werden, die teilweise sogar versuchen, das gleiche Problem zu lösen, wird es sehr schwer, Interoperabilität zu schaffen bzw. zu gewährleisten. Das Problem der dynamischen Adressvergabe wird sich aber nach Einführung von IPv6 von selbst lösen. Die Einführung verzögert sich allerdings noch, sodass zuerst nur kleinere IPv6-Inseln davon profitieren werden.

6.2 Quality-of-Service (QoS)

Nachdem die Sicherheitsplattform mittels IPsec geschaffen wurde, wird in einem nächsten Schritt der notwendige Quality-of-Service (QoS) aufgesetzt, um Echtzeitanwendungen zu unterstützen. Dabei ist die reine Betrachtung der Realisierung von QoS auf der Schicht 3 (Netzwerkschicht) bei der Umsetzung eines QoS-fähigen Netzes nicht ausreichend. Hierfür sind zwei Gründe ausschlaggebend:

- Es werden immer öfter LANs auf der Schicht 2 vermittelt bzw. geswitcht. In diesen Bereichen kommen keine Router zum Einsatz. Dabei werden in den Switches lediglich die L2-Header der Datenpakete ausgewertet. Die IntServ bzw. DiffServ basieren hingegen auf dem Austausch und der Auswertung

62 L2TP- und IPsec-Header
63 Layer-2-Tunneling und IPsec
64 Business-to-Business

der Layer-3-Header und des Payloads der IP-Pakete. Somit müssen zusätzliche Verfahren entwickelt werden, die eine QoS-Realisierung in reinen Layer-2-Netzen ermöglichen.

- Das IP-Protokoll wird über unterschiedliche Protokolle der Schicht 2 übertragen. Dabei legt das verwendete Layer-2-Protokoll oft wesentliche Einschränkungen bei der QoS-Realisierung fest. Wird z.B. das IP-Protokoll in einem Shared Media IEEE-802.3-Netz verwendet, so kann im Netz kein IntServ eingesetzt werden. Lässt das eingesetzte Schicht-2-Protokoll eine Einhaltung der QoS-Parameter zu, so muss der gewählte IntServ- bzw. DiffServ-Dienst so auf das Schicht-2-Protokoll abgebildet werden, dass die gesamte Dienstgüte von einem Teilnehmer bis zum anderen sichergestellt werden kann.

Es müssen also Mechanismen gefunden werden, die eine Anpassung bzw. ein Mapping zwischen den unterschiedlichen Verfahren ermöglichen, damit alle Schichten gleichermaßen adressiert werden können. Dabei ist zu beachten, dass der Best-effort-Verkehr nicht durch den Einsatz von QoS-Verfahren verdrängt wird. Bei der Betrachtung von QoS-Implementierungen ist zu beachten, dass die Abbildungsmechanismen auf Schicht 2 zur Zeit eines der schwächsten Glieder der Kette bei der QoS-Entwicklung darstellen. Einige Ansätze befinden sich allerdings noch im Draft-Stadium, sind im praktischen Betrieb noch nicht getestet und zum Teil auch nur als Architekturmodell spezifiziert worden. Deswegen geht es in diesem Abschnitt eher um die Problematik bei den Abbildungsmechanismen und die grundsätzliche Vorgehensweise bei der Realisierung und als um Einzelheiten der Protokolle.

6.2.1 Subnet Bandwidth Manager (SBM)

Zum Erkennen von Ressourcen in Layer-2-LANs ist von der IETF ein Protokoll zur RSVP[65]-basierten Admission Control entwickelt worden. Dieses Protokoll nach RFC-2814 basiert auf RSVP und wird in Layer-2-Netzen eingesetzt. Dafür sind zusätzlich einige Layer-2-spezifische Objekte für RSVP definiert worden. Es wird dabei vorausgesetzt, dass ein Anfangs- und Endpunkt für Datenströme in LAN-Hosts oder Layer-3-Routern vorhanden sind. Somit wird der Anpassungsbedarf des Protokollstacks in den Layer-3-Einheiten minimal. Das Management der Layer-2-Geräte übernehmen Agenten, die als Subnet Bandwidth Manager (SBM) bezeichnet werden.

Die Funktionsweise ist dabei wie folgt: Innerhalb einer Layer-2-Domäne befinden sich mehrere SBMs. Dabei hat jedes physikalische Segment höchstens einen Designated Subnet Bandwidth Manager (DSBM). Der DSBM steuert die Ressourcenanforderungen für das zuständige Segment. Die restlichen an dieses Segment angeschlossene SBMs können beim Ausfall des DSBM seine Funk-

65 Resource Reservation Protocol

tionen übernehmen. Des Weiteren agieren die SBMs als normale Layer-2-Switches. Ein DSBM kann gleichzeitig mehrere Layer-2-Segmente verwalten. Ein Segment des Layer-2-Netzes, dessen Ressourcen von einem DSBM verwaltet werden, wird als Managed Segment bezeichnet. Eine Netzeinheit, die die Daten über ein Managed Segment sendet und den Dienst eines DSBM nutzt, wird DSBM-Client genannt.

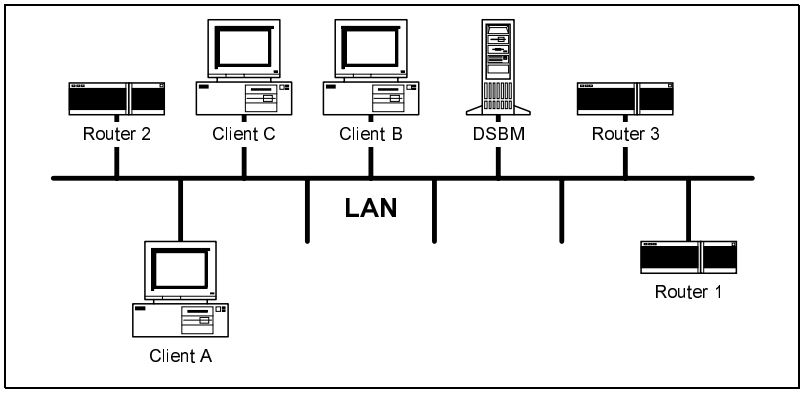


Abb. 6.18
Managed Segment eines SBM-Netzwerks

Das Protokoll durchläuft folgende Phasen: Zuerst wird der DSBM initialisiert. Das heißt, der DSBM werden die notwendigen Informationen, unter anderem die verfügbare Bandbreite und Verzögerung auf dem Managed Segment, mitgeteilt. Diese Informationen können statisch vom Systemadministrator eingetragen werden oder dynamisch ermittelt werden. Anschließend wird der DSBM-Clients initialisiert. Hierbei wird für jede aktive Ausgangsschnittstelle der zuständige DSBM eingetragen. Dafür laufen Prozeduren zur Bestimmung des DSBM ab, was ähnlich dem Spanning-Tree-Algorithmus des Standards IEEE802.1D funktioniert. [IEEE802.1D98]

Die DSBM-basierte Admission Control wird anschließend eingesetzt, was Abb. 6.18 verdeutlicht. Dabei sei Client *A* der RSVP-Empfänger, der RSVP-Sender liegt hingegen außerhalb der Layer-2-Domäne. Die Daten vom RSVP-Sender erreichen die Layer-2-Domäne über den Router *1*. Zur Ressourcenanforderung auf einem Managed Segment muss ein DSBM-Client folgende Schritte vornehmen:

1. Wird eine PATH-Nachricht auf ein Managed Segment gesendet, so wird diese von Router *1* an den DSBM und nicht an den Client *A* gesendet[66]. Der DSBM speichert den PATH-Status sowie die Layer-2- und Layer-3-Adresse des PHOP und leitet die Nachricht weiter zum Ziel. Dabei wird die ADSPEC im DSBM angepasst und als PHOP-Adresse in der PATH-Nachricht die eigene Layer-2-/Layer-3-Adresse eingetragen.

66 Wie das bei dem Standard-RSVP-Ablauf stattfinden würde.

2. Möchte Client *A* eine Reservierung vornehmen, so sendet er eine RESV-Message anhand der RSVP-Verarbeitungsregeln an die im PATH-State-Eintrag gespeicherte PHOP-Adresse[67].
3. Der DSBM prüft die Verfügbarkeit der Ressourcen auf dem Managed Segment und generiert entsprechend eine RESV- oder RESV-ERROR-Nachricht. Diese wird wiederum an die Layer-2- bzw. Layer-3-Adresse des PHOP gesendet[68].
4. Liegen zwischen dem Ziel-Client A und dem Router *1* mehrere Managed Segmente, so wird die PATH-Nachricht Hop-by-hop über die zuständigen DSBM durchgereicht. Somit spielen die DSBM hier die Rolle eines RSVP-Knotens im normalen RSVP-Prozess. Es wird dabei die Admission-Control-Entscheidung aufgrund der Verfügbarkeit der Ressourcen auf dem Managed Segment getroffen.

Zur Realisierung dieses Protokolls müssen allerdings Erweiterungen von RSVP in Kauf genommen werden. Wird eine Netzschnittstelle eines RSVP-Knotens mit einem Layer-2-Netz verbunden, so muss die PATH-Nachricht nicht an die RSVP-Zieladresse, sondern an den DSBM gesendet werden. Hinzu kommt, dass die Layer-2-Einheiten keine Layer-3-Forwarding-Informationen verarbeiten können, weshalb innerhalb der Layer-2-Domäne die Next-Hops sowie Previous-Hops mit ihrer Layer-2-Adresse adressiert werden müssen. Dafür muss ein neues Objekt, das als LAN_NHOP bezeichnet wird, im RSVP eingeführt werden. Weiterhin ist zu beachten, dass auf der Schicht 2 die Datenströme auf Aggregatbasis verarbeitet werden. Dabei haben unterschiedliche Aggregate unterschiedliche Prioritäten.

Zur Realisierung eines IntServ-Dienstes Ende-zu-Ende kann in Layer-2-Netzen die Priorisierung der Layer-2-Datenpakete nach dem IEEE 802.1D vorgenommen werden. Dafür müssen Mapping-Tabellen definiert werden, anhand derer einem bestimmten IntServ-Dienst eine Priorität im IEEE 802.1D-Netz zugeordnet wird. Dabei kann die vom DSBM-Client gewählte Priorität aufgrund der Admission-Control-Abfrage des SBM bestätigt oder geändert werden. Damit diese Informationen mit Hilfe des RSVP übertragen werden können, muss das RSVP um ein TCLASS-Objekt erweitert werden.

Zur Realisierung des SBM-Protokolls werden zwei Multicast-Adressen aus dem lokalen Bereich reserviert. Sie werden entsprechend als DSBM_Logical_Address und ALLSBM_Address bezeichnet. Mit der ersten Adresse wird immer der DSBM eines Managed Segment adressiert, mit der zweiten alle SBMs gleichzeitig. Damit müssen die DSBM-Clients nicht die Unicast-Ethernet-Adresse des DSBM vorerst abfragen bzw. aus einer IP-Adresse mit Hilfe des ARP[69] extrahieren.

67 Also an die Layer-2- bzw. Layer-3-Adresse des DSBM
68 Hier an die Layer-2/Layer-3-Adresse des Router 1
69 Address Resolution Protocol

Vielmehr werden die PATH-Nachrichten auf einer Layer-2-Schnittstelle an die DSBM_Logical_Address gesendet. Der DSBM hört das Medium auf Nachrichten mit dieser Adresse ab und verarbeitet diese anhand der beschriebenen Regeln. Die ALLSBM_Address wird zum Management der Layer-2-Domäne eingesetzt. Alle SBMs empfangen Nachrichten mit dieser Ethernet-Adresse. An diese Adresse sendet z.B. der DSBM die Nachricht I_AM_DSBM aus. Fällt über längere Zeit diese Nachricht aus, so wird eine Managementprozedur zur Bestimmung eines neuen DSBM gestartet. [YHBB+00]

6.2.2 IntServ-Ansatz in IEEE-802-Netzen

IEEE802-Netztechnologien besitzen die größte Verbreitung im LAN und halten auch vermehrt Einzug ins WAN. Das grundsätzliche Problem bei der Realisierung bezüglich QoS-Mechanismen ist allerdings, dass in den LANs keine Behandlung einzelner IP-Datenströme vorgenommen werden kann. Es ist oft problematisch, die einzelnen Charakterisierungsparameter einer Verbindung festzustellen, da beispielsweise der Guaranteed Service feste Parameter verlangt. Bei der Abbildung der IntServ auf ein LAN geht man prinzipiell genauso wie bei der Interoperation zwischen IntServ und DiffServ vor.

Zuerst wird eine Standard-Abbildungsvorschrift eines IntServ-Dienstes auf eine Layer-2-Priorität definiert. Dabei muss im Netz mindestens ein Bandwidth Allocator (BA) bzw. Subnet Bandwidth Manager (SBM) vorhanden sein, der auf der einen Seite die Funktionalität Admission Control für das LAN zur Verfügung stellt und auf der anderen Seite die Charakterisierungsparameter[70] des LAN an die Layer-3-Einheiten exportiert. In einigen IEEE 802-Netzen, wie beispielsweise Token Ring, FDDI etc., ist der Zugriff auf das Medium deterministisch, wodurch Grenzwerte einzelner QoS-Parameter im LAN angegeben werden können. Bei Ethernet ist hingegen der Zugriffsmechanismus stochastisch, dabei kann keine obere Grenze der Verzögerung der Datenpakete angegeben werden. Die Ausnahme bildet lediglich der Fall, dass eine Verbindung von einer einzigen Endstation benutzt wird. Hierfür ist die Paketverzögerung ungefähr gleich der Paketierungsverzögerung, also 1,2 ms bei Ethernet, 120 µs bei Fast-Ethernet und 12 µs bei Gigabit-Ethernet. Der erweiterte IEEE802.1D-Standard von 1998 legt das Ethernet-Switching fest. Dabei wird ein Teil des Frame Delimiter[71] zur Festlegung der Priorität des Datenpakets benutzt. Jeder zu IEEE802.1D konforme Switch muss mindestens zwei Prioritätsklassen mit Trennung der Puffer für jede Klasse realisieren. Dabei kommt im einfachsten Fall eine strikte Priorisierung[72] des höherwertigen Verkehrs zum Einsatz. 1998 wurde die ursprüngliche Bezeichnung IEEE802.1p als Traffic Class Expediting

70 z.B. die C-/D-Parameter der AD_{SPEC}, falls der Guaranteed Service zum Einsatz kommt

71 Erweiterung des Ethernet-Rahmens um ein Frame-Tag, welches die Prioritäts- und VLAN-Angaben enthält

dem IEEE802.1D-Standard hinzugefügt. Dabei wird eine Methode beschrieben, die erläutert, wie man Pakete auf Schicht 2 anhand eines Tags identifiziert, klassifiziert und priorisiert, wie Abb. 6.19 zeigt. Der Tag wird dabei durch einen Client, Switch oder Router in den Rahmen eingefügt.

Tab. 6.3 Standard-Service-Mapping

Priorität	Service
0	Default: Pakete werden als Best-effort behandelt
1	Reserved: weniger als die Dienstgüte Best-effort
2	Reserved: unbenutzt
3	Reserved: unbenutzt
4	Delay Sensitive: Anwendungen ohne feste Verzögerungszusage
5	Delay Sensitive: 100 ms Verzögerung
6	Delay Sensitive: 10 ms Verzögerung
7	Network Control: Netzwerkmanagement-Daten

Optional können in einem IEEE802.1D-Switch bis zu 8 unterschiedliche Prioritäten mit einer erweiterten Queue[73]-Verwaltung wie beispielsweise Weighted Fair Queueing (WFQ) realisiert werden. Dabei ist zu beachten, dass sich die Priorisierung der Datenpakete laut IEEE802.1D nur auf die Reihenfolge der Verarbeitung der Pakete in den Switches bezieht. Kommen in einem IEEE802.3-LAN Switches zum Einsatz, wobei an einem Port immer maximal eine Endstation über eine Full-Duplex-Verbindung angeschlossen ist, so können für solche Konfigurationen unter bestimmten Voraussetzungen die für die IntServ notwendigen Charakterisierungsparameter bestimmt werden. Die wichtigsten Bedingungen sind dabei, dass die Summe aller Datenströme auf den Eingangsports des Switches dessen Verarbeitungskapazität nicht überschreiten und dass die Summe der Datenströme auf einem beliebigen Port nicht über die vorhandene Bandbreite der Verbindung hinausgeht. Um dieses sicherzustellen, ist wiederum der Einsatz einer Instanz im LAN notwendig, die die Funktionalität Admission Control übernimmt. Diese Aufgaben kann ein SBM übernehmen. Trotz der Beschränkungen hat die geswitchte Ethernet-Topologie eine weite Verbreitung bekommen und kann kostengünstig realisiert werden. Das Problem stellen Bandwidth Allocator bzw. SBM dar. Es sind hier zurzeit noch nicht alle Spezifikationen entwickelt worden. [IEEE802.1D98]

72 Eine strikte Priorisierung bedeutet an dieser Stelle, dass keine Datenpakete mit geringerer Priorität verarbeitet werden, solange Daten mit höherer Priorität zur Verarbeitung bereitstehen.

73 Warteschlange

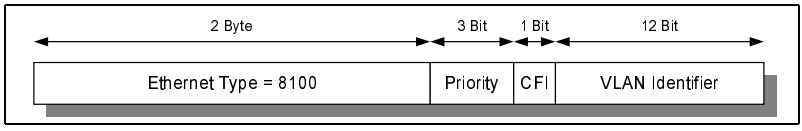


Abb. 6.19 IEEE802.1D-Tag

Es wird in RFC-2815 ein Default Mapping spezifiziert. Dabei werden die QoS-Anforderungen in einige wenige Klassen eingeteilt. Jeder Klasse wird eine IEEE802.1D-Priorität zugeordnet. Sind ausreichend Ressourcen im Netz verfügbar, so wird für den gewünschten Dienst diese Priorität verwendet. Andernfalls kann der SBM für die Anforderung eine andere Priorität[74] bestimmen. Tab. 6.3 zeigt die Zuordnung unterschiedlicher Anwendungsanforderungen der IEEE802.1D-Prioritäten. Die Zeitangaben gelten für jeweils einen Managed Segment. Die Prioritäten 1 und 2 bedeuten laut IEEE 802.1D eine geringere Priorität als die Standard-Priorität. Diese wird für eine Dienstgüte, die geringer als Best-effort ist, verwendet und soll für Datenpakete angewandt werden, die die angemeldete T_{SPEC} nicht einhalten. Obwohl eine solche Maßnahme den Verkehr Best-effort tatsächlich vor Quellen schützen kann, die die vereinbarten Traffic Profiles nicht einhalten, ist dessen Verwendung innerhalb des IETF umstritten, da einzelne Pakete für ihre Priorität markiert werden, sodass es zu Paketüberholungen im Netz kommen kann. Das führt dann zu dem unerwünschten Effekt der Paketreplikation. Eine komplette Degradierung des Datenstromes zu einem Dienst mit sehr geringer Dienstgüte birgt aber auch einige Gefahren, da die Reservierungen in der Regel für zeitkritische Anwendungen vorgenommen werden sollen. Werden im Extremfall wegen eines zu früh ausgesendeten Datenpakets alle nachfolgenden Pakete zum Dienst mit sehr geringer Dienstgüte degradiert, so ist es meistens sinnvoller, die Anwendung abzubrechen, als die Übertragung auf diese Weise fortzusetzen. Deswegen entscheidet man sich oft dafür, nur die einzelnen Pakete, die nicht konform zur T_{SPEC} ankommen, zu verwerfen und die Dienstgüte für die nachfolgenden Pakete unverändert zu lassen. Werden in einem Switch weniger als acht getrennte Queues für unterschiedliche Dienstklassen realisiert, so werden mehrere Prioritäten zu einer Traffic Class gemäß IEEE802.1D zusammengefasst und in einer Warteschlange verwaltet. Tab. 6.5 zeigt, wie die unterschiedlichen Prioritäten bei unterschiedlicher Anzahl von Queues zusammengefasst werden. Dabei stehen die Abkürzungen für folgende Verkehrsarten in Tab. 6.4. [SSCW00]

74 In aller Regel eine geringere

Tab. 6.4 Zusammenfassung unterschiedlicher Prioritätsklassen [IEEE802.1D98]

Unterschiedliche Verkehrsarten nach IEEE 802.1D [IEEE802.1D98]Priorität	Akronym	Verkehrsart
0	BE	Best-effort: Default
1	BK	Background: z.B. Backup
2	-	Reserved: unbenutzt
3	EE	Excellent Effort: geschäftskritische Anwendungen
4	CL	Controlled Load Applications: z.B. Streaming Multimedia
5	VI	Video: < 100 ms Verzögerung und Jitter
6	VO	Voice: < 10 ms Verzögerung und Jitter
7	NC	Network Control: Netzwerkmanagement

Anzahl der Queues im Switch	Verkehrsklassen							
1	BE							
2	BE				VO			
3	BE				CL		VO	
4	BK		BE		CL		VO	
5	BK		BE		CL	VI	VO	
6	BK		BE	EE	CL	VI	VO	
7	BK		BE	EE	CL	VI	VO	NC
8	BK	-	BE	EE	CL	VI	VO	NC

6.2.3 IntServ über ATM

Die Abbildungsmechanismen des IntServ-Ansatzes auf ATM-Netze sind bereits sehr ausführlich spezifiziert. Seit August 1998 liegen einige Arbeiten dazu als RFC-Spezifikationen vor:

1. **RFC-2379:** RSVP over ATM Implementation Guidelines
2. **RFC-2380:** RSVP over ATM Implementation Requirements
3. **RFC-2381:** Interoperation of Controlled-Load Service and Guaranteed Service with ATM
4. **RFC-2382:** A Framework for Integrated Services and RSVP over ATM

Die ATM-Technologie ist dabei von Anfang an unter Berücksichtigung der QoS-Anforderungen spezifiziert und realisiert worden. Deshalb ist die Abbildung auf ATM in vielen Punkten wesentlich einfacher realisierbar als in IEEE802-Netzen. Da die ATM-Spezifikationen zwar immer weiter entwickelt werden, können sich im Interworking auch Definitionen ändern. Inkompatibilitäten werden aber dadurch vermieden, dass seit einigen Jahren Basisfunktionen festgeschrieben worden sind, die sich nicht mehr grundlegend ändern dürfen. Wenn bei ATM Verbesserungen vorgenommen werden, so sind diese rückwärtskompatibel einzubringen. Die Einzelheiten des Mapping ist allerdings von der UNI[75]-Schnittstelle stark abhängig. Die Wahl der einzelnen Parameter beim Mappen eines IntServ-Dienstes unter Verwendung unterschiedlicher UNI-Schnittstellen wird in der Spezifikation RFC-2381 detailliert beschrieben. [GABO98]

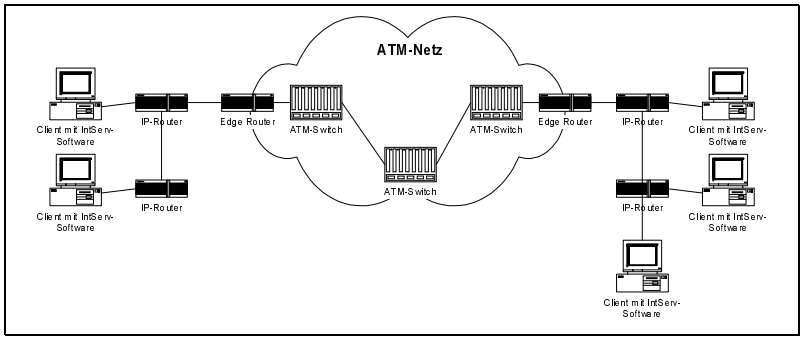


Abb. 6.20 Zusammenspiel von ATM und IntServ

Abb. 6.20 zeigt das Strukturbild für die Zusammenschaltung von IP-Netzen mit Hilfe von ATM-Strecken. Folgende Eigenschaften von ATM werden dabei beim Mappen benutzt:

- **Parameterverhandlung:** Beim Aufbau eines ATM-VC können einzelne Parameter der Verbindung zwischen Teilnehmer und Netz ausgehandelt werden.
- **Datenprofil:** Wird das ausgehandelte Datenprofil nicht eingehalten, so müssen die Datenpakete nicht verworfen werden, sondern das CLP[76]-Bit wird auf eins gesetzt. Ist das Netz überlastet, so werden Zellen mit dem gesetzten CLP-Bit vorrangig verworfen. Somit tritt das Problem der Paketüberholung bei Überschreitung der in der *TSPEC* angemeldeten Flussparameter nicht auf.

75 User Network Interface
76 Cell Loss Priority

- **Multicast:** In ATM können Punkt-zu-Mehrpunkt-Verbindungen aufgebaut werden. Solche Verbindungen werden bei Multicast-RSVP-Sitzungen aufgebaut.
- **QoS-Parameter:** Der ATM-Nutzer kann die einzelnen QoS-Parameter für die ATM-Verbindung über die UNI abfragen. Aus diesen Parameter können danach die einzelnen GCP[77] der *ADSPEC* abgeleitet werden.

Zur Realisierung von IntServ müssen die Edge-Router auf der IP-Netzseite RSVP-Fähigkeit besitzen, während auf der ATM-Netzseite die Netzwerkschnittstellen die ATM-Funktionalität entsprechend des realisierten UNI besitzen. Innerhalb des Knotens müssen Interworking Functions (IWF) zur Umsetzung der Anforderungen realisiert werden. [BERG98a] [BERG98b]

Das ATM-Netz wird aus Sicht des RSVP-Prozesses im Edge-Router als eine einzige Verbindung mit entsprechendem GCP betrachtet. Will ein Sender einen IntServ-Datenstrom anmelden, so sendet er eine PATH-Nachricht. Diese läuft zum Empfänger über das ATM-Netz genauso wie über ein DiffServ-Netz oder ein nicht RSVP-fähiges Teilnetz. Im Edge-Router muss die IWF-Einheit die spezifischen ATM-Größen für Parameter wie Verzögerung für die ATM-Strecke in die GSP-Parameter in Abhängigkeit vom verwendeten Mapping-Mechanismus umwandeln. Anschließend wird die aktualisierte AD_{SPEC}-Struktur an den Austritts-Router weitergeleitet. Sendet der Empfänger eine RESV-Mitteilung, so wird hop-by-hop eine Reservierung vorgenommen. Jetzt muss am Eingangs-Router die IWF-Einheit eine Reservierung durch das ATM-Netz vornehmen.

Folgende Probleme treten trotzdem bei der Abbildung einer IntServ-/RSVP-Sitzung auf ATM-Verbindungen grundsätzlich auf:

1. RSVP sieht die Möglichkeit vor, während einer Sitzung einzelne Parameter der Reservierung zu ändern, zusätzliche Ressourcen zu belegen oder Ressourcen freizugeben. Die gegenwärtige UNI-Spezifikation sieht solche Möglichkeit dagegen nicht vor. Ändert ein Empfänger seine Anforderung, so muss im Allgemeinem ein neuer VC[78] geschaltet werden.
2. Bei einer Multicast-RSVP-Sitzung sind die Reservierungen aller Multicast-Teilnehmer voneinander entkoppelt. Jeder Empfänger kann eine andere Reservierungsanforderung für denselben Datenstrom anfordern. Diese Eigenschaft ist in der Praxis wichtig, da unterschiedliche Teilnehmer einer Multicast-Sitzung über unterschiedliche Systeme und Technologien an das Internet angeschlossen sein können. Eine ATM-Punkt-zu-Mehrpunkt-Verbindung besitzt allerdings die gleichen Parameter auf jedem Zweig des Baumes.

77 Generalized Characterisation Parameter
78 Virtual Channel

3. Der Aufbau einer ATM-Verbindung ist mit einem relativ hohen Managementaufwand verbunden. Deshalb ist man bestrebt, möglichst wenig Änderungen der bestehenden ATM-Verbindungen vorzunehmen.
4. Werden durch das Netz PVC[79]-Verbindungen geschaltet, so ist oft die Summe der zugesagten Bandbreiten in den PVCs größer als die Gesamtbandbreite der Verbindung[80]. Werden nun IntServ-Datenströme über solche PVCs geschaltet, so kann ein IntServ-Dienst nicht gewährleistet werden. In einem solchen Fall hat der Netzadministrator dafür Sorge zu tragen, dass keine Überbelegung der Ressourcen stattfindet.

Unter Berücksichtigung der beschriebenen Fähigkeiten und Einschränkungen eines ATM-Netzes lassen sich die Anforderung wie folgt auf ATM übertragen. Wenn eine RSVP-Reservierung in einem Edge-Router ankommt, so wird ein SVC[81] für diese Anforderung geschaltet. Das Segmentieren wird in der Regel über den AAL-5-Typ realisiert. Dafür kann CLIP, LANE, MPOA oder ein anderes Protokoll verwendet werden. Die allgemeine Empfehlung für die Verwendung eines Dienstes in Abhängigkeit vom angeforderten IntServ-Dienst ist in Tab. 6.6 aufgeführt.

Integrated Services	ATM-QoS-Klassen
Guaranteed Quality-of-Service	CBR oder rt-VBR
Controlled Load Network Element Service	nrt-VBR oder ABR
Best-effort	UBR oder ABR

Tab. 6.5 Mapping von ATM-Diensten zu IntServ-Diensten

Es können allerdings auch andere Dienste für einen IntServ-Service gewählt werden. Die einzelnen Parameter der *FLOWSPEC* werden in Abhängigkeit vom gewählten Dienst in einzelne von der UNI unterstütze Parameter umgewandelt. Ausführliche Tabellen dafür sind in RFC-2382 definiert worden. Wird die Anforderung geändert, so steuert die IWF-Einheit den Aufbau eines neuen SVC an. Nur wenn ein neuer VC aufgebaut ist, wird der alte abgebaut. Scheitert hingegen der SVC-Aufbau, so sendet die IWF-Einheit bei Bedarf eine RESV-ERROR-Nachricht. Für alle Best-effort-Datenströme wird in der Regel ein einziger SVC bzw. PVC geschaltet. Hierbei wird deutlich, dass diese Konfiguration, bei der für jede RSVP-Anforderung ein SVC aufgebaut ist, mit einem extrem hohen Signalisierungsaufwand verbunden ist. Dies kann zu Überlastungen des Netzes bzw. der ATM-Switches führen. Aus diesem Grund kann man eine Alternative wählen. Dabei werden mehrere SVC- bzw. PVC-Verbindungen mit

79 Permanent Virtual Circuit
80 Ausnutzung des Bündelgewinns
81 Switched Virtual Circuit

unterschiedlicher Dienstgüte durch das ATM-Netz geschaltet. Es wird also jeder dieser VCs als ein Tunnel betrachtet. Kommt eine RSVP-Anforderung, so entscheidet die IWF-Einheit, in welchen Tunnel diese Anforderung geleitet wird, bzw. falls alle Ressourcen belegt werden, ob ein neuer Tunnel aufgebaut werden soll. Außerdem ist bei diesem Mapping-Modell eine Änderung der Anforderung nicht zwangsläufig mit einer Verzögerung zum Verbindungsaufbau verbunden. Dafür ist das Management der VCs wesentlich komplizierter, denn es muss eine prädikative Schätzung des ankommenden Datenverkehrs durchgeführt werden, damit VCs frühzeitig geschaltet bzw. nicht belegte Ressourcen nach einer bestimmten Zeit wieder freigegeben werden.

Auf ähnliche Weise werden heterogene Empfänger innerhalb einer RSVP-Sitzung behandelt. Dafür werden zwei Multicast-Bäume gleichzeitig aufgebaut. Jeder Multicast-Baum bietet eine bestimmte Dienstgüte. Es können so beispielsweise Datenströme auf einem Baum mit Best-effort und auf dem anderen mit GQOS mit einer vordefinierten *TSPEC* verteilt werden. Weitere Freiheitsgrade hat man bei der Lenkung der Signalisierungsdaten. Dabei sind folgende Realisierungen möglich:

1. Für jede RSVP-Anforderung wird zusätzlich zum Datenkanal ein zusätzlicher SVC zu Signalisierungszwecken durchgeschaltet. Diese Lösung ist aber zu kostspielig und kann zu Netzwerküberlastungen führen, wenn die SVC-Kapazität der Switches überschritten wird.
2. RSVP-Signalisierungsdaten werden zusammen mit den Best-effort-Daten gesendet. Da RSVP für eine unsichere Übertragung in IP-Netzen konzipiert worden ist, bereiten geringe Verluste von RSVP-Paketen keine großen Probleme für den Service. Kommt es aber im Netz zur Verstopfung, so können RSVP-Daten permanent nicht mehr zum Ziel kommen, wodurch auch die SVCs für reservierte Datenströme wegen Ablauf der Timer abgebaut werden.
3. Es kann ein zusätzlicher Punkt-zu-Punkt- oder Punkt-zu-Mehrpunkt-SVC alleine für Signalisierungsdaten geschaltet werden. Dabei tritt das Problem auf, dass bei jeder Änderung der Topologie des ATM-Netzes eine neue Verbindung für die Signalisierungsdaten aufgebaut werden muss. Die Dienstgüte für diesen Kanal kann ähnlich dem Controlled Load Network Element Service sein.

Durch die erwähnten Schwachstellen erscheint die dritte Alternative bei der IntServ-Realisierung über ATM-Netze die beste Wahl zu sein. Dabei müssten die einzelnen Managementfunktionen der IWF-Einheit detailliert spezifiziert werden. Wird dies nicht getan, kommt nur die Schaltung einzelner SVCs für jede RSVP-Anforderung in Frage. Als Alternative dazu wäre eine direkte Übertragung der Datenströme vom Client über ATM eine Möglichkeit. Zusammenfassend kann man sagen, dass es sich trotz großen Managementaufwands und

erheblichen Overhead lohnt, RSVP-Datenströme auf ATM-Verbindungen anzupassen. Gerade die Hard-State-ATM-QoS-Parameter lassen eine garantierte Dienstgüte zu, die IP ohne Schicht-2-Unterstützung nicht in dieser Form leisten kann. [CBBB+98]

6.2.4 Mapping auf DiffServ-Netze

Das DiffServ-Konzept ist in erster Linie für das innere Netz entwickelt worden. In diesem Bereich werden zwischen den Routern meistens Punkt-zu-Punkt-Kanäle, beispielsweise über SDH geschaltet. Diese Kanäle kann man aus Sicht von DiffServ als serielle Leitung mit festem Datendurchsatz betrachten. Somit ist hier der entscheidende Punkt, das richtige Paket-Scheduling-Verfahren in den Forwarding-Einheiten des Routers zu wählen. In Kapitel 3 wurde bereits auf mögliche DiffServ-Architektur und das Zusammenspiel mit IntServ eingegangen, weshalb hier darüber keine weitere Betrachtung erfolgt.

6.3 MPLS

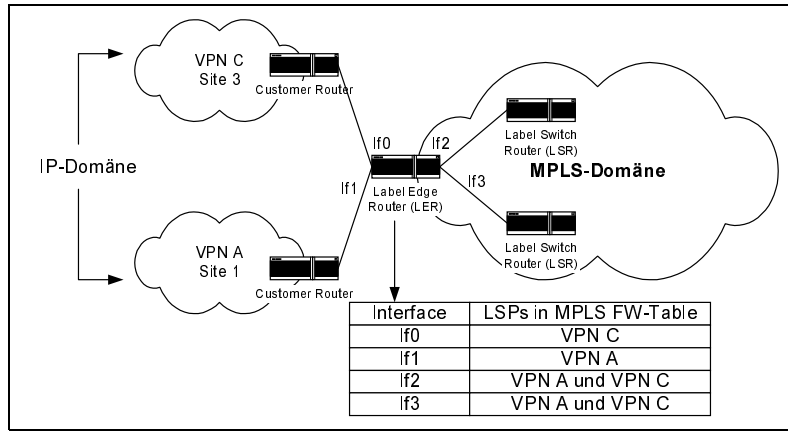
Neben den erwähnten Leistungsmerkmalen im Bereich des Traffic Engineering (TE) bietet das MPLS-Verfahren gezielt die Möglichkeit an, Virtual Private Networks (VPN) aufzubauen. Ein VPN kann man dabei als ein Netzwerk ansehen, in dem die Verbindung der einzelnen Standorte (Sites) des Netzbetreibers über eine geteilte Infrastruktur erfolgt. Bei dieser Infrastruktur kann es sich beispielsweise um ein öffentliches Netz oder um ein Netz eines VPN-Service-Providers handeln. Dieses Konzept existiert bereits seit über 10 Jahren und wird bezüglich Zuverlässigkeit, Sicherheit, Skalierbarkeit und Kosteneffizienz ständig weiterentwickelt. Hinzu kam die Entwicklung des Internets, die in den letzten 10 Jahren ebenfalls einen großen Sprung gemacht hat. Eine weitere Optimierung gegenüber bisherigen Realisierungen von VPNs soll mit der Einführung des MPLS erreicht werden. Die Realisierung eines VPN setzt für das Verbindungsnetz die Fähigkeit voraus, die Daten verschiedener VPNs auf logischer Ebene trennen zu können. Dies ist erforderlich, da aus Gründen der Datensicherheit und Verbindungsqualität sichergestellt werden muss, dass die Daten eines VPN nicht in ein anderes VPN fehlgeleitet werden.

6.3.1 Datentrennung

An dieser Stelle soll ein MPLS-Verbindungsnetz basierend auf dem Border Gateway Protocol (BGP) die logische Trennung der Daten gewährleisten. Dazu muss zunächst der Netzaufbau betrachtet werden. Es wird hierbei ein Peer-to-peer-VPN-Modell verwendet, das in Abb. 6.21 dargestellt ist. Hierbei erfolgt der Peer-Aufbau zwischen den Kunden-Routern und ihren direkt angeschlossenen Provider-Routern. Auf diese Weise können Routing-Informationen zwischen dem Kunden und dem Provider ausgetauscht werden.

Die logische Trennung der Daten verschiedener VPNs erfolgt bei der MPLS/BGP-Lösung über den kontrollierten Austausch dieser Routing-Informationen, ohne die eine Vermittlung auch im MPLS nicht möglich ist. Das Ziel besteht darin, dass ein LER ausschließlich die Routing-Informationen für VPNs verwendet, an die er auch direkt angeschlossen ist. Auf diese Weise werden die Routing-Informationen reduziert, die ein LER[82] verarbeiten muss. Damit verbunden reduziert sich auch die Wahrscheinlichkeit der Datenfehlvermittlung. Zusätzlich wird die Zuverlässigkeit der Datentrennung erhöht, indem jeder LER MPLS-Forwarding-Tabellen auf Interface-Basis führt. Dabei wird für jeden Eingangsport, an den direkt ein VPN angeschlossen ist, eine Forwarding-Tabelle unterhalten, die ausschließlich Einträge für dieses VPN führt. Die Fehlvermittlung von Daten aus einem direkt angeschlossenen VPN in ein anderes VPN ist damit ausgeschlossen. Die MPLS-Forwarding-Tabelle auf dem VPN-Interface enthält nur LSPs[83] für dieses VPN. Die Forwarding-Tabelle eines Interface, an das ein LSR[84] angeschlossen ist, enthält hingegen Einträge für alle direkt angeschlossenen VPNs, nicht aber für alle VPNs, die an die MPLS-Domäne angeschlossen sind. Diese Einträge lassen sich nicht weiter reduzieren. Abb. 6.21 verdeutlicht ebenfalls, in welcher MPLS-Forwarding-Tabelle welche Einträge verwaltet werden.

Abb. 6.21 Peer-to-peer-VPN-Modell mit Forwarding-Tabelle



Der Aufbau der einzelnen MPLS-Forwarding-Tabellen beginnt in einem LER mit dem Empfang von Routing-Informationen aus einem direkt angeschlossenen VPN. Der LER verwendet diese Routing-Informationen, um eine MPLS-Forwarding-Tabelle für dieses VPN bzw. diese Schnittstelle zu erstellen. Anschließend wird die Routing-Information über das BGP-Protokoll weiterge-

82 Label Edge-Router
83 Label Switched Path
84 Label Switch Router

leitet. Das BGP bietet dabei die Möglichkeit, das VPN über das so genannte Community-Attribut zu kennzeichnen. Anhand dieses Attributs und der Kenntnis, welche VPNs direkt angeschlossen sind, können die anderen LERs dann entscheiden, ob sie die über das BGP empfangene Routing-Information verwerten oder ignorieren müssen.

6.3.2 Adressierung

Ein weiteres Problem, das bei der Realisierung mehrerer VPNs über eine MPLS-Domäne oder allgemein über ein Verbindungsnetz berücksichtigt werden muss, besteht in der Adressierung. Normalerweise werden für den Aufbau von MPLS-Forwarding-Tabellen in einer MPLS-Domäne FECs[85] verwendet, die auf Adresspräfixen basieren. Dies ist im Netz eines VPN-Service-Providers aber nicht möglich. Nach der Spezifikation RFC-1918 müssen IP-Adressen in VPNs nur innerhalb des eigenen VPNs eindeutig vergeben sein. Durch diese Festlegungen können in verschiedenen VPNs die gleichen IP-Adressen verwendet werden. Hierdurch ist der eindeutige Aufbau der MPLS Forwarding-Tabelle ohne weitere Mechanismen nicht möglich. [RMKG+96]

Für das Problem der Adressierung in VPNs bietet BGP die Möglichkeit, die lediglich im VPN eindeutigen IP-Adressen in global eindeutige IP-Adressen umzuwandeln. Dies wird erreicht, indem die IP-Adressen mit einem so genannten Route Distinguisher, der aus den Feldern Type, Autonomous System Number und Assigned Number besteht, verkettet wird. Auf diese Weise können die MPLS-Forwarding-Tabellen eindeutig aufgebaut werden. [FROMM01]

6.3.3 LSP-Tunnel

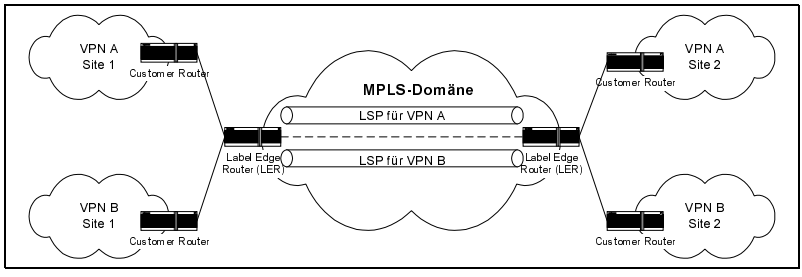


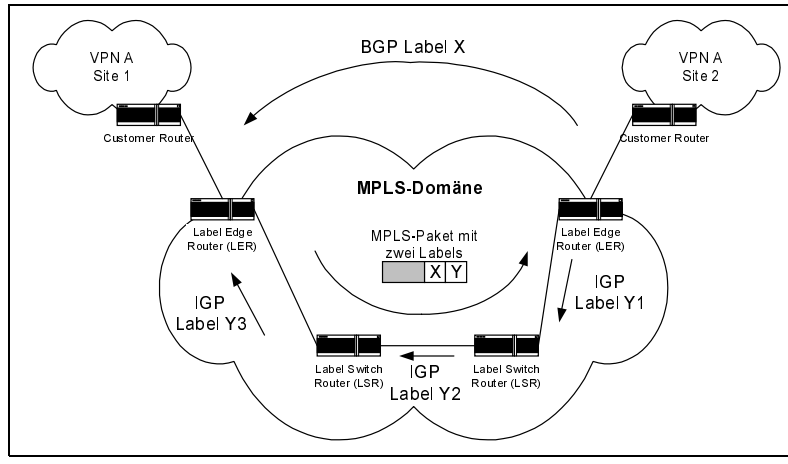
Abb. 6.22 LSP-Tunnel

Die Aufgabe des in Abb. 6.22 dargestellten MPLS-Netzes besteht in der Verbindung der einzelnen Sites der angeschlossenen VPNs. Die Verbindung der Sites der VPNs erfolgt über separate LSPs. Dabei können verschiedene LSPs über den gleichen Pfad aufgebaut werden. Die Abb. 6.22 verdeutlicht das Szenario anhand eines LSP pro VPN.

85 Forwarding Equivalence Class

Die bisher betrachteten MPLS-Forwarding-Tabellen der LER müssen für jedes direkt angeschlossene VPN Einträge unterhalten. Dies gilt ohne weitere Maßnahmen auch für die LSR im Kern des MPLS-Netzes. Die Forwarding-Tabellen können hierbei sogar noch größer werden als die der LER, da diese sich nicht auf die direkt angeschlossenen VPNs beschränken können. Um zu vermeiden, dass die MPLS-Forwarding-Tabellen der LSR zu groß werden, können LSP-Tunnel eingesetzt werden. Hierbei werden die VPN-LSPs in einen übergeordneten LSP abgebildet. Diese übergeordneten LSPs verbinden keine Sites, sondern ausschließlich LERs. Bei einer vollen Vermaschung der LERs ist es möglich, alle VPN-LSPs, unabhängig von der Anzahl der angeschlossen VPNs und Sites, in eine begrenzte Anzahl von übergeordneten LSPs abzubilden. Die LSRs müssen nur Einträge für die übergeordneten LSPs unterhalten.

Abb. 6.23
LSP-Tunnel in einem MPLS-VPN



Die Realisierung eines LSP-Tunnels erfolgt über die Verwendung des Label-Stacks. Das Binding für den VPN-LSP erfolgt über das BGP-Protokoll. Das entsprechende Label wird dem IP-Paket angefügt. Anschließend wird ein zweites Label (Toplabel) in den Label-Stack des Pakets geschrieben. Dieses Label bestimmt den übergeordneten LSP zum Ziel-LER. Das Binding für diesen LSP erfolgt über ein Interior Gateway Protocol (IGP) oder LDP[86]. Anhand dieses zweiten Labels erfolgt die Weiterleitung in den LSRs. Im Ziel-LER wird dieses Label entfernt. Über das erste Label kann der LER dann das Ziel-VPN bzw. die Site bestimmen. Abb. 6.23 zeigt die Funktion des LSP-Tunnels.

86 Label Distribution Protocol

6.3.4 VPN-Anwendung

Die Umsetzung auf ein VPN bzw. Extranet wird zu einer Konvergenz von Netz- und Dienstplattformen führen. Somit werden bekannte Dienste auf alternativen Plattformen eingesetzt und neue, kombinierte (Multimedia-)Dienste wie Videokonferenzsysteme, E-Mail-Handy, IP-Telefonie, LAN-LAN-Verbindungen, Homebanking, Distant Learning, Multimedia Service und Shopping verfügbar sein. Extranets werden heute oftmals mit VPNs gleichgesetzt und deshalb vornehmlich als die Realisierungsform von Corporate Networks (CN) großer Unternehmen angesehen. Dabei gibt es unterschiedliche Möglichkeiten, ein VPN zu etablieren. Über Leased Lines oder Festverbindungen lassen sich eigene VPNs aufbauen. Dabei stehen meistens Einsparungsmöglichkeiten und Verfügbarkeit im Vordergrund. Zusätzlich lassen sich VPNs im Mobilfunkbereich genauso realisieren wie im Festnetzbereich. Ein Extranet beinhaltet aber gegenüber einem VPN die Einbeziehung von Kommunikationsprozessen mit Partnern, Lieferanten und Kunden. Das VPN stellt somit die Basisplattform für ein Extranet dar.

Für die Verbindung der einzelnen Außenstellen werden die genannten Tunneling-Verfahren eingesetzt, mit deren Hilfe sichere, private Verbindungen für Netzapplikationen über ein öffentliches oder ein unsicheres Medium zwischen abgesetzten Netzwerken und/oder einzelnen PC-Arbeitsplätzen zu einem zentralen Datennetz aufgebaut werden. Der Einsatz eines VPN für Unternehmen bedarf der Berücksichtigung unterschiedlicher Anforderungen, die sich in folgende Punkte aufgliedern lassen:

- **Verfügbarkeit:** Die Verfügbarkeit eines Netzes spielt heute für ein Unternehmen eine entscheidende Rolle und besitzt die höchste Priorität gegenüber anderen Anforderungen. Die Nichtverfügbarkeit eines Netzes bzw. die daraus resultierenden Verzögerungen in der Informationsverarbeitung und Kommunikation ist aus unternehmerischer Sicht untragbar, da sie monetäre und rechtliche Auswirkungen haben und dadurch immense Verluste bedeuten können.
- **Sicherheit:** Da das Extranet als Erweiterung des Intranets über die Organisationsgrenzen hinweg anzusehen ist, ist Sicherheit ein wesentlicher Punkt in der Entscheidung für den Einsatz von VPNs. Bei der Nutzung des Internets als Netz kommt diesem Sachverhalt eine besondere Bedeutung zu, da unternehmensinterne Informationen öffentlich zugänglich gemacht werden können.
- **Skalierbarkeit:** Der hohen Dynamik der Kommunikationsbeziehungen in einem VPN muss durch eine flexible Skalierbarkeit begegnet werden. Die Skalierbarkeit muss sich auf die Netzwerkarchitektur, die Netzwerkkomponenten und das übergeordnete Netzwerkmanagement beziehen. Die Anzahl und Zugangsberechtigungen der Teilnehmer und Benutzer müssen

schnell und effizient gemanagt werden können. Das bedeutet, dass VPNs auf bestehenden oder kommenden Standards aufbauen müssen, um eine feste Skalierbarkeit erreichen zu können.

- **Quality-of-Service (QoS):** Hohe Performance ist nur ein Parameter für die Dienstqualität (QoS), die sich im Grunde durch den Durchsatz bzw. die Bandbreite ausdrückt. Eine garantierte Bandbreite mit einer bestimmten Dienstgüte zu erhalten ist für die Adaption von VPNs ein entscheidendes Kriterium.
- **Mobilität:** Die steigende Notwendigkeit der externen Anbindung mobiler Nutzer im Sinne eines „virtuellen Büros“[87] für den Zugriff auf Firmenressourcen bedingt die flächendeckende Bereitstellung von Einwahlmöglichkeiten[88]. Die Anforderungen werden hierbei von Anzahl, Zugriffsverhalten, geografischer Verteilung und Zugangsberechtigungen[89] der Nutzer bestimmt. Hier muss in den Gateways[90] eine angepasste Sicherheits- und Zugangsberechtigung Wirkung finden, da diese Klasse von Nutzern auf Netzressourcen zugreifen.
- **Netzwerkmanagement:** Im Vordergrund eines jeden Dienstes steht das Management des zugrunde liegenden Netzes und Dienstes, welches Komponenten wie Authentifizierungs-Server, Remote Access Server (RAS), Router, Firewall-Systeme usw. umfasst. Weiterhin ist für den täglichen Betrieb, der eine hundertprozentige Verfügbarkeit voraussetzt, ein effizientes Netzwerkmanagement unverzichtbar.
- **Accounting und Billing:** Das Accounting sollte auf den Daten in den Authentifizierungs-Servern aufsetzen, um daraus die Billing-Informationen für die Organisation zu generieren. Die Verarbeitung von Accounting-Informationen für Billing-Mechanismen, d.h. eine Kopplung zwischen Accounting und Billing, ist besonders vorteilhaft, wenn die Authentifizierung beim Service Provider erfolgt.
- **Migrationsfähigkeit und Integrierbarkeit:** Konzepte zum Aufbau von Extranets haben grundsätzlich die Aufgabe, existierende IuK-Strukturen[91] so einzubinden, dass eine einfache und investitionsschützende Migration möglich wird. Das umfasst auch die Unterstützung unterschiedlicher Netzwerkprotokolle wie IP, IPX, ATM, AppleTalk, DECnet, SNA usw.

Diese Anforderungskriterien muss jedes Unternehmen an ein VPN bzw. Extranet individuell berücksichtigen, wenn es effektiv, kostensparend, sicher und leistungsfähig eingesetzt werden soll.

87 Z.B. Außendienstmitarbeiter
88 Dial-In, Remote Access
89 Autorisierung, Authentifizierung
90 Router, Firewall-Systeme, Server etc.
91 Insbesondere Legacy-Systeme

Es lassen sich VPN-Plattformen unterschiedlich realisieren. Hier steht bei der Realisierung IPsec und MPLS im Vordergrund. Diese Technologien müssen unterschiedlichen Szenarios gerecht werden:

1. **Access VPN:** In einem Access VPN wird einer Person der entfernte Zugriff auf ein privates Netzwerk ermöglicht. So können beispielsweise Außendienstmitarbeiter sicher auf ein Unternehmensnetzwerk zugreifen. Zur Etablierung einer sicheren Verbindung unterscheidet man zwei Möglichkeiten (siehe Abb. 6.24):
   - Client-initiated: Bereits durch die Einwahlsoftware auf dem Client-Rechner wird eine sichere IP-Verbindung zum Internet Service Provider (ISP) aufgebaut und so bis in das Unternehmensnetzwerk weitergeleitet.
   - NAS-initiated: Vom Client-Rechner wird eine nicht sichere Verbindung zum ISP aufgebaut und erst der Network Access Server (NAS) stellt eine sichere Verbindung bis ins Unternehmensnetzwerk her.
2. **Intranet VPN:** Ein Intranet VPN dient dazu, den Hauptsitz eines Unternehmens und dessen privates Netzwerk mit Außenstellen zu verbinden. Dabei wird die Sicherheit der Verbindung durch Einsatz entsprechender Router gewährleistet, die in der Lage sind, z.B. durch IPsec die Kommunikation zwischen den beiden privaten Netzwerken zu sichern.
3. **Extranet VPN:** Im Gegensatz zum Intranet VPN bietet ein Unternehmen im Extranet VPN zusätzlich zu den Außenstellen auch Kunden, Lieferanten und Geschäftspartnern beschränkten Zugriff auf das Unternehmensnetzwerk. In diesem Fall ist ein kombinierter Einsatz von geeigneter Hard- und Software nötig, d.h., alle Kommunikationspartner müssen aufeinander abgestimmt sein, was durch die Nutzung eines einheitlichen Standards wie IPsec natürlich einfacher ist.

Der Aufbau eines VPN beinhaltet eine Vielzahl von Anforderungen an die Planung und Konzeption. Gerade die genannten Punkte spielen dabei eine herausragende Rolle und sind für die Umsetzung entscheidend. Die Kostenvorteile gegenüber anderen Netztechnologien und die hohe Flexibilität versprechen in der Zukunft ein hohes Marktvolumen. IPsec und MPLS zeigen hier neue Realisierungsmöglichkeiten auf, um unter Berücksichtigung der genannten Anforderungen herstellerneutral ein VPN bzw. Extranet aufbauen zu können.[92] [DEER01]

92 Detaillierte Strukturierungs-, Realisierungs- und Implementierungsmöglichkeiten findet man unter [DEER01].

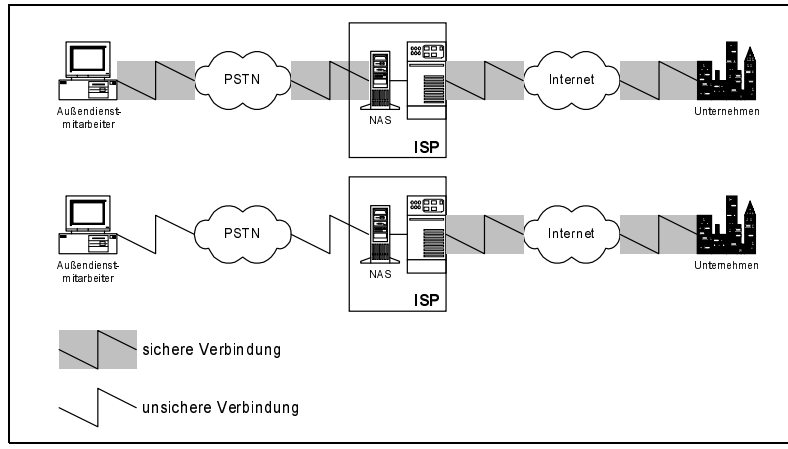


Abb. 6.24
Etablierung einer sicheren Verbindung

6.3.5 Bewertung

An dieser Stelle soll wieder eine kurze Bewertung von MPLS mit POS[93] oder ATM als Schicht-2-Technologie bezüglich einer VPN-Realisierung erfolgen. Das zu betrachtende VPN besteht dabei aus mehreren Teilnetzen, die zum Teil über ein oder mehrere öffentliche Netze verbunden sind. Die Anwendung liegt beispielsweise im Aufbau von Intranets.

6.3.6 ATM-VPN

Für die Realisierung von VPNs über ein öffentliches ATM-Netz wird das Overlay-VPN-Modell verwendet. Es wird im Folgenden von mehreren IP-Subnetzen eines Kunden ausgegangen, die über das ATM-Netz eines Providers verbunden sind. Jedes IP-Subnetz repräsentiert einen Standort (eine Site) des Kunden. Hierbei besteht die einzige Aufgabe des Providers darin, seinen Kunden ATM-VCs für die Verbindung der einzelnen Standorte zur Verfügung zu stellen. Bei diesen VCs kann es sich sowohl um PVCs als auch um SVCs handeln.

IP-Routing-Informationen der einzelnen Standorte werden bei diesem VPN-Modell nur zwischen den Kunden-Routern ausgetauscht. Diese Informationen werden im ATM-Netz wie Nutzinformationen behandelt und übertragen. Die Provider-Router werten diese Routing-Informationen nicht aus und haben aus diesem Grund keine Kenntnis über die interne Struktur des Kunden-IP-Netzes. Da die Aufgabe des Providers ausschließlich in der Verbindung der Kunden-Router liegt, werden diese IP-Routing-Informationen allerdings auch nicht benötigt. Die Routing-Entscheidung für die Datenvermittlung in andere Standorte muss im Overlay-VPN-Modell in den Kunden-Routern über die Wahl des VC getroffen werden, wie Abb. 6.25 verdeutlicht.

93 Packet-over-SONET

Für die Realisierung eines VPN über ein öffentliches MPLS-Netz wird das Peer-to-peer-VPN-Modell verwendet. Der Peer-Aufbau erfolgt hierbei nicht zwischen den Kunden-Routern der einzelnen Sites, sondern ausschließlich zwischen den Kunden-Routern und den direkt angeschlossenen Provider-Routern. Das bedeutet, dass zwischen den Kunden-Routern und den Provider-Routern IP-Routing-Informationen ausgetauscht werden. Auf diese Weise erhält der Provider die notwendigen Informationen, um selbst an der Rout-Findung für die Datenvermittlung an verschiedene Standorte teilzunehmen.

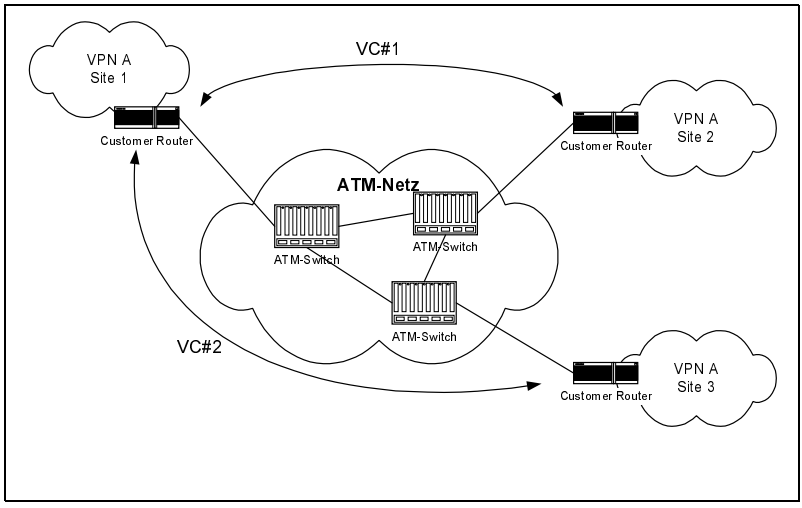


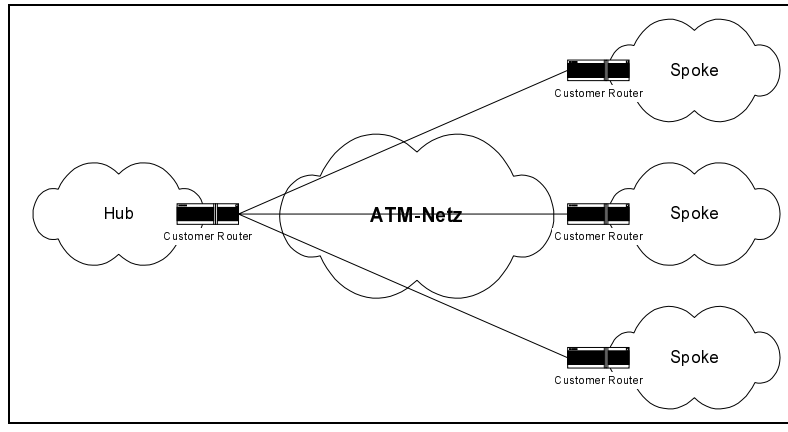
Abb. 6.25 Overlay-ATM-VPN-Modell

Die VPN-Unterstützung ist im MPLS dadurch effektiver möglich. Im Overlay-VPN-Modell, welches von ATM verwendet wird, müssen alle über einen VC verbundenen Router eines VPN untereinander Routing-Informationen austauschen. Bei VPNs mit vielen Sites würde eine volle Vermaschung der einzelnen Sites zu einer großen Menge von zu übertragenden Routing-Informationen führen, was das Netz stark belasten kann. Zusätzlich können die Routing-Tabellen in den einzelnen Routern eines VPN stark anwachsen. Hierdurch werden die Router, insbesondere die Central Processing Unit (CPU) belastet, da die Suche in den Routing-Tabellen aufwendiger wird. Um diesen Problemen entgegenzuwirken, wird üblicherweise auf eine volle Vermaschung verzichtet und eine Hub-and-Spoke-Topologie gewählt. Bei einer Hub-and-Spoke-Topologie erfolgt die Verbindung einzelner Sites (Spokes) über eine bestimmte Site (Hub). Abb. 6.26 verdeutlicht die Topologie.

Der Hub stellt hierbei einen Single-Point-of-Failure (SPOF) dar. Fällt der Hub aus, so sind alle angeschlossenen Sites ohne Verbindung. Des Weiteren reduziert die Hub-and-Spoke-Topologie die Performance, da die Verbindung zwischen zwei Spokes über den Hub und nicht direkt abläuft. Im MPLS ist eine

volle Vermaschung problemlos möglich. Der Grund hierfür liegt in der Verwendung des Peer-to-peer-VPN-Modells. Hierbei erfolgt ein Peer-Aufbau für einen Router einer Site nur mit dem direkt angeschlossenen Label Edge-Router (LER) der MPLS-Domäne. Auf diese Weise wird das Netz im Vergleich zum Overlay-VPN-Modell nur gering mit dem Transport von Routing-Informationen belastet. Zusammenfassend besteht der Hauptvorteil von MPLS im Vergleich zu ATM in der effektiveren VPN-Unterstützung.

Abb. 6.26 Hub-and-Spoke-Topologie



6.3.7 POS-VPN

Für die Realisierung eines VPN über ein POS-Netz kann das IP-Tunneling verwendet werden. Für diese Methode wird das Protokoll IPsec vorgeschlagen und bevorzugt. Für die VPN-Unterstützung durch das IP-Tunneling wird ebenfalls das Overlay VPN-Modell verwendet, das bereits im Zusammenhang mit ATM betrachtet wurde. Diese Realisierung gleicht prinzipiell der ATM-Lösung. Aus diesem Grund bestehen im MPLS die gleichen Vorteile, die schon für den Vergleich MPLS und ATM erarbeitet wurden. Zusätzlich ist zu beachten, dass das IP-Tunneling, welches im POS für die VPN-Unterstützung verwendet wird, den Overhead erhöht und auf diese Weise die Nettobitrate senkt. Hier liegt ein weiterer und wichtiger Vorteil von MPLS. Zusammenfassend bestehen die Hauptvorteile von MPLS im Vergleich zum POS ebenfalls in der effektiveren VPN-Unterstützung.

6.4 Voice-over-IP (VoIP)

Nachdem nun die Plattform geschaffen wurde, um sicher und mit einer garantierten Dienstgüte zu kommunizieren, wird an dieser Stelle die Echtzeitkommunikation mittels VoIP dargestellt und integriert. Dabei ist der Verbindungssteuerung besondere Aufmerksam zu schenken bzw. der Kommunikation

zwischen Gateway und Gatekeeper, da diese grundlegend für ein H.323-Endgerät ist. Dies beinhaltet Funktionen zur Signalisierung des Verbindungsauf- und -abbaus, den Austausch von Endgerätefunktionalitäten, sowie den Nachrichtenaustausch, um weitere Verbindungen zur Übertragung von Audio, Video, oder Daten aufzubauen und zu beschreiben.

6.4.1 Digitalwandlung

Jede Sprachinformation liegt zuerst analog vor. Für eine Vermittlung durch Daten- und Telekommunikationsnetze muss eine Umsetzung in Digitalsignale erfolgen. Dabei werden zur effizienten Übertragung in IP-Netzwerken die Sprachinformationen durch Kompression in der Bandbreite reduziert. Ziel bei der Wahl des Kompressionsalgorithmus ist es, eine hohe Kompressionsrate bei niedriger Verzögerung (Delay) zu erreichen.

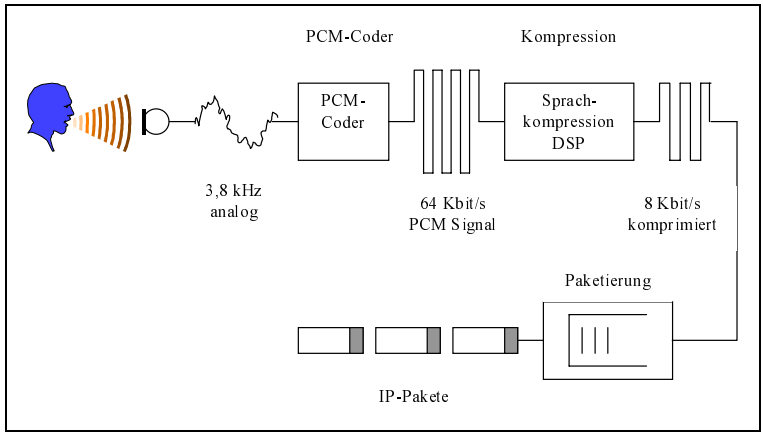


Abb. 6.27 Digitalisierung der Sprache[a]

a. Es wird PCM nur verwendet, wenn keine Kompression angewandt wird.

Die komprimierte Sprachinformation am Ausgang des Kompressors wird in Form von UDP-Paketen in das IP-Netzwerk gesendet. Die Übertragung erfolgt dadurch verbindungslos. So wird zusätzlicher Overhead sowie Empfangsquittierungen durch ein Protokoll wie TCP vermieden. Eine Sendewiederholung ist sowieso bei Sprachdaten nicht sinnvoll. Stattdessen wird bei Paketverlust ein Stummsignal vom Ziel-Terminal ausgegeben. Bei geringen Paketverlustraten[94] wird dies noch kaum wahrgenommen. In dem Ziel-Terminal wird dann die entsprechende Dekompression und gegebenenfalls eine Analogwandlung des digitalen Sprachsignals durchgeführt. Zum Erreichen geringer Paketverlustraten im Trägernetz sind geeignete Maßnahmen wie die Einführung von QoS-Mechanismen, ausreichende Dimensionierung des Netzes und Warteschlangen vorzusehen.

94 Im Bereich 3-5 %

Die Kompressionsalgorithmen können in Form von Microcode auf digitalen Signalprozessoren (DSP) realisiert werden. Dies bietet die Möglichkeit, die Art der Kompression je nach Anwendung frei zu wählen. Sprachpausenunterdrückung[95] vermeidet, dass Audiodaten übertragen werden, wenn einer der Gesprächsteilnehmer gar nichts von sich gibt. Die Telefonkommunikation erfolgt im Halb-Duplex-Betrieb, da beide Seiten sehr selten gleichzeitig reden. Indem man „Silence" auf einer Seite durch einen Standardwert kodiert, lässt sich die benötigte Bandbreite der Telefonverbindung erheblich senken. Dafür muss das System in der Lage sein, auf neue Geräusche, die ein Abweichen vom Ruhezustand darstellen, sofort zu reagieren. Silence Suppression kann zusätzlich zur Datenkompression eingesetzt werden. Die Echokompensations-Algorithmen werden ebenfalls als Microcodes auf DSP-Basis implementiert und stehen dann zusammen mit der Sprachkompression für bandbreiteneffiziente und qualitativ hochwertige Sprachdienste zur Verfügung. [GORE99]

6.4.2 Verbindungsaufbau

Der Verbindungsaufbau bzw. die Gesprächssignalisierung dient zum Aufbau und Schließen von Verbindungen, zum Anfragen nach Bandbreitenänderung und zur Auskunft über den Status der Endpunkte in einer bestehenden Verbindung. Realisiert werden diese Funktionen durch Nachrichten und Prozeduren gemäß der H.225.0-Empfehlung. Man kann zwischen drei verschiedenen Signalisierungsfunktionen unterscheiden: der H.245-Steuerungskanal, der H.225.0/Q.931-Signalisierungskanal und der RAS[96]-Kanal.

Adressierung Jede H.323-Einheit muss durch mindestens eine Netzwerkadresse eindeutig identifizierbar sein. Dabei können sich mehrere Einheiten gemeinsam eine Netzwerkadresse teilen, die entsprechend der Netzwerkumgebung unterschiedlich formatiert sein kann. Ein Endpunkt kann mehrere Netzwerkadressen für verschiedene Kanäle innerhalb einer Verbindung verwenden. Für jede Netzwerkadresse kann ein Endpunkt mehrere Transport Layer Service Access Points (TSAP)[97] besitzen, mit deren Hilfe das Multiplexen mehrerer Kanäle über dieselbe Netzwerkadresse möglich ist. Für einen Endpunkt ist zum Empfangen von Q.931-Nachrichten wie SETUP über den Gesprächssignalisierungskanal die Portnummer 1720 festgelegt, das heißt, ein Endpunkt schickt eine SETUP-Nachricht an diesen Port der IP-Adresse des gewünschten Gesprächspartners. Der anrufende Endpunkt wiederum kann dem angerufenen Endpunkt eine dynamische Portnummer mitteilen, an welcher er die Q.931-Nachrichten empfängt.

95 Silence Surpression
96 Registration Admission Status
97 Entspricht der Portnummer in einer TCP/UDP/IP-Umgebung.

Für den RAS-Kanal eines Gatekeeper ist die Portnummer 1719 als Standardwert festgelegt. Ebenso ist für die Ermittlung der Gatekeeper-Zuständigkeit[98] die Multicast-Adresse 224.0.1.41 mit der Portnummer 1718 festgelegt. Dynamische Portnummern werden für H.245 Kontroll-, Audio-, Video- und Datenkanäle vergeben. Ein Gatekeeper ordnet Gesprächssignalisierungskanälen dynamische Portnummern zu. Während der Registrierungsprozedur kann ein Gatekeeper den RAS-Kanälen auch dynamische Portnummern zuweisen. Aliasadressen bieten alternative Adressierungsmethoden und können sowohl für Endpunkte als auch für Konferenzen vergeben werden, an denen ein Endpunkt teilnimmt. Zur alternativen Adressierung können E.164-Telefonnummern oder H.323-IDs für alphanumerische Zeichen von Namen oder E-Mail-Adressen verwendet werden. Gatekeeper, MC[99] und MP[100] haben keine Aliasadressen. Endpunkte können dagegen mehrere Aliasadressen besitzen. In Systemen ohne Gatekeeper können Aliasadressen nicht verwendet werden. [DOMM98]

Tab. 6.6 Adressierung durch TSAP

TSAP	Übertragung	Adressierung
RAS-Kontrolle	unzuverlässig (UDP)	• festgelegter Standardwert (Port 1719) • dynamisch (kann während Registrierungsprozedur geändert werden)
Q.931-Gesprächssignalisierung	zuverlässig (TCP)	• festgelegt für angerufenen Endpunkt, Port 1720 • dynamisch für anrufenden Endpunkt
H.245-Kontrolle	zuverlässig (TCP)	dynamisch
Audio/RTP	unzuverlässig (UDP)	dynamisch
Audio/RTCP	unzuverlässig (UDP)	dynamisch
Video/RTP	unzuverlässig (UDP)	dynamisch
Video/RTCP	unzuverlässig (UDP)	dynamisch
T.120-Daten	zuverlässig (TCP)	festgelegt/dynamisch

Soll eine Verbindung zum Zielteilnehmer aufgebaut werden, so ist zunächst die IP-Adresse des Gateway zu bestimmen, über welches der Zielteilnehmer angeschlossen ist. Dies kann entweder über eine interne Adressumsetzungstabelle oder eine zentrale Gatekeeper-Funktion erfolgen. Der Weg über interne Gate-

98 Gatekeeper Discovery Procedure
99 Mulipoint Controller
100 Multipoint Prozessor

way-Tabellen bietet kurze Verbindungsaufbauzeiten, bringt jedoch einen erhöhten Verwaltungsaufwand gegenüber einer zentralen Lösung über einen Gatekeeper mit sich. Im Gatekeeper lassen sich zudem Zusatzfunktionen wie die Vergabe von Zugriffsrechten leicht implementieren.

RAS-Kanal Erfolgt die Kommunikation unter H.323 mit Beteiligung eines Gatekeeper, werden im ersten Schritt zwischen diesem und den Endpunkten H.225.0-Nachrichten über den RAS-Kanal ausgetauscht. Der Nachrichtenaustausch vollzieht sich in der Regel so, dass zunächst eine Anfrage (Request – RQ) gesendet wird, die anschließend entweder bestätigt (Confirm – CF) oder abgelehnt (Reject – RJ) wird. Die wichtigsten Nachrichten lassen sich wie folgt unterscheiden:

- **Gatekeeper-Zuständigkeit:** Damit sich ein Endpunkt bei einem Gatekeeper registrieren kann, muss er dessen Transportadresse kennen. Diese ist dem Endpunkt entweder von vornherein durch Konfigurationseinstellungen bekannt oder er sendet die Nachricht Gatekeeper_Request (GRQ) an die ihm bekannte Multicast-Discovery-Adresse, um den zuständigen Gatekeeper ausfindig zu machen. Daraufhin erhält er entweder die Antwort Gatekeeper_Confirm (GCF), in welcher ein oder mehrere Gatekeeper die Transportadresse ihres RAS-Kanals mitteilen, oder er empfängt Zurückweisungen durch die Antwort Gatekeeper_Reject (GRJ). Senden mehrere Gatekeeper die Antwort GCF, kann sich der Endpunkt einen Gatekeeper für die Registrierung aussuchen.
- **Endpunkt-Registrierung:** Bei der Endpunkt-Registrierung meldet sich ein Endpunkt in einer Zone an und teilt einem bestimmten Gatekeeper seine Transportadresse und weitere Aliasadressen mit. Die Registrierung erfolgt durch das Versenden eines Registration_Request (RRQ) an die Transportadresse des RAS-Kanals eines Gatekeepers. Der Gatekeeper antwortet entweder mit einer Registrierungsbestätigung durch die Nachricht Registration_Confirm (RCF) oder mit einer Ablehnung durch Registration_Reject (RRJ). Erhält ein Endpunkt die Antwort RCF, ist er registriert und muss für alle zukünftigen Gespräche den Gatekeeper verwenden. Dieser kann allerdings auch den direkten Nachrichtenaustausch zwischen den Endpunkten zulassen. Die Registrierung eines Endpunkts kann zeitlich befristet sein und muss in diesem Fall nach Ablauf einer festgelegten Zeit durch eine weitere RRQ erneuert werden. Ein Endpunkt oder ein Gatekeeper kann eine Registrierung durch die Nachricht Unregister_Request (URQ) auflösen. Der Empfänger dieser Nachricht schickt daraufhin die Antwort Unregister_Confirm (UCF), welche die Auflösung bestätigt.
- **Endpunkt-Lokalisierung:** Ein Endpunkt, der nur die Aliasadresse eines anderen Endpunkts kennt, kann einen Location_Request (LRQ), welcher die Aliasadresse beinhaltet, entweder an den Gatekeeper, bei dem er registriert ist, oder an die bekannte Multicast Discovery Address schicken. Der

Gatekeeper sendet als Antwort die Nachricht Location_Confirm (LCF), welche weitere Kontaktinformationen über den Endpunkt oder den Gatekeeper des Endpunkts beinhaltet. Die Kontaktinformationen enthalten die Adresse des Gesprächssignalisierungskanals oder des RAS-Kanals, um den gewünschten Endpunkt zu erreichen. Alle Gatekeeper, die eine LRQ über den RAS-Kanal erhalten haben und bei denen der nachgefragte Endpunkt nicht registriert ist, antworten mit der Zurückweisung Location_Reject (LRJ). Falls ein Gatekeeper eine LRQ-Nachricht über die Multicast Discovery Address empfängt und der gesuchte Endpunkt bei ihm nicht registriert ist, schickt er keine Antwort auf die LRQ-Anfrage. Ein Endpunkt kann mit LRQ auch E.164-Telefonnummern im Feld Destination Info mitschicken, um gewünschte Kontaktinformationen von einem Gatekeeper zu erhalten und eine Verbindung zu einem bestimmten Endpunkt aufzubauen. Ein Gatekeeper ist jedoch nicht verpflichtet, Gateways für den Verbindungsaufbau zu den gewünschten Endpunkten zur Verfügung zu stellen.

- **H.323-Zonenzugang:** Damit ein Endpunkt Zugang zu einem LAN über einen Gatekeeper erhält, muss er an den Gatekeeper die Nachricht Admission_Request (ARQ) senden, welche Angaben über die Bandbreite für ein gewünschtes Gespräch[101] enthält. Der Wert bezieht sich auf die Audio- und Video-Bitraten und schließt Protokoll-Header und Overhead aus. Der Gatekeeper schickt entweder eine Zugangsbestätigung durch Admission_Confirm (ACF), die eine modifizierte Bandbreite vorgeben kann, oder er lehnt den Zugang mit der Nachricht Admission_Reject (ARJ) ab, wenn z.B. bereits zu viele Gespräche in der Zone geführt werden und damit nicht mehr genug Bandbreite für ein weiteres Gespräch zur Verfügung steht oder ein in dieser Zone nicht registrierter Endpunkt eine ARQ-Nachricht an den Gatekeeper geschickt hat.
- **Bandbreitenänderung:** Ein Endpunkt oder ein Gatekeeper kann während eines bestehenden Gesprächs mit der Nachricht Bandwidth_Request (BRQ) eine Änderung der Bandbreite beantragen. Die Antwort kann entweder bestätigend, Bandwidth_Confirm (BCF), oder ablehnend, Bandwidth_Reject (BRJ), sein.
- **Statusabfrage:** Eine Anfrage durch die Nachricht Information_Request (IRQ) kann sowohl von einem Endpunkt als auch von einem Gatekeeper gestartet werden. Als Reaktion erfolgt die Antwortnachricht Information_Request_Response (IRR), in welcher Adressen, Referenzwerte und die Bandbreite für die Verbindung mitgeschickt werden.
- **Verbindungsende:** Mit der Nachricht Disengage_Request (DRQ) kann entweder ein Endpunkt oder ein Gatekeeper signalisieren, dass die Verbindung

101 Dies gilt sowohl für die Entgegennahme eingehender Anrufe als auch für das Anrufen anderer Anwender.

aufgelöst wird. Die Bestätigung erfolgt durch Disengage_Confirm (DCF). Falls ein Endpunkt bei einem Gatekeeper nicht registriert war und dennoch eine DRQ empfängt, sendet er die Nachricht Disengage_Reject (DRJ) an den Endpunkt zurück.

- **Access Token:** Einige RAS-Nachrichten und die Setup-Nachricht können Access Tokens (Strings) enthalten, welche zwei Funktionen erfüllen. Zunächst können Endpunkte ihre Transport- und Aliasadresse verbergen, indem sie einem anderen Endpunkt einen einzigen Access Token als Kontaktadresse mitteilen. Der Gatekeeper, der den Access Token aufgrund der Registrierung zuordnen kann, dient dabei als Vermittler zwischen den Endpunkten. Zweitens wird durch die Verwendung von Access Tokens sichergestellt, dass Verbindungen durch H.323-Einheiten korrekt weitergeleitet werden. Ein Gateway kann z.B. anhand eines Access Token feststellen, ob ein Endpunkt die Erlaubnis zur Nutzung von Ressourcen besitzt.

H.225.0-Signalisierungskanal

Der Signalisierungskanal ist ein zuverlässiger Kanal und transportiert H.225.0-Nachrichten für die Verbindungskontrolle. Hierbei wird das Q.931-Protokoll[102] verwendet, um eine Verbindung zwischen zwei Terminals aufzubauen. In Netzwerken ohne Gatekeeper werden die Nachrichten direkt zwischen den beteiligten Endpunkten über die Transportadressen des Gesprächssignalisierungskanals ausgetauscht. Ist ein Gatekeeper vorhanden, teilt dieser nach erfolgter Zonenzugangsanfrage ARQ in seiner Antwort ACF dem Endpunkt mit, ob die Signalisierung direkt mit dem anderen Endpunkt stattfinden soll oder über den Gatekeeper geleitet wird. Die H.225.0-Nachrichten werden dann entweder an die Transportadresse des Gesprächssignalisierungskanals des entsprechenden Endpunkts oder des Gatekeepers geschickt.

Gesprächsidentifizierung

Alle Nachrichten über den RAS-Kanal und über den Signalisierungskanal beinhalten einen jeweils unterschiedlichen Call_Reference_Value (CRV), damit die Nachrichten einer bestimmten Verbindung eindeutig zugeordnet werden können. Alle in eine Verbindung einbezogenen Einheiten verwenden den gleichen CRV für den entsprechenden Kanal. Bei einer neuen Verbindung, wenn beispielsweise ein Gesprächspartner zu einer bestehenden Verbindung hinzukommt, wird ein neuer CRV sowohl für den RAS-Kanal als auch für den Gesprächssignalisierungskanal zugeordnet. Im Unterschied zum CRV enthalten alle Nachrichten unabhängig vom Kanal zusätzlich eine Call_ID. Diese dient der Zuordnung aller Nachrichten über den RAS- und den Signalisierungskanal innerhalb einer Verbindung. In Konferenzen enthalten alle Nachrichten eine Conference_ID (CID), um Nachrichten den entsprechenden Konferenzen zuordnen zu können. Beim Nachrichtenaustausch innerhalb von

102 Für weitere Informationen siehe [Q.931(98)].

Konferenzen kann mit dem Feld Conference_Goal der Zweck der Verbindung durch die Nachrichten Create, Join, Invite, Capability_Negotiation und Call_Independent_Supplementary_Service mitgeteilt werden.

6.4.3 Kommunikationsphasen

Zum Aufbau der Verbindung zwischen den Endteilnehmern wird zunächst eine Verbindung zwischen den betreffenden Gateways initiiert. Hierbei können über den Call_Setup hinaus weitere Verbindungsparameter wie die Wahl des Kompressionsalgorithmus ausgetauscht werden. Der Kommunikationsablauf einer H.323-Sitzung lässt sich dabei in folgende Phasen einteilen:

- **A:** Verbindungseinrichtung
- **B:** Austausch von H.245-Kontrollnachrichten
- **C:** Öffnung logischer Kanäle für audiovisuelle Kommunikation
- **D:** Gesprächsdienste während der Verbindung
- **E:** Gesprächsbeendigung

Phase A: Verbindungseinrichtung

Die Verbindungseinrichtung erfolgt mit den Q.931-Nachrichten der H.225.0-Empfehlung. Für die Synchronisation des Verbindungsaufbaus zweier Endpunkte gibt es keinen Mechanismus, das heißt, beide Endpunkte können zur gleichen Zeit eine Setup-Nachricht senden. Ein entsprechender Regelmechanismus bleibt den Anwendungsentwicklern überlassen.

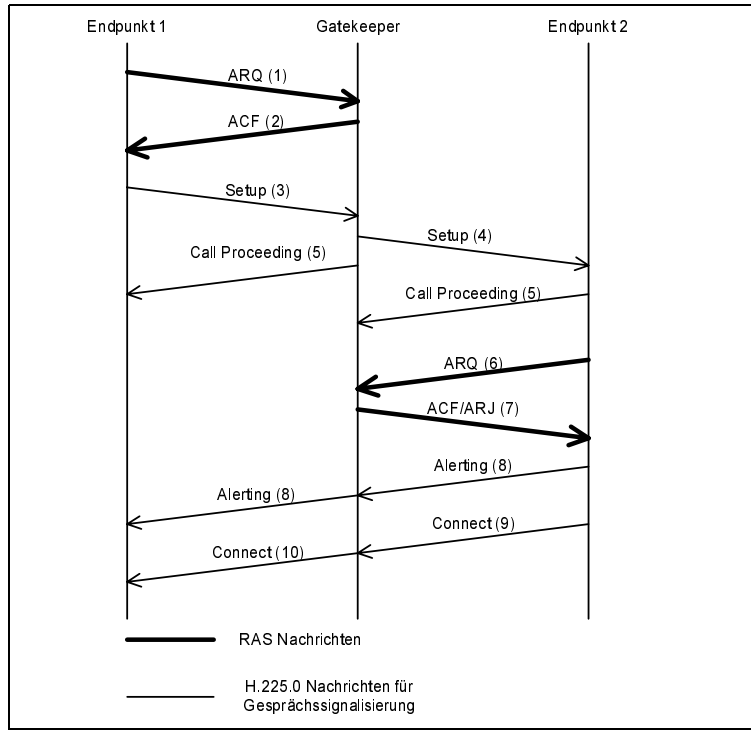
Tab. 6.7 Verbindungsaufbau mit Gatekeeper

Registrierung	Verbindung
Beide Endpunkte sind bei dem gleichen Gatekeeper registriert.	• Direkte Verbindung zwischen den Endpunkten • Routing über Gatekeeper
Nur der Endpunkt 1 ist bei dem Gatekeeper registriert.	
Nur Endpunkt 2 ist bei dem Gatekeeper registriert.	
Endpunkt 1 ist bei Gatekeeper 1 und Endpunkt 2 bei Gatekeeper 2 registriert.	• Direkte Verbindung zwischen den Endpunkten • Routing über Gatekeeper 1 • Routing über Gatekeeper 2 • Routing über beide Gatekeeper

Sind zwei Endpunkte bei keinem Gatekeeper registriert, erfolgt die Kommunikation direkt zwischen den Endpunkten. Der Anrufer am Endpunkt 1 sendet dann eine Setup-Nachricht unmittelbar nach der Einrichtung einer TCP-Verbindung an den bekannten TSAP des Signalisierungskanals des Angerufenen am Endpunkt 2, um eine Gesprächsverbindung aufzubauen. Die Setup-Nach-

richt enthält den Namen des Anrufers und dessen IP-Adresse. Nach dem Erhalt der Setup-Nachricht antwortet Endpunkt 2 sofort entweder mit Alerting, Connect, Call Proceeding oder Release Complete. Wenn der Angerufene mit Alerting geantwortet hat, bleiben ihm 3 Minuten für die Annahme des Gesprächs durch das Senden der Nachricht Connect oder dessen Ablehnung durch Release Complete. Die Antwort Connect enthält die Adresse des H.245-Kontrollkanals von Endpunkt 2. Bei Beteiligung eines Gatekeepers am Verbindungsaufbau gibt es eine Vielzahl von Konstellationen, die in Tab. 6.8 dargestellt sind, wobei Endpunkt 1 als Anrufer eine Verbindung zu Endpunkt 2 aufbauen möchte. Exemplarisch wird der Verbindungsaufbau für zwei Endpunkte dargestellt, die beim gleichen Gatekeeper registriert sind und der das Routing übernimmt.

Abb. 6.28 Routing über Gatekeeper



Falls ein externes Terminal eine Verbindung zu einem Endpunkt in einem Netzwerk über ein Gateway aufbauen möchte, läuft der Verbindungsaufbau zwischen Endpunkt und Gateway genauso wie beim direkten Verbindungsaufbau zwischen zwei Endpunkten ab. Ein Gateway erkennt die Nachrichten externer Terminals nach Tab. 6.9.

Externes Terminal	Erkennen von
H.310 im H.321-Modus, H.320-, H.321-, H.322-Netzwerk	SBE-Nummern
H.310- und H.324-Netzwerk	H.245-Nachricht User_Input_Indication optional DTMF-Nummern

Tab. 6.8 Erkennung externer Anrufsignale

Falls ein Endpunkt im Netzwerk ein externes Terminal anrufen möchte, läuft der Verbindungsaufbau zwischen Endpunkt und Gateway genauso wie beim direkten Verbindungsaufbau zwischen zwei Endpunkten ab. Das Gateway empfängt die E.164- oder Party_Number-Adresse mit der Setup-Nachricht des Endpunkts. Mit dieser Adresse versucht das Gateway, die Verbindung herzustellen. [H.323(99)]

Phase B: Austausch von H.245-Kontrollnachrichten

Nach erfolgtem Nachrichtenaustausch für den Verbindungsaufbau wird über eine neue TCP-Verbindung ein H.245-Kontrollkanal zwischen den Endpunkten eingerichtet. Über diesen werden zunächst die technischen Eigenschaften mit der H.245-Nachricht Terminal_Capability_Set ausgetauscht. Daran schließt sich die Master-/Slave-Bestimmung[103] an. Falls eine der beiden genannten Prozeduren erfolglos ist, sollten diese mindestens zweimal wiederholt werden, bevor ein Endpunkt die Verbindung beendet. Hat ein Terminal nicht die Möglichkeit des H.245-Kontrollkanals, wird die Verbindung abgebrochen. [PAKO99]

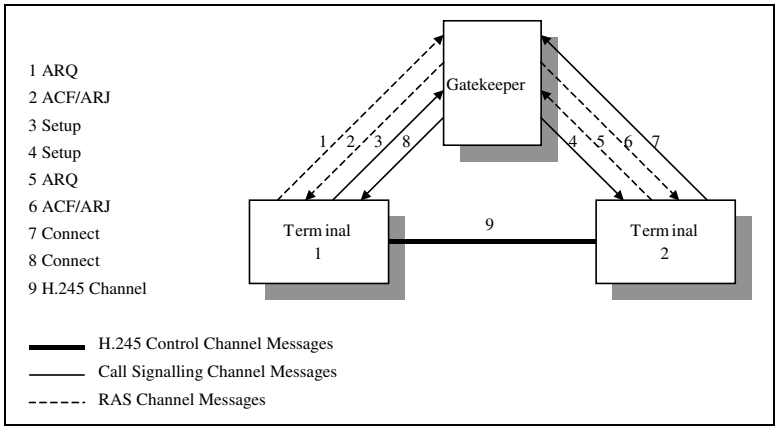


Abb. 6.29 Direkter H.245-Kontrollkanal zwischen den Terminals[a]

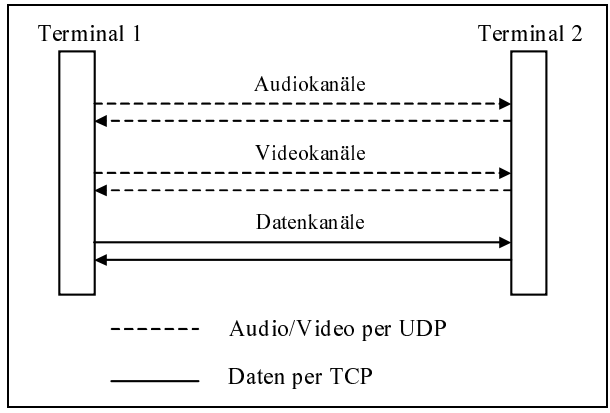
a. Indirekter Verbindungsaufbau über einen Gatekeeper

103 Es werden Informationen wie H.323.Einheiten (Terminal, Gatekeeper etc.) ausgetauscht und festgelegt, wer in der Kommunikation Master und wer Slave ist.

Phase C: Öffnung logischer Kanäle für audiovisuelle Kommunikation

Es folgen H.245-Prozeduren mit TSAP, um logische Kanäle für die unterschiedlichen Informationsströme (Audio, Video und Daten) zu öffnen. Der Anrufer sendet die H.245-Nachrichten Open_Logical_Channel, woraufhin der Angerufene jeweils mit Open_Logical_Channel_Ack antwortet, in welcher er die Transportadresse eines logischen Kanals mitteilt.

Abb. 6.30 Audio-, Video- und Datenkanäle zwischen den Terminals



Für die Audioübertragung zwischen zwei Endpunkten werden jeweils zwei unidirektionale logische Kanäle für den Austausch der Sprachsignale mit RTP und ein bidirektionaler logischer Kanal für die Kontrolle der Audioübertragung mit RTCP eingerichtet. Falls Video von den beteiligten Endpunkten unterstützt wird, werden wie bei Audio zwei unidirektionale Kanäle für den Austausch von Videosignalen mit RTP und ein bidirektionaler Kanal zur Steuerung der Videoströme mit RTCP eingerichtet. Für die Datenübertragung werden entweder ein oder zwei unidirektionale[104] Datenkanäle geöffnet, die jedoch keinen zusätzlichen Steuerkanal benötigen. [DOMM98]

Die Audiodaten bestehen weiterhin aus digitalisierter und komprimierter Sprache. Jedes Terminal muss dabei den G.711-Standard unterstützen; andere Standards sind optional. Während die Videoübertragung ebenfalls optional ist, muss jedes H.323-Terminal, welches diese verwendet, den H.261-Standard unterstützen; H.263 ist ebenfalls optional. H.261 wird dabei in Verbindungen von nx64Kbit/s verwendet. H.263 ist abwärtskompatibel zu H.261 und bietet eine verbesserte Bildqualität sowie eine Optimierung für geringe Übertragungsraten. Die Verwendung von Datenkonferenzen innerhalb einer H.323-Kommunikationsverbindung ist ebenfalls optional. Werden sie unterstützt, so sind damit unterschiedliche Arten der Zusammenarbeit möglich, wie beispielsweise ein gemeinsames Arbeiten an Dokumenten mittels Application Sharing nach T.120. [T.120(98)]

104 Für den bidirektionalen Betrieb

Phase D: Gesprächsdienste

Während einer aktiven Verbindung stehen den Kommunikationsteilnehmern verschiedene Dienste wie Bandbreitenänderung oder Statusabfragen über den RAS-Kanal zur Verfügung. Darüber hinaus können bestehende Punkt-zu-Punkt-Verbindungen über den Gesprächssignalisierungskanal zu Ad-hoc-Konferenzen erweitert werden. Voraussetzung ist, dass ein MC entweder in einem Endpunkt oder einem Gatekeeper vorhanden ist. Die Einleitung oder Erweiterung einer Konferenz geschieht durch das Einfügen des Elements Conference_Goal=<value> in der Setup-Nachricht beim Verbindungsaufbau über den Gesprächssignalisierungskanal. Zunächst bauen zwei Endpunkte eine Punkt-zu-Punkt-Konferenz auf, indem Conference_Goal=create gesetzt wird. Die Erweiterung zu einer Multipoint-Konferenz kann dabei auf zwei Arten geschehen. Entweder lädt ein Endpunkt über den MC einen weiteren Endpunkt zur Konferenzteilnahme mit Conference_Goal=invite ein oder ein neu hinzukommender Endpunkt ruft einen Endpunkt in der Konferenz an, indem er in der Setup-Nachricht das Element Conference_Goal=join einfügt. Die Anordnung der H.245-Kontrollkanäle hängt davon ab, ob die Endpunkte direkt oder über einen Gatekeeper kommunizieren.

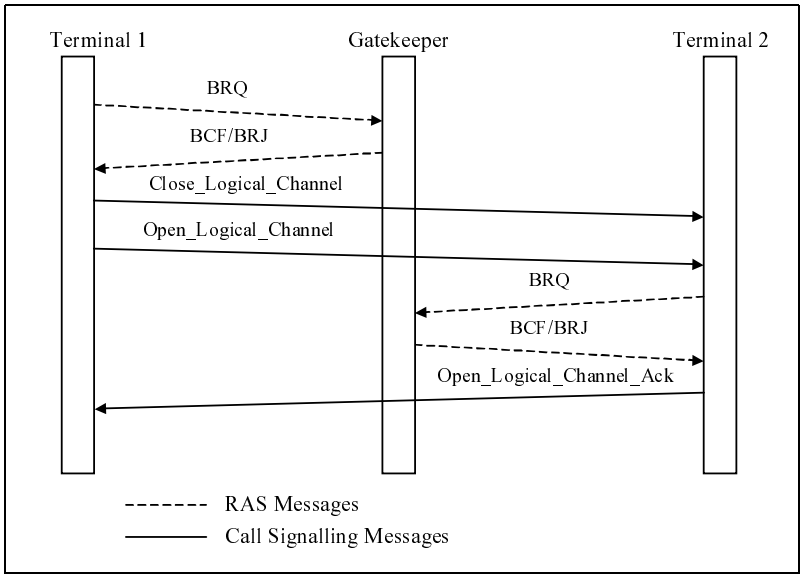


Abb. 6.31 Bandbreitenänderungen durch den Sender

Die Bandbreite wird anfänglich festgelegt und vom Gatekeeper überwacht. Die Terminals müssen in der Lage sein, die vereinbarte Bandbreite senden und empfangen zu können. Diese Bandbreite beinhaltet die Audio- und Videoströme, die wiederum jegliche RTP-Header, Netzwerk-Header und weiteren Overhead beinhalten. Zu jeder Zeit kann durch die Terminals oder den Gatekeeper die Bandbreite verändert werden. Ein Terminal kann ohne Signalisie-

rung an den Gatekeeper die Bandbreite verändern, sofern die Gesamtbandbreite nicht überschritten wird. Als Erstes wird eine Anfrage an den Gatekeeper gestellt, ob die neue Bandbreite akzeptiert werden kann, wie Abb. 6.31 verdeutlicht. Bei Annahme wird der logische Kanal geschlossen und ein neuer mit neuer Bandbreite geöffnet. Nun meldet Terminal 2 dem Gatekeeper die Annahme der neuen Bandbreite und sendet Terminal 1 eine Quittierungsmeldung zurück. Der Gatekeeper hat die Möglichkeit, mit den Meldungen Information_Request (IRQ) und Information_Request_Response (IRR) den Status der Terminals abzufragen, um so eine Fehlerdiagnose durchzuführen [PAKO99]

Phase E: Gesprächsbeendigung

Jeder Endpunkt kann ein Gespräch gemäß der folgenden Prozedur beenden:

1. Beendigung der Übertragung von Video am Ende eines kompletten Bildes und Schließen aller logischen Kanäle für Video
2. Beendigung der Übertragung von Daten und Schließen aller logischen Kanäle für Daten
3. Beendigung der Übertragung von Audio und Schließen aller logischen Kanäle für Audio
4. Senden der Nachricht End_Session_Command über den H.245-Kontrollkanal
5. Abwarten der gleichen Nachricht als Antwort des anderen Endpunkts und anschließendes Schließen des H.245-Kontrollkanals
6. Ist der Gesprächssignalisierungskanal geöffnet, wird dieser nach Versenden der Nachricht Release_Complete geschlossen.

Empfängt ein Endpunkt die H.245-Nachricht End_Session_Command, ohne diese vorher gesendet zu haben, beendet er das Gespräch gemäß der oberen Prozedur ohne Schritt 5. Das Ende eines Gesprächs ist nicht mit dem einer Konferenz gleichzusetzen. Letzteres wird mit der H.245-Nachricht Drop_Conference eingeleitet. Dabei müssen die Endpunkte warten, bis der MC die Schritte 1-6 abgearbeitet hat. In Netzwerken mit Gatekeeper müssen diese über das Ende von Gesprächen benachrichtigt werden, um über frei werdende Bandbreite informiert zu sein. Nachdem jeder Endpunkt die oberen Schritte 1-6 abgearbeitet hat, schickt er die Nachricht DRQ an seinen Gatekeeper, der daraufhin mit DCF antwortet. Ab diesem Zeitpunkt ist das Gespräch beendet. Ein Gatekeeper kann ein Gespräch auch selbst beenden, indem er DRQ an einen Endpunkt schickt. Der Endpunkt beginnt anschließend die Schritte 1-6 und beendet mit dem Versenden von DCF an den Gatekeeper das Gespräch. Handelt es sich um das Schließen einer Multipoint-Konferenz, sendet der Gatekeeper an jeden der Endpunkte die Nachricht DRQ.[105] [H.323(99)]

105 Detailliertere Beschreibungen der Signalisierungen und des Datenaustauschs sind in den jeweiligen Standards nachzuschlagen.

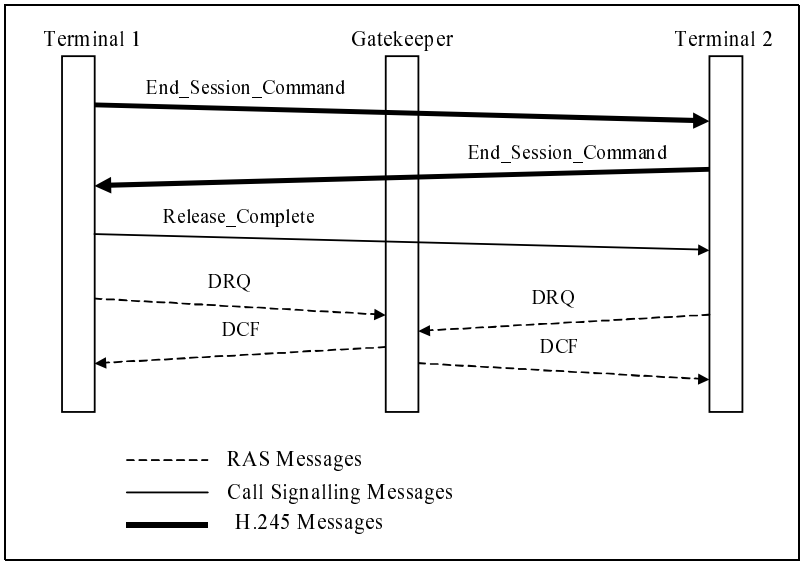


Abb. 6.32 Verbindungsabbau, der durch ein Terminal initiiert wird

6.4.4 Betrachtung der Kommunikationsstrecke

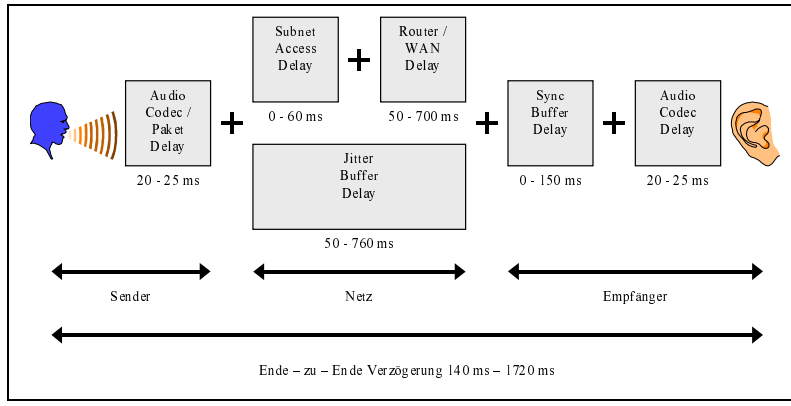
Der H.323-Standard verwendet sowohl gesicherte als auch ungesicherte Verbindungen. Kontroll- und Dateninformationen erfordern eine sichere Übertragung, um auszuschließen, dass Pakete verloren gehen oder in einer falschen Reihenfolge empfangen werden. Audio- und Videoinformationen verlieren ihren Wert, wenn sie nicht zum genau richtigen Zeitpunkt empfangen werden. Wenn ein Paket verspätet empfangen wird, hat es keine Bedeutung mehr und darf nicht mehr in den gerade abgespielten Audiodatenstrom eingefügt werden. Daher werden für die Übertragung von Audio und Video ungesicherte Verbindungen verwendet, die effizienter sind.

In IP-basierten Netzen wird für gesicherte Verbindungen das Protokoll TCP verwendet, welches eine fehlerfreie Übertragung in der richtigen Reihenfolge garantiert, aber dafür Verzögerungen bewirkt und einen geringeren Durchsatz hat. H.323 verwendet TCP-Verbindungen für das Signalisierungsprotokoll Q.931, für die Datenübertragung T.120, und die Verbindungssteuerung H.245. Für ungesicherte Verbindungen wird innerhalb von IP das UDP-Protokoll verwendet. Ungesicherte Übertragung bedeutet verbindungslose Übertragung ohne Überwachung, ob und wann ein abgesendetes Paket empfangen wird. H.323 verwendet UDP für die Übertragung von Audio und Video sowie für das RAS-Protokoll.

In Konferenzen mit mehreren Audio- und Videodatenströmen wird für die ungesicherte Übertragung über UDP das IP-Multicast und das Real-time Transport Protocol (RTP) verwendet. IP-Multicast wird zur ungesicherten Übertragung gleichzeitig an mehrere Empfänger eingesetzt. RTP verwendet IP

Multicast und wurde auf die Belange der Übertragung von Audio und Video über das Internet zugeschnitten. Unter Verwendung von Zwischenpuffern, Zeitstempeln und Folgenummern ermöglicht RTP es der empfangenen Station, fehlende, doppelte oder in falscher Reihenfolge empfangene Pakete zu erkennen und in geeigneter Weise den Empfangsstrom zu korrigieren. Des Weiteren kann durch RTP die Synchronisation zwischen den Audio-, Video- und Dateninformationen hergestellt und die kontinuierliche Ausgabe ermöglicht werden. Abb. 6.33 zeigt die großen Verzögerungen, die Ende-zu-Ende auftreten können. Diese sind für eine Echtzeitkommunikation nicht tolerierbar und müssen durch QoS-Mechanismen und ausreichend Bandbreite so klein wie möglich gehalten werden. Aus diesem Grund wird an dieser Stelle der Einsatz von RSVP[106] empfohlen, um bestimmte Bandbreiten reservieren zu können. RSVP ist nicht Bestandteil des H.323-Standards.

Abb. 6.33 Netzgesamtverzögerung auf dem Kommunikationspfad



IP lässt sich unabhängig von der Schicht-2-Technologie einsetzen. Aus diesem Grund ist VoIP auch nicht abhängig von Ebene 2 und kann über SDH, POS, ATM, GE etc. genutzt werden. Trotzdem ergeben sich durch die verbindungsorientierte Arbeitsweise von ATM Vorteile, weswegen hier die Übertragungsmöglichkeit von ATM mittels AAL-Typ1 exemplarisch untersucht wird, da hier ISDN transparent übertragen werden kann.

Übertragung über ATM-AAL1-Verbindungen

Die Anpassungsschicht vom Typ 1 (AAL-1) von ATM wurde mit dem Ziel entwickelt, Übertragung von Anwendungen mit konstanter Bitrate zu ermöglichen. Dabei werden verlorene und fehlerhafte Daten nicht wiederholt übertragen oder korrigiert. Die zu übertragenden Protokolldaten werden in 47-Byte-Blöcken (CS-PDU[107]) übertragen und mit einem 1-Byte-Header versehen. Der Header enthält Informationen über die Taktfrequenz des Sendebitstroms und

106 Resource Reservation Protocol
107 Convergence Sublayer – Packet Data Unit

wird als SAR-PDU[108] im ATM-Informationsfeld übertragen. Zur Übermittlung der Taktfrequenz kann die Synchronous Residual Time Stamp (SRTS) verwendet werden. Damit kann die Taktinformation am Empfänger mit Hilfe des CS-PDU-Header wieder synchron herausgefiltert werden. Abb. 6.34 zeigt das Zusammenspiel von H.323-Zonen und einem ATM-Netz.

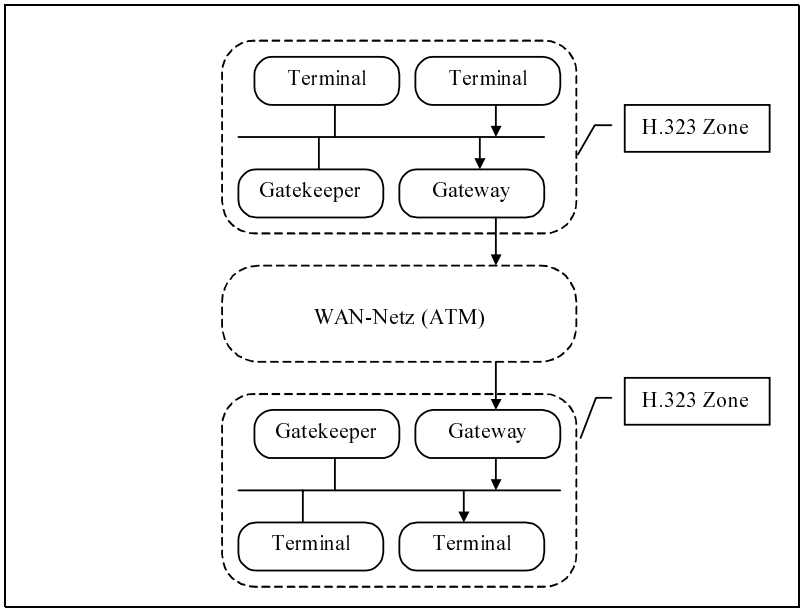


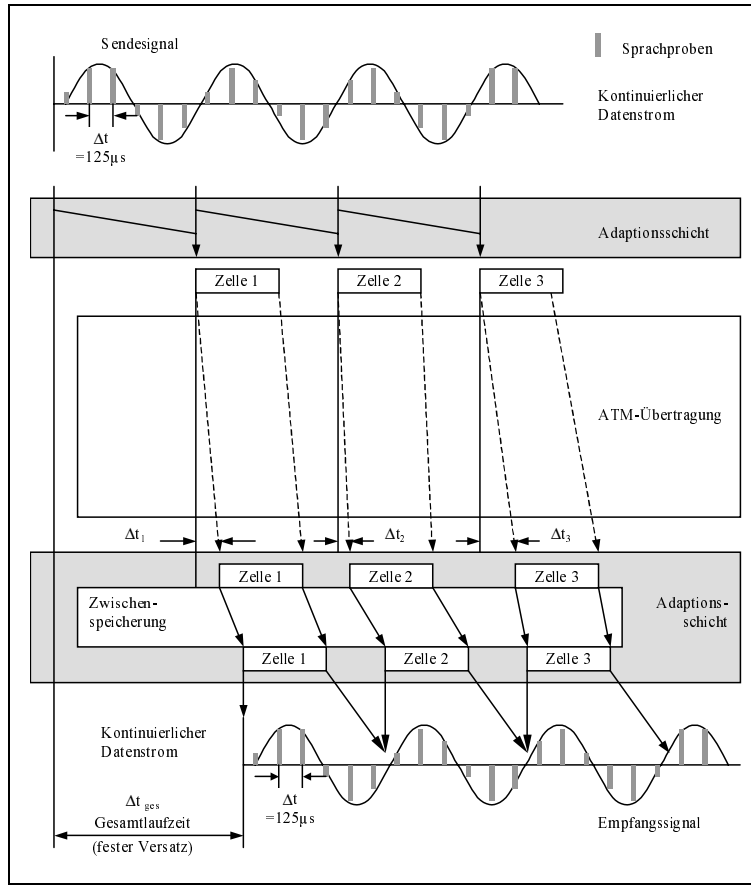
Abb. 6.34 Signalverlauf von IP-Sprachdaten über ein ATM-Netz

Die auf dem Weg durch das ATM-Netz entstehenden Verzögerungsschwankungen zwischen den Zellen einer Verbindung[109] werden mit Hilfe eines elastischen Puffers ausgeglichen. Dabei werden die ersten Dateneinheiten der Verbindung vor dem Ausspielen so lange im Pufferspeicher gehalten, bis ein gewisser Füllgrad erreicht ist. Dieser Füllgrad ist so dimensioniert, dass selbst bei der maximal erwarteten Verzögerung immer eine Dateneinheit zum Ausspielen verfügbar ist. Falls der Puffer trotzdem leer läuft, müssen Leerinformationen mit dem Wert 1 ausgespielt werden, um die Synchronisation des Empfängers oberhalb des AAL zu erhalten. Wenn der Füllstand des Puffers zu groß wird, müssen eventuell Dateneinheiten verworfen werden. Der Füllgrad verursacht einen festen Versatz beim Empfänger. Dieser muss natürlich in einem für den Dienst akzeptablen Rahmen[110] bleiben. Bei großen Laufzeiten[111] muss eine Echounterdrückung beim Empfänger vorgesehen werden. Zellverzöge-

108 Segmentation and Reassembly – Packet Data Unit
109 Cell Delay Variation
110 Bei Sprache < 25 ms in einer Übertragungsrichtung
111 Ab 25 ms: 15 ms im öffentlichen Netz plus jeweils 5 ms im Zugangsnetz

rungsschwankungen innerhalb einer Verbindung[112] sind bei der Dimensionierung von Netzressourcen[113] zur Bedienung einer ATM-Verbindung zu berücksichtigen. Traffic Shaping lässt sich einsetzen, um Zellverzögerungsschwankungen innerhalb eines Zellstroms zu verringern.

Abb. 6.35 Laufzeitausgleich bei Sprachübertragung



Die an der SAR-Subschicht weitergegebenen CS-PDU werden mit einer 3-Bit-Folgenummer versehen[114]. Anhand dieser Folgenummer und anhand der Ergebnisse der Sicherungsprozedur für die Folgenummer in der SAR-Subschicht kann die empfangene CS-Subschicht Zellverluste und -einfügungen[115] erkennen. Fälschlich eingefügte Zellen werden verworfen, Zellverluste können durch Einfügen von Dummy-Informationen behoben werden.

112 Cell Delay Variation
113 Übertragungskapazität bzw. Speicher
114 Modulo 8
115 Diese können durch Mehrfachfehler im Zellenkopf entstehen, wenn diese vom HEC-Mechanismus nicht erkannt werden und der VIP/VCI betroffen ist.

Strukturierte und unstrukturierte Übermittlung

Bei der strukturierten Übermittlung verschiedener Signale wird die Struktur des ankommenden isochronen (Multiplex-)Signals analysiert und einzelne Kanäle werden getrennt behandelt, das heißt, in unterschiedliche ATM-Verbindungen verpackt und an unterschiedliche Ziele geschickt. Dadurch können beispielsweise in einem 45-Mbit/s-Signal (DS3) 1,5-Mbit/s-Signale (DS1) und 64-Kbit/s-Kanäle (DS0) getrennt behandelt werden. Im ISDN wurden Übermittlungsdienste definiert, für die mehrere 64-Kbit/s-Kanäle zu einer höheren Übertragungskanalrate zusammengefasst werden können, um z.B. Video- oder Audiosignale mit besserer Qualität übertragen zu können. Diese Übermittlungsdienste können durch einen n-x-64-Kbit/s-Dienst auf ATM-Basis emuliert werden. Dabei werden *N* beliebige Zeitschlitze[116] aus einem PCM-Rahmen zusammengefasst und in die ATM-Zellen einer Verbindung eingekapselt. Aus den Zeitschlitzen eines PCM-Rahmens können dabei völlig flexibel *M* Gruppen aus Zeitschlitzen gebildet werden, die in entsprechend vielen ATM-Verbindungen abgebildet und damit zu unterschiedlichen Zielen weitervermittelt werden können. Durch die Verwendung des für den AAL-Typ-1 definierten strukturierten Datentransfers wird nicht nur ein transparenter digitaler Kanal mit einer Kapazität von n x 64 Kbit/s zur Verfügung gestellt, sondern der Empfänger kann bei Bedarf die empfangenen Daten wieder den Ursprungszeitschlitzen zuordnen und damit wieder in einzelne 64-Kbit/s-Kanäle zerlegen.

Eine Einsatzmöglichkeit dieser Dienste ist die Sprachübermittlung zwischen zwei Standorten, wobei die Abtastproben mehrere Sprachkanäle in eine gemeinsame ATM-Verbindung paketiert, die beim Empfänger wieder getrennt werden. Damit kann erreicht werden, dass die bei der Sprachpaketierung auftretende Verzögerung reduziert wird. Bei solchen Anwendungen wird die Zeichengabeinformation nicht notwendigerweise über einen zentralen, meldungsorientierten Zeichengabekanal übermittelt, sondern es können auch bestimmte Bits einzelnen Kanälen zur Übermittlung ihrer Zeichengabeinformation fest zugeordnet werden[117]. In diesem Fall müssen die entsprechenden Bits getrennt und so in die ATM-Zellen der unterschiedlichen Verbindungen eingefügt werden, sodass der Empfänger die Zeichengabebits den Kanälen wieder richtig zuordnen kann. Entsprechende Erweiterungen der Dienste werden in der Spezifikation Circuit Emulation Service (CES) des ATM-Forums ebenfalls definiert.

Wenn das ATM-Netz nur als reine Transportinfrastruktur ohne Vermittlungsfunktion für Schmalbandverkehr eingesetzt werden soll, können mit der Spezifikation Circuit Emulation Service (CES) Datenraten von 1,544 Mbit/s (DS1) oder 2,048 Mbit/s (E1) mit einer physikalischen Schicht nach der ITU-T Empfehlung G.703 transparent über ein ATM-Netz hinweg transportiert wer-

116 $n \leq 31$ für PCM 30/32

117 Channel Associated Signalling (CAS)

den. Dabei werden am Eingang des ATM-Netzes die aufeinander folgenden Bits in die Zellen einer einzigen ATM-Verbindung paketiert, ohne auf eine eventuell vorhandene Rahmenstruktur Rücksicht zu nehmen[118]. Am Ausgang des ATM-Netzes werden die variablen Verzögerungen ausgeglichen und die Bits werden wieder an ein entsprechendes Übertragungssystem übergeben. Da die Rahmenstruktur nicht analysiert wird, gibt es keinerlei Möglichkeit, die eventuell in dem Signal enthaltenen Zeichengabekanäle zu identifizieren und zu bearbeiten. In der CES-Spezifikation des ATM-Forums ist auch ein entsprechender Dienst auf der Basis der plesiochronen Datenraten 45 Mbit/s (DS3) bzw. 34 Mbit/s (E3) definiert. [ATMF97c]

Anwendung finden diese Dienste beispielsweise zur Vermaschung von Firmenstandorten oder Schmalband-Vermittlungsstellen sowie zur Verbindung von TDM[119]-Multiplexern. Das ATM-Netz wird hierbei als rein übertragungstechnische Anordnung eingesetzt und ersetzt getrennte Übertragungssysteme oder Mietleitungen für unterschiedliche Dienste. Aufgrund der weiten Verbreitung von Mietleitungen, vor allem im Bereich von 2 Mbit/s, und der relativ einfachen Emulationsfunktionen im ATM-Netz ist dies einer der attraktivsten Bereiche für eine Netzkonsolidierung. Dies gilt insbesondere, da die Mietleitungsnetze oft relativ alte Geräte beinhalten, sodass allein durch das verbesserte Netzmanagement die Investitionen gerechtfertigt sein können.

Fehlerkorrektur

Das Fehlerkorrekturverfahren[120], welches für die empfindliche Video-/Audio-Übertragung zusätzlich implementiert wurde, besteht aus einer Zusammensetzung von Reed-Solomon-Code und Byte-Umschichtung. Dieser Code kann aus einem Block von jeweils 128 Byte bis zu zwei gefälschte Byte korrigieren. Wenn die Position der gefälschten Bytes innerhalb des Blocks bekannt ist, können sogar vier gefälschte Bytes korrigiert werden.

Vor dem Senden werden aufeinander folgende Blocks reihenweise in einen Matrixspeicher mit 128 Spalten mit jeweils 1 Byte und 47 Reihen eingeschrieben. Die SAR-PDU-Nutzlast wird dann durch spaltenweises Auslesen generiert. Dadurch wird erreicht, dass beim Verlust einer Zelle in jedem Block nur ein Byte gefälscht wird, dessen Position außerdem bekannt ist, und eine Korrektur ist möglich. Durch den zusätzlichen Schutz der Kodierung ist allerdings ein weiterer Overhead entstanden, der durch den RS-Code von 4 Byte allerdings nur 3,1%[121] pro 124 Informationsbyte beträgt. Auch die nachfolgende Zellenverzögerung von 128 Byte, die sich durch das vollständige Ausfüllen einer CS-PDU ergibt, erlaubt noch eine effektive Umsetzung der Datenströme. Das

118 Unstrukturierter Datentransfer
119 Time Division Multiplexing
120 Forward Error Correction (FEC)
121 4 Bit zu 128 Bit

Generatorpolynom, das der Reed-Solomon-Code zur Berechnung der 124 Datenbyte benötigt[122], ist dabei folgendermaßen definiert:

$$G(x) = (x - \alpha^{120}) \cdot (x - \alpha^{121}) \cdot (x - \alpha^{122}) \cdot (x - \alpha^{123})$$

mit $\alpha^2 = x^8 + x^7 + x^2 + x + 1$

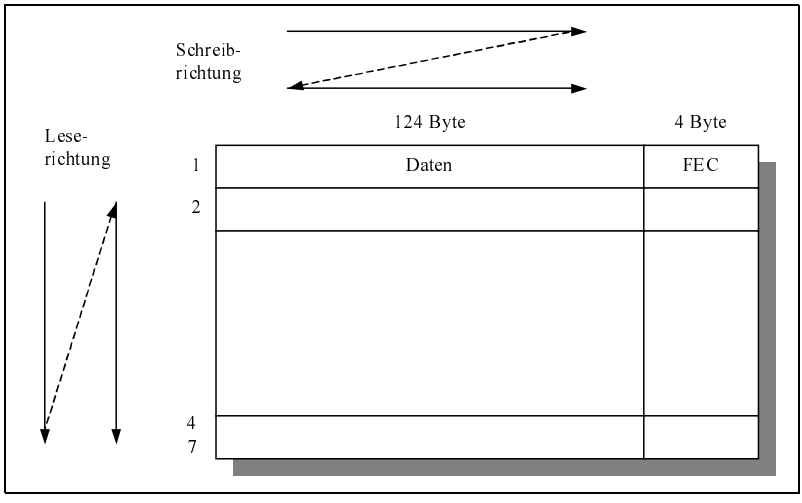


Abb. 6.36 Struktur des Matrixspeichers

IP-ATM Gateway

Unter einem Gateway versteht man Hard- und Software, um verschiedene Netze miteinander zu verbinden oder an andere Netze durch Protokollumsetzung anzuschließen. Ein Gateway hat die Aufgabe, Nachrichten von einem Rechnernetz in ein anderes zu übermitteln, wofür vor allem die Übersetzung der Kommunikationsprotokolle notwendig ist; es kann also auch als Protokollkonverter betrachtet werden. Dies bezieht sich auch auf die Verknüpfung von nicht normkonformen Netzen wie IP, ISDN, ATM usw.

Ein Gateway ist jeweils auf der kleinsten gemeinsamen Schicht der miteinander zu verbindenden Netze angesiedelt. Das Gateway ist in der Lage, beide Protokolle zu bearbeiten, und ist in beiden Welten ein adressierbarer Netzknoten. Die vollständige Umwandlung beinhaltet die Umsetzung von Adressen, Formaten, Konvertierung der Kodierung, Zwischenpufferung der Datenpakete, Paketbestätigung sowie Flusskontrolle und Geschwindigkeitsanpassung.

Ein Gateway ermöglicht aufgrund der vollständigen Bearbeitung aller Kommunikationsschichten für die verbundenen Protokollwelten oft eine höhere Funktionalität hinsichtlich Terminal-Emulation, Grafikfähigkeit, Programm-zu-Programm-Kommunikation, Dateitransfer und Anzahl parallel

122 4 Byte ist der RS-Kode selbst lang, weshalb man auf insgesamt 128 Byte kommt.

möglicher Sitzungen als gemeinsam benutzbare Standardprotokolle. Nachteilig ist die Beschränkung auf zwei verschiedene Protokolle, was bei dem Einsatz von *n*-Protokollen n x (n-1)/2 Gateways erfordert und damit einer exponentiellen Steigerung gleichkommt.

Abb. 6.37 Beispiel für ein IP-ATM Gateway

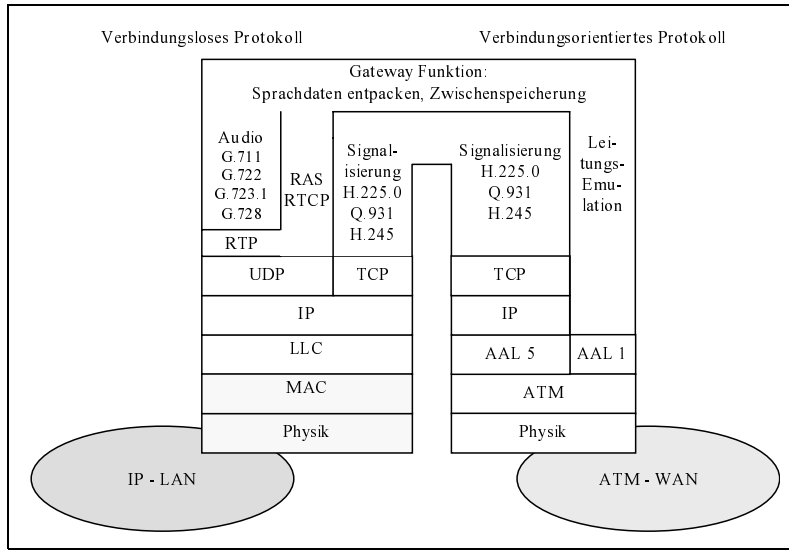
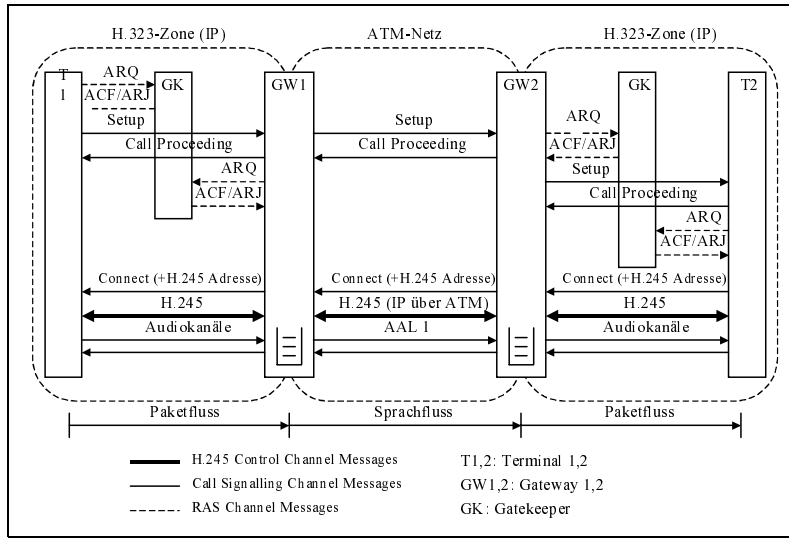


Abb. 6.38 Verbindungsaufbau und Signalfluss



Die Kommunikation mit dem Gatekeeper erfolgt über ungesicherte RAS-Meldungen. Der Verbindungsaufbau erfolgt über gesicherte TCP-Verbindungen zwischen den IP-Terminals und dem Gateway. Die Sprachdaten werden dann

als ungesicherte UDP-Pakete und über das ATM-Netz als AAL-1-Sprachstrom versandt. Ziel ist es, Sprachdaten im Vollduplex-Betrieb von Terminal 1 zu Terminal 2 zu übertragen.

Zum Verbindungsaufbau initiiert Terminal 1 das ARQ und ACF/ARJ mit dem Gatekeeper über eine RAS-Nachricht. Der Gatekeeper übermittelt die Call-Signalisierungskanal-Transportadresse von Gateway 1 in der ACF-Nachricht an Terminal 1 zurück. Terminal 1 sendet dann die Setup-Nachricht zu Gateway 1[123], wobei er die übermittelte Transportadresse benutzt. Wenn Gateway 1 das Setup annehmen kann, initiiert er ein ARQ und ACF mit dem Gatekeeper. In der ARJ-Nachricht meldet der Gatekeeper dem Gateway 1 eine Release_Complete-Mitteilung mit dem Terminal 1. Gateway 1 antwortet dem Terminal 1 mit der Call_Proceeding-Nachricht über die Bearbeitung des Setups. Das Signalisierungsprotokoll Q.931 wird im Gateway 1 nach Q.2931[124] umgesetzt, um den Verbindungsaufbau zum Gateway 2 zu bewerkstelligen. Nun kann Gateway 1 seine Setup-Nachricht an Gateway 2 senden. Gateway 2 wiederum antwortet mit der Call_Proceeding-Nachricht Gateway 1. In Gateway 2 findet nun eine Rücktransformation von Q.2931 nach Q.931 statt. Anschließend beginnt die gleiche Prozedur zwischen Gateway 2 und Terminal 2 wie zwischen Terminal 1 und Gateway 1. Es antwortet Terminal 2 mit der Connect-Nachricht, die wiederum eine H.245-Kontrollkanal-Transportadresse zur Verwendung in der H.245-Signalisierung enthält. Diese wird bis zum Terminal 1 zurückgemeldet, mit entsprechender Adressumsetzung in dem Gateways. Es ist möglich, das die Warnmeldung Alerting gesendet wird, wenn Terminal 2 nicht innerhalb von vier Sekunden antworten kann.

Sobald der Verbindungsaufbau abgeschlossen ist, werden die Terminals den H.245-Kontrollkanal öffnen. Hat ein Terminal dazu nicht die Möglichkeit, wird die Verbindung abgebrochen. Zwischen den Gateways wird der H.245-Kontrollkanal über eine AAL-1-Lösung umgesetzt. Über einen VC[125] mit AAL-Typ5 werden die H.245-Pakete über das ATM Netz getunnelt. Nun können Terminal 1 und Terminal 2 aushandeln, welche Möglichkeiten sie für die Übertragung vereinbaren. Nach dem Möglichkeitsaustausch werden die logischen Kanäle für die Audiodaten über die durch H.245 festgelegten dynamischen TSAPs geöffnet. Zwischen den Gateways werden AAL-1-Verbindungen aufgebaut. Für die Audiokanäle werden die Sprachdaten in UDP-Pakete übertragen. In Gateway 1 werden die Header entfernt und die reinen Sprachdaten in einen Pufferspeicher geschrieben.

Für die Übertragung von Audio über ATM wird ein kontinuierlicher Datenstrom über die Netzverbindungen vorausgesetzt. Wichtig sind Verzögerungsmanagement und das Timing zwischen Quelle (Gateway1) und Ziel

123 Direkter Verbindungsaufbau mit Gateway
124 Da Q.2931 eine Erweiterung von Q.931 ist, ist die Umsetzung unproblematisch.
125 Virtual Channel

(Gateway 2). AAL-Typ1 stellt eine konstante Bitrate[126] mit 64 Kbit/s (PCM-Format) zur Verfügung und sichert zudem die korrekte Sequenzfolge übertragener AAL 1-PDU. Dabei tritt ein Problem auf, da das Gateway 1 wissen muss, wie viele AAL1-Verbindungen zwischen den Gateways aufgebaut werden sollen. Dies wird aber zwischen den Endterminals ausgehandelt und ohne Kenntnis durch die Gateways vereinbart. Ein weiteres Problem ist, dass diese Sprachdaten komprimiert am Gateway ankommen können, je nachdem was zwischen den Terminals vereinbart wurde. Auf die Wahl der Kompressionsart haben die Gateways so keinen Einfluss, da der H.245-Kontrollkanal getunnelt übertragen wird. AAL-Typ1 erwartet die Sprachdaten aber im PCM-Format und würde sie mit einer Geschwindigkeit von 64 Kbit/s auslesen. Sind sie nicht konform, würden keine brauchbaren Daten mehr übertragen. Da dem Gateway die Wahl der Komprimierungsart nicht bekannt ist, kann er sie auch nicht in PCM-Daten zurückwandeln. Würden die Sprachdaten im PCM-Format zum Gateway geliefert werden, tritt ein weiteres Problem auf, da das Gateway 1 die Sprachdaten über das ungesicherte Protokoll UDP erhält. Dies stellt nicht sicher, dass im Pufferspeicher immer Sprachdaten zur Verfügung stehen, da die Übertragung und Reihenfolge nicht garantiert wird. Hierzu müsste man das RTP mit auswerten, um zu entscheiden, ob die gelieferten Pakete noch in den Sprachfluss eingefügt werden können oder ob fehlende Pakete durch Füllbits ersetzt werden müssen. Sonst besteht die Gefahr, das der Pufferspeicher keine Daten für den kontinuierlichen Sprachfluss von AAL-Typ1 zur Verfügung hat.

Es werden alle 5,875 ms 47-Byte-Sprachdaten für die Bildung einer AAL1-SAR-PDU[127] benötigt, um den Sprachfluss aufrecht zu erhalten. Die auf dem Weg durch das ATM-Netz entstehenden Verzögerungsschwankungen[128] zwischen den Zellen einer Verbindung werden mit Hilfe eines Pufferspeichers im Gateway 2 ausgeglichen. Dann kann der Sprachfluss wieder in einen Paketfluss zurückgewandelt werden. Zusätzlich müsste auch das RTP-Protokoll wieder neu generiert werden, da es Bestandteil von H.323 Terminals ist und dies in Terminal 2 wieder mit ausgewertet wird. Um die Probleme in den Griff zu bekommen, muss das H.245-Protokoll im Gateway 1 ausgewertet werden, um daraus Rückschlüsse auf den AAL1-Verbindungsaufbau zu ziehen. Hierzu müssen die H.245-Nachrichten Logical_Channel_Signalling_Messages im Gateway zur Übertragung zum Gateway 2 bereitgestellt werden und dem Gateway 1 zur Auswertung übergeben werden. Folgende Informationen sind in diesen Nachrichten enthalten:

- **Forward_Logical_Channel_Number:** zeigt die logische Kanalnummer des vorwärts gerichteten logischen Kanals an, der geöffnet werden soll.

126 Constant Bit Rate (CBR)
127 Unstrukturierte Nutzlast
128 Cell Delay Variation

- **Forward/Reverse_Logical_Channel_Parameters:** Beinhaltet Informationen über die Art des vorwärts und rückwärts gerichteten logischen Kanals (uni-directional oder bi-directional).
- **Port_Number:** Die ist ein Parameter, der zwischen Endnutzern ausgetauscht werden kann, um ein Input-/Output-Port oder eine höhere Schicht-Kanalnummer mit dem logischen Kanal zu verbinden.
- **Data_Type:** Zeigt die Art der Daten an, die auf dem logischen Kanal übertragen werden sollen. Null_Data signalisiert, dass keine Mediendaten transportiert werden sollen, sondern nur Adaptation-Layer-Informationen.

Aus diesen Informationen können nun im Gateway Rückschlüsse über die Anzahl der AAL1-Verbindungen gezogen werden. Anschließend kann entschieden werden, ob zusätzlich noch andere AAL-Vorbindungen für Video (AAL 2) oder Daten (AAL 5) geöffnet werden sollen. Generell sollte darauf geachtet werden, dass erst der AAL1-Verbindungsaufbau zwischen den Gateways abgeschlossen ist, bevor der Sprachdatentransfer beginnt. Damit auf jeden Fall sichergestellt ist, dass die Verbindung zwischen den Gateways vorhanden ist, wenn sie benötigt wird.

Des Weiteren ist die Wahl des Kodierverfahrens entscheidend, um eine Kommunikation herzustellen. Hierzu tauschen die Terminals während des Terminal_Capability_Set Tabellen aus, in denen die Komprimierungsalgorithmen aufgelistet sind, die die Terminals unterstützen. Weiterhin kann ein bevorzugter Modus übermittelt werden. Die endgültige Wahl wird in der Nachricht Receive_And_Transmit_Audio_Capability übertragen. Diese Nachricht ist für das Gateway entscheidend, um durch die Wahl des Kodierverfahrens eine Entscheidung zu treffen, ob die strukturierte oder die unstrukturierte Datenübertragung verwendet werden sollte. Für G.711, G.722 und G.726 wird die unstrukturierte Datenübertragung benutzt. Für G.723-1, G728 und G.729 ist die strukturierte Datenübertragung zu benutzen. Die Struktur wird benötigt, um die Daten im Empfänger wieder zu dekomprimieren. Die Sprachdaten werden hier in Blöcken übertragen, die so am Empfänger wieder benötigt werden.

Bei der strukturierten Übertragung ist zu beachten, dass die Sprachdaten in der Payload von den IP-Paketen so übernommen werden, da diese im Bereich der Strukturgröße von AAL-Typ1 (bis 93 Byte) liegen werden. Eine Analyse der IP-Pakete muss nicht vorgenommen werden, da durch den Sender[129] schon eine Entscheidung über die Paketgröße getroffen wurde. Das kann mehr als ein Sprachdatenblock sein, da die optimale Paketgröße zwischen 38 und 87 Byte liegt. Das IP-Paket könnte bei G.723-1 zwei bis drei Datenblöcke enthalten, bei G.729 vier bis acht Datenblöcke.

129 IP-Terminal

H.245 ist zusätzlich in der Lage, durch eine Round Trip Delay Determination die Ende-zu-Ende-Verzögerung zu messen. Dies kann den Terminal auch Auskunft über die volle Funktionsfähigkeit des anderen Terminals geben. Dadurch können Puffergrößen in den Terminals angepasst werden. Eine größere Verzögerung bedeutet, dass die Pufferspeicher im Ziel-Terminal größer dimensioniert werden müssen. Dies vergrößert wiederum die Gesamtverzögerung der Übertragung, verhindert aber ein Aussetzen der Sprachinformation. Die größere konstante Verzögerung hat keinen Einfluss auf die Sprachqualität, sondern verschlechtert mit zunehmender Länge immer stärker die Gesprächsqualität.

Die Lieferung der IP-Pakete bis zum Gateway sollte durch eine ausreichende Dimensionierung des IP-Netzes genügend gewährleistet sein. Zusätzlich wird generell davon ausgegangen, dass H.323 verwendet wird. Obwohl die Sprachdaten von einem unzuverlässigen UDP-Protokoll geliefert werden, ist davon auszugehen, dass die Pakete in ausreichender Geschwindigkeit und richtiger Reihenfolge am Gateway ankommen. Klarer Vorteil bei dieser Variante ist, dass der entstehende Header von RTP/UDP/IP nicht über das ATM-Netz mit übertragen werden braucht. Eine Anpassung an ISDN-Endgeräte ist ebenfalls sehr einfach, da der AAL1-Datenstrom direkt als B-Kanal von ISDN übernommen werden kann. [PAKO99]

H.323 über ATM Um die Vorteile für Echtzeitdaten über ATM auf AAL-Typ5 noch besser zu nutzen, kann die im Anhang C der Empfehlung H.323 dargestellte Methode benutzt werden. Generell kann H.323 durch CLIP[130], LANE[131] oder MPOA[132] über das ATM-Netzwerk transportiert werden. Solche Verfahren wurden bereits beschrieben. Das hier aufgezeigte Verfahren verbessert bzw. ergänzt die bisherigen IP-over-ATM-Methoden, weil die eigentlichen Daten (Audio oder Video) direkt mit AAL-Typ5 über einen ATM Virtual Channel (ATM-VC) übertragen werden können. Dabei kann QoS auf ATM-VCs eingesetzt werden.

Bei einem direkten Vergleich zwischen dem ATM-Modell und H.323 fallen einige Ähnlichkeiten auf. Bei H.323 wird H.225 verwendet, um einen Kanal für H.245 zu öffnen. ATM benutzt die SAAL, um einen Signalisierungskanal zu öffnen. H.245 handelt die Kommunikationsmöglichkeiten[133] zwischen Sender und Empfänger aus. Bei ATM überträgt der Sender seine Daten mit bestimmten Parametern[134]. Durch die Signalisierung wird unter anderem auch überprüft, ob das ATM-Netz und der Empfänger die Anforderungen, welche in den Parametern enthalten sind, bereitstellen können. Beides, Möglichkeitsabgleich und Signalisierung, sorgt dafür, dass die Mediendaten bestmöglich übertragen

130 Classical IP
131 LAN Emulation
132 Multi-Protocol-over-ATM
133 Kodierverfahren, Videoeinsatz usw.
134 Z.B. Peak Cell Rate (PCR)

werden. Erst nachdem H.245 die Möglichkeiten abgeglichen hat, wird mit H.225 ein Kanal für die Medienströme geöffnet. In ATM wird nach Abschluss der Signalisierung ebenfalls ein neuer Kanal für die Daten geöffnet. Aber die Ähnlichkeit besteht nicht nur in dem Verbindungsaufbau, sondern auch in den Protokollen selbst. H.225 beinhaltet die Signalisierung Q.931 von ISDN. Die Signalisierung für ATM Q.2931 ist eine Erweiterung von Q.931.

H.323	ATM
H.225 öffnet einen Kanal für H.245.	SAAL öffnet einen Signalisierungskanal.
Kommunikationsmöglichkeiten werden über H.245 ausgehandelt.	Signalisierung erfolgt durch Q.2931.
H.225 öffnet einen Kanal für den Medienstrom.	Signalisierung öffnet einen Kanal für den Medienstrom.

Tab. 6.9 Direkter Vergleich zwischen ATM und H.323

Die Systemarchitektur ist so aufgebaut, dass H.323 mit allen spezifizierten Protokollen benutzt werden kann. Darüber hinaus kann mit diesem System der Vorteil von ATM, garantierter QoS bei hoher Bandbreite, genutzt werden. Weiterhin wird bei der Übertragung der Sprachdaten direkt ein ATM-VC genutzt. Nur RTP wird zusätzlich benötigt. Aber der zusätzliche Overhead von UDP und IP entfällt an dieser Stelle, wie Abb. 6.39 verdeutlicht.

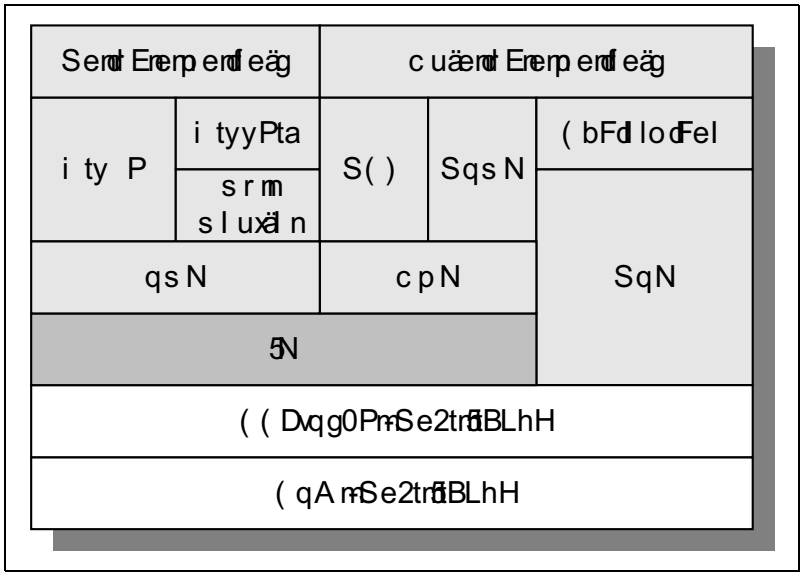


Abb. 6.39 Systemarchitektur H.323 über ATM

Der Verbindungsaufbau einer H.323-Sitzung über ATM durchläuft nun die folgenden Stationen:

- **H.225 über ATM:** H.225 wird mit den AAL-Typ5-Verfahren CLIP, LANE oder MPOA vom Sender zum Empfänger übertragen. Dabei ist keine bestimmte Methode zu bevorzugen. Der Empfänger erwartet die H.225-Daten an dem TCP-Port, der nach der Vereinbarung in H.225 festgelegt wurde. Wenn die H.225-Verbindung bestätigt ist, wird diese für H.245 genutzt. H.245 wird ebenfalls mit einem dieser Verfahren übertragen (siehe Abb. 6.40, Schritt 1).
- **H.245 über ATM:** Der durch H.225 geöffnete Kanal wird für H.245 verwendet. Für die Übertragung von H.245 wird ebenfalls ein AAL-Typ5-Verfahren ausgewählt. Mit H.245 wird der Möglichkeitsabgleich durchgeführt, wie er in H.323 vorgesehen ist. Außerdem erkennt H.245, ob ein ATM-VC für den Transport geöffnet werden kann. Wenn dies der Fall ist, muss H.245 beim Möglichkeitsabgleich die Transportmöglichkeiten von ATM nach I.371 berücksichtigen. Deshalb bedarf H.245 eines Zusatzes, ohne den der Transport von H.323 über AAL-Typ5 nicht möglich ist. H.323 wurde so konfiguriert, dass der H.245-Kanal und der Kanal für die Mediendaten zu unterschiedlichen Adressen aufgebaut werden können. Während des Abgleichs wird die Adresse für den Medienkanal vom Empfänger an den Sender gegeben. An dieser Stelle ist das die ATM-Adresse (siehe Abb. 6.40, Schritt 2).
- **Übergabe von H.225 an ATM-Signalisierung:** Nachdem der Möglichkeitsabgleich von H.245 durchgeführt worden ist, wird von H.225 ein neuer Kanal für die Mediendaten geöffnet. Dabei führt H.225 die Informationen mit, die für die ATM-Signalisierung von Bedeutung sind, einschließlich der ATM-Adresse. Diese Informationen werden von H.225 an die Signalisierung von ATM übergeben. Eine solche Information wäre z.B. die Peak Cell Rate (PCR). H.245 ist auf ein Kodierverfahren zwischen den Übertragungspartnern abgeglichen. Daraus lässt sich die maximale Bitrate für den Übertragungskanal ableiten. Wenn das maximal mögliche Kodierverfahren G.728[135] ist, wäre die Spitzenbitrate für den Übertragungskanal auf 16 Kbit/s festzulegen. Allerdings kann die Spitzenbitrate in einem IP-LAN nicht festgelegt werden, weil hier keine festen Bitraten vergeben werden können. In ATM wird aber für eine vereinbarte Bitrate sogar garantiert. Deshalb wird die aus dem vereinbarten Kodierverfahren gewonnene Information der PCR an Q.2931 übergeben. Weil H.225 die ISDN-Signalisierung Q.931 enthält und die ATM-Signalisierung Q.2931 aus Q.931 entstanden ist, ist die Übergabe einfach. Einige Übersetzungen von Q.931 zu Q.2931 müssen laut H.225 allerdings noch weiter untersucht werden (siehe Abb. 6.40, Schritt 3).

135 Mit einer Bitrate von 16 kBit/s

- **ATM-Aktivitäten:** Nach der Übergabe der Informationen, wird zunächst ein ATM-Signalisierungskanal geöffnet (siehe Abb. 6.40, Schritt 4). Danach tritt die Q.2931 in Aktion (siehe Abb. 6.40, Schritt 5). Q.2931 prüft mit den übernommenen H.225-Informationen, ob das ATM-Netz und der Empfänger die Parameter bereitstellen können, die für die Übertragung der Mediendaten entscheidend sind. Es wird also ein bestimmter QoS gefordert. Wenn dieser vom ATM-Netz und vom Empfänger bestätigt wird, kann durch die Kontrollfunktionen des ATM-Netzes im Management die Einhaltung des QoS garantiert werden. Nachdem der QoS zwischen Sender, Empfänger und ATM-Netz vereinbart ist, kann der ATM-VC für die Mediendaten geöffnet werden (Abb. 6.40, Schritt 6). Wenn der ATM-VC aktiv ist, werden die Mediendaten direkt über diesen VC übertragen. Nur das RTP befindet sich noch zwischen Anwendungsschicht und der Adaptionsschicht von ATM. Zusätzlich kann der QoS des ATM-VC voll genutzt werden.

Dieser Verbindungsaufbau und die Übertragung der Mediendaten direkt über einen ATM-VC sind nur durchführbar, wenn sich beide Endgeräte in einem ATM-Netz befinden. Die Zusammenarbeit mit anderen H.323-Endgeräten, die beispielsweise auf IP basieren, ist ohne Gateway-Funktionen möglich. Allerdings müssen Endgeräte, die H.323, Anhang C, entsprechen wollen, die Mediendaten sowohl über ATM-VCs als auch durch UDP/IP empfangen können. Wenn also ein Endgerät aus einem Ethernet mit UDP-/IP-Sprachdaten zu einem Endgerät in einem ATM-Netz schicken will, werden alle Daten durch eine AAL-Typ5-Methode übermittelt. Denn das Endgerät im ATM-Netz arbeitet nach diesem Anhang C und kann alle Daten auch von TCP oder UDP übernehmen. Nur wenn sich beide Endgeräte im ATM-Netz befinden, können die Mediendaten über ein ATM-VC direkt übertragen werden. [H.323(99)]

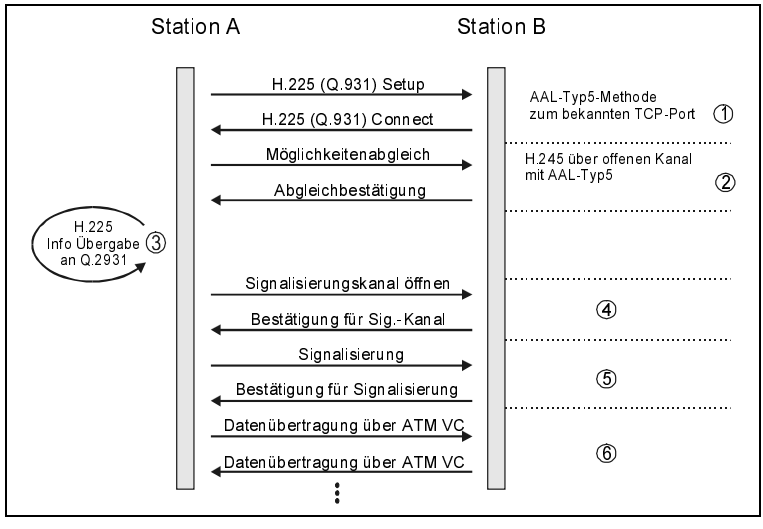


Abb. 6.40 Verbindungsaufbau von H.323 über ATM

6.4.5 Fazit

Mit den beschriebenen Ansätzen sollte eine Möglichkeit geschaffen werden, die Vorteile von ATM mit der bestehenden Infrastruktur der LANs auf IP-Basis zu koppeln. Für Echtzeitanwendungen sind der QoS und die Bandbreitenreservierung des ATM-Netzes von gesteigertem Interesse. Ebenfalls konnte so ein guter Überblick über die Funktionsweise eines Gateway geschaffen werden, welches beim Übergang von IP auf ISDN ebenfalls notwendig ist.

Die Vorteile des QoS von ATM können bei IP-over-ATM-Verfahren wie CLIP und LANE nicht genutzt werden. ATM dient hier nur als reines Transportmittel. Das ATM-Netz kann auch als Tunnel für Ethernet-Rahmen aufgefasst werden. Gerade für Echtzeitdaten wirkt sich dies nachteilig aus. Der garantierte QoS würde hier die Qualität solcher Daten wesentlich verbessern. MPOA[136] und MPLS[137] bieten die Möglichkeit, die meisten Vorteile von ATM zu nutzen. Durch die Verteilung der Routing-Funktionen über das ATM-Netz bei MPOA wurde die Konzentration auf eine zentrale Stelle vermieden, wie sie bei CLIP und LANE vorherrscht. Durch die Verteilung der Router-Funktionen wird es außerdem möglich, auf die Anforderungen flexibel zu reagieren. Während MPOA verschiedene Subnetze von IP mittels der Shortcut-Funktion überwindet und dadurch QoS und geringe Laufzeiten möglich macht, schafft MPLS mittels Layer-2-Unabhängigkeit und Priorisierung ein Traffic-Management für das Kernnetz mit den gleichen Zielen.

MPOA setzt direkt auf der Netzwerkschicht 3 auf. Damit ist ATM für IP nicht mehr transparent, sondern mit MPOA können die Eigenschaften von ATM in den bestehenden Infrastrukturen besser genutzt werden. Das wird durch neue Möglichkeiten der Konfiguration und des Managements gewährleistet. Zusätzlich werden bestehende Protokolle wie NHRP[138] oder RSVP integriert, die zusätzliche Eigenschaften von ATM für den Datentransport nutzbar machen und darüber hinaus auch die IP-Eigenschaften verbessern. RSVP ermöglicht in IP-Netzen das Verwalten von reservierten Ressourcen. Es wird durch RSVP mehreren Sendern und Empfängern ermöglicht, sequentiell die gleiche Netzressource zu nutzen. Diese Funktion ist ähnlich der von ATM angebotenen, dass nämlich für die vereinbarte Bitrate garantiert wird. Durch MPOA können die Informationen von RSVP für die ATM-Verbindungen nutzbar gemacht werden. Für Echtzeitdaten sind die Shortcuts mit dem entsprechenden QoS besonders wichtig. Aber auch die Nutzung von RSVP in den IP-Subnetzen trägt zur Verbesserung der Qualität solcher Daten bei. Allerdings wirkt sich die Verteilung der Router-Funktionen über das ATM-Netz und die Unterstützung der verschiedenen Protokolle auch nachteilig aus. Das MPOA wird dadurch

136 Multi-Protocol-over-ATM
137 Multi-Protocol-Label-Switching
138 Next-Hop Resolution Protocol

sehr komplex und damit schwierig zu konfigurieren und zu administrieren. Auch die Flexibilität des Netzes bezüglich der Ausdehnung ist nicht befriedigend gelöst. MPLS geht hier einen anderen Weg, der eine wesentlich bessere Skalierbarkeit und Performance verspricht, was bereits ausführlich dargelegt wurde. MPLS könnte im Zusammenspiel mit ATM die Vorteile von MPOA aufgreifen und erfolgreich umsetzen.

Ein weiterer Aspekt der Echtzeitdaten ist die Empfehlung H.323 der ITU-T. Diese Empfehlung verbessert die Übertragung von Echtzeitdaten in den IP-LANs. Sie bietet aber auch die Möglichkeit, die IP-over-ATM-Verfahren zu nutzen. Darüber gibt es die Möglichkeit, innerhalb eines ATM-Netzes direkt einen ATM-VC mit AAL-Typ5 zu nutzen. Dadurch entfällt ein Teil des Overhead und der QoS von ATM kann für den VC vereinbart werden.

Mit VoIP steht eine Technik zur Verfügung, die zur Zeit vor allem im Unternehmensbereich sehr gute Möglichkeiten für die Integration von Audio, Video und Daten in einem IP-basierten Netz bietet. In diesem Bereich lässt sich vor allem eine höhere Produktivität der Mitarbeiter erreichen, die durch eine einheitliche Kommunikationsplattform für Telefonie, E-Mail und Fax schneller Informationen abrufen und verarbeiten können. Durch den Einsatz der IP-Telefonie lassen sich auch Infrastrukturkosten senken, da durch eine gemeinsame Netzstruktur für Sprache und Daten sämtliche Kosten für ein separates Telefonnetz entfallen. In diesem Zusammenhang kann durch eine dynamische IP-Adressvergabe über DNS[139]/DHCP[140] an die IP-Telefonie/Soft-Clients der administrative Aufwand verringert werden. Dieses bedeutet unter anderem höhere Mobilität und Kostensenkung, da der Mitarbeiter unter seiner Rufnummer und seinen eingerichteten persönlichen Profilen ohne aufwendige Umstellungen wie bei einer traditionellen TK-Lösung erreichbar bleibt. Mit Hilfe der Sprach- und Datenkonvergenz können Leistungsmerkmale leichter und kostengünstiger implementiert werden, durch die völlig neue Anwendungen etabliert werden können, beispielsweise Voice-Mail (Unified Messaging), Virtual Call Center bzw. Web Call Center (E-Commerce).

Dabei ist die Verbindungsqualität das Maß für die Akzeptanz der IP-Telefonie. Durch diese Kriterien werden derzeit Grenzen für einen sinnvollen Einsatz der VoIP-Technologie in Netzen ohne QoS gesetzt. In Unternehmensnetzen sind die zulässigen und akzeptablen Werte für Laufzeit (Echo), Paketverlust und Jitter auf Grund administrativer Möglichkeiten leichter zu erreichen. Bei der IP-Telefonie über das öffentliche Internet sind vor allem hohe Anforderungen an die ISP und an die verschiedenen Netzbetreiber (Netzübergänge) gestellt, die VoIP-Technologie einsetzen. Hier sind vor allem noch Fragen einer zentralen Netzma-

139 Domain Name Service
140 Dynamic Host Control Protocol nach RFC-2131

nagementplattform und zur unterschiedlichen Lastsituation des Netzes zu klären. Weiterhin als problematisch kann sich die Interoperabilität bei dem Einsatz von standardkonformen VoIP-Produkten unterschiedlicher Hersteller erweisen.

7

Messungen

Nachdem bisher die theoretischen Grundlagen erarbeitet und die Standards untersucht wurden, findet in diesem Kapitel eine praktische Betrachtung bzw. Evaluierung der genannten Ansätze statt. Es werden die Verfahren in Messungen und Tests untersucht und miteinander verglichen, schließlich können Theorie und Praxis oftmals unterschiedliche Ergebnisse hervorbringen. Auch hier wird es daher zu einer Unterteilung der Bereiche Security, QoS, TE und VoIP kommen. Im Vordergrund steht auf der einen Seite die Performance bei den durchgeführten Messungen. Auf der anderen Seite spielen auch andere Kriterien wie Handhabung, Einhaltung der Standards, Interoperabilität und Skalierbarkeit eine entscheidende Rolle. Einige Tests wurden von mir selbst in Laborumgebung durchgeführt, andere sind von Testcentern umgesetzt worden.

7.1 Security

Bei den Messungen der Security standen die Performance im Vordergrund, die durch die Sicherheitsmechanismen in jedem Fall eingeschränkt wird, und die Standardkonformität, um interoperabel zu unterschiedlichen Produkten zu sein. Dabei wurde der Einsatz von **Secure Socket Layer** (SSL) für einen gesicherten Webzugriff und von IPsec für den Aufbau einer Sicherheitsplattform speziellen Tests unterzogen.

7.1.1 Messungen der Performance von SSL

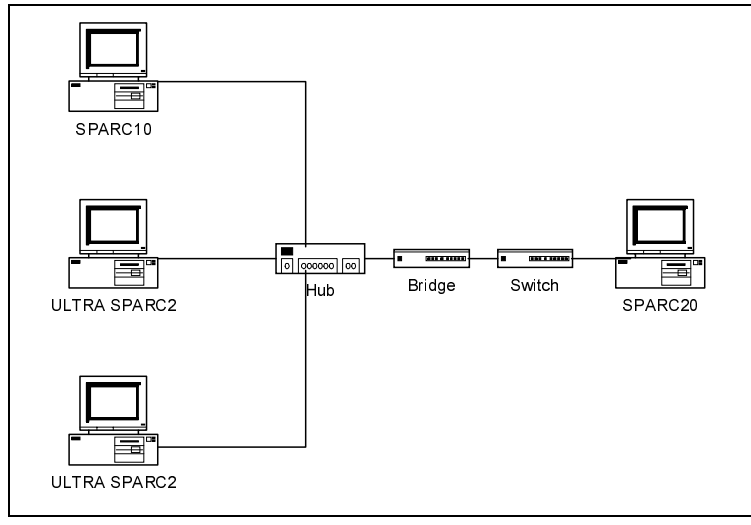
Die Zielsetzung der Messungen war es zu zeigen, wie sich der Einsatz von SSL auf die Performance einer Anwendung auswirkt. Dies sollte für verschiedene Rechnerleistungen gezeigt werden. Um Eckpunkte für die Bewertung zu erhalten, wurde auch eine Vergleichsmessung mit einer herkömmlichen Implementierung der Anwendung FTP durchgeführt. Es kamen dabei die SSL-Implementierung von Eric Young, SSLeay-0.6.6[1], und die dazugehörige FTP-Anpassung von Tim Hudson, SSLftp-0.8, zum Einsatz. Dabei handelt es sich zwar nicht um

1 SSLeay-0.6.6 ist eine Implementierung von SSL Version 2

die letzte SSL-Version; trotzdem lassen sich gute Aussagen bezüglich der Performance treffen, da diesbezüglich keine Verbesserungen in die Version 3.0 eingeflossen sind.

Messaufbau Bei den Messungen wurde nur die Transferzeit für die Übertragung einer Testdatei aufgenommen. Der SSL-Handshake, der nur beim Login auf den FTP-Server erfolgt, bleibt dabei unberücksichtigt. Die Messungen gliederten sich in zwei Teile: Bei den Messungen wurde ausschließlich das SSL-FTP-Paket verwendet, um nur die Auswirkung von SSL auf die Performance eines Dateitransfers und nicht die Performance-Unterschiede zwischen verschiedenen FTP-Implementierungen zu messen. Dazu mussten die Messungen auch ohne den Einsatz von SSL durchgeführt werden. Für die Messungen wurden Solaris-Workstations von Sun verwendet. Dabei kamen SPARC10, SPARC20 und UltraSPARC zum Einsatz. Die Workstations waren über ein Ethernet-LAN miteinander verbunden.

Abb. 7.1
SSL-Messaufbau



Zuerst wurde auf einer ULTRRA SPARC2 und der SPARC10 ein FTP-Server eingesetzt, während die anderen beiden Rechner als Client fungierten. Dabei wurden beide Server jeweils mit und ohne SSL gemessen. Für die UltraSPARC als Server und die UltraSPARC als Client wurden die Messungen über 10- und 100-Mbit/s-Ethernet durchgeführt. Als Messergebnisse dienten die vom FTP-Client in einer Statuszeile gelieferten Werte für die Dauer der Übertragung und die erzielte Datenrate. Beide Werte werden vom FTP-Client in Fliesskommazahlen angegeben, deren Genauigkeit auf zwei Stellen beschränkt sind. Damit ist die Auflösung bzw. Genauigkeit größerer Werte eingeschränkt. Der Dateitransfer erfolgte stets im Binary-Modus.

Es wurden unterschiedliche Kombinationen der verschiedenen FTP-Implementierungen gemessen. Bei jeder Messreihe wurde für jeden Client eine FTP-Verbindung zum Server aufgebaut. Danach erfolgten nacheinander drei Übertragungen einer Testdatei. Es handelte sich dabei um eine binäre Datei, die ca. 100 MByte groß war. Im zweiten Teil wurden die Messungen auch mit einer zweiten Testdatei durchgeführt, um zu bestätigen, dass bei dieser SSL-Umsetzung keine Komprimierung der Daten vorgenommen wird. Es wurde dazu eine komprimierte Datei der Größe 40 Mbyte verwendet. Die Messergebnisse bestätigten diese Annahme, da die Übertragung im Verhältnis zur Größe der Dateien die gleiche Zeit benötigte; sie werden deshalb hier nicht explizit dargestellt.

Um Plattenzugriffe weitgehend zu verhindern, wurde die Testdatei auf dem Client gleich nach */dev/null* geschrieben. Die UltraSPARC2 verfügte über einen Hauptspeicher von 256 Mbyte und konnte damit die gesamte Datei in den Hauptspeicher laden. Die SPARC10 mit einem Hauptspeicher von 64 Mbyte könnte dies nicht, sodass sich Zugriffe auf die Festplatte nicht vermeiden lassen könnten. Weiterhin wurden der Verschlüsselungsalgorithmus RC4 und die Hash-Funktion MD5 verwendet. Die Messungen wurden zu einem Zeitpunkt mit minimaler Netzlast in Laborumgebung durchgeführt. Somit ließen sich im Allgemeinen maximale Datenraten mit den FTP-Verbindungen erzielen.

Messergebnisse

Tab. 7.1 Messungen mit/ohne SSL über 10/100-Mbit/s-Ethernet, Teil 1

Client/Server	Messung	Ethernet [Mbit/s]	Dauer [s] Mit/ohne SSL	Datenrate [Kbyte/s] Mit/ohne SSL
SPARC10/UltraSPARC2	1	10	130/97	790/1000
	2	10	130/97	760/1000
	3	10	130/97	760/1100
SPARC20/UltraSPARC2	1	10	120/110	820/910
	2	10	130/110	800/910
	3	10	120/110	800/890
UltraSPARC2/UltraSPARC2	1	10	92/91	1100/1100
	2	10	92/93	1100/1100
	3	10	91/90	1100/1100
UltraSPARC2/UltraSPARC2	1	100	32/12	3100/8300
	2	100	32/12	3100/8300
	3	100	32/12	3100/8300

Die Messungen erfolgten mit einer 100-MByte-Testdatei. Für die Messreihen wurde die UltraSPARC2 als FTP-Server eingesetzt. Die anderen Workstations dienten abwechselnd als Clients. Die Messreihen wurden mit dem SSL-FTP-Paket durchgeführt, wobei mit und ohne SSL getestet wurde. Dabei kamen abwechselnd die UltraSPARC2 und die SPARC10 als Server zum Einsatz. Der Dateitransfer bei der Vergleichsmessung erfolgte immer im Binary-Modus.

Tab. 7.2 Messungen mit/ohne SSL über 10-Mbit/s-Ethernet, Teil 2

Client/Server	Messung	Ethernet [Mbit/s]	Dauer [s] Mit/ohne SSL	Datenrate [Kbyte/s] Mit/ohne SSL
UltraSPARC2/SPARC10	1	10	260/150	380/650
	2	10	240/160	420/640
	3	10	240/150	410/680
SPARC20/SPARC10	1	10	240/150	410/650
	2	10	240/150	410/660
	3	10	240/160	410/630
SPARC10/SPARC10	1	10	280/150	360/680
	2	10	280/140	360/710
	3	10	280/140	360/710

Auswertung Wie erwartet kommt es beim Einsatz von SSL zu Performance-Einbußen, wie es sich an den Messergebnissen aus Tab. 7.1 und Tab. 7.2 ablesen lässt. Der Vergleich von unverschlüsselter zu verschlüsselter Übertragung ergibt dabei, dass die Übertragungszeit bei einem Datentransfer mit SSL je nach Bemessungsgrundlage um ca. 32 bis 39 % zunimmt. Es lässt sich erkennen, dass die Übertragungszeit für den verschlüsselten Transfer mit der Leistungsfähigkeit des Clients abnimmt.

Für die SPARC10 nimmt die Übertragungszeit bei verschlüsseltem Transfer um 34% im Vergleich zum unverschlüsseltem Transfer zu. Bei der SPARC20 sind es 12% und bei der UltraSPARC2 0 %. Obwohl man bei der SPARC20 eine kürzere Zeit für die Übertragung ohne SSL als bei der SPARC10 erwartet, tritt genau der umgekehrte Fall ein. Dies lässt sich dadurch erklären, dass die SPARC20 nicht direkt über den Hub mit den anderen Workstations verbunden war (siehe Abb. 7.1). Damit erreicht die SPARC20 nicht die hohen Datenraten bei unverschlüsseltem Transfer wie die SPARC10 und hat somit eine geringere prozentuale Zunahme der Übertragungszeit bei dem Transfer mit SSL.

Für den Client UltraSPARC2 erweist sich die Bandbreite von Ethernet schon bei der verschlüsselten Übertragung als Engpass. Deshalb liegt die Zunahme der Übertragungszeit bei 0%. Ergänzend dazu ist die Messung über

Fast-Ethernet aus Tab. 7.1 zu betrachten. Hier benötigt die verschlüsselte Übertragung mehr als das 2,5-fache der Übertragungszeit für den unverschlüsselten Transfer.

Die Messungen mit der SPARC10 als Server sind nur bedingt aussagekräftig, da es unabhängig vom Client immer die gleiche Übertragungszeit gibt. Dies lässt sich auf zwei Punkte zurückführen. Zum einen ist der nicht ausreichende Hauptspeicher entscheidend, da dadurch die Übertragungszeit maßgeblich durch die Performance der Festplatte bestimmt wird, und zum anderen ist die SPARC10 unter den verwendeten Workstations die mit der geringsten Rechenleistung.

Abschließend bleibt festzuhalten, dass bei höherer Netzlast, als sie bei diesen Messungen vorhanden war, der Unterschied zwischen den Transferzeiten bei verschlüsselter und unverschlüsselter Übertragung kleiner wird, da sich die hohen Datenraten wie bei den Messungen ohne SSL nicht mehr erzielen lassen. Wenn beispielsweise die effektiv zu erzielende Datenrate unter der mit SSL zu erzielenden Datenrate liegt, gleichen sich die Übertragungszeiten für den Transfer mit und ohne SSL an. Deshalb muss die zu erreichende Datenrate immer mit betrachtet werden.

7.1.2 Messung der Performance von IPsec

Der Standard IPsec stellt heute die Grundlage für den einheitlichen Aufbau von VPNs dar. Eine ganze Reihe von Herstellern implementiert inzwischen IPsec, sodass die Anwender nicht mehr auf proprietäre Lösungen angewiesen sind. Allerdings ergeben sich durchaus noch Probleme in der Interoperabilität, da die Spezifikationen mitunter unterschiedlich interpretiert werden bzw. proprietäre Funktionen hinzukommen. Um die Kompatibilität zwischen den Produkten zu erhöhen und die Sicherheit zu garantieren, stellt das Sicherheitsgremium ICSA (http://www.icsa.net) Zertifizierungen aus. Auf diese Zertifizierungen sollte man achten, auch wenn es nachweislich momentan noch keine Garantie für die Interoperabilität gibt.

Das liegt daran, dass IPsec eine sehr junge Spezifikation ist, die noch weiterentwickelt werden muss. Aus diesem Grund wird bereits an der nächsten Version gearbeitet, die u.a. die Einbindung von Certificate Authorities und den standardisierten Tunnelaufbau über variierende Subnetze einbezieht. Grundsätzlich werden IPsec-Tunnel durch zwei Endpunkte bzw. Subnetze definiert. Es müssen vorab aber auch zusätzliche Angaben wie Festlegung der Algorithmen für Authentifizierung und Verschlüsselung gemacht werden. Manche Hersteller haben eigene Datenkompression in die Softwarelösung integriert, wodurch es zu Problemen beim Einrichten der Tunnel kommen kann.

Messaufbau Am Telecommunications Software and Multimedia Laboratory (TCM) der Helsinki University of Technology (HUT) wurde bereits 1995 ein Projekt zur Studie von IPsec gestartet. In dieser konkreten Implementierung wurde der IPsec-Kontrollmechanismus unterhalb der IP-Schicht eingefügt. Das hat den Vorteil, dass man unabhängig vom jeweiligen TCP/IP-Stack arbeiten kann, andererseits mussten Funktionen wie Fragmentierung und IP-Checksummenbildung neu implementiert werden. Abb. 7.2 zeigt die Architektur des Prototyps. Weitere Algorithmen zur Authentisierung und Verschlüsselung können hier leicht durch weitere Module (Treiber) hinzugefügt werden. Da es sich um einen Prototyp handelte, wurde kein automatisches, sondern nur ein manuelles Key-Management unterstützt. Beim Performance-Test wurde eine große Datei über ein 10-Mbit/s-Ethernet zwischen zwei Sun-SPARC-Workstations kopiert.

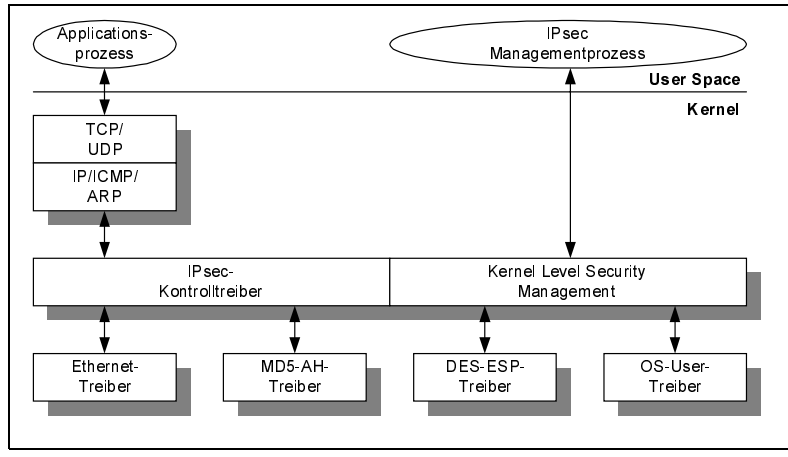


Abb. 7.2 IPsec-Implementierung beim TCM

Ein anderer Test wurde im WAN mittels Cisco Router 2600 und 3600 durchgeführt. Als vorteilhaft erwiesen sich dabei das Erkennen der Quelle, des Empfängers oder der Daten aus dem gesamten Datenstrom, weil die IP-Pakete gekapselt werden und die neuen IP-Pakete nur noch Source und Destination des IPsec-Knotens enthalten. Ebenfalls war ein bestimmter Sicherheitsgrad von einem Netzwerk zum anderen aufzubauen und aufrechtzuerhalten. Dabei konnte unabhängig von der Anwendung eine Verschlüsselung vorgenommen werden. Das Einrichten eines neuen IPsec-Tunnels erzeugt weiterhin keine Verzögerungszeiten, da der neue Tunnel bereits aufgebaut wird, bevor die Lebenszeit des alten Tunnels abgelaufen ist. Aus Sicherheitsgründen muss der Tunnel in bestimmten Zeitintervallen gewechselt werden.

Ein dritter Test wurde im eigenen Labor durchgeführt. Dabei sind zwei Rechner als Client und als Gateway verwendet worden, um die Software zu installieren. Es wurde eine direkte Verbindung mittels Cross-over-Kabel reali-

siert, um keine Beeinträchtigung durch das Netzwerk zu erhalten. Die verwendete NCP-Software besteht aus einer Remote-Access-Lösung für unternehmensübergreifendes Enterprise Networking und ist sowohl für Intranets als auch Extranets entwickelt worden. Neben der Verbindung von verteilten LANs beispielsweise in Außenstellen und der Möglichkeit des Teleworking hat man auch die Integration mobiler Teilnehmer berücksichtigt. Zusätzlich bietet NCP den direkten Tunnelauf- und -abbau durch den Client zum firmeneigenen Remote Access Server (RAS), unabhängig von der ISP-Infrastruktur an. Auf diese Weise wird eine sehr sichere End-to-end-Kommunikation auch ohne IPsec ermöglicht, sodass das gesamte Sicherheitsmanagement in den Händen des Unternehmens und nicht bei den jeweiligen ISPs liegt. Der Remote-Client des Teilnehmers wird während des Bestehens der Verbindung vollständig in das Unternehmensnetz integriert. Connectivity besteht über alle öffentlichen Netze wie ISDN, PSTN (analoges Fernsprechnetz), X.25, GSM, HSCSD, GPRS, xDSL, xDSL Sat und Internet. Protokollseitig werden neben IP auch IPX, SNA und NetBIOS unterstützt und transparent im WAN übertragen. Als Tunnelverfahren kann das Protokoll Layer 2 Forwarding (L2F), Layer 2 Tunneling Protocol (L2TP) und IPsec eingesetzt werden. Das L2TP-Tunneling-Protokoll ist, so wie es NCP implementiert hat, in der RFC-2716 als L2sec dokumentiert. In der höchstmöglichen Sicherheitsstufe unterstützen die NCP-Komponenten PKI-Infrastrukturen nach X.509v3. Die Einbindung der Lösung in private oder öffentliche Trust-Center-Umgebungen[2] ist bereits in der Praxis erprobt worden.

Tab. 7.3 Verwendete NCP-Testkomponenten

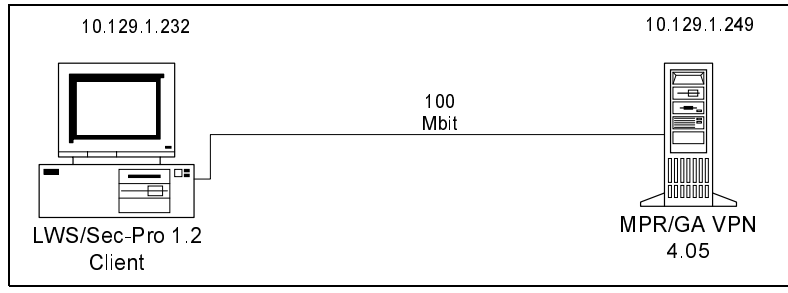
Komponenten	Geräte
LWS/Sec Client	NT-Server [Pentium II/400 MHz/128 MByte]
	NCP LWS/Sec Pro ver.1.2
	NIC: 3Com EtherLink10/100 PCI NIC(3C905C-TX)
	IP: 10.129.1.232
MPR/GA Gateway	NT-Server [AMD/500MHz/128 MByte]
	NCP MPR/GA VPN 4.05 + NCP Enterprise Manager 4.02
	NIC: 3Com EtherLink10/100 PCI NIC(3C905C-TX)
	IP: 10.129.1.249

Nach der LWS/Sec-Pro-Installation wurde zu dem physikalischen der virtuelle Netzwerkadapter (NCP LWS/Sec-Pro Adapter) installiert. Bei der Konfiguration der Netzwerkadapter gibt es zwei Varianten:

2 Z.B. T-Telesec, Deutsche Post Signtrust

- **Variante A:** Ein Gateway-Eintrag wird lediglich beim physikalischen Adapter verwendet. Hierdurch werden nur die im LWS/Sec-Monitor angegebenen Netze über Tunnel erreicht. Die Konfiguration kann durch Einsicht in die Routing-Tabelle kontrolliert werden. Der restliche Datenverkehr wird über das Default-Gateway geroutet. Bei einer Verbindung zum Server (MPR/GA-Gateway) bedarf es jedoch eines weiteren Routing-Tabellen-Eintrags auf dem Server (siehe MPR/GA-Konfiguration). Dieser Eintrag sorgt dafür, dass alle Daten mit virtuellem Zielnetz über das (virtuelle) Gateway geroutet werden.
- **Variante B:** Verwendet im Gegensatz zur Variante A einen Gateway-Eintrag lediglich beim virtuellen Adapter. Dies bewirkt, dass der gesamte Datenverkehr über den Tunnel stattfindet. Es wird jedoch ein Eintrag in der Routing-Tabelle benötigt, der für den erstmaligen Tunnelaufbau vom LWS/Sec-Client zum MPR/GA-Gateway verantwortlich ist. Hierdurch wird über das Gateway eine Verbindung zum MPR/GA-Gateway hergestellt, welches dann dem Client eine virtuelle IP-Adresse zuweist. Der folgende Datenverkehr findet dann über den Tunnel statt.

Abb. 7.3 Messaufbau der VPN-Messungen



Wegen der Direktverbindung beider PCs wurde jedoch lediglich Variante A verwendet. Damit alle IP-Pakete mit Ziel 172.17.1.0 vom MPR/GA- zum LWS/Sec-Client durch den Tunnel übertragen werden, muss die Routing-Tabelle um den folgenden Eintrag ergänzt werden: route add -p 172.17.1.0 mask 255.255.255.0 10.1.1.1. Alle Pakete mit dem Zielnetz 172.17.1.0 werden dann über das VPN-Gateway gesendet. Zur Administration des Remote Access Server (RAS) wurde der NCP Enterprise Manager benötigt. Dieser kann auf jedem PC, der in Verbindung mit dem NAS steht, installiert werden. Im Enterprise Manager wird neben dem WAN-Link für Default User Tunnel-End-Point auch der für den User1 konfiguriert. Um Daten verschlüsselt zu übertragen, muss im LWS/Sec-Client und MPR/GA-Gateway der gleiche Verschlüsselungsalgorithmus und Schlüssel ausgewählt werden. Die FTP-Messungen wurden hingegen mit Celeron-Prozessoren PII und 450 MHz ausgeführt.

Messergebnisse

Die Performance im eigenen Intranet wird nicht ausschließlich von den Sicherheitsmechanismen von IPsec beeinträchtigt. Das liegt daran, dass diese nur innerhalb des Extranets und am Netzrand in der Firewall oder dem Router angewendet werden. Allerdings kommt es natürlich zu Einschränkungen zwischen den Teilnehmern eines Extranets. Auch wenn bei dieser prototypischen Implementierung noch keine Optimierung hinsichtlich der Geschwindigkeit getroffen wurde, lässt sich doch ein deutlicher Overhead durch Verschlüsselung und Authentisierung der IP-Pakete bemerken. Das ist wiederum eine Chance für Tunnelmechanismen mit proprietärer Verschlüsselung oder für Hardware-Beschleuniger, die IPsec mit höherer Performance ausstatten. Es ist daher immer wichtig, einen Performance-Test durchzuführen, wenn man ein Extranet oder VPN aufbauen bzw. realisieren möchte. Ansonsten könnte man immer Probleme erleben, beispielsweise einen schlechten Datendurchsatz durch Software-Implementierungen.

Performance-Werte	Durchsatz
ohne IPsec	315 Kbyte/s
IPsec ohne AH und ESP	47 Kbyte/s
IPsec mit AH	26 Kbyte/s
IPsec mit ESP (Transport- oder Tunnelmodus)	26 Kbyte/s
IPsec mit AH und ESP (Transportmodus)	20 Kbyte/s
IPsec mit AH und ESP (Tunnelmodus)	18 Kbyte/s

Tab. 7.4 Performance-Werte der IPsec-Implementierung beim TCM

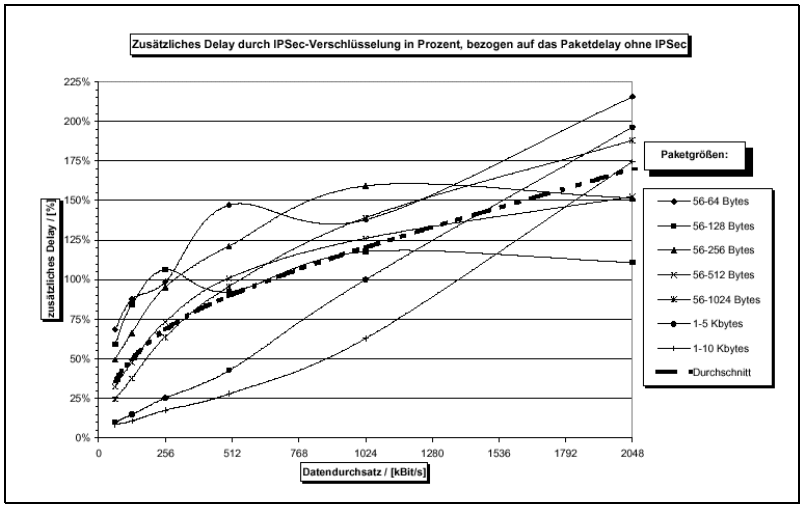


Abb. 7.4 Verzögerung durch IPsec-Verschlüsselung bei Cisco-Routern

In einem anderen Beispiel wurden Performance-Messungen zwischen zwei Linux-Systemen mittels ttcp durchgeführt. Das Testtool ttcp ermittelte auf einem Pentium-133-MHz-Rechner mit 32 MByte RAM über eine Fast-Ethernet-Verbindung 88 Mbit/s im unverschlüsselten Modus. Mittels AH (HMAC-MD5) sank die Performance auf nur noch 20 Mbit/s, während sich mit ESP (Triple-DES-MD5) der Durchsatz auf 2,56 Mbit/s verringerte. Bei Einsatz beider Verfahren wurde der Wert 2,48 Mbit/s erreicht. Hieran wird deutlich, dass ESP den Datenverkehr erheblich beeinflussen kann.

Abb. 7.5 NCP-Performance-VPN-Test

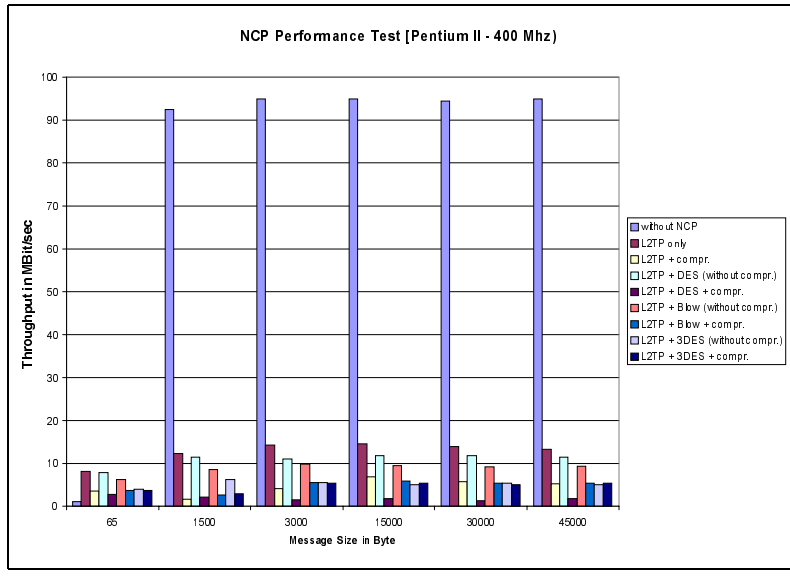


Abb. 7.4 zeigt die Ergebnisse der Performance-Messungen, die mit Cisco-Routern mittels einer Dialogapplikation vorgenommen wurden. Die Prozentwerte beziehen sich auf die Paketverzögerung ohne Verschlüsselung. Wurde beispielsweise für Datenpakete der Größe 56-64 Byte bei einem Durchsatz von 1 Mbit/s ohne Verschlüsselung eine Verzögerung von 3,72 ms gemessen und mit Verschlüsselung 8,87 ms, entspricht dieser Wert einem Zuwachs von immerhin 137%. Ein weiteres Problem besteht darin, dass ein vermehrter Einsatz von Sicherheitsfunktionen auf höheren Schichten beispielsweise über SSL/TLS vorgenommen wird. Wenn dies der Fall ist, braucht auf IP-Ebene keine Sicherung mehr durchgeführt zu werden. Hinzu kommt die Gefahr einer Doppelverschlüsselung, die weitere Leistungseinbußen nach sich ziehen würde.

Abb. 7.5 und Abb. 7.6 gehen auf die dritte Messung genauer ein und zeigen die Performance-Einbrüche bei verschiedenen Varianten der Verschlüsselung. Dabei wurde einmal die Performance direkt gemessen und ein anderes Mal durch die Übertragung einer FTP-Datei ermittelt. Die besten Ergebnisse wur-

den in beiden Fällen erzielt, wenn ausschließlich L2TP eingestellt ist. Die größten Einbrüche sind mit L2TP + (3)DES + Komprimierung festzustellen. IPsec konnte hier noch nicht getestet werden, da es zum Testzeitpunkt nicht zur Verfügung stand. Als Ergebnis lässt sich aber feststellen, dass die Performance auf unter 1/5 zurückgeht. Dies kann sich auch bei geringeren Datenraten negativ bemerkbar machen.

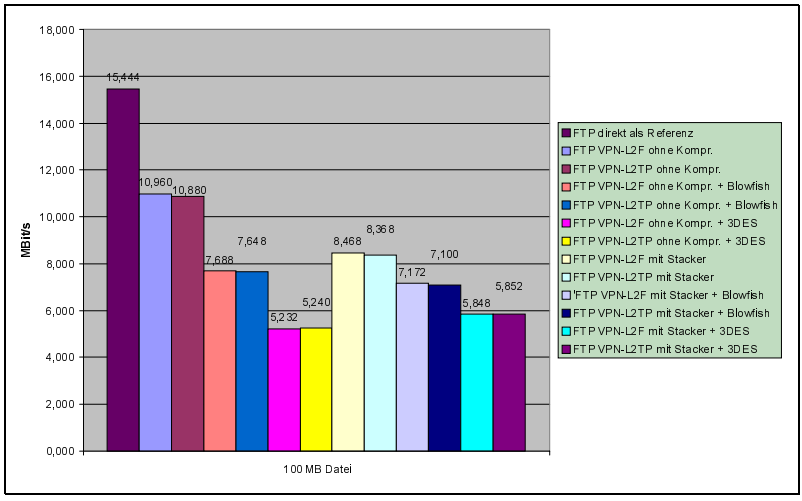


Abb. 7.6 FTP-Durchsatz ohne Disk-IO und mit einer 4-Mbit/s-Datei

Auswertung

Der Transportmodus von IPsec schützt vorrangig höhere Schichtenprotokolle, was wiederum für den Einsatz von End-to-end-Kommunikation spricht. Für ein VPN ist allerdings sowohl die Authentifizierung als auch die Verschlüsselung bedeutsam. Deshalb wird der Einsatz von ESP und damit dem Tunnelmodus eine höhere Bedeutung zukommen. IPsec bietet ebenfalls Schutz gegen Replay-Attacken. Das heißt, ein altes Paket, welches in den Datenstrom wieder eingebracht wird, kann aufgrund der Sequenznummern nicht mehr als gültig erkannt werden und wird abgewiesen. Um eine höchstmögliche Sicherheit bezüglich der eingesetzten Anwendungen erhalten zu können, muss AH und ESP im Tunnelmodus eingesetzt werden. ESP allein würde nur die unveränderlichen Daten im Originalpaket inklusive des ESP-Header und Trailer absichern.

Asymmetrische Verschlüsselungsverfahren sind ungefähr um den Faktor 1000 langsamer als die symmetrische Variante. Allerdings kann man eine höhere Sicherheit durch asymmetrische bzw. Public-Key-Verfahren und durch größere Schlüssellängen erreichen. Kryptographische Algorithmen, die auf Hardware implementiert werden, können die Performance erheblich steigern. Hier muss letztendlich zwischen Kosten/Nutzen bzw. Leistung/Sicherheit entschieden werden. Wenn man eine höhere Sicherheit, sprich Verschlüsselung, erhalten will, muss man Einschränkungen in der Performance hinnehmen.

Hinzu kommt, dass Authentifizierung und Verschlüsselung die Leistungsfähigkeit mehr einschränken, als wenn man nur über AH arbeiten würde.

Nachteilig wirkte sich allerdings der erhebliche Performance-Verlust aus, der beispielsweise durch die Router-Verschlüsselung verursacht wurde. Obwohl im Back-to-back-Betrieb getestet wurde, lag der Leistungsverlust bei einer FTP-Anwendung bis 500 Kbit/s bei nur 6%, stieg dann aber auf 28% bei 1 Mbit/s bzw. 62,9% bei 2 Mbit/s an. Bei der Übertragung einer Dialoganwendung, das heißt bei einem Austausch von Paketen geringer Größe, lag der Leistungsverlust bei 64 Kbit/s bereits bei 22,2% und stieg dann kontinuierlich auf 63,7% bei 2 Mbit/s an. So kam es zu einer durchschnittlichen, minimalen Verzögerung von 12,57 % bei 64 Kbit/s und bis zu 175,9 % bei einem Datendurchsatz von 2 Mbit/s. Bei dem Einsatz des Cisco-Routers der 3600er Serie fiel der Test etwas besser aus. Trotzdem konnte man das Fazit ziehen, dass IPsec bei heutigen Software-Implementierungen nur bedingt für Echtzeitanwendungen geeignet ist. Aus diesem Grund sollte man über geeignete Hardware-Lösungen nachdenken, wenn man eine Echtzeitverschlüsselung benötigt.

Als Endresultat bleibt festzustellen, dass die Performance bei Verschlüsselung und Authentifizierung kein Produktproblem ist, sondern ein generelles Problem durch die abzuarbeitenden Algorithmen. Diese stellen die Software auf eine harte Probe, sollen sie doch auf der einen Seite möglichst hohe Schlüssel für einen hohen Sicherheitsgrad verwenden, auf der anderen Seite aber die teuren WAN-Leitungen nicht beeinträchtigen. Hier ist sicherlich ein Kompromiss zwischen Leistung und Sicherheit erforderlich. In Zukunft wird man aber nicht an der Implementierung von IPsec & Co. in Hardware vorbeikommen, um eine höhere Leistung zu erhalten. Dabei ist man aber wieder in der Bewegungsfreiheit eingeschränkt. Hier muss man einen Kompromiss finden, der auf die jeweiligen Anforderungen des Unternehmens abgestimmt sein muss.

7.2 Quality-of-Service (QoS)

Nachdem die Sicherheitsplattform gemessen wurde, findet hier ein Test der garantierten Dienstgüte statt. Dabei wurde eine ATM-basierte Lösung als Beispiel für Hard-State QoS und DiffServ für Soft-State QoS verwendet. Ebenfalls wird hier der DiffServ-Ansatz mit ATM-QoS verglichen. Dabei müssen vorab zwei wichtige Eigenschaften multimedialer Anwendungen beachtet werden:

1. Multimediale Anwendungen werden in der Regel über das UDP-Protokoll übertragen. Diese Datenströme verhalten sich „aggressiv“ im Netz. Da im UDP-Protokoll keine Maßnahmen zur Flusskontrolle vorgesehen sind, reagiert das Protokoll nicht auf Paketverluste. Tritt auf einer Verbindung eine Konkurrenz zwischen einem TCP und einem UDP Datenstrom auf, so wird der TCP-Datenstrom verdrängt.

2. Es wird speziell bei der Videoübertragung eine Flusskontrolle auf Anwendungsebene durchgeführt. Treten Verluste im Netz auf, so werden Bilder neu angefordert; diese werden zusätzlich zu den regulären Rahmen übertragen. Somit kommt es vor, dass in Zeiten der Überlast im Netz diese Anwendungen die Überlastsituation noch verschärfen. Man muss in solchen Anwendungen Mechanismen zur Anpassung der Datenrate in Zeiten der Überlast im Netz vorsehen. [BBCD98]

Gerade der erste Punkt ist unbedingt zu beachten, da es bislang kaum Erfahrungen gibt, was einen großen UDP-Anteil im Internet betrifft. An einem Beispielszenario sei die Bedeutung noch einmal herausgehoben: Zwei Datenquellen beginnen gleichzeitig mit jeweils 6 Mbit/s Daten zu senden. Beide Datenströme laufen dabei über einen 10 Mbit/s-Trunk. Da die Summe der Übertragungsraten größer als die verfügbare Datenrate auf dem Trunk ist, läuft die Ausgangswarteschlange zu einem bestimmten Zeitpunkt über. Hier kann man davon ausgehen, dass die Datenpakete zeitlich äquidistant ausgesendet werden. Somit ist auch die Wahrscheinlichkeit, dass ein Paket verworfen wird, für beide Datenströme gleich. Bei Beginn des Überlaufs würden also jeweils 20 % der Pakete von jedem Datenstrom verworfen. Sobald Pakete des TCP-Datenstroms verworfen werden, kommen die Flusskontroll-Mechanismen von TCP zum Einsatz[3], wodurch der Sender die Datenrate heruntersetzt. Der UDP-Sender reagiert jedoch auf den Paketverlust nicht und sendet weiter mit 6 Mbit/s. Der TCP-Sender wird die Datenrate erhöhen, bis wiederholt ein Paket aus diesem Datenstrom verworfen wird. Nach einer gewissen Zeit stellt sich ein quasistationärer Zustand ein, in dem maximal 12% der UDP-Pakete verloren gehen und 8% der TCP-Pakete wiederholt gesendet werden. Der UDP-Durchsatz wird ca. 53% der Trunk-Datenrate und der TCP-Datenstrom 47%[4]. Mit steigender Überlast auf dem Trunk wird die Verschiebung der Datenratenverteilung zugunsten des UDP-Datenstroms immer stärker. [SIEM00]

7.2.1 ATM-QoS

Messaufbau

Das eingesetzte Testequipment und die Netzwerkanalysatoren waren für die Erfassung der gewünschten Parameter ausschlaggebend. Die Messungen fanden in eigener Laborumgebung statt und beinhalteten einen Großversuch, an dem folgende Hersteller teilnahmen:

- **Bay Networks (Nortel Networks):** ATM-Switch Centillion 100 mit ATM-Modul ATMSpeed MDA MCP; Durchsatz: bis 3,2 Gbit/s; Non-Blocking; Shared Memory Switching, 4-Mbyte-Paketpuffer

3 Congestion-Avoidance-Mechanismen im TCP

4 In der Realität werden die Datenpakete nicht ideal äquidistant gesendet, wodurch bereits viel größere Unterschiede gemessen wurden; somit handelt es sich bei dieser Betrachtung immer um eine Unterschätzung!

- **Bay Networks (Nortel Networks):** ATM-Switch Centillion 1600 mit ATM OC-3 Modul 155M-SMFS; Durchsatz: bis 10 Gbit/s; Non-Blocking; Output-Input-Buffering: 65.536 Zellen/Slot
- **Cabletron (Enterasys):** ATM-Switch 9A000 mit ATM-Modul NM-4/155MMSCC; Durchsatz: bis 63 Gbit/s; Non-Blocking; Output-Buffering mit 13.300 Zellenpuffer
- **Cabletron (Enterasys):** ATM-Switch 6A000-02 mit ATM-Modul 6E132-25; Durchsatz: bis 2,45 Gbit/s, Non-Blocking; Shared Memory Buffering
- **Cisco Systems:** ATM-Switch Lightstream 1010 inkl. ATM155-Modul (OC3); Durchsatz: bis 5 Gbit/s; Non-Blocking; Output Buffering mit 65.536 Zellen/Port
- **Fore Systems (Marconi):** ATM-Switch ASX-200BX mit ATM-Modul NM-4/155MMSCSL; Traffic Shaping von mehreren Kanälen und Pfaden möglich (bis 12.000mal); Durchsatz: bis 2,5 Gbit/s; Non-Blocking; Output-Buffering mit 65.536 Zellen/Port
- **Madge (Networks):** ATM-Switch Meritage 1000 inkl. ATM155-Modul (OC3); Durchsatz: bis 10 Gbit/s; Non-Blocking; Output/Input Buffering mit 65.536 Zellen/Slot
- **Newbridge Networks (Alcatel SEL):** ATM-Switch CS3000 inkl. ATM155-Modul (OC3); Durchsatz: bis 6,4 Gbit/s; Non-Blocking; 128.000 Zellenspeicher pro Karte; Control Card: 32 MByte Standard SIMM-basiertes DRAM – erweiterbar auf 64 MByte, 8 MByte FLASH Memory, 1 MByte Batterie Non-Volatile Memory, 512 kByte Boot EPROM, 8 Queues pro Port
- **Olicom (Madge Networks):** Olicom: ATM-Switch CrossFire OC-9100 inkl. ATM155-Modul (OC3); Durchsatz: bis 2,5 Gbit/s; Non-Blocking (5 Gbit/s Switching RAM); Output Buffering: 32.000 Zellenspeicher

Alle Hersteller halfen aktiv an der Konfiguration der eigenen Switches mit, um Fehler bei der Bedienung zu vermeiden und um sicherzustellen, dass die Messszenarien in der Lage sind, die gewünschten Ergebnisse zu liefern. Auf Wunsch wurden alle Switches nur mit 155-Mbit/s-Ports mit Multimode-LWL-Kabeln und SC-Steckern ausgestattet, um auf einen gemeinsamen Nenner aufzusetzen und somit einen direkten Vergleich zu ermöglichen. Dabei sind die aufgeführten Geräte nach unterschiedlichen Testmethoden und -szenarien analysiert worden. Inzwischen sind nicht mehr alle Geräte verfügbar, die hier nachgemessen wurden. Es wurden sicherlich auch Weiterentwicklungen in Hard- und Software vorgenommen. Hier geht es allerdings um die grundsätzlichen Aussagen und nicht um die spezielle Heraushebung einzelner Geräte.

QoS-Parameter lassen sich nur im laufenden Betrieb testen. Dafür gibt es zwei Überwachungsmöglichkeiten, den Inband- und Outband-Service-Test. Bei der ersten Möglichkeit analysiert man den Bandbreitenverlauf, die HEC[5]-

5 Header Error Control

Fehler und die fehlerhaften Sektoren auf die Bitübertragungsschicht bezogen. Dabei muss der Analysator in der Lage sein, sich in einen passiven Monitormodus einzuschleifen. Beim Outband-Service-Test verwendet man spezielle Testzellen nach der Empfehlung ITU-T O.191. Durch diese speziellen Testzellen, die für die ATM-Messungen erzeugt werden, lassen sich die gewünschten Parameter ermitteln. Dazu werden zwei Analysatoren benötigt, zwischen denen ein virtueller Kanal aufgebaut wird. Die Testzellen werden dann über den Übertragungspfad gesendet und auf der Empfangsseite ausgewertet. Die bereits beschriebenen Verkehrsparameter werden durch die eingestellte Last signifikant beeinflusst. Interessante Parameter sind dabei die Zellenverzögerungsschwankung CDV[6], die bei steigender Last zunehmen kann, und das Zellenverlustverhältnis CLR[7], welches sich verschlechtert. Der Switch muss während der Messungen so belastet werden, dass die Lastgrenze eines realen Betriebs erreicht wird. Dadurch lassen sich Aussagen bezüglich der Einsatzgebiete, Leistungsfähigkeit und Ausfallsicherheit treffen. Zusätzlich ist die mögliche Überbelastung (Overload) einer Verbindung interessant. Hierbei ist der ATM-Switch gezwungen, seine internen Puffer einzubeziehen, da er sonst die Datenrate nicht bewältigen kann. Wächst hierbei der CDV-Wert kontinuierlich, wird er bei konstanter Belastung ab einem bestimmten Zeitpunkt überlastet und bricht die Verbindung ab. Interessant ist hierbei, welche Datenmengen die jeweiligen Switches dann noch verkraften können.

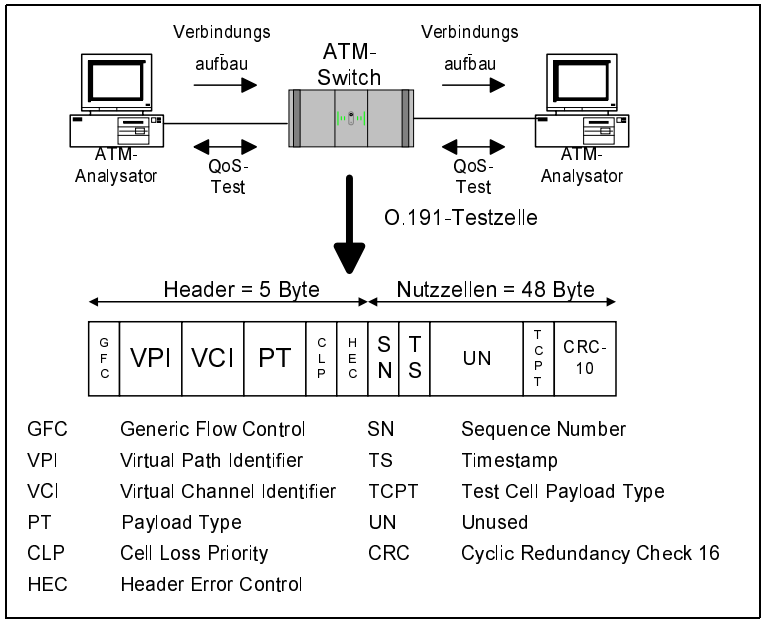


Abb. 7.7
Test der QoS-Parameter

6 Cell Delay Variation
7 Cell Loss Ratio

Die ATM-Zellen der virtuellen Kanal- und Pfadverbindungen werden bei diesem Szenario ebenfalls einem Test unterzogen. Dadurch kann man aktive Kanäle sofort erkennen. Weiterhin lassen sich Zellfilter einsetzen, um die Gesamtzahl der Zellen bzw. die Zellrate je Kanal zu ermitteln. Hierbei helfen definierte Lastprofile mit Hilfe der Testzellen bei den QoS- und Stresstest-Messungen. Abb. 7.7 zeigt die O.191-Testzelle, durch die auf der ATM-Schicht der QoS nachgewiesen werden kann. O.191 beschreibt weiterhin ein Diagnosemodell zur Performance-Analyse, für die Testzellen über eine vereinbarte virtuelle Verbindung transportiert werden. Dabei ist zu beachten, dass es sich um Performance-Messungen auf Zellenbasis handelt und somit auf der ATM-Schicht. Die Funktionalität und Leistungsfähigkeit der einzelnen Anpassungsschichten (AAL) müsste man separat betrachten. Bei der Testzelle ist besonders auf die Felder Sequence Number (SN) und Time Stamp (TS) zu achten. SN ist für die Erfassung der Zellverluste (Cell Loss) und Zellintegrität (Cell Integrity) verantwortlich, während TS die Zellenverzögerung (Cell Delay), den Zellen-Jitter (Cell Jitter) und die Zellenverteilung (Cell Distribution) misst. Darüber hinaus lässt sich auch der Wert CTD erfassen. Das Feld Test Cell Payload Type (TCPT) zeigt die Versionsnummer der O.191-Zelle an. Das Feld CRC-16 ist für die Erkennung von Bitfehlern zuständig. Das heißt, alle fehlerhaften Zellen werden aus dem Datenstrom entfernt und nicht einbezogen. Korrekturen können dabei nicht vorgenommen werden.

Stresstests

Durch die Definition der Testszenarien kann erst eine Bewertung der ATM-Switches als Netzwerkelemente unter Verwendung der NPP auf der ATM-Schicht stattfinden. Parameter auf den AAL-Schichten werden dabei nicht erfasst, da für den Anwenderverkehr das AAL-Protokoll in reinen ATM-Einrichtungen üblicherweise nur als End-to-end-Protokoll gefahren wird.

Zur Bewertung der ATM-Switches als Netzwerkelemente werden diese mit geeignetem Testverkehr gefahren. Die Switches werden hierbei als „Black Box" betrachtet. Zur Charakterisierung werden die relative Transportrate[8] und die Verzögerung in Abhängigkeit von der Verkehrslast herangezogen. Am Anstieg dieser Kennwerte in Abhängigkeit der Verkehrslast lassen sich Blockierungsmechanismen und damit die Leistungsgrenze der Switches erkennen. Die Switches werden lokal betrachtet, also als isolierte Elemente betrieben. Als kritische Lastsituationen des Switches werden Multiplex- und Demultiplexbetrieb mit unterschiedlichen Dienstklassen (CBR[9], VBR[10] und UBR[11]) betrieben. Dadurch las-

8 Verhältnis empfangener Zellen zu gesendeten Zellen am Zielport
9 Constant Bit Rate
10 Variable Bit Rate
11 Unspecified Bit Rate

sen sich unterschiedliche Verkehrsprofile am Switch untersuchen, um die Leistungsfähigkeit beurteilen zu können.

Im ersten Schritt generiert das Analysesystem einen CBR-Datenstrom und leitet ihn an zwei Ports des ATM-Switches weiter. Dadurch wird die Zellenverzögerung erfasst. Das Verkehrsmanagement bleibt während der Messungen ausgeschaltet, weil nur ein Datenstrom von konstant 7,488 Mbit/s anliegt. Danach folgen Messungen mit den Dienstklassen VBR und UBR. Anschließend wird die Nettoübertragungsrate auf 149,8 Mbit/s gesteigert und erneut der CBR -, VBR- und UBR-Verkehr gemessen, diesmal jedoch an vier Ports (Abb. 7.8). Mit zunehmender Belastung des Switches steigt die Zahl der Time-outs; auch die Latenzzeiten werden größer. Das Gerät sollte jedoch diese Übertragungsrate verkraften können, wenn es seine internen Puffer heranzieht. Die virtuelle Verbindung wird durch einen PVC (VPI=1,VCI=33) über den ABT-20 aufgebaut. Latenzzeiten sind vor allem für Echtzeitapplikationen problematisch. Switches sollten daher die Zellen schnell und konsistent übertragen. Ob sie das tun, zeigt CDV, die mit Jitter vergleichbar ist. Als Referenzwert dient die Standardabweichung der Zellenverzögerung, die bei einer Abtastung von 2,5 Millionen Zellen entsteht.

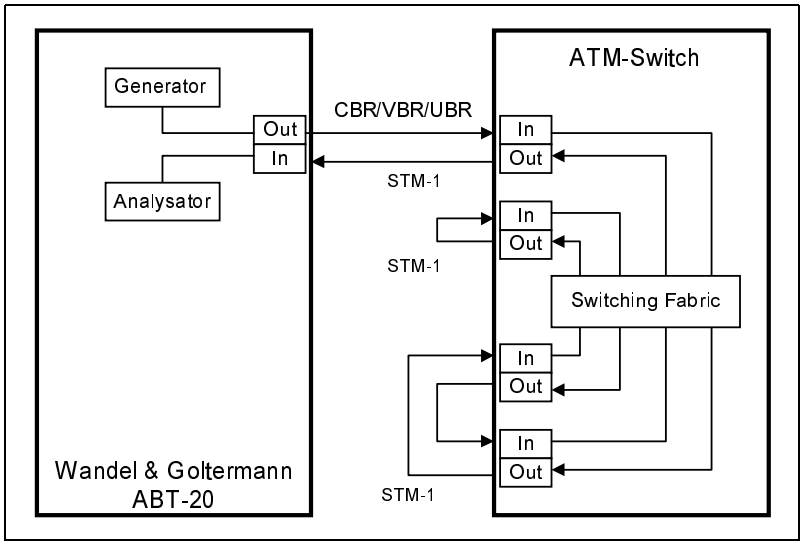


Abb. 7.8 Switching zwischen vier Ports

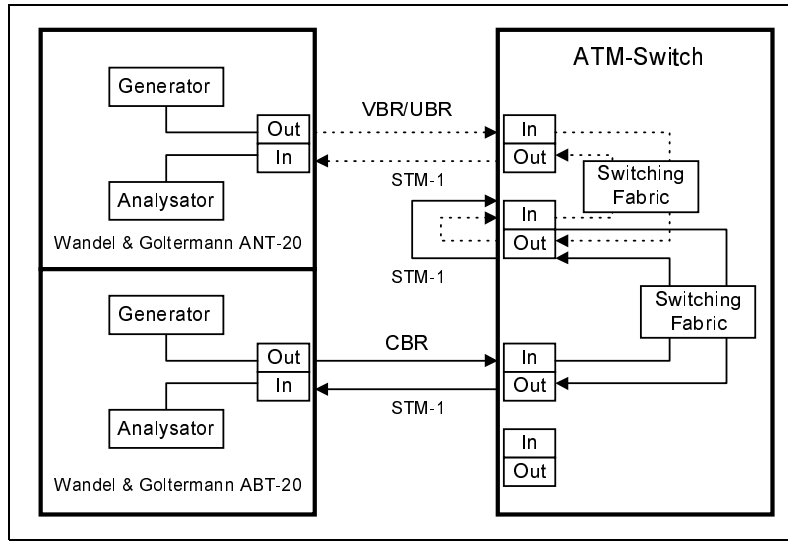
Verkehrsmanagement (Traffic Policing)

Switching ist weit mehr, als Zellen so schnell wie möglich vom Eingangs- zum Ausgangs-Port zu leiten. Der Switch muss mit unterschiedlichen Verkehrsarten, Datenraten und Belastungen zurechtkommen. Eine Schlüsselrolle spielt das Verkehrsmanagement. Seine Qualität kann der Experte mit Hilfe von zwei Datenströmen untersuchen: Burst-artiger VBR-/UBR-Verkehr und verzögerungsempfindlicher CBR-Verkehr.

Der UBR-Datenstrom simuliert IP-Traffic, während VBR für eine zusätzliche Datenlast sorgt. Der Versuch macht deutlich, ob ein Switch CBR-Verkehr ohne zusätzliche Latenzzeiten und Unterbrechungen aufrecht erhalten kann. Dazu wird der VBR-/UBR-Verkehr kontinuierlich von 10 auf 90 % Last erhöht. Als Folge davon steigt die Latenzzeit des CBR-Verkehrs. Die Messwerte zeigen, wie die Mechanismen zur Flusskontrolle und Zwischenspeicherung zusammenarbeiten. Switches verwenden dabei folgende Verfahren:

- Separate Pufferwarteschlangen für hohe Priorität des Datenverkehrs
- Usage Parameter Control (UPC), um Bandbreite zu reservieren
- Cell Loss Priority (CLP), um Datenverkehr zu kennzeichnen, der verworfen werden kann
- Proprietäre Fairness-Algorithmen, um dedizierte Bandbreite für spezielle Verbindungen bereitstellen zu können

Abb. 7.9 Switching unter Einbeziehung des Verkehrsmanagements



Stehen separate Pufferwarteschlangen zur Verfügung, werden CBR-Daten mit höchster Priorität verarbeitet und in die Ausgangspuffer geleitet. VBR-Daten erhalten eine niedrigere Priorität. UPC ist ein Teil des Traffic Management 4.0 und besteht aus Algorithmen, die mit den virtuellen Verbindungen zusammenarbeiten, um Bandbreite zuzuweisen. Weil ATM keine physikalische Begrenzung der Zugriffsrate kennt, ist der Zellenstrom jeder Pfad- und Kanalverbindung zu überwachen.

Eine Kontrolle für UBR-Daten bietet der Leaky-Bucket-Algorithmus. Er verwirft jede Zelle, die den definierten Durchsatz nicht einhält. Der zweifache Leaky-Bucket-Algorithmus gilt zusätzlich für VBR-Zellen. Er prüft, ob die Peak

Cell Rate (PCR) und die kontinuierliche Zellrate SCR[12] eingehalten werden. Fairness-Mechanismen erlauben einer Verbindung, die ausgehandelten Anforderungen zu überschreiten, so lange nicht andere VCCs am selben Port davon betroffen sind. Ein entsprechender Algorithmus sorgt beispielsweise dafür, dass eine virtuelle Verbindung eine feste Bandbreite erhält, sobald der Switch die Zellen nur noch mit Verzögerung weiterleiten kann.

Um Verkehrsspitzen abzufangen, legte die ITU-T zwei Möglichkeiten fest. Eine verordnet Zellen eine höhere Cell Loss Priority (CLP=1) und wirft sie weg, wenn die PCR überschritten wird. Die zweite ist das Traffic Shaping: Es verschiebt Zellen in einen Zeitabschnitt mit geringerer Belastung. Der Nachteil dabei ist, dass die Daten zwischengespeichert werden müssen, was wiederum zu Verzögerungen führen kann. Die verwendeten Messgeräte besaßen einen Traffic Shaper. Er überprüft, ob ein virtueller Kanal seinen Verkehrsvertrag einhält. Der Shaper rechnet nach, ob die gesendeten Zellen im Einklang mit den Kontraktparametern stehen. Wenn nicht, passt er den Verkehr an und versucht dabei, die Datenrate zu halten. Das klappt jedoch nicht immer; etwa dann nicht, wenn bei der Datenquelle eine höhere Zellrate eingestellt wurde, als der Traffic Contract zulässt. Dann muss der Shaper Zellen verwerfen. Bei ausgeschaltetem Shaper kann man prüfen, bis zu welchem Grad ein ATM-Switch und somit das Netz eine Überbuchung zulässt.

Non-Blocking-Mechanismus

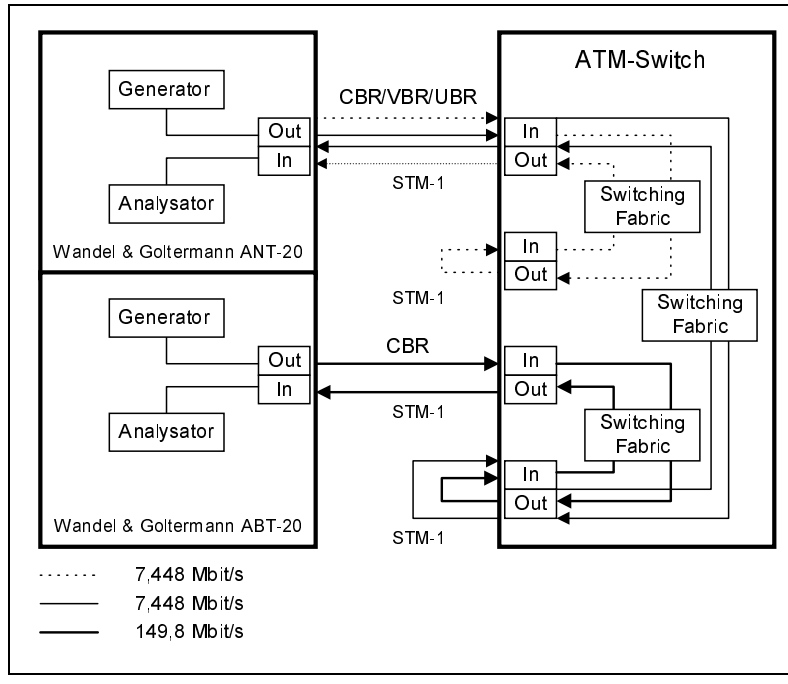
In der Praxis kommt es immer wieder vor, dass auf einem Ausgangs-Port Überlast auftritt, die den Durchsatz an einem unbelasteten Port beeinflusst. Dieses wird Head-of-the-Line-Blocking (HoLB) genannt und zeigt die Grenzen einer Switch-Architektur auf, die unterschiedliche Konzepte aufweisen kann:

- **Geräte mit Eingangsspeicher:** häufiges Blockieren möglich, wenn mehrere Eingänge auf den gleichen Ausgang zugreifen, wodurch nur eine maximale Verkehrslast von 58% bei 16 Eingängen erreicht werden kann.
- **Systeme mit Ausgangsspeicher:** ebenfalls häufigeres Blockieren möglich, bei Überbelegung eines Ausgangs-Ports. Allerdings kann hier eine bessere Performance von bis zu 80% Verkehrslast bei 16 Eingangs-Ports erreicht werden.
- **Switches mit zentralem Speicher:** Weniger empfindlich, da über ihn alle Zugriffe von Eingangs-Ports auf den gleichen Ausgangs-Port laufen. Zellen gehen nur dann verloren, wenn das Fassungsvermögen des gesamten Speichers erschöpft ist. Nachteilig ist eine komplizierte Steuerung und eine hohe Zugriffsgeschwindigkeit auf den Speicher.

12 Sustainable Cell Rate

- **Switches mit verteiltem Speicher:** Ähnliches Arbeitsverhalten wie bei zentralem Speicher. Es wird aber mehr Speicher benötigt, weil die Puffer verteilt sind. Dagegen ist das gesteuerte Auslesen der verteilten Speicher weniger aufwendig als beim zentralen Speicher.

Abb. 7.10 Testszenario Head-of-the-Line-Blocking



Um Non-Blocking-Mechanismen zu testen, empfiehlt sich ein Szenario mit zwei Datenströmen zu einem Eingangs-Port. Beide Ströme werden auf zwei Ausgangs-Ports geswitcht. Ein weiterer Datenstrom einer zweiten Quelle mit 149,8 Mbit/s sorgt anschließend dafür, dass an einem Port eine Überlastsituation eintritt.

Jitter

Unter Jitter versteht man normalerweise die zeitliche Abweichung der Flanken eines digitalen Signals gegenüber seiner ursprünglichen Position. Jitter sind durch ihre Amplitudengröße und Frequenz bestimmbar. Bei sehr kleinen Frequenzen wird der Jitter auch als Wander bezeichnet. Davon betroffene Signale werden außerhalb des normalen Abtastzeitpunkts abgetastet, was zu vereinzelten Fehlern oder ganzen Fehlerbüscheln führen kann.

Es sind unterschiedliche Ursachen für die Entstehung von Jitter verantwortlich. Es wird zwischen systematischen und nicht systematischen Jitter unterschieden. Die erste Art ist abhängig von der übertragenden Bitfolge.

Durch Leitungsverzerrungen kann ein musterabhängiges Impulsnebensprechen entstehen, welches durch leicht fehlerhafte Signalentzerrungen Jitter erzeugt. Die zweite Art von Jitter entsteht durch elektromagnetisches Übersprechen interner oder externer Störsignale (Rauschen, Reflexionen, Nebensprechen usw.). Bei heutiger Übertragung in digitalen Systemen wird die Jitter-Problematik noch deutlicher, da hier entstehende Signalabweichungen stärkere Störungen verursachen. Deshalb müssen in digitalen Übertragungssystemen die Eigen-Jitter möglichst gering gehalten werden. Zusätzlich dürfen Netzkomponenten vorhandene Jitter nicht noch zusätzlich verstärken. [DETK99b]

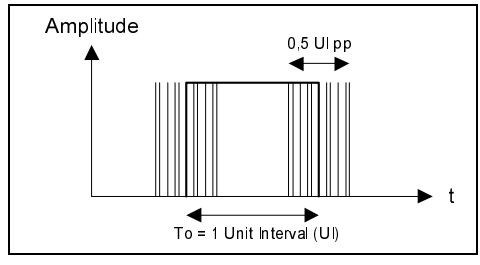


Abb. 7.11 Kennzeichnung des Unit-Intervalls (UI)

Messergebnisse

Nun erfolgt die Auswertung der gemessenen Ergebnisse der unterschiedlichen Bereiche. Ein Vergleich der unterschiedlichen Messungen und Werte ist durch die konsequente Verwendung gleicher Messgeräte möglich geworden. Am Ende der Auswertung kann deshalb eine Abschätzung erfolgen, die die Leistungsfähigkeit der ATM-Switches, Verfügbarkeit und Ausfallsicherheit betrifft. Zusätzlich lassen sich Rückschlüsse auf die zu unterstützenden Applikationen schließen.

2/4-Port-Messungen

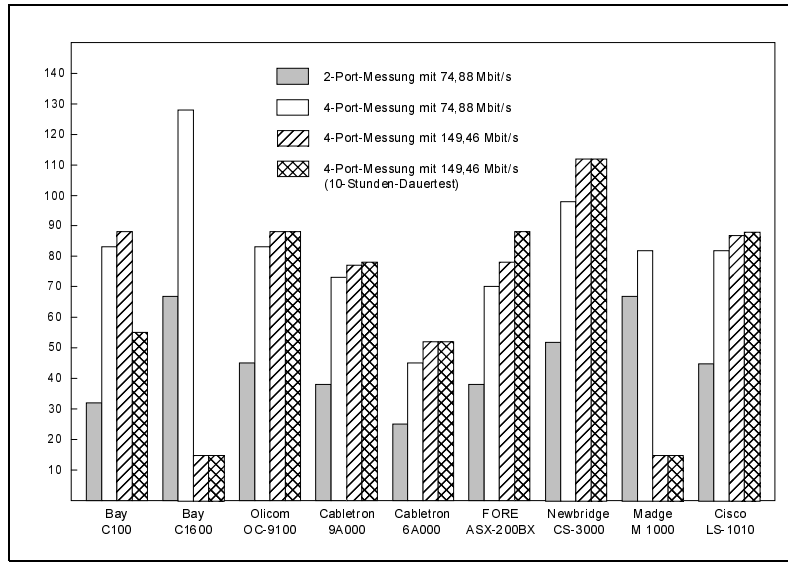
Um einen ersten Eindruck von der Leistungsfähigkeit der Switches zu erhalten, wurden die ersten Messungen mit einer geringen Belastung von 74,88 Mbit/s durchgeführt. Audio- und Videodatenströme stellen allerdings besonders hohe Anforderungen: Die Verzögerung CTD darf dabei max. 50 ms betragen. Die Testgeräte schafften in diesem Stadium Werte zwischen 25 µs (Cabletron 6A000) und 66 µs (Centillion 1600 und Meritage 1000), liegen also deutlich unter dieser Schwelle. Da es sich bei der Messanordnung um eine Schleifenverbindung handelte, sind die gemessenen Werte der 2-Port-Messung noch durch den Faktor zwei zu teilen. Damit schnitten die Testgeräte besser ab, als es zunächst den Anschein hatte.

Im nächsten Schritt wurde die Anzahl der Ports auf vier und die Nettodatenrate auf 149,46 Mbit/s erhöht. Damit gerieten die Switches bereits in den Bereich der Grenzbelastung. Aus diesem Grund stieg der CTD-Wert steil an. Besonders deutlich wird dieser Effekt beim Fore-Switch ASX-200BX und beim

Cabletron 9A000. Nachdem die Signalisierung ausgeschaltet wurde, näherten sich die Werte wieder ihrer normalen Höhe. Die Ähnlichkeit zwischen diesen beiden Switches ist im Übrigen kein Zufall: Cabletron bezog seinen 9A000 als OEM von Fore Systems. Die Systeme sind aber nicht genau baugleich.

Der LS-1010 von Cisco ließ sich als einziger Switch nicht einmal für die maximale Nettodatenrate von knapp 150 Mbit/s konfigurieren. Das Cisco-Betriebssystem IOS erteilte keine Auskunft darüber, warum das Gerät die Verbindung nicht annahm, sondern gab lediglich die Fehlermeldung „Connection is different" aus. Deshalb wurde der Switch für nur 139 Mbit/s konfiguriert, aber dennoch in seiner PCR mit der vollen Nettodatenrate angesteuert. Erfreulicherweise kam es dabei nicht zu Zellverlusten; auch die Latenzzeiten stiegen nicht weiter an.

Abb. 7.12 /4-Port-Messung, Mean. CTD [µs]



Mit dieser Messanordnung wurde anschließend einen 10-Stunden-Messung durchgeführt. Die Geräte von Bay Networks, Cisco, Newbridge, Madge und Olicom zeigten hierbei keinerlei Ermüdungserscheinungen. Sogar ein dreitägiger Dauerbetrieb konnte den Switch von Olicom nicht dazu bewegen, auch nur einen einzigen Zellenfehler zu verursachen. Schlechter lief es bei den Cabletron-Switches: Unter dem Dauerstress verfälschten sie einige Testzellen; der 9A000 verlor auch eine Reihe von Zellen. Der 6A000 des gleichen Herstellers verursachte keine Zellverluste, wartete dafür aber mit Zellenfehlern auf, die sich ebenfalls negativ auswirken.

Das Phänomen war zwar nicht gravierend; mit einer auf den Gesamtdatenstrom bezogenen Fehlerrate von 5,04 x 10^{-9} oder besser liegen die Geräte im

zulässigen Rahmen. Im realen Betrieb können diese Fehler dennoch deutlich ins Gewicht fallen, weil der Verlust einer Zelle auf der ATM-Schicht die Zerstörung einer Protocol Data Unit (PDU) mit einem Umfang bis zu 65 Kbyte zur Folge hat. Diese muss dann nochmals gesendet werden, was den effektiven Durchsatz senkt. Außerdem lässt der Umgang mit den Daten Rückschlüsse auf die Verfügbarkeit und Ausfallsicherheit zu. Parallel zu den bisherigen Messungen wurden Schwankungen im Laufzeitverhalten der Switches aufgezeichnet. Dieser Wert sagt etwas über die subjektiv vom Teilnehmer wahrgenommene Dienstgüte aus, vor allem bei der Übertragung von Audio- und Videodaten. Ein starker Anstieg deutet darauf hin, dass der Switch an seiner Lastgrenze betrieben wird. Die Messungen ließen keine Probleme erkennen.

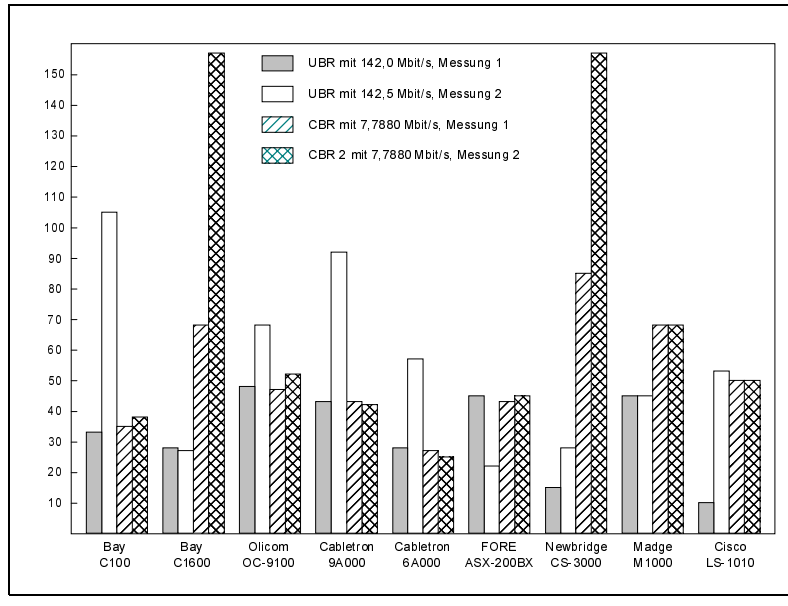
3-Port-Messungen

Ein wichtiges Merkmal von ATM ist die garantierte Bandbreite für bestimmte Verkehrsarten. Für die Praxis ist es wichtig zu wissen, inwieweit ein Switch diese Verpflichtung auch dann noch einhält, wenn er mehr Daten zur Weiterleitung erhält, als er eigentlich verarbeiten kann. Daher wurden hier drei Datenströme erzeugt, die auf einen gemeinsamen Ausgangs-Port aufliefen und diesen überlasteten. Bei dieser Messreihe wurden die Switches mit einem Datenstrom hoher Priorität mit CBR von rund 7,5 Mbit/s und einem Datenstrom mit niedriger Priorität mit UBR von 142,5 Mbit/s angesteuert. Damit kam eine Überlastsituation zustande. Als erste Gegenmaßnahme nehmen die Geräte dann den internen Puffer in Anspruch. Wenn der Pufferspeicher die Überbelastung nicht mehr kompensieren kann, beginnt der Switch, Zellen zu verwerfen. CBR-Datenströme sind dabei in jedem Fall zu schützen; Jitter, Verzögerung und Laufzeitschwankungen dürfen nicht über Gebühr ansteigen. Der Switch darf nur den UBR-Verkehr zurücknehmen.

Hierbei fiel der ASX-200BX von Fore Systems positiv auf: Mit seinem umfangreichen Pufferspeicher (65.536 Zellen) schaffte er es, volle anderthalb Minuten lang der Überbelastung standzuhalten, bevor er Zellen verwerfen musste. Das ist für den praktischen Betrieb von großem Vorteil und wirkte sich auch auf den UBR-Verkehr sehr positiv aus. Weiterhin fiel auf, dass der 6A000 von Cabletron auch dem CBR-Datenstrom geringe Verluste zumutet, während alle anderen Switches ausschließlich UBR-Verkehr verwerfen. Das hängt damit zusammen, dass dieses Gerät den Eingangs-Port des UBR-Verkehrs bei Überbelastung abschaltet. Das heißt, nachdem der Puffer an seine Leistungsgrenze gekommen ist, bricht der Port die Verbindung ab. Der CBR-Verkehr ist davon zwar nicht direkt betroffen, es kommt allerdings auch hier zu geringen Zellverlusten. Wenn der Eingangs-Port abgeschaltet ist, treten keine weiteren Probleme mehr auf. Die Abschaltung des Port ist allerdings nur durch einen Reset zu beheben. Das darf im praktischen Einsatz nicht passieren.

Am besten schnitt in diesem Punkt der Madge Meritage 1000 ab, der beide Datenströme ohne Verluste meistert. Auch das Cabletron-Modell 9A000 zeigte sich von seiner positiven Seite: Es verwarf nur rund 100.000 Zellen des UBR-Verkehrs. Das ist ein sehr gutes Ergebnis. Als deutlich schwächer erwies sich in dieser Disziplin der CS3000 von Newbridge: es wurden über eine halbe Million Zellen verworfen. Dadurch konnte er nicht die gleiche effektive Übertragungsrate erreichen wie die anderen Switches. Der Centillion 100 von Bay Networks berücksichtigte mit der vorliegenden Hardware keine QoS-Merkmale; deshalb musste der Tester sich mit einer einfachen Prioritätsvergabe begnügen. Ein direkter Vergleich mit den anderen Testgeräten ist auf dieser Basis allerdings nicht möglich.

Abb. 7.13
Port-Messung UBR/ CBR, Mean. CTD [µs]



Im nächsten Schritt wurden zwei CBR-Verkehrsströme auf denselben Ausgangs-Port geleitet. Die Datenraten betrugen 142,5 Mbit/s und 7,79 Mbit/s. Weil in diesem Fall beide Ströme die gleiche Priorität aufwiesen, konnte sie der Switch nicht mehr gleichermaßen schützen. In dieser Situation sind drei Verhaltensweisen denkbar:

- Der Switch ermöglicht wenigstens einem der beiden Datenströme, die eingestellte Dienstgüte zu halten.
- Der Switch mutet beiden Verbindungen Verluste und Verzögerungen zu.
- Der Switch lehnt die gesamte Verbindung ab, um eine Überbuchung gar nicht erst zuzulassen.

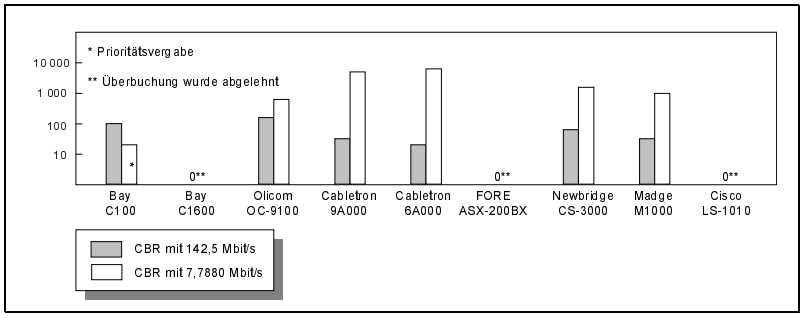


Abb. 7.14
Port-Messung CBR/ CBR, Mean. CTD [µs]

Die einzelnen Testgeräte reagieren sehr unterschiedlich auf die Überbuchung. Der Centillion 100 von Bay Networks und der ASX-200BX von Fore Systems besaßen die größte Beständigkeit[13]. Gleiche Priorität mit Überbuchung ließ sich nicht konfigurieren, da der Switch solche Verbindungen ablehnte. Das ist sinnvoll, wenn man ein komplexes ATM-Netz betreiben will, weil der Administrator nicht über jede permanente PVC-Verbindung Bescheid wissen kann und sich überlagernde temporäre SVC-Verbindungen zusätzlich auf die Gesamtbelastung auswirken können.

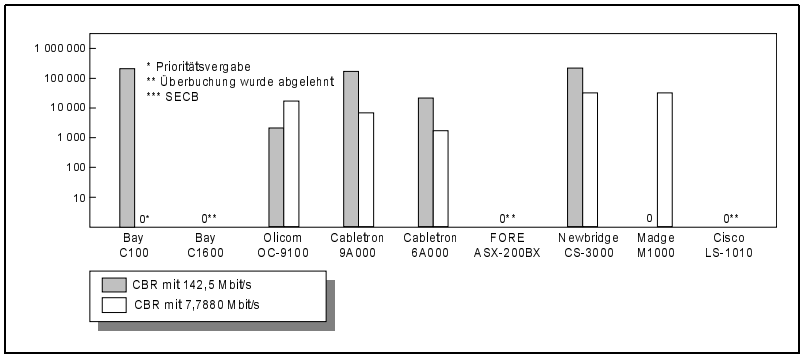


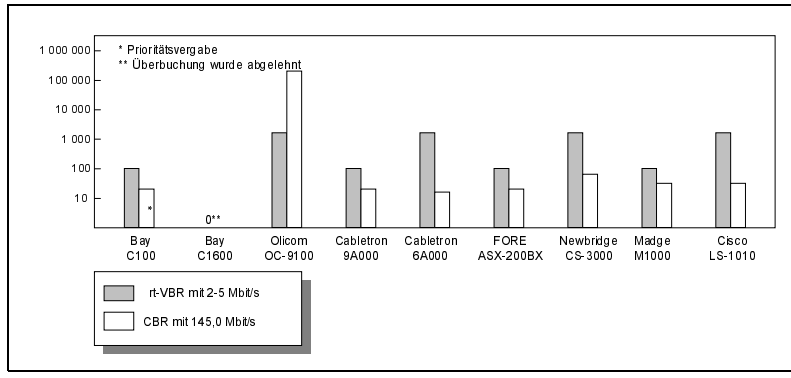
Abb. 7.15
Port-Messung CBR/ CBR, Cell Loss

Der Fore-Switch weist die geringste Zellentransferverzögerung auf. Er arbeitet in dieser Überlastsituation nur geringfügig langsamer als bei Normalbelastung. Auch der OC-9100 von Olicom machte einen sehr soliden Eindruck, obwohl hier die Latenzzeit auf beiden CBR-Verbindungen stark zunimmt. Die beiden Cabletron-Geräte legen sich auf einen CBR-Datenstrom fest: Sie halten jeweils die Verbindung mit der hohen Datenrate konstant. Das geht zu Lasten der Verbindung mit der niedrigen Datenrate, obwohl sie die gleiche Priorität besitzt. Dasselbe passiert beim CS3000 von Newbridge und dem Meritage 1000 von Madge. Unter bestimmten Überlastbedingungen können sich die Puffer mehrerer Ports gegenseitig beeinflussen. Dabei kann es vorkommen, dass auch

13 Unter der Berücksichtigung, dass der Centillion 100 kein QoS besitzt

unbelastete Ports Zellen verwerfen, obwohl sie noch über freie Kapazitäten verfügen[14]. In die Bewertung des Überlastverhaltens gehen allerdings auch die Einflüsse der Pufferarchitektur ein.

Abb. 7.16
Port-Messung VBR-CBR, Mean. CTD [µs]



Head-of-the-Line-Blocking (HoLB)

Um zu testen, ob dieser Effekt bei den getesteten Switches eintritt, wurde ein weiterer Datenstrom hinzugefügt und belastete damit die Ausgangs-Ports noch stärker als bisher. Bei der ersten Messung wurden nacheinander drei CBR-Verkehrsströme etabliert. Eine der drei Verbindungen soll dabei unbedingt geschützt werden, weil sie auf einen unbelasteten Ausgangs-Port geführt wird, während die beiden anderen Verkehrsströme eine Überlast über die maximal zulässige Nettodatenrate hinaus verursachen. Im ersten Moment kann ein Switch die Überbelastung wieder durch seinen Pufferspeicher abfangen. Ist dieser voll, so ist er aber genötigt, einzelne Zellen oder ganze Zellenblöcke zu verwerfen. Es darf dabei nicht zur Beeinflussung anderer Ausgangs-Ports kommen.

In dieser Situation zeigte sich, dass die meisten Switches den größeren Datenstrom mit einer höheren Priorität versehen als die beiden Verbindungen mit 7,448 Mbit/s. Der 6A000 von Cabletron setzt die Prioritäten allerdings umgekehrt. Zwei Geräte, der ASX-200BX von Fore Systems und der Bay Centillion 1600, ließen überhaupt keine Überbuchung auf einen Ausgangs-Port zu. Dies ist vom Hersteller auch so beabsichtigt und hat durchaus seinen Sinn, da eine Überbuchung einen Switch immer überlastet.

Als Nächstes wurde die Klassifizierung für den zweiten Datenstrom von CBR auf die Verkehrsart nrt-VBR geändert, die eine größere Flexibilität hinsichtlich ihres Zeitverhaltens toleriert. Das sollte die Switches in die Lage versetzen, die beiden verbleibenden CBR-Verkehrsströme ohne Verluste weiterzuleiten. Die Messungen belegten, dass alle Switches dieser Erwartung entsprachen. Ein Blocking der Ports trat nicht auf.

14 Head-of-the-Line-Blocking (HoLB)

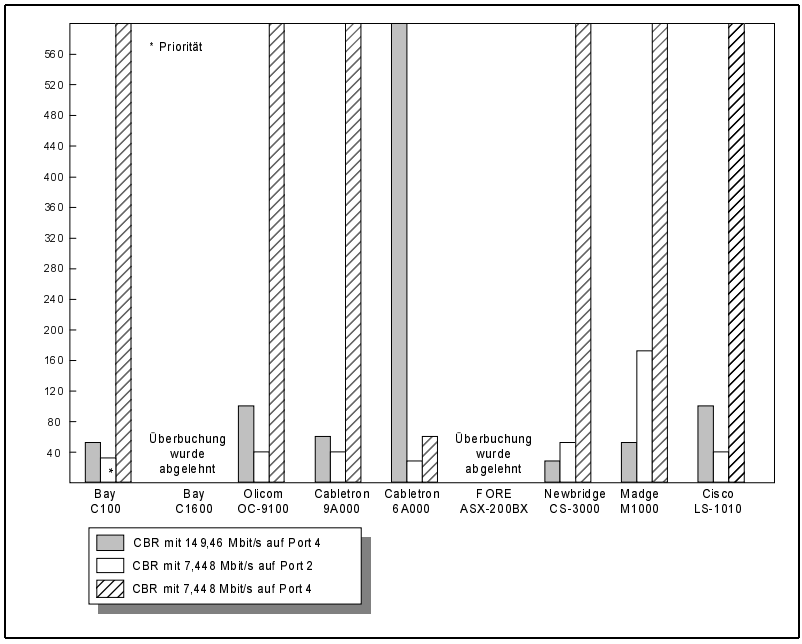


Abb. 7.17
Non-Blocking-Messung CBR-CBR-CBR, Mean. CTD [µs]

Der Newbridge CS3000 verwirft ganze Zellenblöcke des VBR-Verkehrs, nachdem der entsprechende Pufferspeicher voll ist. Beide CBR-Datenströme bleiben davon unberührt. Die Laufzeitschwankungen sind im Vergleich mit den anderen Switches relativ hoch, bleiben aber in einem akzeptablen Rahmen. Der Meritage 1000 kommt auch diesmal an seine Leistungsgrenze, ohne dass allerdings die CBR-Datenströme darunter leiden müssten. Ein ähnliches Ergebnis erzielt der LS-1010 von Cisco.

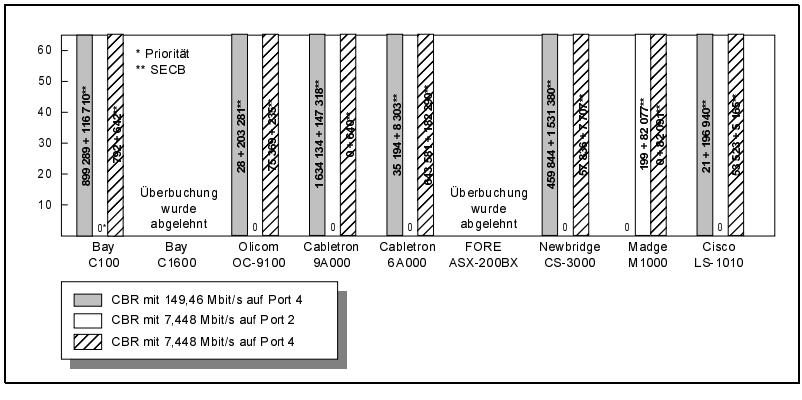
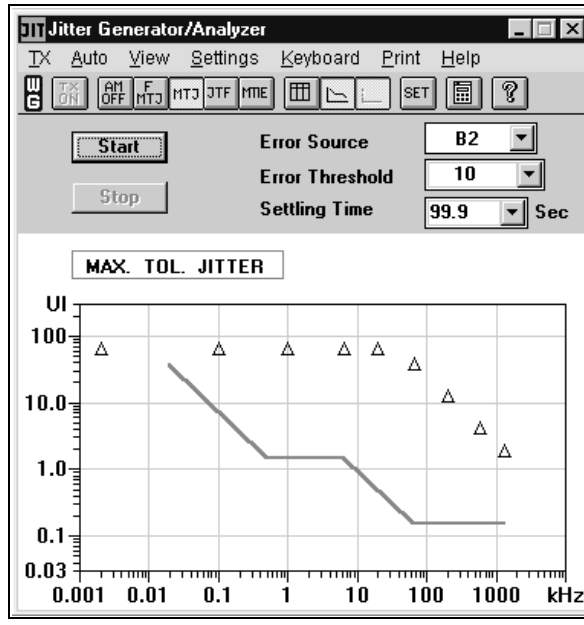


Abb. 7.18
Non-Blocking-Messung CBR-CBR-CBR, Cell Loss

Jitter

Die Jitter-Verträglichkeit ist eine Standardmessung, die nach der Installation von Systemen und Komponenten durchgeführt wird. Sie ist sowohl für das Übertragungsverhalten des Gesamtsystems als auch für einzelne Komponenten wichtig. Es wird hierbei angezeigt, ob Störereignisse wie Bitfehler, Bit-Slips oder Alarme durch zulässige Jitter-Werte verursacht werden. Die ITU-T hat hier Toleranzen definiert, die die minimale Jitter-Toleranz in Abhängigkeit von der Jitter-Frequenz beschreiben. Die Verträglichkeit wurde bei allen ATM-Switches bei sinusförmiger Jitter-Modulation gemessen.

Abb. 7.19 Max. Toleranz Jitter (MTJ)



Alle ATM-Switches besaßen bei den Messungen ähnliche Kennlinien. Dabei war bei höheren Jitter-Frequenzen der zurückgewonnene Takt nicht mehr in der Lage den schnellen Phasenänderungen zu folgen. Es entstanden so Falschabtastungen und Bitfehler. Die Jitter-Verträglichkeit nahm bei höheren Frequenzen ab. Alle Jitter befanden sich allerdings nach der Abb. 7.19 innerhalb der Toleranzwerte, wodurch keine Eigen-Jitter Probleme entstanden. [DETK99b]

Auswertung

Die ATM-Tests wurden im eigenen Labor durchgeführt, um die Performance und Qualität der ATM-Switches unter Beweis zu stellen. Dabei bezog sich die Hauptkritik auf die Dienstgüte von ATM, welche nach wie vor neben der Skalierbarkeit der größte Vorzug gegenüber anderen Technologien darstellt. Alle gemessenen Switches erfüllten die Anforderungen hinsichtlich des QoS, wobei auch Abstriche zu machen sind.

Der Switch von Olicom zeigt ein sehr robustes Verhalten und gute Messergebnisse. Nur die Transitzeiten CDT lagen teilweise über denen der anderen Switches. Das könnte sich aber selbst bei Sprachdatenübertragung nicht negativ bemerkbar machen, da hier 25 ms Verzögerung pro Richtung noch akzeptabel sind. Der Fore-Switch bietet aufgrund seines großen Speichers ein sehr gutes Handling der verschiedenen Datenströme. Das Gerät hatte keinerlei Zellverluste zu verzeichnen. Die Messungen ergaben durchweg gute Resultate. Der Madge Meritage 1000 besaß ebenfalls gute Messergebnisse. Auch die geringen Verzögerungszeiten und Schwankungen wurden positiv vermerkt. Nach Einstellung von Call Admission Control (CAC) konnte auch eine sehr effektive Überbuchung erfolgen, die sogar die mögliche Nettodatenrate übertraf.

Der CS3000 von Newbridge Networks hatte etwas schlechtere Werte. Er konnte in den Transitzeiten und der Verzögerung nicht ganz mithalten. Dieses Problem konnte er jedoch durch Robustheit und Traffic-Management ausgleichen. Der Cisco LS-1010 hinterließ einen durchaus positiven und soliden Eindruck. Hinsichtlich der Messergebnisse braucht er keineswegs hinter den anderen Modellen des Tests zurückzustehen.

Zusammenfassend hinterließen alle Switches einen recht guten Eindruck. Allerdings dürfen Zellverluste und -fehler sowie die Zulassung von Überbuchungen zweier CBR-Verbindungen auf einem Ausgangs-Port eigentlich nicht auftreten. Ansonsten kann man bei dieser Switch-Generation auch endlich QoS-Merkmale im Kernbereich des eines Netzes ausnutzen. Wenn man Anpassungs- und Integrationsverfahren ebenfalls mit diesen Merkmalen erweitert und die Anwendungssoftware anpassen würde, hätte man einen QoS-Pfad durch ein gesamtes Netz funktionsfähig geschaltet. Dabei würde allerdings ATM als Kerntechnologie die Voraussetzung sein.

7.2.2 IP-QoS

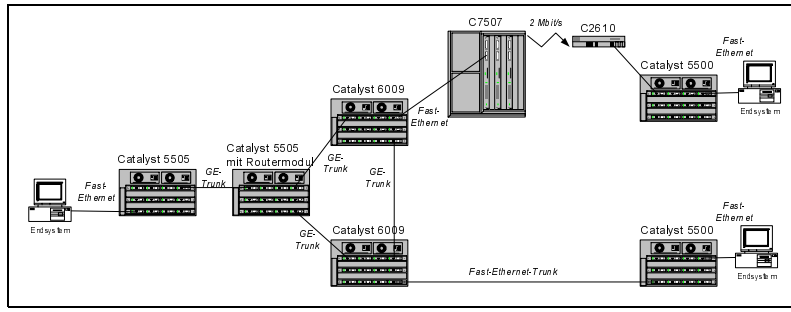
Nachdem ATM im Vordergrund der Betrachtung bei QoS stand, werden hier nun eine DiffServ-Testplattform und deren Messergebnisse dargestellt.

Messaufbau

Es wurde bereits in der Einleitung auf das Problem der Verdrängung von TCP-Datenströmen hingewiesen. Dieser Fall des Datenstroms mit UDP ist der häufigste, aber auch schwierigste Anwendungsfall für ein Queue-Management. Aus diesem Grund wurden UDP-Datenströme gemessen. Beginnt ein Sender die UDP-Datenübertragung, bevor das Aushandeln der Übertragungsparameter anderer Clients über die TCP-Kontrollverbindung abgeschlossen ist, so wird der Datenstrom der Kontrollverbindung verdrängt. Im Extremfall setzt der zuletzt gestartete Sender das Aushandeln der Parameter nur nach Beenden aller UDP-Datenströme fort. Aus diesem Grund muss man also mit großer Sorgfalt auf das gleichzeitige Starten einzelner Test-Clients achten.

Bei der Laufzeitmessung ist vor allem auf die Symmetrie des Versuchaufbaus inklusive der Priorisierungen zu achten. Gerade bei Paketverlusten infolge von Überlastung auf einer Verbindung ist ein gleicher Füllgrad der Warteschlangen in der hin- und rücklaufender Richtung nicht gegeben. In diesem Fall kann man die Verzögerung in einer Queue lediglich schätzen. Als Layer-2-Technologie wird hier Fast- und Gigabit-Ethernet (GE) verwendet. Trotzdem stößt man hier an Performance-Grenzen einzelner Komponenten[15], die noch nicht diese Datenraten verarbeiten können. Somit ist man darauf angewiesen, die Messungen auf Verbindungen mit 10 Mbit/s durchzuführen und daraus das Verhalten bei höheren Datenraten zu extrapolieren. Da die GE-Trunks mit den software basierten Messtools nicht näherungsweise füllen konnte, bleibt es ebenfalls unklar, ob das Queuing und Policing bei diesen Datenraten ohne Probleme funktioniert. Tendenzen ließen sich aber ableiten.

Abb. 7.20 Testplattform des QoS-Netzes nach Priorisierungsverfahren



Bei der Berechnung der Warteschlangen und der Verzögerungen wurde immer der schlechteste Fall angenommen. Ist das Verhalten des Betriebssystems bekannt, so kann man damit bei einem oder zwei Datenströmen deterministisch rechnen. Für statistische Betrachtungen waren es zu wenige Datenströme. Auch bei den Messungen wird immer versucht, einen Worst-Case[16] nachzubilden. Möchte man eine bestimmte Dienstgüte gewährleisten können, so darf auf keiner Teilstrecke ein Shared-Media zum Einsatz kommen. Diese Bedingung ist in vielen Backbone-Systemen erfüllt. Es ist zu beachten, dass alle Verbindungen im Vollduplex-Modus betrieben werden. Im Testbed ist dieses auf der Ethernet-Schnittstelle des Cisco-7500-Routers nicht möglich gewesen, weshalb auf diesen Schnittstellen nur Messungen im Halbduplex-Modus gemacht werden konnten. Werden die Endsysteme nicht im Überlastzustand betrieben, so wird die Verzögerung im IP-Protokollstapel des Betriebssystems vernachlässigt.

15 Sowohl der Endgeräte als auch der Router und Switches
16 Z.B. durch Senden mit mehrfacher Verbindungsdatenrate

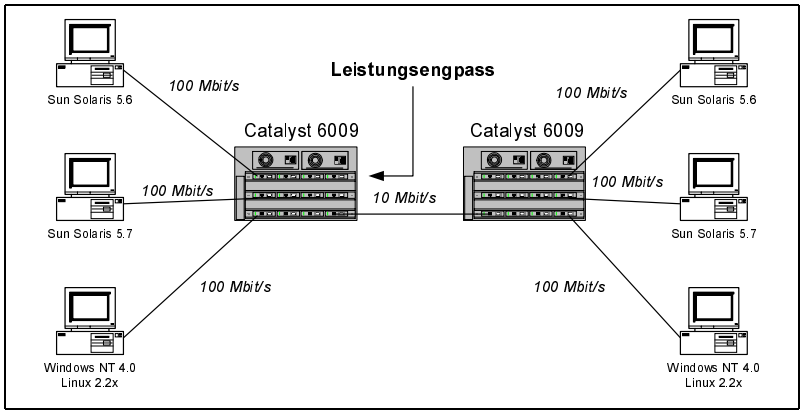


Abb. 7.21
Schicht-2-Testszenario

Es wurde für die QoS-Messungen eine Testplattform aufgesetzt, die am Rechenzentrum der Universität Hannover etabliert wurde (siehe Abb. 7.20). Um die einzelnen Warteschlangen-Strategien zu untersuchen, ist dieses Netz in kleinere Teile unterteilt worden, wie Abb. 7.21 und Abb. 7.22 zeigen. Die verfügbare Datenrate im gesamten Netz kann dann durch eine Minimumbildung ermittelt werden, die Gesamtverzögerung durch Summation der Verzögerungen auf einzelnen Teilstrecken. Folgende Geräte wurden in der Testplattform verwendet:

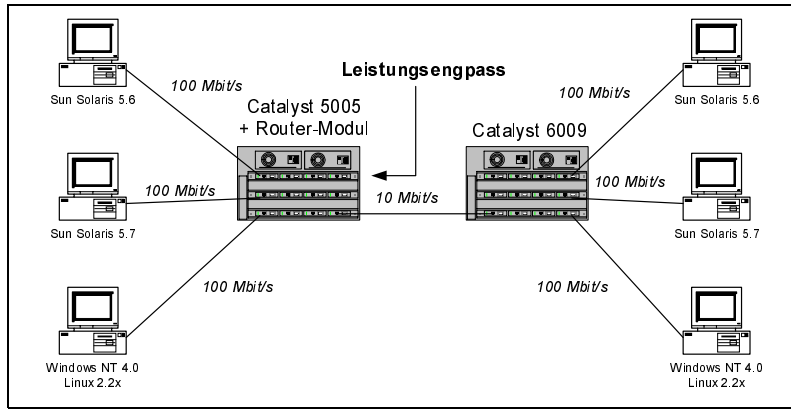
- Zwei Cisco Catalyst5500, einer davon mit einem Router Switch Modul (RSM)
- Zwei Cisco Catalyst6009 ohne jegliche Layer-3-Funktionalität
- Cisco Router 7500
- Access Device Cisco 2600

Die meisten Leistungsmerkmale auf der Schicht 3 sind in dem IOS-Betriebssystem auf dem Router realisiert worden. Dies hat den Vorteil, dass die Konfiguration von QoS-Merkmalen auf unterschiedlicher Cisco-Hardware mit Router-Funktionalität ähnlich abläuft. Auf der anderen Seite sind diese Merkmale stark von dem eingesetzten Software-Release abhängig. Hinzu kommt, dass sich gerade die QoS-Features noch im Entwicklungsstadium befinden. Folglich funktionieren viele davon nicht immer einwandfrei, was ja auch die ATM-QoS-Messungen widergespiegelt haben. Mit jedem neuen Release kann sich dies allerdings ändern.

Als Endsysteme standen vier Sun-SPARC-Ultra-1-Maschinen mit Solaris 5.6 bzw. 5.7 und drei PCs zur Verfügung, die wahlweise unter Linux 2.2.x oder Windows NT 4.0 betrieben wurden. In allen Testszenarien war mindestens ein Engpass vorhanden, in dem die Schnittstellen entsprechend gewählt wurden. Untersucht wurde ausschließlich der DiffServ-Ansatz, da die hier genannten Geräte auf der einen Seite nicht in der Lage sind, den IntServ-Ansatz umzusetzen[17], und auf der anderen Seite IntServ aufgrund der fehlenden Skalierbarkeit

nicht alleine verwendet werden sollte. In diesen Szenarien wird die Priorisierung der Datenpakete durch das DiffServ Code Point (DSCP) vorgenommen.

Abb. 7.22
Schicht-3-Testszenario



Die Abbildung dieser Priorisierung auf ein Schicht-2-Protokoll kann durch Verwendung des IEEE-802.1Q-Rahmens realisiert werden. Cisco Systems hat ebenfalls eine proprietäre Variante eingeführt, die in dieser Umgebung eingesetzt werden könnte: Inter Switch Link (ISL) heißt dieses Protokoll, welches zum Ziel hat, die Algorithmen von IEEE 802.1Q zu verbessern. In beiden Protokollen sind jeweils drei Bits zur Kennzeichnung der Priorität eines Datenpakets vorgesehen. Da diese Protokolle hinsichtlich QoS lediglich Informationen zur Priorisierung übertragen, sind sie für unsere Untersuchungen gleichwertig. Bei dem ISL ist dafür der Spanning-Tree-Algorithmus gegenüber dem IEEE 802.1Q etwas erweitert worden und erzielt dadurch eine bessere Lastverteilung auf den Ports. Deshalb ist bei den Messungen im Schicht-2-Szenario überwiegend ISL angewandt worden.

Hierbei stellt sich bei der Realisierung die Frage, auf welcher Schicht man eine Priorisierung vornimmt. Dabei werden die folgenden Grundaussagen getroffen:

1. Befinden sich zwischen zwei Router-Instanzen keine aktiven Komponenten, so reicht eine Markierung der Datenpakete im IP-Header aus.
2. Kann zwischen den Endgeräten, die Priorisierung verwenden, ausschließlich auf der Schicht 2 vermittelt werden[18], so reicht eine Priorisierung im IEEE-802.1Q-Header bzw. im ISL-Header aus.

17 Es fehlt die Kopplung zwischen RSVP und einem passenden Scheduling-Verfahren: RSVP lässt sich im IOS zur Zeit nur in Verbindung mit WFQ einsetzen, welches aber keine Möglichkeit zum sicheren Aufsetzen der Netzressourcen bietet.

18 Dieses ist in der Regel möglich, wenn der Sender und Empfänger sich im selben Subnetz bzw. VLAN befinden.

3. Befinden sich zwischen zwei Routern Schicht-2-Switches, so muss man darauf achten, dass die Priorität im IP-Header korrekt auf die Schicht-2-Priorität abgebildet wird.
4. Befinden sich im Netz nicht QoS-fähige Komponenten, so dürfen diese Komponenten die Prioritätsinformationen nicht verändern.
5. Besonders ist darauf zu achten, welche Eigenschaften die nicht QoS-fähigen Abschnitte besitzen. Hier darf keine Überlastung vorkommen und die Paketierungsverzögerung muss so gering wie möglich bleiben.
6. Die eigentliche Priorisierung von Datenpaketen wird in den aktiven Komponenten durch den Einsatz bestimmter Warteschlangen-Strategien realisiert. Man muss also genau wissen, wie der einzelne Mechanismus abläuft, damit die entsprechenden Parameter korrekt gesetzt werden können.

Der DiffServ-Ansatz hängt sehr stark von den Warteschlangen-Mechanismen der einzelnen aktiven Komponenten ab. Aus diesem Grund wird hier noch kurz auf die unterschiedlichen Strategien eingegangen. Dazu muss der Vermittlungsvorgang in den Switches bzw. im Router verdeutlicht werden. Abb. 7.23 zeigt den vereinfachten Aufbau eines Layer-2-Switches.

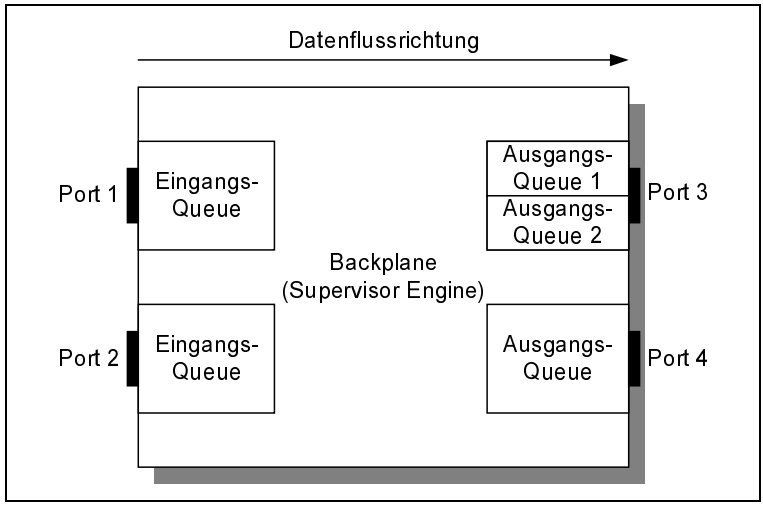


Abb. 7.23 Struktur eines Layer-2-Switch

Bei der Arbeitsweise nimmt man an, dass die Datenpakete auf den Ports 1 und 2 ankommen und auf den Ports 3 und 4 den Switch wieder verlassen. Beim Einlauf der Datenpakete in den Switch gelangen sie in die so genannte Input-Queue, aus der sie entnommen werden, sobald die Backplane sie verarbeiten kann. Die Backplane bzw. Supervisor Engine setzt die einzelnen Felder des Rahmens und trifft die Entscheidung, auf welchem Port das Datenpaket ausgegeben wird. Besitzt ein Ausgangs-Port mehrere Warteschlangen, so wird außerdem entschieden, in welche davon das Datenpaket gestellt wird.

Unter strenger Beachtung der Standards RFC-2597[19], RFC-2598[20] und RFC-2638[21] müssten zur Ermöglichung eines QoS-Dienstes sowohl am Eingangs- als auch am Ausgangs-Port mehrere Warteschlangen realisiert werden. Außerdem muss für jede QoS-Klasse eine separate Queue eingerichtet werden. Dabei sollte allerdings bei allen aktiven Komponenten beachtet werden, in welchen Betriebszuständen die einzelnen Queues sich füllen.

Beim Eintreffen eines Pakets auf einem Port wird er schnellstmöglich auf die Backplane gegeben. Die Input-Queue füllt sich also lediglich bei der Überlast der Backplane bzw. der Supervisor Engine. Ein Switch der Catalyst6xxx-Serie kann nach Herstellerangaben bis zu 36 Gbit/s verarbeiten. Bei der Annahme, dass ein Catalyst6009 mit acht Fast-Ethernet-Modulen ausgestattet ist und zwei GE-Uplink-Ports sich auf der Supervisor Engine befinden, könnte eine theoretische Last von ca. 40 Gbit/s erzeugt werden. Also ist es in wenigen Fällen möglich, dass die Input-Queues wegen mangelnder Kapazität überlaufen werden. Müssen die Datenpakete auf vielen Ports markiert und remarkiert, gefiltert und anders verarbeitet werden, kann es beispielsweise zur Überlast der Switching Engine kommen.

Anders ist hingegen die Situation mit der Output-Queue. Hier werden die Datenpakete maximal mit der Portgeschwindigkeit entnommen. In folgenden Betriebsfällen kann es daher zum Überlauf der Output-Queues kommen:

- Wenn Datenströme von mehreren Eingangs-Ports auf einen Ausgangs-Port ausgegeben werden (z.B. auf einen Trunk).
- Wenn der Ausgangs-Port eine geringere Datenrate als der Eingangs-Port hat.
- Bei einer Kombination der oben geschilderten Betriebszustände.

Es ist zu beachten, dass die Switches immer nur komplette Datenpakete verarbeiten können. Diese Tatsache bereitet einige Schwierigkeiten bei der Queue-Dimensionierung für einzelne Warteschlangen-Verfahren. Die oben angegebenen Betriebszustände sind oft beim Übergang aus einem LAN in das Backbone zu treffen. Dieser Fall ist auch in den einzelnen Testszenarios nachgebildet worden.

Prinzipiell gelten die bisherigen Betrachtungen sowohl für die Schicht-2- als auch für die Schicht-3-Betrachtungen. Bei der Verarbeitung der Schicht-3-Informationen[22], müssen viel längere Bitstrings ausgewertet und umfangreiche Paketfilterungen vorgenommen werden. Somit ist die Überlast der Prozessoren bei den Routern ein häufiges Problem. Folglich muss bei den Schicht-3-Instanzen auch auf das Queue Scheduling auf den Eingangs-Ports geachtet werden. [SIEM00]

19 *Assured Forwarding PHB Group* [HBWW99]
20 *Expedited Forwarding PHB* [JNP99]
21 *Two-bit Differentiated Services Architecture for the Internet* [NJZ99]
22 Bei QoS-Realisierung zusätzlich die Schicht-4-Informationen

Messergebnisse

Folgende Grundvorgaben wurden bei den Messungen definiert:

- Bei unterschiedlichen Priorisierungen im System wird der Datenstrom zwischen den Testrechnern von Sun höher als andere Datenströme priorisiert.
- Die in den Diagrammen angegebenen Datenraten in Bytes/s beziehen sich auf den Nutzdatenstrom. Möchte man die Link-Belastung berechnen, so muss der jeweilige Wert ca. mit dem Faktor 1,1[23] multipliziert werden.
- Die maximale Nutzdatenrate auf einem Vollduplex-Ethernet-Link beträgt ca. 1,13 x 10^6 byte/s

Dabei wurden die folgenden Fälle gemessen:

1. **TCP-Datenströme mit Priorisierung:** Standardkonfiguration bleibt fast unverändert[24]. Es werden zehn nicht priorisierte Datenströme gegen einen priorisierten Datenstrom gesendet. Die Datenrate einzelner Datenströme werden nur durch die TCP-Congestion-Control-Mechanismen bestimmt.
2. **UDP-Datenströme:** Das Bandwidth Borrowing wurde hier untersucht. Die Warteschlangenpuffer werden dabei nach unterschiedlichen Kriterien konfiguriert.
3. **Burstiness:** Das Burstiness-Problem beinhaltet eine gravierende Benachteiligung eines Datenstroms, trotz gleichwertigen Datenverkehrs.
4. **WRR-Parameter:** Proportionale Veränderung der Parameter des Verfahrens Weighted Round Robin (WRR)
5. **Verzögerungsbetrachtung der Schicht 2:** Es wird die maximale Verzögerung und Jitter einbezogen, um die zeitkritischen Daten auszuwerten.
6. **QoS auf Schicht 3:** Unter Berücksichtigung des Queueing und der Filterung wurden QoS-Merkmale auf der Schicht 3 einbezogen.

Bei den ersten Messungen wurde ausschließlich TCP/IP eingesetzt. Abb. 7.24 zeigt dabei bereits das Ergebnis der Sende- und Empfangsdatenrate der Test-Sun-Solaris-5.7. Auf der Sun Solaris 5.6 wurden dabei 6 WRR-Parameter und 32 kByte als Puffergröße in den Queues eingestellt, während die Sun Solaris 5.7 255 WRR-Parameter mit 9,6 kByte beinhaltete. In den ersten Sekunden läuft die Übertragung mit voller Link-Datenrate; danach kommt eine leichte Einsackung der Datenrate. Man kann daran erkennen, dass die Datenströme von der ersten Test-Sun etwas später gestartet worden sind, was an der Synchronisationsproblematik liegt. Hieran erkennt man auch die Datenrate, die der ersten Test-Sun durch das WRR zur Verfügung gestellt wird. Um das Ergebnis noch übersichtlich erscheinen zu lassen, werden in Abb. 7.25 nur drei der insgesamt 10 gesendeten

23 Dieser Faktor hängt von der Nutzdatenlänge ab. Möchte man diese Werte genau berechnen, so müssen alle Header und die Synchronisationszeiten in das Verhältnis zur Baudrate (10 bzw. 100 Mboud) gesetzt werden. Da es bei den Messungen große Toleranzen gab, wurde diese Korrektur nicht für jede Paketgröße neu berechnet.

24 Es wird nur die Gewichtung der Queue 1 von 4 auf 6, gegenüber 255 für die zweite Queue, erhöht.

Datenströme gezeigt. Man sieht, dass diese weniger als 10% der Link-Datenrate untereinander teilen. Nach ca. 250 Sekunden hat die zweite Test-Sun das Senden beendet, danach teilen sich die 10 Datenströme von der ersten Test-Sun die volle Link-Datenrate. Da es sich bei diesem Versuch um TCP-Datenströme handelt, ist die Senderate gleich der Empfangsrate. Dieses Verhalten war auch zu erwarten, da sich mit den TCP-Datenströmen keinerlei Queue-Überläufe oder ähnliche Probleme bemerkbar machen. Bei allen weiteren Messungen auf Schicht 2 wurden die Messdaten über das UDP-Protokoll übertragen.

Abb. 7.24 Priorisierter TCP-Datenstrom auf der Sun Solaris 5.7

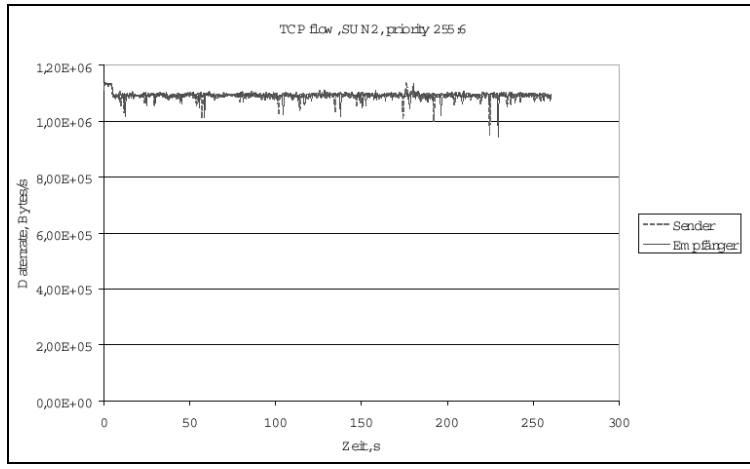
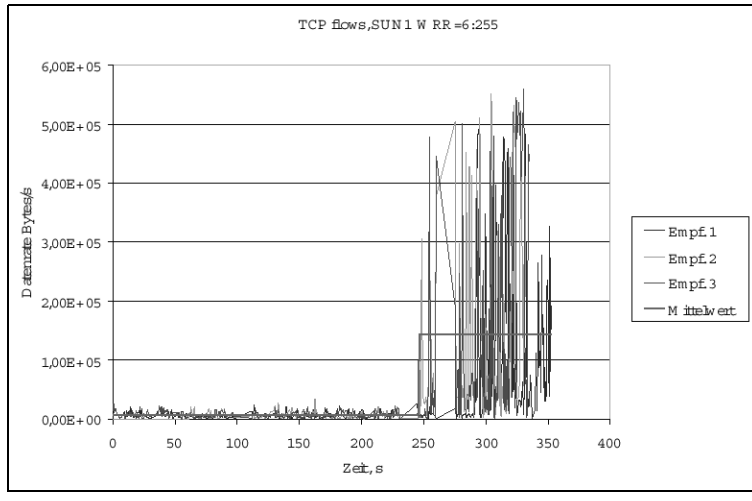


Abb. 7.25 Nicht priorisierte TCP-Datenströme auf der Sun Solaris 5.6



Die Konfiguration der WRR-Parameter mit den dazugehörigen Queues bleibt bei der anschließenden UDP-Messung identisch. Der Datenstrom von der Sun

Solaris 5.7 wird mit der Priorität 6 markiert und kommt somit in die Queue 2. Von der Sun Solaris 5.6 werden die Daten mit COS = 0 markiert und gelangen in die Queue 1. Am Anfang wird von beiden Maschinen jeweils ein Datenstrom mit der Datenrate 500 Kbyte/s gesendet. Das entspricht einer Auslastung des Links um ca. 90%. Dabei werden dem Datenstrom von der Sun Solaris 5.6 lediglich 25 Kbyte/s durch das WRR zugeteilt.

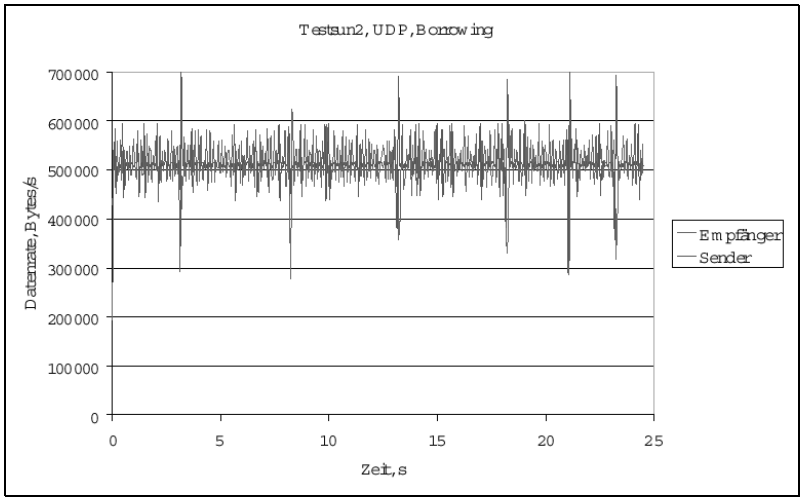


Abb. 7.26 Bandwidth Borrowing mit 25 Kbyte/s in einem 255:6-WRR-Verhältnis

Abb. 7.26 zeigt das Messergebnis. Das Bandwidth Borrowing funktioniert, da die von der Sun Solaris 5.7 nicht genutzte Datenrate der Sun Solaris 5.6 zugeteilt wird. Bei der Nutzung einer höheren Datenrate gehen allerdings immer mehr Pakete verloren. Dies liegt zum einen daran, dass mehr Daten gesendet werden als verarbeitet werden können[25], und auf der anderen Seite, dass wegen der Paketlänge von 520 Bytes zuzüglich aller Header in jeder Zeitscheibe mindestens 4 und nicht 3 Datenpakete von der Queue 1 verarbeitet werden.

Bei der Burstiness-Messung werden als Erstes einzelne Datenströme nach dem ersten Testszenario nach Abb. 7.21 gesendet. Jedes Mal wird mit über 90 % der maximalen Link-Datenrate gesendet. Alle Datenpakete haben die Priorität null. Mit einer Queue Size von 50% und einen Drop Threshold von 40% steht effektiv eine Warteschlangenkapazität von ca. 9 Kbyte zur Verfügung. Da keine anderen Daten über die Verbindung gesendet werden, ist zu erwarten, dass keine Verluste bei der Datenübertragung auftreten. Diese Erwartung bestätigte sich bei der Messung auf der Sun Solaris 5.6, allerdings nicht auf dem PC1, wie Abb. 7.28 verdeutlicht.

25 Funktioniert das WRR korrekt, so liegt dies daran, dass die Systeme ihre Daten mit Bursts in der Größenordnung von mindestens 10 Kbytes senden.

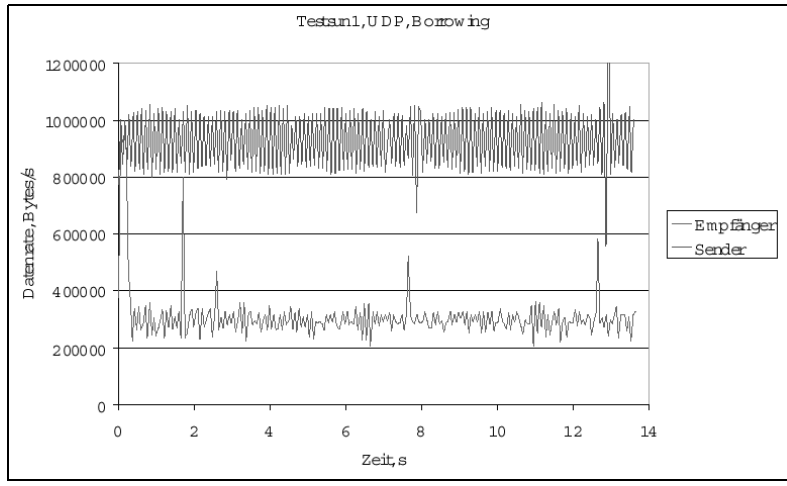


Abb. 7.27 Bandwidth Borrowing mit 900 Kbyte/s und 6:255-WRR-Verhältnis

Dort kommen gerade mal etwas mehr als die Hälfte der Datenpakete ans Ziel. Da sonst alle Parameter identisch sind, kann der Unterschied nur am Verhalten des Betriebssystems liegen. Anschließend wird die Queue Size und die Drop Threshold auf den Ports auf 80% vergrößert. Es stehen jetzt ca. 25 Kbyte[26] für das Queueing zur Verfügung. Abb. 7.28 und Abb. 7.29 zeigen die Ergebnisse. Die Veränderung der Queue Size hat hier wirklich Abhilfe geschaffen. Nun kann das Aussenden der Datenpakete vom Endsystem genauer betrachten werden, um den zeitlichen Abstand zwischen den Datenpaketen zu messen. Hieran kann man das Problem der Verdrängung des Datenstroms sehen, da die Datenpakete in Bursts gesendet werden. Der zeitliche Abstand zwischen zwei Datenpaketen der Länge 1428 Byte beträgt im Burst im Mittel 125 ms. Damit ergibt sich eine Datenrate von fast 100 Mbit/s in den Bursts. Wenn mit einer bestimmten Datenrate gesendet wird, so werden die Daten also eine Zeit lang mit voller Link-Datenrate gesendet. Anschließend wird die CPU für andere Prozesse freigegeben.

Nach einer betriebssystemabhängigen Idle-Zeit wird der Testsoftware wieder die CPU-Zeit zugewiesen und berechnet, wie viele Datenpakete ausgesendet werden müssen, damit die geforderte Datenrate eingehalten wird. Es kommt also auf die Idle-Zeit an. Die Messungen ergaben, dass diese unter Linux ca. zweimal länger als bei einem Sun-Betriebssystem sind. In Zeiten solcher Bursts ist die Ankunftsrate zehnmal größer als die Ausgangsrate. Somit benötigt man bei einem einzelnen Datenstrom eine Warteschlange die ca. 90% der Burst Size beträgt. Die Bursts von der Sun-Workstation waren 8658 Byte

26 Rechnerisch ist der Wert für die Queue Size größer, da aber ein Teil der Queue für andere Zwecke verwendet wird, ist der verfügbare Wert 25000 Byte.

lang, weshalb keine Verluste bei einer Queue Size von ca. 9600 Byte auftraten. Bei dem PC waren die Bursts 22,848 Kbyte groß, weshalb massive Verluste bei dieser Queue Size auftraten und keine bei einer Queue-Size von 25 Kbyte.

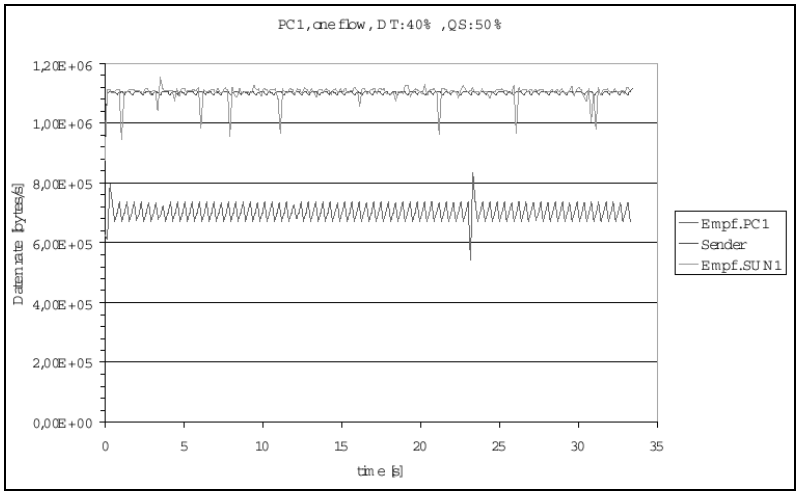


Abb. 7.28 Burstiness bei einer Queue Size von 9500 Byte

Wird die Queue über eine längere Zeitspanne überlastet, so ist der mittlere Füllgrad unabhängig von der Queue Size permanent knapp unter 100%. Somit ist die Wahrscheinlichkeit für das Droppen der Frames bei größeren Bursts[27] höher als bei kurzen Datenpaketen. Dieses Ergebnis hat weit gehende Folgen:

1. Bei diesem Effekt kommt es im Wesentlichen auf das Verhältnis der Burst Size zur Queue Size an. Dieses Verhältnis wird mit der Teilung des gesamten Pufferplatzes in mehrere Queues immer größer. Die Verdrängung der Datenströme war z.B. wesentlich schwächer ausgeprägt, wenn QoS auf dem Switch abgeschaltet war. Die Ursache dafür ist, dass in diesem Fall insgesamt eine Queue pro Port vorhanden war, die dann entsprechend größer wird. Das bedeutet, dass man das Burstiness-Problem mit der Einführung von QoS verschärft!
2. Die Queues arbeiten mit Flussaggregationen. Folglich werden in allen Queues, in die Datenströme mit unterschiedlicher Burstiness bzw. unterschiedlicher Paketlänge einlaufen, ähnliche Verhaltensweisen gegeben sein.
3. Die Burstiness ist sowohl system- als auch anwendungsabhängig. Somit kann man das Verhalten der Endsysteme bei der Netzadministration nicht berücksichtigen.
4. Bei Einsatz von Traffic Shaper müssen diese mit einer Auflösung von wenigen Millisekunden arbeiten.

27 Das gilt auch bei größeren Paketen.

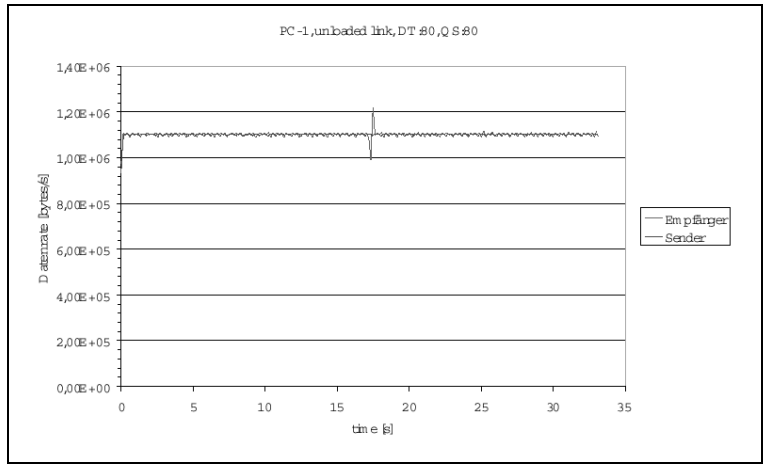


Abb. 7.29 Burstiness bei einer Queue Size von 25 kByte

Abschließend könnte man zu den Burstiness-Messungen nur den Einsatz von Traffic Shaper empfehlen. Diese müssen aber mit sehr hoher Zeitauflösung arbeiten. Man kann außerdem die Queue Size überdimensionieren, was aber in Zeiten der Überlast im Netz kaum Abhilfe schaffen wird.

Die bisherigen Messungen wurden immer mit den gleichen WRR-Parametern durchgeführt. Nun wird an dieser Stelle eine proportionale Veränderung eingeführt. Dabei soll das Verhältnis der zugeteilten Datenraten unverändert bleiben. Der Sun Solaris 5.6 wird dabei eine Paketgröße von 520 Byte mit einer Queue von 36 kByte und einer Senderate von 1,1 Mbyte/s zugeteilt, während die Sun Solaris 5.7 zwar die gleiche Paketgröße und Senderate, aber eine Queue Size von nur 9,6 Kbyte erhält. Die Messergebnisse für unterschiedliche WRR-Parameter sind für die Sun Solaris 5.6 (nicht priorisiert) und Sun Solaris 5.7 (priorisiert) in Abb. 7.30 und Abb. 7.31 zusammengefasst worden.

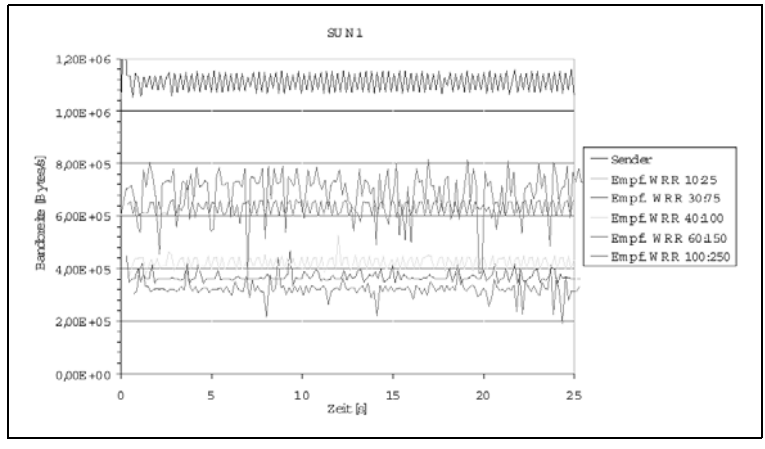


Abb. 7.30 Sun Solaris 5.6 mit veränderten WRR-Parametern

Man kann feststellen, dass bei kleineren WRR-Parametern das Verhältnis der Datenrate in etwa dem Verhältnis 2,5:1 entspricht. Geringfügige Abweichungen kann man auf die etwas ungünstige Rahmenlänge von ca. 580 Byte auf Ethernet-Ebene zurückführen. Doch ab der WRR-Größen von 100:40 kommen wesentliche Einbußen in der Übertragungsrate der Sun Solaris 5.7 vor. Auch dieses liegt wiederum an den Queue-Überläufen. Verändert man beispielsweise die Queue Size lediglich um 10% bei den WRR-Parametern 40:100, hat man dasselbe Verhalten wie bei der Wahl der WRR-Parameter 10:25. Offensichtlich fehlten dem priorisierten Datenstrom gerade diese 4,5 kByte. Eine weitere Veränderung der Queue bis auf 25 kByte brachte dann auch keine Änderung mehr in der Verteilung der Übertragungsraten.

Aus den bisherigen Messungen könnte man den Schluss ziehen, dass man die Warteschlangen möglichst proportional zur zugeteilten Übertragungsrate, das heißt proportional zu den WRR-Parametern einteilen sollte. Zieht man die maximalen Verzögerungen und Jitter mit ein, so kann man feststellen, dass gerade für die zeitkritischen Daten die Queues nach Möglichkeit minimiert werden müssen. Die Verzögerung schwankt bei voller Last ständig zwischen einem Minimum und einem Maximum. Das Maximum wird dabei durch die Queue Size bestimmt, das Minimum durch die Serialisierungsverzögerung und die Verarbeitungszeit der Datenpakete. Dabei können die Verarbeitungszeiten im Switch aus den Messungen extrahiert werden.

Als Erstes wird die Verbindung überlastet, da 10 Mbit/s auf der Anwendungsebene gesendet werden. Die Queue ist 44.032 Byte groß, wobei keine anderen Daten auf dem Link gesendet werden. Es ist zu beachten, dass die Ergebnisse sich auf die Daten oberhalb des UDP-Protokolls beziehen. Der entstehende Datenverlust ist auf die Überlast des Links zurückzuführen. Von besonderem Interesse sind dabei die minimalen, die maximalen und die mittleren Laufzeiten. Das erste Datenpaket läuft noch in eine leere Queue hinein. Somit kann man anhand des ersten Datenpakets die Verarbeitungszeit im Switch abschätzen. Die Serialisierungsverzögerung eines Datenpakets auf einem 10-Mbit/s-Port in dieser Messung beträgt:

$$1446 \cdot 100ns \cdot 8 = 1{,}16ms$$

Die Datenpakete im Testbed wurden viermal auf 100 Mbit/s-Ports serialisiert und zweimal auf 10 Mbit/s-Ports. Somit kann man die Serialisierungsverzögerung folgendermaßen bestimmen:

$$1{,}16 \cdot 2{,}4 = 2{,}77ms$$

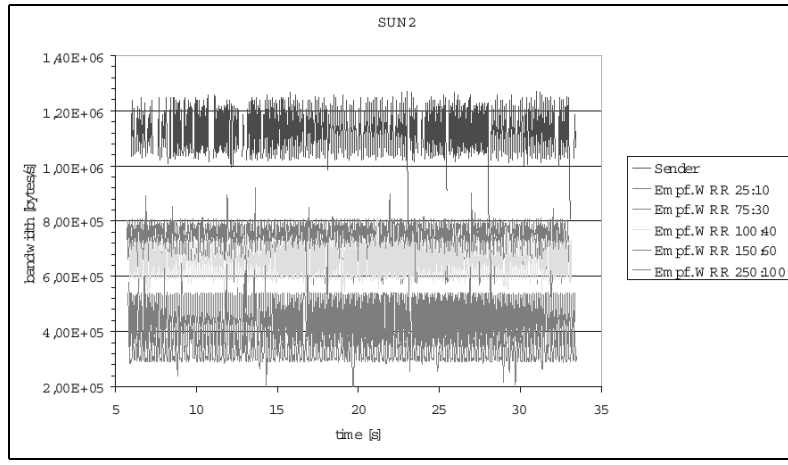


Abb. 7.31 Sun Solaris 5.7 mit veränderten WRR-Parametern

Die Differenz der Laufzeiten ergibt die Verzögerungen im Betriebssystem und im Switch. Mit einer Abwandlung dieser Messung[28] konnte man eine Verarbeitungszeit im Switch abschätzen. Diese beträgt etwas unter 100 µs. Es wurden bei den Messungen durchschnittlich 15 Rahmen gesendet, bis ein Überlauf stattfand und ein Datenpaket verworfen wurde. Das ergibt eine Verlustrate von 6,7%[29]. Berechnet man die Serialisierungsverzögerung auf einem 10 Mbit/s-Link für eine volle Queue, so ergibt sich ein Wert von ca. 37 ms. Die Verzögerung war relativ gering. Bei Einsatz von 100-Mbit/s-Ports lassen sich die gemessenen Verzögerungswerte ca. um den Faktor 8 weiter reduzieren. Bei den Messungen konnte man aber einen 100 Mbit/s-Port zur Überlast bringen, somit waren auch durchgehend Verzögerungen im Bereich von 1 ms gemessen worden.

Bei Einsatz einer Layer-3-Umgebung mit QoS-Mechanismen kann man von folgenden Anforderungen ausgehen:

- Das Queueing benötigt mehr CPU-Last als eine Realisierung auf Schicht 2. Somit ist mit langen Verarbeitungszeiten zu rechnen.
- Die Filterung über Access-Listen kann Router zur Überlast bringen.

QoS-Merkmale auf dem Catalyst 5500 mit einem RSM-Modul lassen sich für einzelne VLANs konfigurieren. Diese werden als logische Schnittstellen im Route Switch Modul (RSM) behandelt. Doch alle Messungen liegen im unsicheren Bereich. Das Queueing hat dabei weder brauchbare noch reproduzierbare Ergebnisse gebracht. Die Ursache dafür liegt in dem zu schwachen Prozessor. Alle Queueing-Mechanismen sind mit der Definition von Access-Listen verbunden. Sobald man eine Access-Liste auf ein 10-Mbit/s-Interface anwendet, steigt allerdings die Prozessor-Last bei der Verarbeitung der Datenströme

28 Durch Zuschalten eines zusätzlichen Switches

29 Der Gesamtverlust liegt bei 6%, da der erste Paketverlust nur nach Versenden von ca. 450 Datenpaketen stattfindet.

auf zeitweise über 98%. Schaltet man den RSM in den Debugging-Modus, so sieht man, dass viele Frames ohne Anwendung der Access-Listen direkt auf das Interface gegeben werden. Es ließe sich lediglich das Traffic Shaping auf die Datenströme anwenden. In beiden Fällen lag die durchschnittliche Round Trip Time (RTT) im zweiten Testszenario bei über 100 ms! Beim Einsatz von Traffic Shaping auf einer überlasteten 10-Mbit/s-Schnittstelle stieg die RTT auf über 300 ms. Interessant war auch hier, dass man eine Konkurrenz von Datenströmen mit unterschiedlicher Burstiness sehen konnte. Auch hierfür wurde ein Datenstrom von einer Sun und einem PC auf den Trunk gegeben. Es kam wiederum zur Benachteiligung der Datenpakete vom PC, das Verhältnis der Übertragungsraten war dabei aber 1:1,7[30]. Dieses kann man damit erklären, dass man auf der Schicht 3 ein größeres Verhältnis der Queue Size zur Burst Size hat. Hat man Traffic Shaping auf beide Datenströme angewandt, so war die Verdrängung stärker ausgeprägt. Das liegt mit großer Wahrscheinlichkeit daran, dass der Zeitintervall für das Shaping mit 10 ms zu groß war. Interessant war die Untersuchung des Weighted Fair Queueing (WFQ). Hier trat das Burstiness-Problem am 2-Mbit/s-Interface des Cisco 2600 nicht auf. Der Sun-Solaris-Workstations und der PC waren die Übertragungsrate auf der Schnittstelle jeweils zur Hälfte zugeteilt worden. Dieses ist dadurch möglich, dass jeder Datenstrom über eine eigene Queue abgearbeitet wird. [SIEM00]

Auswertung

Bei den Messungen wurde bei allen Betrachtungen der schlechteste Fall angenommen. Diese Betrachtungsweise ist gerade bei der Realisierung zeitkritischer Dienste im Internet legitim. Man möchte ähnlich zum Telefonnetz einen laufenden Dienst nicht wegen mangelnder Ressourcen im Netz unterbrechen. Doch der Worst Case tritt im richtig dimensionierten Netz nicht oft und nur kurzzeitig auf. Es ist also naheliegend, im Internet eine gewisse Flusskontrolle zusätzlich zu den aufwendigen Queueing-Mechanismen einzuführen. Würde das gelingen, so könnte in bestimmten Fällen eine effiziente Flusskontrolle ein komplexes Queueing-Verfahren komplett ersetzen. Das hätte den Vorteil, dass nur in Zeiten der Überlast bestimmte Datenströme durch explizite oder implizite Signalisierung abgebremst wären. Solche Möglichkeiten zur Flusskontrolle sind beispielsweise im X.25-Standard sowie im ATM zu finden. Im Internet-Protokoll ist aber ursprünglich kein Mechanismus zur Flusskontrolle vorgesehen worden. Nur durch die Erweiterung von TCP mit zusätzlichen Algorithmen wird die Datenrate einer TCP-Übertragung an die aktuelle Belastungssituation im Netz dynamisch angepasst. Da diese Methoden zur Vermeidung von Verstopfungen im Netz eingesetzt werden, werden sie im IP-Umfeld als Congestion Avoidance bezeichnet. Dies wird aber nur bei TCP vorgenommen. UDP zur Unterstützung von Echtzeitprozessen besitzt keinerlei Mechanismen.

30 In einer Layer-2-Umgebung war das Verhältnis ca. 1:5.

Das Verfahren Drop Threshold ist bei Cisco Systems als Queue-Management-Verfahren implementiert, obwohl es nur eine Congestion Avoidance bereitstellt. Dabei werden für unterschiedliche Verkehrsklassen innerhalb einer Warteschlange so genannte Drop Thresholds definiert. Eine Drop Threshold von 50% bedeutet beispielsweise, dass Datenpakete bei einem Füllgrad von mehr als 50% verworfen werden. Die Abb. 7.32 zeigt vier Drop Thresholds. Es wird auf je zwei Verkehrsklassen, die durch unterschiedliche Prioritäten im COS-Feld gekennzeichnet sind, eine Drop Threshold angewendet. Ist die Warteschlange zu 20% gefüllt, so werden alle ankommende Pakete, die mit der Priorität 0 bzw.1 gekennzeichnet sind, verworfen. Lediglich Datenpakete mit der Priorität 6 und 7 kommen durch den fast vollen Puffer durch. Dabei stellt sich die Frage, wie man die Drop Threshold für eine Echtzeitanwendung einstellen sollte. Möchte man die maximale Verzögerung minimieren, so muss der geringste Drop Threshold für Echtzeitanwendungen vorgesehen werden. Das hat aber die Konsequenz, dass bei einer steigenden Belastung des Netzes alle Datenpakete dieses Stroms trotz verfügbarer Ressourcen verworfen werden. Man stößt somit beim Versuch, Echtzeitanwendungen mit Drop Thresholds zu priorisieren, auf einen Widerspruch bei der Konfiguration. Nimmt man nun zuerst an, dass die vier Drop Thresholds in einem Switch eingerichtet wurden, wobei nur Best-effort-Daten transportiert werden, die mit der COS = 0 markiert sind, stehen für diese effektiv lediglich 25% von der gesamten auf dem Port realisierten Warteschlange zur Verfügung. Tritt eine Burstiness auf oder ist die Ausgangsverbindung stark belastet, so werden die Datenpakete zu früh verworfen. Die Hardware wird also nicht optimal genutzt.

Wenn man genauer betrachtet, was passiert, wenn nur UDP-Datenpakete gesendet werden, kommt man zu dem Schluss, dass ein Paketverlust keinerlei Auswirkungen auf den Sender hat. Weiterhin wird angenommen, dass sich im Netz ein quasistationärer Zustand eingestellt hat. Das bedeutet, dass alle Sender im Mittel mit einer konstanten Datenrate senden und die Datenpakete im Mittel mit einer konstanten Datenrate der Warteschlange entnommen werden. Somit ist auch die Anzahl der Datenpakete in der Queue im Mittel konstant. Damit das etwas anschaulicher wird, soll die Warteschlange 10 kByte groß und im Mittel ca. 20% gefüllt sein. Betrachtet werden nun zwei Datenströme, auf die die Drop Threshold 1 angewendet wird. Einer davon soll 300 Byte und der andere 1500 Byte große Pakete senden. Es ist klar, dass die 300-Byte-Datenpakete viel seltener als die 1500-Byte-Pakete verworfen werden. Als Resultat bekommt man unterschiedliche Datenraten für Datenströme, die derselben QoS-Klasse angehören, zugeteilt. Unterscheiden sich zwei betrachtete Datenströme nicht in der Paketgröße, sondern in ihrer Burstiness, so kann der Unterschied noch größer werden und es werden Datenpakete mit der Priorität 2 und 3 öfter als die mit der Priorität 0 und 1 verworfen. Somit ist die Anwendung von

Drop Thresholds, angewandt auf UDP-Datenströme, eine schlechte Wahl. Weiterhin ist zu beachten, dass die Funktionsweise dieser Drop Thresholds nicht mit den im RFC-2597[31] empfohlenen Drop Precedences übereinstimmt. Unterschiedliche Drop Precedences werden auf Datenpakete innerhalb einer Queue angewendet, wenn ein Datenpaket wegen mangelnder Ressourcen verworfen werden muss. Die Anwendung von Drop Thresholds dagegen bewirkt, dass Datenpakete verworfen werden, wenn noch Ressourcen im Netz verfügbar sind.

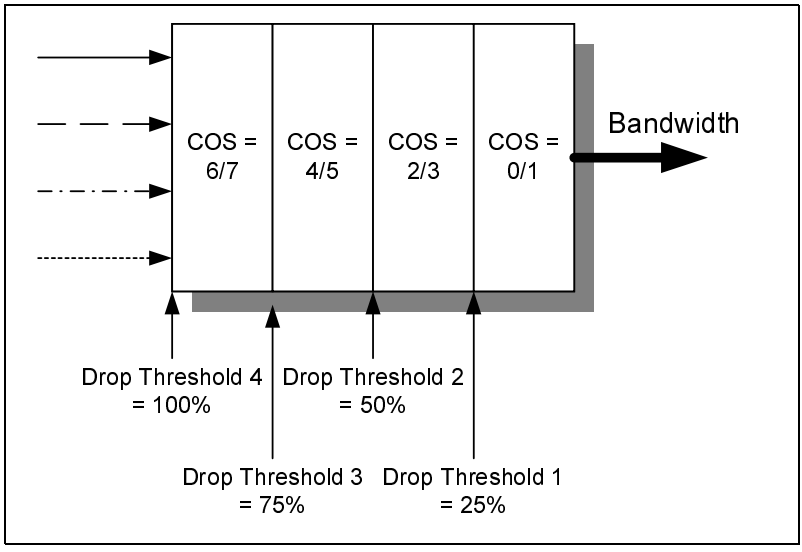


Abb. 7.32
Drop Thresholds

Das Verfahren Weighted Random Early Detection (WRED) eignet sich besser zur Vermeidung von Überlastsituationen: Wenn eine Warteschlange reine TCP-Verbindungen abarbeitet und es hier zu einer Verkehrskonzentration kommt, sodass die Queue überläuft, werden die Datenpakete aller einlaufenden Datenströme verworfen. Dieses hat zur Folge, dass bei allen beteiligten Datenflüssen die Bestätigung ausfällt. Folglich warten alle beteiligten Sender auf eine Bestätigung bis zum Time-out. Zweitens beginnen die Sender nach dem Time-out mit einer geringeren Datenrate zu senden. Da aber der Überlauf nur kurzzeitig stattgefunden hat, ist es nicht notwendig, alle Datenströme abzubremsen. Um die Queue bei TCP-Datenströmen besser auszunutzen, wird das WRED-Verfahren eingesetzt, welches bewirkt, dass die Füllung der Queues zufällig geschieht. Dabei werden für unterschiedliche Klassen unterschiedliche Parameter gesetzt. Dieses Verfahren ist zwar in seiner Wirkungsweise sehr interessant, kann aber nur in Verbindung mit TCP-Datenströmen gewisse Vorteile bei der

31 *Assured Forwarding PHB Group* [HBWW99]

Lastverteilung bringen. Für die meisten multimedialen Anwendungen ist WRED somit nicht einsetzbar.

Der Einsatz von WFQ ist aber hinsichtlich QoS nicht unproblematisch. Es bekommen beispielsweise IP-Telefonie-Datenpakete wegen geringer Datenrate und geringer Paketlänge einen Vorrang, wobei große Datenpakete einer Videoübertragung kaum eine Chance hätten, auch mit einer geringen Verzögerung ans Ziel zu kommen. Weitere Probleme sind:

- Das Verhalten von WFQ ist nicht vorhersagbar. Es werden zwar bestimmte Gewichtungen berechnet, diese ändern sich aber mit der Veränderung der Gesamtsituation im Netz permanent.
- Der Algorithmus, anhand dessen die Gewichtungen aus den einzelnen Parametern bestimmt werden, ist nicht bekannt bzw. wird von Cisco nicht bekannt gegeben
- Es ist denkbar bzw. in großen Routern sogar regulär, dass mehr als 255 Datenströme gleichzeitig über eine Schnittstelle laufen. In diesem Fall ist die Einteilung in einzelne Flows bzw. einzelne Queues unklar.

In Anbetracht der dargestellten Verfahrensanalyse und der Messergebnisse kann man folgende Implementierungsempfehlungen geben:

- Da im Catalyst5xxx keine Trennung der Queues möglich ist, ist in einer Catalyst5xxx-Umgebung keine QoS-Realisierung für multimediale Dienste realisierbar.
- Da für den IP-Telefonie-Dienst die Laufzeit eine kritische Größe darstellt, sollte man möglichst auf Verbindungen mit 10 Mbit/s und weniger verzichten.
- Die Queue Size sollte beim Einsatz von WRR groß genug, aber nicht übermäßig groß gewählt werden.
- Für eine Videoübertragung soll die Reservierung mit Reserve vorgenommen werden, da sonst Queue-Überläufe unvermeidbar sind.
- Für jeden QoS-Dienst wird je eine zusätzliche Warteschlange benötigt.

7.3 Traffic Engineering (TE)

An dieser Stelle sollen Traffic-Engineering-Verfahren wie MPOA und MPLS genauer auf ihre Echtzeitfähigkeit hin betrachtet werden.

7.3.1 MPOA

Um die MPOA-Fähigkeiten zu testen, steht die Performance unter Beachtung der QoS-Parameter im Vordergrund. Die Durchsatzmessungen untersuchten das Verhalten der Switches bei Erhöhung der Verkehrslast. Die Messungen sollten die Rahmentransportrate in Abhängigkeit der Last erfassen und darüber

den Durchsatz, also die maximale Rahmentransportrate ohne Verluste, bestimmen. Diese Messungen sind ähnlich den bereits durchgeführten Stresstests auf Schicht 2. Diesmal wurde aber nicht nur die zweite ATM-Schicht gemessen, sondern auch die Anpassungsschicht AAL-5, die für die Anpassungs- und Integrationsverfahren verwendet wird. Auch hierbei können Fehler im Puffermanagement des Switch ermittelt werden.

Die Erfassung der Latenzzeiten wurde für die Bestimmung der mittleren Durchlaufzeiten eines Rahmens durch den Switch bei unterschiedlichen Hintergrundlasten getestet. Hierbei wird zwischen unterschiedlichen Ethernet-Rahmen (Unicast und Broadcast) unterschieden, da sie das Netz bzw. die Switches unterschiedlich belasten. Es ist interessant, wie schnell die Umsetzung der Pakete von IP-MAC-ATM bei LANE bzw. MPOA geschieht und welcher Ansatz Geschwindigkeitsvorteile aufgrund der unterschiedlichen Funktionalität besitzt. Bei den Messungen handelt es sich um eigene Laborwerte.

Messaufbau

Die Empfehlung I.356 der ITU-T beschreibt die Schicht-2-Parameter bei ATM. Diese Parameter sind wichtig für die Zellenverzögerung sowie -verluste, die bei der Übermittlung auftreten können. Im Gegensatz zur Teststellung im Jahr 1998 werden nun auch Parameter der dritten Schicht untersucht, da MPOA stärker beachtet werden soll. Dabei ist es wichtig zu erfahren, wie sich verschiedene Protokolltypen auf die Performance der einzelnen Geräte auswirken. MPOA setzt UNI3.1 oder höher, LANEv2 und NHRP voraus. Dies sagt aber noch nichts darüber aus, mit welcher Performance der virtuelle Router arbeitet, die sich aus dem Zusammenspiel der MPOA-Clients und -Server ergibt.

Die MPOA-Messungen werden unterteilt in zwei Testszenarien: Intra-ELAN- und Inter-ELAN-Test. Der Intra-ELAN-Test evaluiert die grundlegende Interoperabilität zwischen LANEv2-Clients und -Servern sowie MPOA-Clients und -Servern. Der Inter-ELAN-Test deckt hingegen die Interoperabilität der lang- bzw. kurzlebigen Datenverbindungen ab und überprüft die MPOA-Performance aus Sicht der Endgeräte. Das verwendete Testgerät SmartBit von Netcom Systems ist dabei in der Lage, bis zu 1000 Endteilnehmer zu simulieren und die Performance zwischen Ethernet-zu-Ethernet und die zwischen Ethernet und ATM zu ermitteln. Der Test enthält weiterhin die Überprüfung des Shortcut-Verhaltens, Cache Processing und zumutbare Anfragen, wie MPOA-Adressauflösung, aus der Sicht des Servers.

Die Messungen wurden mit unterschiedlichen Paketgrößen vorgenommen, um die Performance und Latenzzeiten korrekt wiedergeben zu können. Fragmentierung, die durch unhandliche Paketgrößen vorgenommen werden muss, beeinflusst erheblich die Messungen. Typische Paketgrößen von IP-Paketen sind die Werte 64 Byte, 1500 Byte und 65 kByte. Kleine Datenpakete fallen bei Client-/Serveranwendungen häufig an und stellen somit die Netzbelastung im Regelfall dar. Pakete mit der Größe von 1500 Byte passen genau in die maxi-

male Rahmengröße eines Ethernet-Frames. Hierbei muss nicht zusätzlich fragmentiert werden, wodurch die optimale Performance zu erwarten ist. Letztendlich wird noch 64 kByte eingestellt, welches die maximale Paketgröße des Internetprotokolls darstellt. Diese muss fragmentiert werden, wodurch die Latenzzeiten zunehmen und die Performance heruntergesetzt wird.

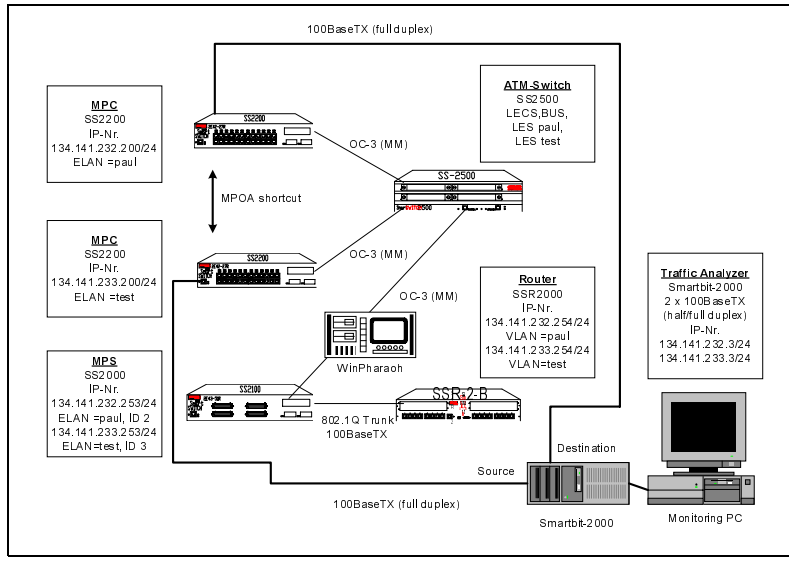


Abb. 7.33 MPOA-Testübersicht der Cabletron (Enterasys)-Komponenten

Als Testequipment wurden zwei Smart Switches 2200 als MPOA-Client und ein Smart Switch 2000 als MPOA-Server eingesetzt. Ein Smart Switch Router 2000 ergänzte das Szenario noch um die fehlende Layer-3-Funktionalität. Da ein LEC/MPOA-Client mit dem LES/MPOA-Server kommunizieren muss, um Verbindungen bei LANE/MPOA automatisch aufbauen zu können, geht die Kommunikation über den Backbone-Switch sowie das entsprechende Messgerät. Zusätzlich wird an den Ethernet-Schnittstellen der Edge Devices der nötige Datenverkehr erzeugt, um die Ethernet-Performance zwischen dem Generator und Analysator End-to-End messen zu können. Durch die Messanordnung der Abb. 7.33 ließen sich die Tests Performance von Ethernet-zu-Ethernet und die Ermittlung der Paketverluste durchführen.

MPOA wird mit seinen spezifischen Parametern (P-NNI, LANEv2, NHRP) in Betrieb genommen. Das Messgerät wird an die Ethernet-Ports der Edge Devices angeschlossen, um den Traffic erzeugen zu können. Zusätzlich wird das Messgerät zwischen die beiden ATM-Switches geschaltet, um die Performance ATM-ATM messen zu können. Letztendlich lässt sich auch mit dieser Anordnung die Performance des Route-Servers ermitteln, der ähnliche Aufgaben wie der BUS bei LANE besitzt. Um die Messungen mit und ohne MPOA-Shortcut

durchführen zu können, wurde jeweils auf den MPC Devices der Wert für Shortcut Frame Count variiert. Dieser Wert stellt den Threshold für die Anzahl der Frames pro Sekunde dar, ab wann ein Shortcut etabliert werden soll. Er wurde bei Messungen mit Shortcut auf 10 und bei Messungen ohne auf 65000 gesetzt.

Messergebnisse

Die Messungen lassen sich hauptsächlich in Latency- und Frame-Loss-Messungen aufteilen. Dabei wurde die Last in 10-Mbit/s-Schritten von 10 auf 100 Mbit/s auf der Fast-Ethernet-Schnittstelle erhöht. Als Dauer für die Messung wurden jeweils 10 sec eingestellt, um die Gesamtzeit der Messungen zu begrenzen. Es wurden dabei unidirektionale Verbindungen am Smart Bit eingestellt. Die Rahmengröße variierte von 64 zu 512 und 1518 Byte, da dies die signifikantesten Frames im Netz darstellen.

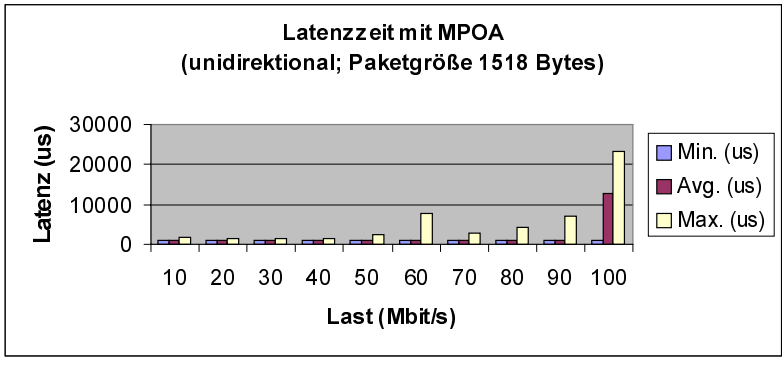


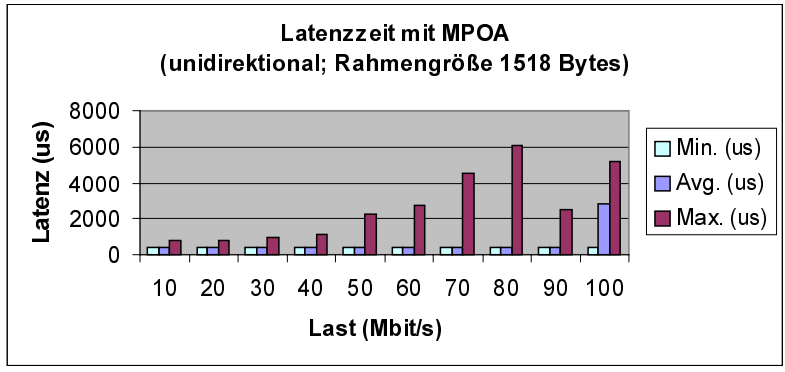
Abb. 7.34 Unidirektionale Latenzzeit bei 1518-Byte-Paketen ohne MPOA

Zuerst wurden 64-Byte-Pakete eingesetzt. Hierbei konnte klar ermittelt werden, dass ohne MPOA die Latenzzeiten zwischen 10.000 µs und 60.000 µs variierten, während bei Einsatz vom MPOA mit Shortcuts die Werte zwischen 40 und 28.000 µs lagen. Besonders bei geringer Auslastung waren die Messergebnisse sehr gut. Allerdings zeigten die Messungen auch, dass MPOA bei geringen Lasten nicht notwendig zu sein scheint, da der Unterschied zwischen den Ergebnissen minimal war. Erst ab 50% Auslastung nehmen die Werte sprunghaft zu, pendelten sich aber dann wieder auf geringeren Verzögerungszeiten ein.

Dass es zwischen den Messungen kaum Unterschiede bei geringer Last gibt, ist zunächst nicht verwunderlich. Allerdings stellen 64-Byte-Pakete bei diesen Messungen noch nicht so hohe Anforderungen an die Netzperformance, sodass dieser Trend sich bei größeren Paketen im Grunde fortsetzen müsste. Dass dies der Fall ist, beweisen Abb. 7.34 und Abb. 7.35. Hier werden 1518-Byte-Pakete mit und ohne MPOA mit unidirektionalen Verbindungen bei gleichmäßig ansteigender Last verwendet. Dabei sind die Unterschiede in den Verzögerungszeiten mit/ohne MPOA wesentlich größer. Das heißt, die Short-

cuts bei MPOA zeigen Wirkung und sorgen durchgängig für die sehr schnelle Weiterleitung der Pakete. Da dieser Pakettyp optimal zu den Ethernet-Rahmen passt, dürften sich selbst ohne MPOA keine sehr hohen Verzögerungen ergeben. Dies ist auch der Fall; erst bei 90-100% Auslastung kommen maximale Verzögerungszeiten von ca. 23.000 µs zustande. Anders hingegen bei einer direkten Verbindung zwischen den virtuellen Clients. Hier sind nur ca. 6.000 µs als maximale Latenzzeit zu beobachten, die bei einer Auslastung von 80% entstehen. Die durchschnittliche Verzögerung ist als minimal zu bezeichnen.

Abb. 7.35 Unidirektionale Latenzzeit bei 1518-Byte-Paketen mit MPOA



Wenn man die Performance von MPOA begutachtet, dürfen auch die Rahmenverluste nicht außer Acht gelassen werden, da ansonsten eine falsche Einschätzung hinsichtlich der Verzögerung vorgenommen wird. Die MPOA-Szenarien sollen schließlich alle Pakete handhaben können und gleichzeitig eine geringe Verzögerung besitzen. Aus diesem Grund wurden die Messungen mit den gleichen Einstellungen noch einmal unter dieser Perspektive durchgeführt.

Abb. 7.36 Verluste einer unidirektionalen Verbindung mit 1518-Byte-Paketen

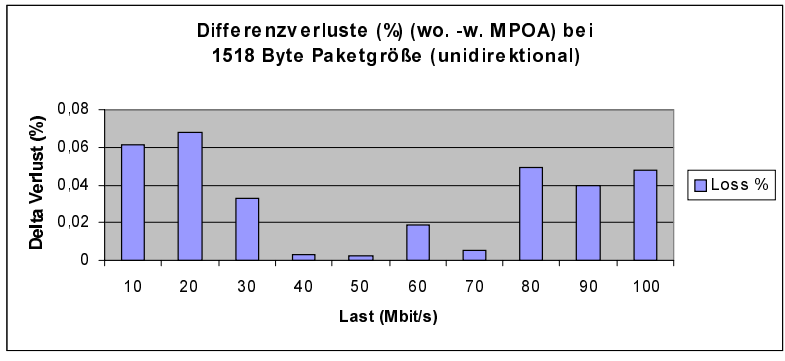


Abb. 7.36 zeigt die Messung der Rahmenverluste in der Differenz der Ergebnisse auf. Das heißt, die Messergebnisse „ohne MPOA“ (wo. = without) sind von den Messergebnissen „mit MPOA“ (w. = with) abgezogen worden. Bei 64-Byte-

Paketen in unidirektionalen Verbindungen fällt auf, dass Rahmenverluste bis 70% Auslastung gar nicht erst entstehen. Erst darüber kommt es insbesondere bei kleinen Paketen zu Paketverlusten, die mal mit und mal ohne MPOA zustande kommen. Während 512-Byte-Pakete erst bei einer 100%-igen Auslastung verworfen werden, geschieht dies bei 1518-Byte-Paketen fast kontinuierlich. Allerdings entstehen diese Verluste erstens ohne die Nutzung von MPOA und zweitens handelt es sich dabei um sehr geringe Werte, die ebenfalls unter 0,1% liegen!

Auswertung

Das MPOA-Szenario von Cabletron (heute Enterasys) wies nach, dass MPOA mittels Shortcut wirklich schneller IP durch den ATM-Backbone schaltet. Die Shortcut-Funktionalität wurde sehr schnell[32] bereitgestellt, sodass bereits bei einer Messdauer von 10 sec pro Session eindeutige Werte erzielt werden konnten. Durch die Messgeräte konnten eindeutig die Performance-Gewinne in einer LAN-Umgebung dargestellt werden. Schnelles Layer-3-Switching war dadurch möglich und zugleich wesentlich effektiver bei unterschiedlichen Paketgrößen als ohne MPOA. Durch die Lösung von Cabletron war man sogar sehr flexibel in der Auswahl der Technologien. Da der Smart Switch ausschließlich Layer-3-Switching unterstützt, kann dieses Gerät auch in einer Gigabit-Ethernet-Umgebung eingesetzt werden. Gerade der Smart Switch Router 8600 besitzt mit 30 Mio. Paketen/s eine hohe Performance. In ATM-Umgebung wird dabei ein weiterer Switch für die Umsetzung von Layer-3/MPOA und IEEE802.Q/QoS sowie IEEE802.1Q/ELAN benötigt. Somit ist zum ersten Mal der QoS von ATM einsetzbar; vorausgesetzt man verbindet sich direkt mit dem ATM-Backbone. Diese Lösung sieht dabei erst relativ komplex aus, besticht aber durch die Möglichkeit, GE und ATM gleichermaßen einsetzen zu können. So ist man in der Lage, im Etagenbereich mit Fast-Ethernet zu planen und im Backbone ATM einzusetzen. Ein Wechsel zu GE ist dabei jederzeit möglich, wenn das gewünscht wird.

7.3.2 MPLS

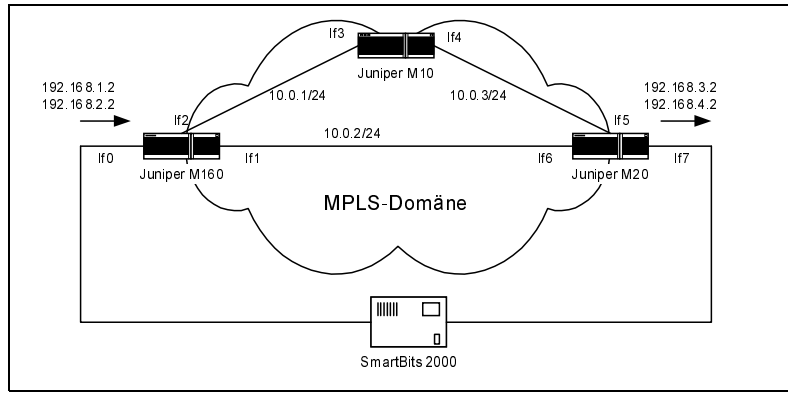
Der Schwerpunkt der praktischen Betrachtung von MPLS wird auf das Constraint Based Routing (CBR) gelegt. Das CBR wurde gewählt, da es sich hierbei um eine Hauptanwendung des MPLS handelt und es in den Testgeräten zur Unterstützung der IP-QoS-Modelle oder von VPNs zur Verfügung stand. Dabei ist das CBR über MPLS und BGP immer noch Gegenstand aktueller Standardisierungsbemühungen der MPLS-Working Group. Aus diesem Grund beruht die Untersuchung auf einer proprietären Implementierung, wobei die Basisspezifikationen abgeschlossen sind.

32 Nach 5 Paketen

Messaufbau Die Implementierung des Herstellers Juniper orientiert sich an den bereits festgelegten Spezifikationen. Für die Darstellung wird die Anwendung des Traffic Engineering und der QoS-Unterstützung betrachtet. Die QoS-Unterstützung darf hierbei aber nicht mit einer Unterstützung der IP-QoS-Modelle IntServ oder DiffServ verwechselt werden. Es handelt sich vielmehr um einen „isolierten" QoS, der ausschließlich für dieses Netz und nicht von der Quelle bis zur Senke realisiert wird.

Die für die Darstellung des Constraint Based Routing (CR) verwendete Teststellung basiert auf den drei Juniper-Routern M160, M20 und M10. Als weiteres Element der Teststellung wird das Messgerät SmartBit 2000 des Herstellers Netcom Systems verwendet. Es wird in dieser Teststellung der Transport von zwei FECs über zwei CR-LSPs betrachtet. Die Aufgabe des SmartBit in dieser Teststellung besteht in der Erzeugung von zwei kontinuierlichen Datenflüssen, welche die beiden FECs darstellen werden. Zu diesem Zweck werden über den SmartBit vier Hosts simuliert. Die Hosts 192.168.1.2 und 192.168.2.2 bilden die Quellen und die Hosts 192.168.3.2 und 192.168.4.2 die Senken. Das SmartBit wird über eine ATM-Schnittstelle an den M160 angeschlossen. An dieser Stelle erfolgt die Simulation der sendenden Hosts. Über das zweite ATM-Interface wird das SmartBit mit dem M20 verbunden. Hierbei simuliert das SmartBit die empfangenden Hosts.

Abb. 7.37
MPLS-Testszenario



Die drei Juniper-Router werden untereinander über PoS-Interfaces verbunden. Als Übertragungstechnik wird hierbei das SONET mit einer Übertragungsrate von 155 Mbit/s verwendet. Die Aufgabe der Juniper-Router besteht in dieser Teststellung im Transport der vom SmartBit generierten Datenflüsse vom M160 zum M20. Dieser Transport soll über zwei LSPs erfolgen. Aus diesem Grund wird auf den PoS-Schnittstellen MPLS konfiguriert. Die drei Juniper-Router bilden nun eine MPLS-Domäne. Als Label-Distribution-Protokoll muss für CBR in der Juniper-Implementierung RSVP verwendet werden. Die Aufgabe

des Routing-Protokolls übernimmt das Intermediate System to Intermediate System (IS-IS). Beide Protokolle werden ebenfalls auf den PoS-Interfaces konfiguriert. In einem letzten Schritt müssen die beiden LSPs im M160 mit dem Ziel M20 konfiguriert werden. Diese beiden LSPs werden mit den Namen LSP_Direkt und LSP_Indirekt bezeichnet. Jedem LSP wurde der Transport eines Datenflusses fest zugewiesen. Auf diese Weise kann die Funktion eines LSP über das Empfangen des entsprechenden Datenflow verifiziert werden. [FROMM01]

Messergebnisse

In diesem Testszenario sollte das Traffic Engineering als Anwendung des Constraint-Based-Routing gezeigt werden. Das Traffic Engineering erlaubt das kontrollierte Einrichten einzelner LSPs mit dem Ziel, das Netzwerk gleichmäßiger zu belasten. Zu diesem Zweck ist es im MPLS möglich, einzelnen Verbindungen unterschiedliche Übertragungsklassen mittels administrativer Attribute zuzuweisen. Über die Konfiguration eines LSP kann für dessen Aufbau die Verwendung bestimmter Links vorgeschrieben oder ausgeschlossen werden.

Das Testszenario soll diese Funktion für den Aufbau von LSPs in der praktischen Anwendung zeigen. Zu diesem Zweck wird ein LSP als Constraint-based Routing LSP konfiguriert. Der zweite LSP wird zum Vergleich als konventionell gerouteter LSP aufgesetzt. Es wird hierbei der Aufbau der beiden LSPs bei folgenden Vorgaben untersucht:

- intakte Verbindungen
- Trennung der Verbindung M160-M20
- Trennung der Verbindung M160-M10

Als Basis für dieses Testszenario wurde die beschriebene Teststellung verwendet. Für die Anwendung des TE wird aber zusätzlich der Verbindung M160-M20 die Übertragungsklasse *gold* und der Verbindung M160-M10-M20 die Übertragungsklasse *silver* zugewiesen. Dies wird über die Konfigurationen der Schnittstellen des M160 realisiert. Auf diese Weise darf der Aufbau dieses LSP nicht über Verbindungen der Übertragungsklasse *gold* erfolgen. Dieser LSP realisiert das TE. Die Konfiguration des LSP_Direkt hingegen enthält außer der Zieladresse keine weiteren Argumente. Das Routing für diesen LSP basiert auf konventionellem Routing.

Die Bestimmung des Pfades, über den ein LSP aufgebaut wird, erfolgt durch eine Berechnung des Algorithmus Constraint Shortest Path First (CSPF). Dieser Algorithmus basiert auf dem Algorithmus Shortest Path First, welcher beispielsweise von OSPF verwendet wird. Das CSPF ist aber zusätzlich in der Lage, weitere Parameter für die Routing-Entscheidung zu berücksichtigen. Für den Aufbau des LSP_Direkt bei intakten Verbindungen sollte der CSPF-Algorithmus die Verbindung mit der geringsten Anzahl von Hops (M160-M20) wählen. Für den LSP_Indirekt dürfte diese Verbindung nicht verwendet wer-

den, da seine Konfiguration die Nutzung von Links der Übertragungsklasse *gold* ausschließt. Dies sollte in den Berechnungen des CSPF-Algorithmus berücksichtigt werden. Der Aufbau des LSP_Indirekt sollte nur über den Pfad M160-M10-M20 möglich sein.

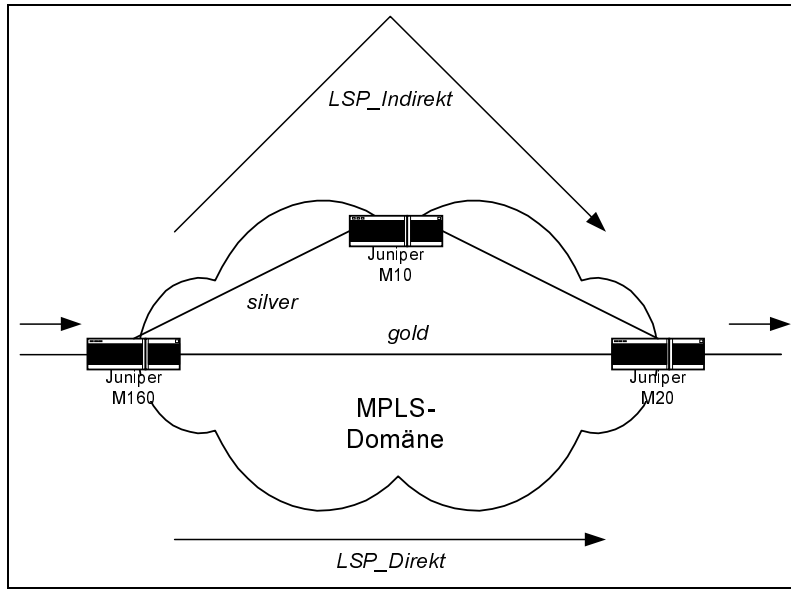


Abb. 7.38 Aufbau eines LSP bei intakter Verbindung

Die Trennung der Verbindung M160-M20 ist automatisch mit dem Abbau des LSP_Direkt verbunden. Über konventionelles Routing sollte für diesen LSP der Alternativpfad über die Verbindung M160-M10-M20 gewählt werden. Zunächst wurde die Verbindung M160-M20 wiederhergestellt. Die Trennung der Verbindung M160-M10 ist automatisch mit dem Abbau der LSPs LSP_Direkt und LSP_Indirekt verbunden. Für den LSP_Direkt sollte über konventionelles Routing der Alternativpfad über die Verbindung M160-M20 gewählt werden. Für den LSP_Indirekt sollte es über den CSPF-Algorithmus nicht möglich sein, einen Alternativpfad zu berechnen, da die Verwendung des Links M160-M20 über die Konfiguration ausgeschlossen wurde. Der LSP_Indirekt sollte sich nicht mehr aufbauen können.

Der CSPF-Algorithmus berechnete anschließend die Pfade für die beiden LSPs. Das Ziel des CSPF bestand hierbei in der Bestimmung eines Explicit Route Objects (ERO). Das ERO ist eine Adressliste und steuert den Aufbau der LSPs. Die LSPs wurden über die Knoten aufgebaut, die das ERO beinhaltet. Die Ergebnisse der CSPF-Berechnungen für die EROs der beiden LSPs ließen erkennen, dass der LSP_Direkt zum Knoten 10.0.2.2/32 und der LSP_Indirekt über 10.0.1.2/32 zum Knoten 10.0.3.2/32 aufgebaut wurde. Neben den CSPF-

Berechnungen gab ebenfalls ein Auszug aus der Routing-Tabelle des M160 Aufschluss über die gewählten Pfade der beiden LSPs, wie Abb. 7.40 zeigt. Der Auszug aus der Routing-Tabelle zeigt, dass der LSP_Direkt auf der Schnittstelle SO-0/1/3.0 und der LSP_Indirekt auf dem Interface SO-0/1/1.0 aufgebaut wurde.

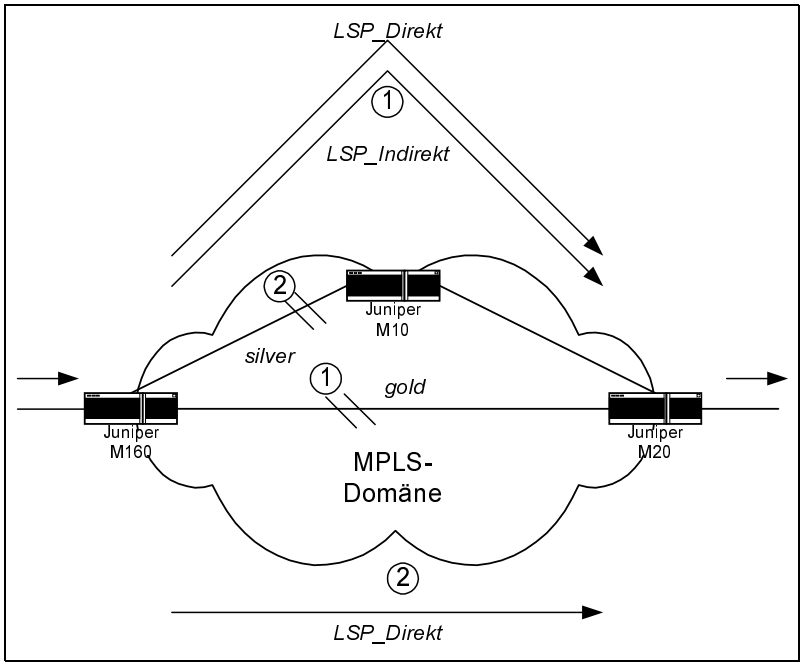


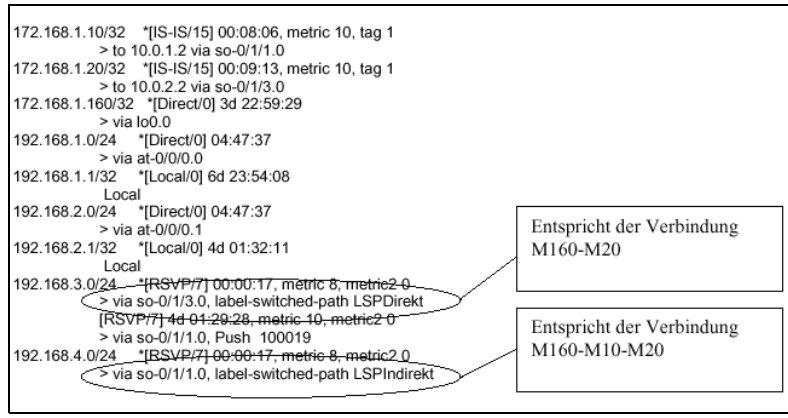
Abb. 7.39
LSP-Rerouting

Die erste CSPF-Berechnung ließ den LSP_Direkt über 10.0.1.2/32 zum Knoten 10.0.3.2/32 aufbauen. Eine weitere CSPF-Berechnung ließ über das Rerouting den LSP_Direkt die Verbindung zum Knoten 10.0.2.2/32 wiederholt aufbauen. Für den LSP_Indirekt hingegen ist es nicht möglich, einen Pfad zu bestimmen, wie Abb. 7.41 beweist.

In weiteren Tests wurde die Unterstützung von QoS durch das Constraint-based Routing (CR) untersucht. Diese basiert im CR auf einer Bandbreitenanforderung, die einem LSP zugewiesen werden kann. Der Aufbau dieses LSP darf dabei nur dann erfolgen, wenn es dem CSPF möglich ist, einen Pfad zu bestimmen, welcher diese Bandbreitenanforderung auf allen Verbindungen erfüllt. Mit dem Aufbau des LSP ist ebenfalls die Reservierung der geforderten Bandbreite verbunden. Zusätzlich ist es bei CR möglich, einem LSP eine Pre-emption zuzuordnen. Die Pre-emption stellt eine Priorisierung des LSP dar und gliedert sich in Setup- und Holding-Priorität. Mögliche Werte der Pre-emption bewegen sich zwischen 0 und 7, wobei 0 den höchstmöglichen Wert darstellt. Die Werte können für Setup- und Holding-Priorität voneinander abweichend

konfiguriert werden. Über die Pre-emption ist es einem LSP möglich, einen zweiten LSP abzubauen. Die Voraussetzung hierfür besteht in der höheren Setup-Priorität des ersten LSP im Vergleich zur Holding-Priorität des zweiten LSP. Der Abbau eines zweiten LSP erfolgt dabei nur dann, wenn für den ersten LSP kein Pfad gefunden werden kann, der seine Anforderungen wie eine bestimmte Bandbreite auf allen Links erfüllt. Durch den Abbau eines zweiten LSP können die frei werdenden Ressourcen für den Aufbau des ersten LSP verwendet werden. Das Rerouting des zweiten LSP ist anschließend von seinen Anforderungen und der Pre-emption abhängig.

Abb. 7.40
Routing-Tabelle



Das Testszenario soll die Funktion der Bandbreitenanforderung für den Aufbau von LSPs in der praktischen Anwendung zeigen. Zu diesem Zweck wurden beiden LSPs LSP_Direkt und LSP_Indirekt definierte Bandbreitenanforderungen zugewiesen. Um die Funktionsweise der Pre-emption aufzuzeigen, wurden den beiden LSPs unterschiedliche Prioritäten zugeordnet. Folgende Szenarien ließen sich dadurch ableiten:

- vollständig intakte Verbindungen
- Trennung der Verbindung M160-M10
- Trennung der Verbindung M160-M20

Das Basisszenario wird für die Tests weiterhin beibehalten. Für die QoS-Unterstützung wurde den beiden LSPs aber zusätzlich eine Bandbreitenanforderung von jeweils 100 Mbit/s zugewiesen. Um die Auswirkungen der Pre-emption auf das Rerouting darzustellen, wurden den LSPs unterschiedliche Prioritäten zugeordnet. Dem LSP_Direkt wird eine Setup- und Holding-Priorität von jeweils 0 zugewiesen, während dem LSP_Indirekt eine Setup- und Holding-Priorität von jeweils 7 zugewiesen wird. Auf diese Weise wurde der Aufbau des LSP_Direkt bevorzugt.

```
Jun  5 12:51:39 mpls lsp LSPIndirekt primary Down
Jun  5 12:51:39 mpls lsp LSPDirekt primary No Route
Jun  5 12:51:39 mpls lsp LSPIndirekt primary No Route
Jun  5 12:51:39 mpls lsp LSPDirekt primary Deselected as active
Jun  5 12:51:39 MPLS lsp LSPDirekt down on primary()
Jun  5 12:51:39 mpls lsp LSPIndirekt primary Deselected as active
Jun  5 12:51:39 MPLS lsp LSPIndirekt down on primary()
Jun  5 12:51:41 TED free LINK j2-re0.00(172.168.1.160)->Transit.00(172.168.1.10)
Jun  5 12:51:41 TED_2_CSPF Start
Jun  5 12:51:41 mpls lsp LSPDirekt primary CSPF: link down/deleted
Jun  5 12:51:41 CSPF adding path LSPDirekt(primary ) to CSPF queue 1
Jun  5 12:51:41 CSPF creating CSPF job
Jun  5 12:51:41 mpls lsp LSPIndirekt primary CSPF: link down/deleted
Jun  5 12:51:41 CSPF adding path LSPIndirekt(primary ) to CSPF queue 1
Jun  5 12:51:41 TED_2_CSPF end elapsed time 0.000159s
Jun  5 12:51:41 CSPF job starting
Jun  5 12:51:41 CSPF for path LSPDirekt(primary ), starting at j2-re0.00
Jun  5 12:51:41 CSPF final destination 172.168.1.20
Jun  5 12:51:41 CSPF starting from j2-re0.00 (172.168.1.160) to 172.168.1.20, hoplimit 254
Jun  5 12:51:41 CSPF Reached target
Jun  5 12:51:41 CSPF completed in 0.000083s
Jun  5 12:51:41 CSPF ERO for LSPDirekt(primary ) (1 hops)
Jun  5 12:51:41        node 10.0.2.2/32
Jun  5 12:51:41 mpls lsp LSPDirekt primary CSPF: computation result accepted
Jun  5 12:51:41 mpls lsp LSPDirekt primary Clear Call
Jun  5 12:51:41 CSPF job starting
Jun  5 12:51:41 CSPF for path LSPIndirekt(primary ), starting at j2-re0.00
Jun  5 12:51:41        path exclude: 0x00000004
Jun  5 12:51:41 CSPF final destination 172.168.1.20
Jun  5 12:51:41 CSPF starting from j2-re0.00 (172.168.1.160) to 172.168.1.20, hoplimit 254
Jun  5 12:51:41        constrains exclude 0x00000004
Jun  5 12:51:41 CSPF completed in 0.000077s
Jun  5 12:51:41 CSPF couldn't find a route to 172.168.1.20
Jun  5 12:51:41 mpls lsp LSPIndirekt primary CSPF failed: no route toward 172.168.1.20
Jun  5 12:51:41 CSPF job done!
Jun  5 12:51:41 mpls lsp LSPDirekt primary Up
Jun  5 12:51:41 mpls lsp LSPDirekt primary Record Route: 10.0.2.2 S
Jun  5 12:51:41 mpls lsp LSPDirekt primary Selected as active path
Jun  5 12:51:41 MPLS lsp LSPDirekt up on primary() Route  10.0.2.2 S
Jun  5 12:51:41 TED free LINK Transit.00(172.168.1.10)->j2-re0.00(172.168.1.160)
Jun  5 12:51:44 mpls lsp LSPIndirekt primary 10.0.2.2: Explicit Route: wrong delivery
Jun  5 12:51:53 mpls lsp LSPIndirekt primary No Route
```

Neuer Pfad für *LSPDirekt*

Kein alternativer Pfad für *LSPIndirekt* gefunden

Abb. 7.41
Auszug aus den CSPF-Berechnungen

Auf Grund der höheren Setup-Priorität sollte der CSPF-Algorithmus mit der Berechnung des Pfades für den LSP_Direkt beginnen. Da auf allen Links eine Bandbreite von 155 Mbit/s zur Verfügung stand, wurde für diesen LSP die Verbindung mit der geringsten Anzahl von Hops[33] gewählt. Auf diesem Link wurden 100 Mbit/s für den LSP_Direkt reserviert, wodurch sich die verfügbare Bandbreite auf 55 Mbit/s reduzierte. Für die anschließende Berechnung des Pfades für den LSP_Indirekt, welcher ebenfalls über eine Bandbreitenanforderung von 100 Mbit/s verfügte, bestand in der Verbindung M160-M20 keine Option mehr. Aus diesem Grund musste der CSPF-Algorithmus den alternativen Pfad M160-M10-M20, welcher auf allen Links eine verfügbare Bandbreite von 155 Mbit/s aufwies, für den Aufbau des LSP wählen. Auch an dieser Stelle sollte die Reservierung der geforderten 100 Mbit/s erfolgen. Abb. 7.42 zeigt die Bandbreitenreservierung auf den einzelnen Interfaces über das RSVP.

```
RSVP interface: 2 active
                  Active Subscr- Static    Available  Reserved   Highwater
Interface   State resv  iption BW        BW         BW         mark
so-0/1/1.0  Up       1  100%  155Mbps    55Mbps     100Mbps    100Mbps
so-0/1/3.0  Up       1  100%  155Mbps    55Mbps     100Mbps    100Mbps
```

Verfügbare Bandbreite

Reservierte Bandbreite

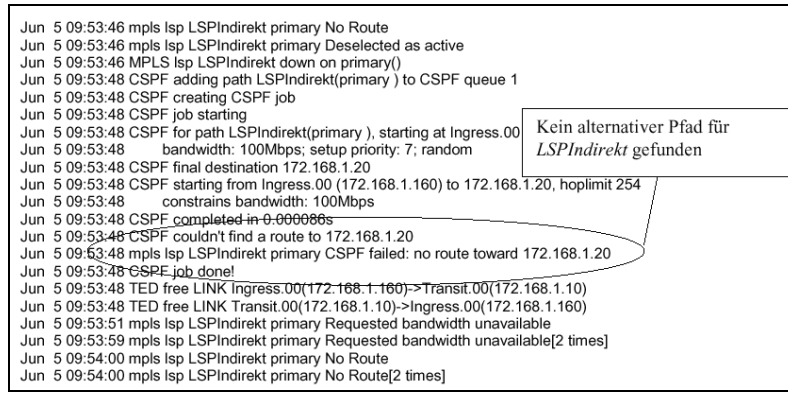
Abb. 7.42
Reservierung der Bandbreite mittels RSVP

33 M160 zu M20

Die Trennung der Verbindung M160-M10 war automatisch mit dem Abbau des LSP_Indirekt verbunden. Für das Rerouting des LSP_Indirekt musste der CSPF-Algorithmus einen alternativen Pfad zum M20 berechnen, welcher eine verfügbare Bandbreite von 100 Mbit/s aufwies. Die einzig mögliche Verbindung bestand dabei im Link M160-M20. Aufgrund der Reservierungen des LSP_Direkt standen hier aber nur 55 Mbit/s zur Verfügung. Aus diesem Grund sowie der geringen Setup-Priorität des LSP_Indirekt durfte hier ein Rerouting nicht möglich sein. Ein Auszug aus den CSPF-Berechnungen zeigt nach Abb. 7.43, dass es für den LSP_Indirekt nicht möglich war, einen Pfad zu bestimmen, wie vorausgesehen wurde.

Anschließend wurde die Verbindung M160-M10 wiederhergestellt. Als Ausgangssituation wurde dann die Trennung der Verbindung M160-M20 initiiert, die automatisch mit dem Abbau des LSP_Direkt verbunden ist. Für das Rerouting des LSP_Direkt musste der CSPF-Algorithmus einen alternativen Pfad zum M20 berechnen, welcher eine verfügbare Bandbreite von 100 Mbit/s aufwies. Die einzig mögliche Verbindung bestand im Link M160-M10-M20. Aufgrund der Reservierungen des LSP_Indirekt standen hier aber nur 55 Mbit/s zur Verfügung. Da die Setup-Priorität des LSP_Direkt aber höher als die Holding-Priorität des LSP_Indirekt ausgelegt war, musste der LSP_Indirekt abgebaut und seine Ressourcen für den Aufbau des LSP_Direkt zur Verfügung gestellt werden. Das darauffolgende Rerouting des LSP_Indirekt war nun aufgrund der verfügbaren Ressourcen und der Pre-emption-Werte nicht mehr möglich, was ebenfalls die Messungen bestätigten. [FROMM01]

Abb. 7.43
Auszug aus einer CSPF-Berechnung



Ein weiterer Test wurde von der Network Test, Inc., mit den Core Routern Cisco 12416 (12.0(14)SX) und Juniper Networks M160 (4.2R2.4/4.2E) durchgeführt. Bei einigen Tests waren auch Charlotte's Networks und Foundry dabei. Diese Geräte konnten aber nicht gleichermaßen in allen Szenarios eingesetzt werden,

weshalb sie hier nur erwähnt werden. Dabei wurde ebenfalls ein SmartBit 6000 Analysator mit 3505Terametrics-Karten als Messgerät verwendet sowie als PoS-Schnittstellen OC-48c (2,5 Gbit/s) und OC-192c (10 Gbit/s). Die Testplattformen bestanden aus drei Core und drei Edge Interfaces (OC-48c) sowie aus drei Core und 12 Edge Interfaces (OC-192c), sodass die Randschnittstellen beider Testplattformen die Kernkapazitäten ohne Überlast bewältigen konnten. Um eine möglichst realistische Abbildung des Internetverkehrs zu bekommen, wurde ein Verkehrsmix von 40 Byte (55%), 1500 Byte (23%), 576 Byte (17%) und 52 Byte (5%) erzeugt, welcher dem realen Verkehrsverhalten möglichst nah kommen sollte.

Zuerst wurde die maximale Forwarding-Datenrate sowie der zu erreichende Datendurchsatz beider eingesetzter Schnittstellen gemessen. Dabei konnten die Geräte von Cisco Systems und Juniper Networks den Datenverkehr mit Wirespeed bei der OC-48c-Schnittstelle handhaben, während die beiden anderen Hersteller Ausfälle aufwiesen. Dabei wurden 40-Byte-Pakete und der erwähnte Internet-Mix eingesetzt. Die OC-192c-Schnittstelle wies dann aber auch hier die ersten Einbrüche auf, obwohl Cisco den Internet-Mix ohne Verzögerung handhaben konnte und Juniper nur 10% Einbrüche hatte. Allerdings machte Cisco die 40-Byte-Datenrate zu schaffen, sodass hier nur 52% der Gesamtperformance erreicht werden konnten, während Juniper immerhin auf 92,2% kam. Beim Forwarding sahen die Ergebnisse der beiden unterschiedlichen Verkehrsströme schon anders aus. Hier konnte bei allen Geräten eine gute Performance der 40-Byte-Datenströme erreicht werden, während der Internet-Mix-Verkehr stark absackte, da dieser schwerer zu handhaben ist, wie Abb. 7.44 verdeutlicht.

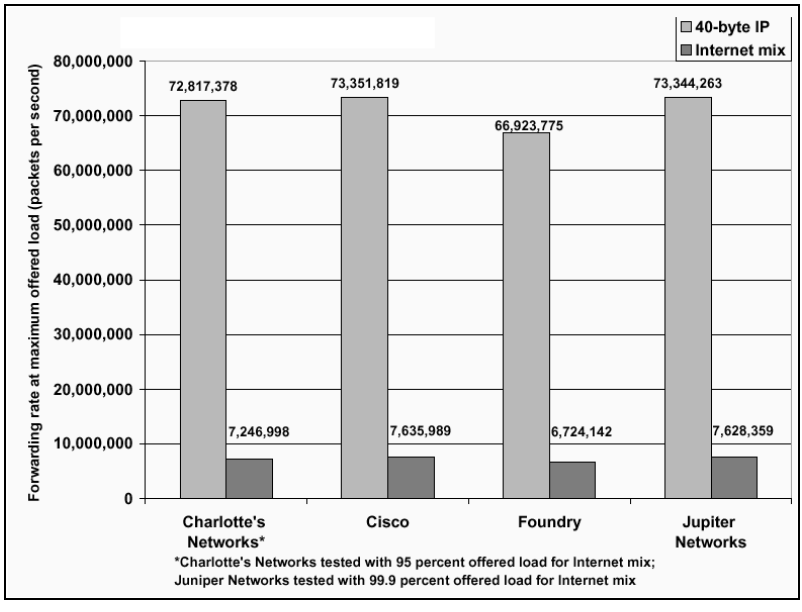


Abb. 7.44
OC-48c Forwarding Rate bei IP-Routing

Im nächsten Schritt wurde MPLS auf den möglichen Datendurchsatz überprüft. Während vorher IP-Routing zum Einsatz kam, konnten jetzt direkte Pfade etabliert werden. Die Datenrate lag ungefähr bei 100% bei allen Testkomponenten. Einzige Ausnahme war Juniper bei dem Internet-Mix-Verkehr, wo nur 88% erreicht werden konnte. Anschließend wurden die Verzögerungen auf der OC-48c-Schnittstelle gemessen. Hier verhielt sich Juniper ebenfalls konstant und konnte die gleichen Ergebnisse erzielen wie beim IP-Routing. Cisco hatte allerdings Probleme, da die Verzögerung von 2 ms auf 16 ms bei 40-Byte-Paketen anstieg, wenn die Pakete über MPLS transportiert wurden. Der Internet-Mix stieg im Vergleich dazu unwesentlich von 250 auf 650 µs. Junipers Werte lagen dagegen im Bereich von 170 µs. Bei der OC-192c-Schnittstelle verhält sich die Messsituation anders. Die maximale Verzögerung von Juniper beläuft sich auf ca. 11 ms bei 40-Byte-Paketen. Dies entsprach einer ungefähren Verdopplung der Verzögerung im Vergleich zu Cisco. Man kann die Verzögerung zwar weiter senken. Dies würde allerdings bedeuten, dass man Paketverluste in Kauf nehmen müsste, was bei dieser Messung nicht getan wurde. Bei Einsatz von MPLS verhielt sich Juniper identisch zu den vorher ermittelten Werten. Ciscos Verzögerung stieg dagegen bei 40-Byte-Paketen von 26,4 µs auf bis zu 13 ms an. Beim Internet-Mix stieg der Wert ebenfalls von 500 µs auf 2,5 ms an. Die Tests machten hier deutlich, wie wichtig der korrekte Einsatz der Pufferspeicher ist.

Weiterhin spielt die Kapazität von MPLS bezüglich der zur Verfügung stehenden Tunnels eine entscheidende Rolle. Da MPLS verbindungsorientiert arbeitet, muss ein Router zuerst einen Tunnel[34] aufsetzen, bevor Daten ausgetauscht werden können. Bei den Tests schaffte der Cisco-Router 5000 LSPs, während Juniper bereits auf 10.000 kam. Beide Werte sind in Ordnung, können in einem Providernetz aber an ihre Leistungsgrenze stoßen.

Abschließend wurden die QoS-Merkmale untersucht. Ähnlich wie bei den ersten MPLS-Szenarien wurden die CoS-Klassen *gold*, *silver* und *bronze* konfiguriert. Dabei war das Verhältnis zwischen den Klassen 70:20:5. Um QoS zu testen, wurde auch hier eine Verbindungsunterbrechung durchgeführt und eine Überbuchung von 150% eingestellt. Bei den Tests konnte Cisco eine unwesentlich höhere Forwarding-Datenrate verbuchen als Juniper. Die Abb. 7.46 zeigt die Forwarding-Datenrate beider Gerätetypen. In beiden Fällen (OC-48c und OC-192c) wurde der *gold*-Verkehr klar bevorzugt, während *silver* und *bronze* ebenfalls in der richtigen Reihenfolge priorisiert wurde. Allerdings mussten auch durch die Überbuchung Pakete verworfen werden. Ciscos Router verlor 0,1% der *gold*-Pakete, während Juniper 2,3% verwarf. Der höher prioriserte Datenverkehr ist somit sehr gut durch beide Geräte geschützt worden. Bei

34 Label Switched Path

Betrachtung des *silver*-Datenverkehrs wurden allerdings höhere Paketverluste erzielt. Hier verlor der Cisco-Router zu viele *silver*-Pakete, um in der voreingestellten Priorisierungsrate zu bleiben. Das Verhältnis von Cisco belief sich auf 70:16:5, während Juniper auf 70:14:5 kam.

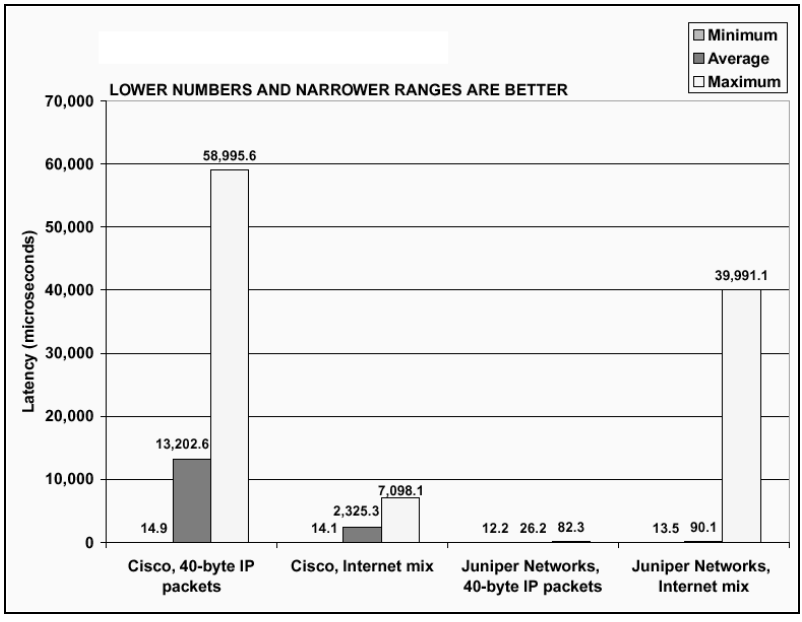


Abb. 7.45
OC-192c-Verzögerung über MPLS

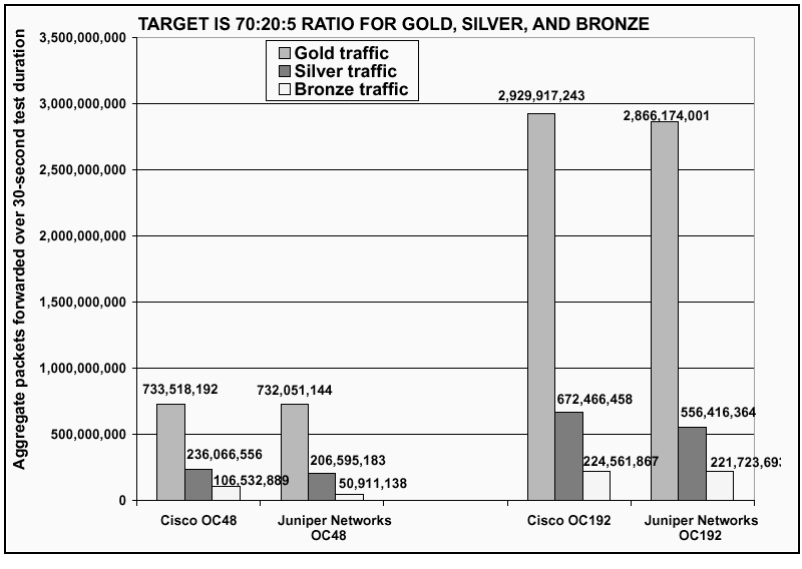


Abb. 7.46
IP Class-of-Service Forwarding-Datenrate

Die Tests zeigten insgesamt, dass kein MPLS-Router bislang in der Lage ist, eine bestimmte Serviceklasse ohne Paketverluste bei Überbuchungen oder Überlast zu meistern. Es wurde allerdings annähernd das beste Ergebnis erzielt, was ebenfalls stark von dem eingesetzten Interface abhängig war. Durch verbesserte Pufferung der Daten erhofften sich beide Hersteller eine bessere Einhaltung der CoS-Randbedingungen. Dies würde aber bei Echtzeitverkehr nicht zum Tragen kommen. [NEWM01]

Auswertung Die Ergebnisse wurden in verschiedenen Testszenarien durchgemessen. Sowohl der Aufbau als auch das Rerouting waren fehlerfrei durchzuführen. Dieser Mechanismus arbeitet bereits in bestehenden Produkten. Auf diese Weise konnte die Funktionalität des Constraint-based Routing für die Anwendungen des Traffic Engineering und der QoS-Unterstützung vollständig nachgewiesen werden.

Der zweite Abschnitt der Messungen zeigt anschließend auf, dass durch MPLS teilweise sehr geringe Verzögerungen umgesetzt werden können, aber auch Paketverluste teilweise in Kauf genommen werden mussten. Diese Verluste entstehen trotz Priorisierung durch Überbelastung, welche sich auch durch größere Pufferspeicher nicht komplett unterdrücken lassen. Trotzdem lassen sich Echtzeitapplikationen dadurch wesentlich besser unterstützen, als dies durch überzogene Bandbreiten möglich wäre.

7.4 Voice-over-IP (VoIP)

Nachdem die Plattform eingehend in verschiedensten Messverfahren untersucht wurde, wird abschließend die Echtzeitapplikationen einem Testverfahren unterzogen. Dies geschieht auf der einen Seite, um die Adaptierbarkeit sicherstellen zu können, und auf der anderen Seite, um die paketbasierte Übertragung unter optimalen Bedingungen kritisch zu hinterfragen.

Im Folgenden soll nun eine Aussage über die Qualität gemacht werden. Hierbei wird nur VoIP im Intranet oder Internet verglichen. Im Intranet ist die Qualität relativ gut, da bei der Kommunikation zwischen zwei Stellen nur wenige Router-Hops liegen. Dadurch bleiben Ende-zu-Ende-Verzögerungen bei einem Anruf innerhalb eines Netzwerks niedrig. Unternehmensnetzwerke können außerdem besser konfiguriert werden, beispielsweise durch die Errichtung eines Prioritätsprotokolls für die Router, welches die Verzögerungen reduziert und so eine gute Gesprächsqualität von Endgerät zu Endgerät garantiert. Darüber hinaus treten in derartigen Netzwerken verhältnismäßig geringe Paketverluste auf. Eine vorausschauende Fehlerverbesserung[35] gleicht zusätzlich den gelegentlich auftretenden Verlust aus, ohne dass der Anwender eine

35 Forward Error Correction

Qualitätsminderung bemerkt. VoIP ermöglicht PSTN-Qualität in Unternehmensnetzwerken, wobei auf bestehende Geräte des Sprach-/Daten-Netzwerks[36] zugegriffen werden kann.

Dagegen ist die Sprachqualität im Internet deutlich reduziert, da die Sprachkommunikation hier anderen Rahmenbedingungen unterliegt, die durch die Art des Netzwerks geprägt sind. Zahlreiche und sich ständig ändernde Richtungssprünge haben Verzögerungen und die zerhackte Übertragung von Sprache zur Folge. Diese Unterbrechungen und Störungen können bis zu mehreren Sekunden dauern, was sich im Bereich der Sprachqualität deutlich bemerkbar macht. Paketverluste von bis zu 10% übersteigen auch die Möglichkeiten eines FEC-Mechanismus. FEC ist somit nicht in der Lage, die verlorenen Stimmsignale zu kompensieren. Die Sprachqualität im Internet kann also keine gleich bleibende Qualität gewährleisten. Unterschiedliche Verkehrsaufkommen im Internet erlauben zu einem bestimmten Zeitpunkt einen hochwertigen Anruf, zu anderer Zeit eine deutlich schlechtere Verbindung. Dies liegt am fehlenden Quality-of-Service (QoS), was bereits ausführlich dargestellt wurde.

Zum Vergleich unterschiedlicher Implementierungen mit Fokus auf die Sprachqualität und Verzögerung wurde ATM als Basisübertragungsnetz mit AAL-Typ5 (typische Paketvermittlung) und AAL-Typ1 (Video- und Audioübertragung) verwendet. Dadurch fällt auch ein Vergleich zu Broadcast-Medien wie Ethernet leichter, da AAL-Typ5 nicht die ATM-spezifischen Eigenschaften nutzt, während AAL-Typ1 die höchste Güte innerhalb eines ATM-Netzes darstellt. Diese Güte kann in paketvermittelten Netzen nicht vergleichbar realisiert werden, weshalb hier die Limitierungen der VoIP-Technik aufgezeigt werden können.

7.4.1 Messaufbau

Die Verzögerung stellt für Echtzeitdaten eine sehr wichtige Größe dar. Sie kann berechnet werden, um verschiedene Verfahren im Voraus vergleichen zu können. Es wird bei der Berechnung davon ausgegangen, dass aus einem LAN über ein WAN in ein LAN gesendet wird. Berücksichtigt wird die Verzögerung des Sprachpaketsenders (SPS), der beiden LANs, des WAN und des Sprachpaketempfängers (SPE).

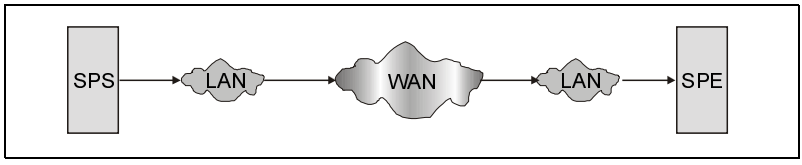


Abb. 7.47 Netzaufbau

36 Wie z.B. bereits installierte Telefone

Folgende Kodierverfahren werden für die Berechnung berücksichtigt:

- PCM (G.711) mit 64 Kbit/s
- LD-Celp (G.728) mit 16 Kbit/s
- A-Celp (G.723.1) mit 5,3 Kbit/s

Die Verzögerung bezieht sich auf ein Paket einer Sprachanwendung mit 80 Byte. Daraus ergeben sich für den SPS typische Werte *tSD* von:

$$t_{SD} = \frac{Paketgröße}{Bitrate}$$

Woraus sich für die Kodierverfahren folgende Verzögerungszeiten ergeben:

G.711 $t_{SD} = \frac{80\,Byte}{64\,Kbit/s} = \frac{640\,Bit}{64\,Kbit/s} = 10\,ms$

G.728 $t_{SD} = \frac{80\,Byte}{16\,Kbit/s} = \frac{640\,Bit}{16\,Kbit/s} = 40\,ms$

G.723.1 $t_{SD} = \frac{80\,Byte}{5{,}3\,Kbit/s} = \frac{640\,Bit}{5{,}3\,Kbit/s} = 120{,}8\,ms$

Für den SPE ergeben sich die Verzögerungen *tPD* aus der Größe des Puffers und den Bitraten. In IP-Netzen haben die Puffer üblicherweise eine Größe von 256 Byte:

$$t_{PD} = \frac{Puffergröße}{Bitrate}$$

Damit ergibt sich die Pufferverzögerung *tPD-IP* für:

G.711 $t_{PD-IP} = \frac{256\,Byte}{64\,Kbit/s} = \frac{2048\,Bit}{64\,Kbit/s} = 32\,ms$

G.728 $t_{PD-IP} = \frac{256\,Byte}{16\,Kbit/s} = \frac{2048\,Bit}{16\,Kbit/s} = 128\,ms$

G.723.1 $t_{PD-IP} = \frac{256\,Byte}{5{,}3\,Kbit/s} = \frac{2048\,Bit}{5{,}3\,Kbit/s} = 386{,}4\,ms$

Nach dem ITU-T Standard E.735 kann der Puffer für ATM klein gewählt werden, da der Jitter mit dem Serviceparameter Cell Delay Variation Tolerance (CDVT) festgelegt wird. Bei einer Speichergröße von einer ATM-Zelle mit einem 48-Byte-Payload ergibt sich die Pufferverzögerung *tPD-ATM* wie folgt:

G.711 $t_{PD-ATM} = \frac{48Byte}{64Kbit/s} = \frac{384Bit}{64Kbit/s} = 6ms$

G.728 $t_{PD-ATM} = \frac{48Byte}{16Kbit/s} = \frac{384Bit}{16Kbit/s} = 24ms$

G.723.1 $t_{PD-ATM} = \frac{48Byte}{5{,}3Kbit/s} = \frac{384Bit}{5{,}3Kbit/s} = 72{,}5ms$

Für ein reines IP-Netz würden sich deshalb die in Abb. 7.48 gezeigten folgenden Verzögerungen einstellen.

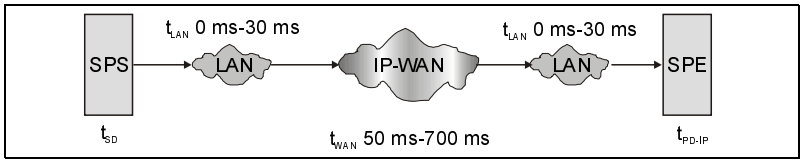


Abb. 7.48 Optimale Verzögerungswerte in einem IP-Netz

Durch Addition der Werte ergibt sich eine minimale Gesamtverzögerung von *tGDmin*:

$$t_{GD\min} = t_{SD} + 2 \cdot t_{LAN\min} + t_{WAN\min} + t_{PD-IP}$$

G.711 $t_{GD\min} = 10ms + 50ms + 32ms = 92ms$

G.728 $t_{GD\min} = 40ms + 50ms + 128ms = 218ms$

G.723.1 $t_{GD\min} = 120ms + 50ms + 386{,}4ms = 556{,}4ms$

Die maximale Verzögerung *tGDmax* ergibt:

$$t_{GD\max} = t_{SD} + 2 \cdot t_{LAN\max} + t_{WAN\max} + t_{PD-IP}$$

G.711 $t_{GD\max} = 10ms + 2 \cdot 30ms + 700ms + 32ms = 802ms$

G.728 $t_{GD\max} = 40ms + 2 \cdot 30ms + 700ms + 128ms = 928ms$

G.723.1 $t_{GD\max} = 120ms + 2 \cdot 30ms + 700ms + 386{,}4ms = 1266{,}4ms$

In IP-Netzen spielt die benötigte Bandbreite und die Auslastung des Netzes die wichtigste Rolle, da dadurch die Übertragungszeit maßgeblich beeinflusst wird. Deshalb werden Komprimierungsverfahren mit geringer Bandbreite genutzt.

Dem widerspricht allerdings die notwendige hohe Paketierungs- und Pufferzeit für die einzelnen Verfahren, wie hier dargestellt wurde.

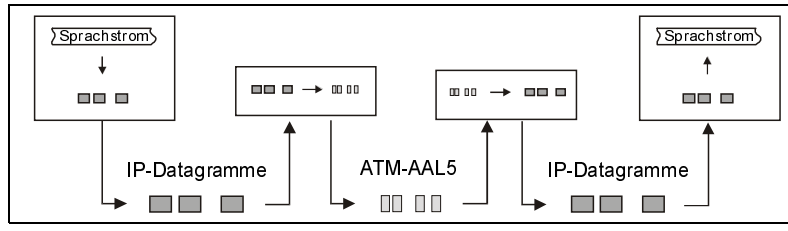


Abb. 7.49 Anpassung mittels AAL-Typ5 ohne Sprachstromwandlung

Die bereits erwähnten Anpassungsverfahren CLIP, LANE und MPOA dienen zur Übertragung von IP-Daten über ein ATM-Backbone (IP-over-ATM). Sie haben alle eines gemeinsam: sie verwenden AAL-Typ5 als gemeinsame Übertragungsbasis. Auch MPLS muss diese Adaptionsschicht von ATM nutzen, da IP asynchrone und verbindungslos gesendete Datagramme (Pakete) benutzt. Diese Pakete können mit dem ebenfalls verbindungslosen AAL-Typ5 von ATM angepasst werden.

Tab. 7.5 Schichten-Overhead

Schichten	**Header [Byte]**
Real-Time Protocol (RTP)	16
User Datagram Protocol (UDP)	8
Internet Protocol Version 4 (IP)	20
Logical Link Control (LLC)	8
LAN Emulation (LANE)	16
∑ Overhead	68

Die IP-Pakete werden am ersten Übergang von IP nach ATM segmentiert und in ATM-Zellen übertragen. Am zweiten Übergang werden die IP-Pakete wieder aus den ATM-Zellen zurückgewonnen. Um Echtzeitdaten wie Sprache über das ATM-Netz zu übertragen, ist aber im Grunde AAL-Typ1 vorgesehen, das für Sprache den optimalen Transport garantiert. Allerdings müssten die IP-Pakete vor dem ATM-Netz in den Sprachstrom zurückgewandelt werden, um beim Übergang zum IP-Empfänger-Netz wiederum in Datagramme zurückgewandelt werden zu können. Für diesen Lösungsansatz ist ein Gateway notwendig.

Diese Umwandlung entfällt bei den Verfahren mit AAL-Typ5, da sie erst im Empfänger vorgenommen wird. Wenn aber ATM eingesetzt wird, sollte für jeden Dienst der optimale Transport gewährleistet werden, was für Sprachübertragung AAL-Typ1 ist. Neben den besseren Transportmöglichkeiten für Sprachdaten wird der Overhead erheblich verkleinert. Bei den klassischen Verfahren

von IP-over-ATM werden mindestens die TCP (bzw. UDP) und IP-Header zusätzlich zu den Headern, die sich aus dem ATM-Referenzmodell ergeben, übertragen. Wenn H.323 benutzt wird, kommt zusätzlich der RTP-Header hinzu. Tab. 7.5 führt die einzelnen Protokolle auf, die verwendet werden müssen.

Der Transport von einem hohen Overhead-Verhältnis belastet zusätzlich das Netzwerk und erhöht die benötigte Bitrate. Durch die höhere Bitrate werden die Ressourcen von ATM auch schneller erschöpft. Diese zusätzlichen Header entfallen, wenn die Sprachdaten über AAL-Typ1 übertragen werden. Denn durch die bedingte Rücktransformation in den Sprachstrom entstehen hier nur ATM-Header bei der Übertragung.

Bei AAL-Typ1 ist im Vergleich zu AAL-Typ5 der Verbindungsaufbau noch aufwendiger. Die Zeit für den Verbindungsaufbau kann aber trotz der größeren Komplexität schneller vonstatten gehen, wenn im WAN ATM eingesetzt wird, welches geringere Verzögerungen aufweist. Die Signalisierung zwischen den End-Terminals muss den Gateways auch kenntlich gemacht werden, damit sie dementsprechend darauf reagieren können. Dies kann den Verbindungsaufbau wiederum etwas verzögern, da zuerst die AAL1-Verbindungen aufgebaut werden sollten, damit sie bei Bedarf sofort zur Verfügung stehen. Ist der Verbindungsaufbau abgeschlossen, ergibt sich eine Verringerung der Verzögerung durch AAL-Typ1.

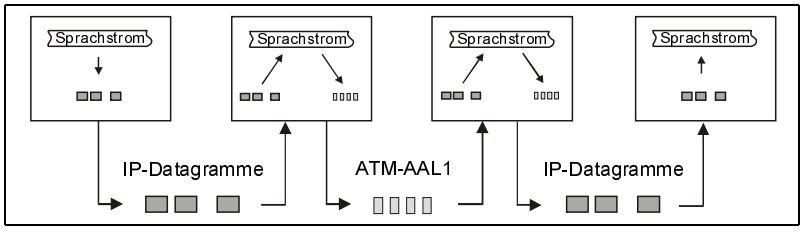


Abb. 7.50 Sprachübertragung über AAL-Typ1

Wenn im WAN ein ATM-Backbone mit AAL-Typ1 verwendet wird, verändern sich die Zeiten erheblich. Es müssen in den Gateways, die jeweils einem SPE und SPS entsprechen, die jeweiligen Verzögerungen der Puffer addiert werden. Der Pufferspeicher kann aber im Gegensatz zum IP-Netz eine geringere Speicherkapazität haben. Er kann im Bereich von 1 bis 2 Paketen[37] liegen. Der geringere Speicher in den LANs kann deshalb gewählt werden, weil es sich um kleinere, in den Ressourcen überschaubare Netze handelt, in denen für ausreichend Bandbreite gesorgt werden kann. Daraus ergeben sich neue Verzögerungen für *tPD-IP* von 20ms (G.711), 80ms (G.728) und 241,5ms (G.723.1). Die Verzögerungen des ATM-WAN wurden aus dem B-WIN entnommen, die für PVCs die jeweiligen Werte angeben. [PAKO99]

37 80-160 Byte Payload des IP-Datagramms

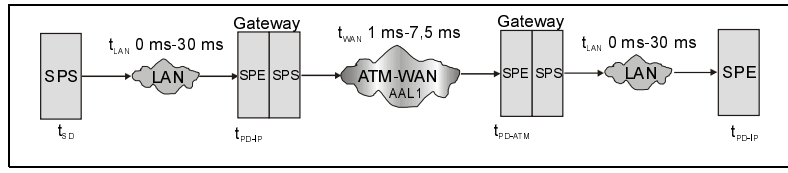


Abb. 7.51 Verzögerungswerte mit AAL-Typ1 im Backbone

Durch Addition der Werte ergibt sich eine minimale Gesamtverzögerung *tGDmin* von:

$$t_{GD\,\min} = t_{SD} + 2 \cdot t_{LAN\,\min} + 2 \cdot t_{PD-IP} + t_{WAN\,\min} + t_{PD-ATM}$$

Die maximale Gesamtverzögerung hingegen beläuft sich auf:

$$t_{GD\,\max} = t_{SD} + 2 \cdot t_{LAN\,\max} + 2 \cdot t_{PD-IP} + t_{WAN\,\max} + t_{PD-ATM}$$

Wenn im ATM-WAN mit AAL-Typ5 gearbeitet wird, ergeben sich kleinere Verzögerungen, weil die zusätzliche Pufferung, die in den Gateways benötigt wird, entfällt.

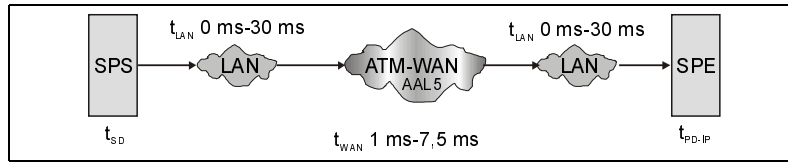


Abb. 7.52 Verzögerungswerte mit AAL-Typ5 im Backbone

Durch Addition der Werte ergibt sich eine minimale Gesamtverzögerung *tGDmin*:

$$t_{GD\,\min} = t_{SD} + 2 \cdot t_{LAN\,\min} + t_{WAN\,\min} + t_{PD-IP}$$

Die maximale Gesamtverzögerung *tGdmax* lässt sich folgendermaßen berechnen:

$$t_{GD\,\max} = t_{SD} + 2 \cdot t_{LAN\,\max} + t_{WAN\,\max} + t_{PD-IP}$$

7.4.2 Messergebnisse

Zusammenfassend werden die Verzögerungswerte in Tab. 7.6 dargestellt. Es kann deutlich festgestellt werden, dass mit zunehmender Komprimierung[38] die Verzögerung zunimmt. Das Verfahren G.723.1 ist nach den Werten nur für das Verfahren mit AAL-Typ5 geeignet. Denn bei den anderen beiden Verfahren liegt

38 Kleinere Bitrate

die Verzögerung deutlich über 250 ms[39], die noch für gute Gesprächsqualität stehen, wenn Kompensationsverfahren eingesetzt werden.

Verfahren		IP	ATM-AAL1	ATM-AAL5
G.711	Minimum	92	57	31
	Maximum	802	123,5	97,5
G.728	Minimum	218	225	121
	Maximum	928	291,5	187,5
G.723.1	Minimum	556,4	676,5	326,5
	Maximum	1266,5	74,3	429

Tab. 7.6 Verzögerungswerte der verschiedenen Backbone-Netze

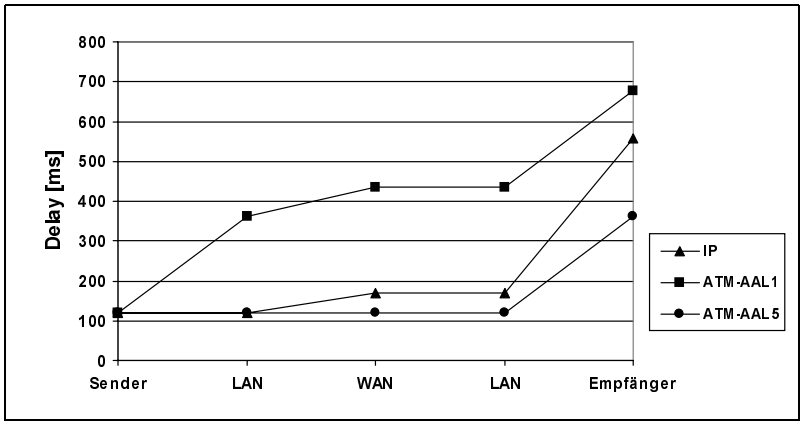


Abb. 7.53 Gesamte Verzögerung eines Übertragungspfades nach G.723.1

In Abb. 7.53 wurde die Verzögerung vom Sender bis zum Empfänger aufsummiert. Hier wird deutlich, dass die Verzögerung vor allem vom Paketierer (Sender) und Puffer (Empfänger) herrührt.

Abb. 7.54 zeigt hingegen die Verzögerung in Abhängigkeit von der Paketgröße. Hier wird deutlich, dass durch kleinere Pakete als die hier angenommenen 80 Byte auch die Verzögerung gesenkt werden kann. Auch durch kleinere Puffer könnte die Verzögerung gesenkt werden.

Gerade für IP sind kleine Bitraten wichtig, denn diese belasten das Netz weniger. Durch geringe Belastung entstehen kleinere Jitter auf der Übertragungsstrecke, weil mehr freie Ressourcen zur Verfügung stehen. Wenn der Jitter klein ist, kann auch der Puffer kleiner dimensioniert werden, wodurch die Verzögerung wiederum sinkt. Also haben Kodierverfahren mit geringer Bitrate vor allem bei IP-Übertragung ihre Berechtigung.

39 Stellt nach ITU G.114 noch eine gute Sprachqualität unidirektional zur Verfügung.

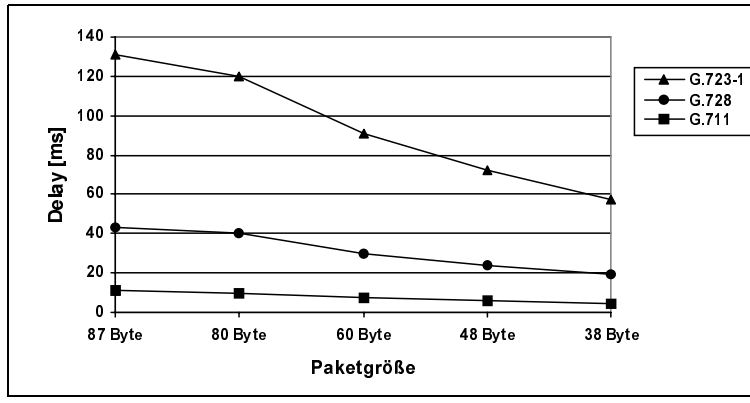


Abb. 7.54 Verzögerung in Abhängigkeit von der Paketgröße

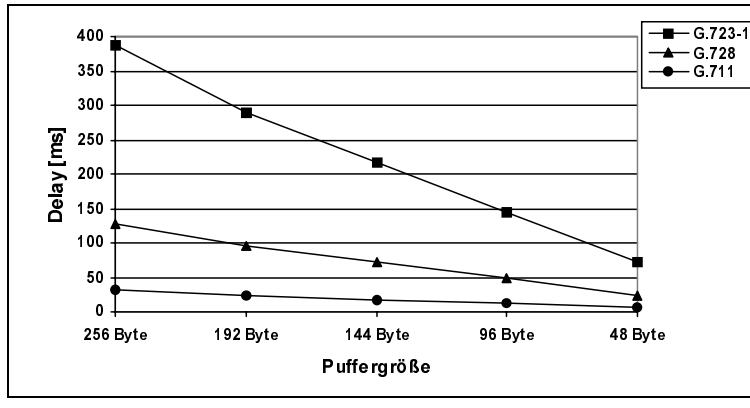


Abb. 7.55 Entstehender Delay durch die Puffergröße

In den ATM-Netzen ist dagegen der Einfluss der Belastung auf die Verzögerung nur gering. Denn durch die garantierten Service-Parameter wird auch der Delay und Jitter auf einen geforderten Wert begrenzt. Abb. 7.56 verdeutlicht die Unterschiede im direkten Vergleich mittels des Kodierverfahrens G.723.1.

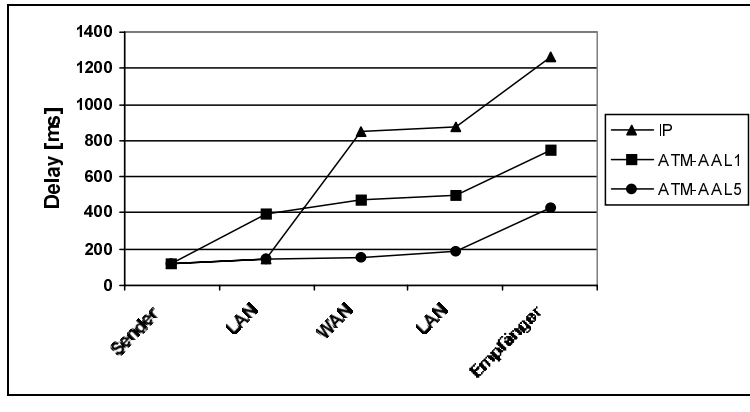


Abb. 7.56 Aufsummierter maximaler Delay für G.723.1

7.4.3 Auswertung

Bei der IP-Übertragung gibt es eine starke Verzögerungsabweichung auf der Übertragungsstrecke (LAN-zu-LAN). Mit der Strecke ATM-WAN ist die Verzögerung wesentlich konstanter. Damit lässt sich auch erklären, warum die minimalen Delays mit IP etwas geringerer sind als die mit AAL-Typ1. Denn vor dem Hintergrund der starken Schwankungen der IP-Verzögerungen muss hier der mittlere Delay betrachtet werden.

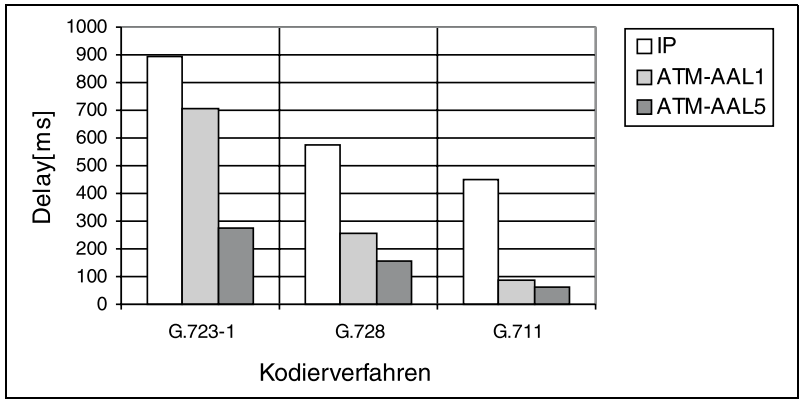


Abb. 7.57 Mittlere Gesamtverzögerung

Durch Abb. 7.57 wird deutlich, dass der IP-Delay durchaus höher ist als die Verzögerung im ATM-WAN. Mit den beschriebenen Methoden[40] kann das IP-Netz verbessert werden. Es bleiben aber die Verzögerungen auf jeden Fall erhalten, die sich durch die Eigenschaften des IP-Protokolls und der Verarbeitungszeit ergeben.

Bei ATM besteht die Alternative zwischen AAL-Typ1 und AAL-Typ5. Seit kurzem kann auch AAL-Typ2 für komprimierte Sprach-/Datenübertragung eingesetzt werden. Nach den ermittelten Werten besitzt AAL-Typ5 den geringsten Delay. Allerdings rührt die größere Übertragungszeit von AAL-Typ1 von den zusätzlichen Puffern her. Durch bessere Messungen mit der Gateway-Methode könnten durch bessere Auslegung der Puffer[41] geringere Verzögerungen erzielt werden. Außerdem ist zu berücksichtigen, dass bei der AAL-Typ5-Methode mit LANE kein QoS festgelegt werden kann. Somit ist die Verzögerung wie bei IP, abhängig von der Belastung des Netzes. Mit MPOA könnte hingegen ein QoS festgelegt werden. AAL-Typ1 besitzt allerdings die optimale Anpassung für die Übertragung von Echtzeitdaten über ATM.

40 RSVP, IntServ, DiffServ, MPLS usw.
41 Dynamische Wahl

Weiterhin spricht der hohe Overhead von den herkömmlichen IP-over-ATM-Methoden (CLIP, LANE und MPOA) gegen die Übertragung von Sprache mit AAL-Typ5. Dieser Overhead entfällt bei Einsatz mit der Gateway-Methode mit AAL-Typ1. Auch Daten können mit der Gateway-Methode mit geringerem Overhead transportiert werden, wenn bei der Signalisierung AAL-Typ5 vereinbart wird. Zusammengefasst ergeben sich folgende Vor- und Nachteile für die AAL-Typ1-Lösung:

1. **Vorteile:**

a) Die entstehenden Header von RTP/UDP/IP müssen nicht mit über das ATM-Netz übertragen werden.
b) Es gibt eine Delay-Verringerung im WAM-Netz.
c) Es kann der Quality-of-Service von ATM-AAL1 genutzt werden.
d) Eine Anpassung an ISDN-Endgeräte ist sehr einfach, da der AAL-Typ1-Datenstrom direkt als B-Kanal-Datenstrom von ISDN übernommen werden kann.

2. **Nachteile:**

a) Fehler aus dem ersten LAN-Netz können am Empfänger des zweiten LAN-Netzes nicht mehr behoben werden.
b) Die Gateway-Entwicklung ist sehr aufwendig.

Diese Auswertung von VoIP sollte nicht ATM- mit IP-Lösungen vergleichen. Vielmehr sollten auf die Verarbeitung, Pufferung und Kodierung eingegangen werden, die jeweils Grundverzögerungen nach sich ziehen, sodass VoIP immer dadurch beeinflusst wird.

8

Evaluierung und Aussicht

Nachdem die einzelnen Ebenen für eine sichere, verzögerungsgeringe Kommunikation aufgebaut und durchgemessen wurden, wird abschließend in diesem Kapitel eine abschließende Bewertung mit einem Ausblick über die jeweiligen Themenschwerpunkte gegeben.

8.1 Sicherheitsinfrastruktur

Beim Aufbau einer Sicherheitsinfrastrukturen bzw. Public Key Infrastructure (PKI), die unterschiedliche Anwendungsbereiche unterstützen soll, sind heute unabhängig von IPsec noch eine Reihe von Problemen zu lösen. Diese Probleme ergeben sich auf der einen Seite durch die teilweise mangelhafte Interoperabilität und Funktionalität der auf dem Markt angebotenen Produkte und andererseits sind sie durch Schwächen in den Standards enthalten. Das heißt, unterschiedliche Anwendungsgebiete werden in sehr unterschiedlicher Weise von einzelnen Produkten, aber auch von den in diesen Produkten verwendeten Standards unterstützt. Die Wahl eines bestimmten Produkts als Basis für den Zertifizierungsdienst limitiert hier möglicherweise Anwendungsbereiche. Nach dem Aufsetzen der Echtzeitplattform wird hier nochmals das Virtual Private Network (VPN) auf Basis von IPsec sowie die Sicherung von Webzugriffen über SSL inklusive des Schutzes von E-Mail durch Verschlüsselung und digitale Signatur betrachtet. Dabei ist eine gewisse Breite der Betrachtungsweise unabdingbar, da zum erfolgreichen Aufbau einer PKI eine insgesamt tragfähige Basis geschaffen werden muss. Die mangelnde Unterstützung eines einzigen wesentlichen Aspekts kann zur Folge haben, dass eine PKI in der Praxis versagt oder Verschlüsselungsverfahren von Benutzern nicht akzeptiert werden.

8.1.1 Zertifikate

Zertifikate werden heute nach dem Standard X.509 vergeben, welcher in der Version 3 vorliegt. Durch diesen Zertifikatstandard ist man in der Lage, Zertifizierungsstellen sowie Zertifikatinhaber zweifelsfrei zu identifizieren. Außerdem können unterschiedliche Bindungsmerkmale festgelegt werden. Dabei muss die

Beurteilung der Gültigkeit und Anwendbarkeit eines Zertifikats in jedem Fall gegeben sein. Neben den standardisierten Feldern sind ebenfalls Bereiche für Erweiterungen vorgesehen, die im schlimmsten Fall zu inkompatiblen Lösungen führen können. Folgende Erweiterungen lassen sich unterscheiden:

- **Nicht kritische Erweiterungen:** Diese geben zusätzliche Informationen, um z.B. den Zertifikateinsatz zu optimieren. Dazu gehören z.B. weitere Informationen zur Identifikation des Zertifikatinhabers. Implementierungen, die diese Erweiterungen nicht benötigen oder verarbeiten können, dürfen sie ignorieren.
- **Kritische Erweiterungen:** Eine Implementierung, welche ein Zertifikat verwenden will, muss alle darin definierten kritischen Erweiterungen berücksichtigen. Falls sie dazu nicht in der Lage ist, muss die Nutzung des Zertifikats nach dem Standard abgelehnt werden.

Dadurch ist der Zertifikataussteller in der Lage, unterschiedliche Steuerungsmöglichkeiten zu nutzen, die Einfluss auf die Verwendung der von ihm ausgestellten Zertifikate nehmen kann. Auf der anderen Seite ist hier eine mögliche Quelle für Inkompatibilitäten zwischen unterschiedlichen X.509-Implementierungen gegeben. [X.509(00)]

Demgegenüber stehen die PGP[1]-Schlüssel, welche oftmals einfacher aufgebaut sind. Hier wird kein Verwendungszweck angegeben, weshalb das RSA-Format eingesetzt werden kann, um eine digitale Signatur oder Verschlüsselung zu unterstützen. Beispielanwendungen sind die Dateisicherung, sichere E-Mail oder VPN. Dabei kann nur die Gültigkeitsdauer eingestellt werden. Neuere Schlüsselpaare, die von PGP ab Version 5[2] bzw. Version 7[3] zusätzlich unterstützt werden, enthalten dagegen mehrere öffentliche Schlüssel bzw. Schlüsselpaare. Der Basisschlüssel dient dabei ausschließlich zur digitalen Signatur. Daneben können ein oder mehrere Schlüssel auch mit unterschiedlichen Gültigkeitsdauern eingesetzt werden. [CDFG98]

PGP-Schlüssel können ebenfalls als Zertifikat betrachtet werden. Es ist möglich, durch eine digitale Signatur eines PGP-Schlüssels dessen Gültigkeit in nachprüfbarer Form zu bestätigen, wobei der signierende Schlüssel die Rolle des Schlüssels der Zertifizierungsstelle übernimmt. Im Gegensatz zu X.509 ist allerdings die Anzahl der signierenden Schlüssel nicht auf einen Schlüssel begrenzt. Deshalb ist es durch PGP möglich, eine Vertrauensstruktur aufzubauen, die nicht wie bei X.509 auf eine Baumstruktur beschränkt ist. Zusätzlich lassen sich mehrere Namen für den Schlüsselinhaber angeben, wobei die Gültigkeit des Namens durch Signaturen bestätigt werden kann. Dadurch wird es einer Person ermöglicht, unterschiedliche E-Mail-Adressen mit dem gleichen

1 Pretty Good Privacy
2 Diffie-Hellman, Digital Signature Standard (DSS)
3 Neues RSA-Format

Schlüssel zu benutzen. Die Adressen können zusätzlich unterschiedlichen Vertrauenswürdigkeiten zugeordnet werden. Jeder PGP-Schlüssel signiert sich selbst, wodurch verhindert wird, dass die Schüsseleigenschaften durch Dritte verfälscht werden können.

Während weiterhin X.509 davon ausgeht, dass Zertifikate zentral von Trust Centern erzeugt werden, favorisiert PGP ein dezentrales Modell. Dabei bezieht PGP aber auch auf Wunsch Zertifizierungshierarchien mit ein. Die unterschiedlichen Ansätze zur Identifikation von Zertifikaten bedingen unterschiedliche Möglichkeiten:

- **X.509:** Bei der Erzeugung eines X.509-Zertifikats vergibt die erzeugende Zertifizierungsstelle eine eindeutige Seriennummer, sodass jedes Zertifikat unter Rückbezug eindeutig identifizierbar ist. Diese Identifikation kann jedoch erschwert werden, wenn das Zertifikat des Trust Centers nicht verfügbar ist.
- **PGP:** Die dezentrale Erzeugung von PGP-Schlüsseln durch die jeweiligen Schlüsselinhaber macht die Vergabe von Seriennummern zur Identifikation unbrauchbar. PGP bildet deshalb eine kryptographische Prüfsumme über den Schlüssel und verwendet einen Teil dieser Prüfsumme[4], zusammen mit dem Erzeugungsdatum, zur Identifikation von Schlüsseln. Dadurch lässt sich zwar keine Eindeutigkeit garantieren, doch die Wahrscheinlichkeit, dass zwei unterschiedliche, am gleichen Tag erzeugte Schlüssel über dieselbe Key-ID verfügen, ist sehr gering.

Neben der Identifikation von Zertifikaten ist die Identifikation des Zertifikatinhabers entscheidend. Während PGP für jeden Identifikator nur ein Feld für den Namen und ein weiteres Feld für die E-Mail-Adresse vorsieht, ohne dabei den Inhalt zu prüfen, sieht X.509 strukturierte Namensfelder vor. Allerdings werden die Namensfelder innerhalb derselben Implementierung zum Teil unterschiedlich gefüllt und interpretiert. Dadurch kann es bei beiden Zertifikaten passieren, dass das Auffinden eines Zertifikats zu einem bestimmten Benutzer bei unterschiedlichen Implementierungen nicht möglich ist.

Die Nutzung von Zertifikaten über die Implementierungsgrenzen hinweg setzt die Möglichkeit eines Austauschs voraus, wofür die geeignete Import- und Exportformate benötigt werden. Für PGP ist eine weitgehende Rückwärtskompatibilität gegeben, da es eine Referenzimplementierung mit einem Austauschformat gibt, das zu allen früheren Versionen ab 2.6.3 kompatibel ist. X.509-Zertifikate sind komplexer zu handhaben, da hier eine Vielzahl unterschiedlicher Formate definiert wurden, die von den einzelnen Implementierungen zum Teil in höchst unterschiedlichem Umfang unterstützt werden. Es lassen sich dabei unter anderem die folgenden Formate nennen:

4 Die so genannte Key-ID

- **PKCS#12:** privater Schlüsselaustausch, welcher auch den Private Key enthält.
- **X.509:** enthält nur das Zertifikat in den Kodierungen.
- **PKCS#7:** kann Zertifizierungspfad enthalten.
- **PEM:** Format der Privacy Enhanced Mail

Ein Austausch der Zertifikate kann dabei nur erfolgen, wenn das gleiche Austauschformat verwendet wird. Momentan existieren sehr eingeschränkte Möglichkeiten der Interoperabilität zwischen PGP und X.509. PGP kann Zertifikate im PKCS#12-Format lesen, wobei dazu immer der Private Key in PGP importiert werden muss. Soll dieser entfernt werden, so muss dies in PGP durchgeführt werden. Ein Export von PGP-Schlüsseln in eine X.509-Hierarchie ist im Gegensatz dazu nicht möglich. Allerdings kann aus PGP heraus ein X.509-Zertifikat zu einem PGP-Schlüssel bei verschiedenen Trust Centern angefordert werden. Aufgrund der Mehrfachsignierung stellt PGP den offeneren Standard dar, der auch in Zertifizierungshierarchien eingesetzt werden kann. Allerdings muss dann darauf geachtet werden, dass das Signieren fremder Schlüssel wirklich nur nach einer eingehenden Prüfung erfolgt, da sonst der Verlust der Vertrauenswürdigkeit droht.

8.1.2 VPN auf Basis von IPsec

Auf die bestehenden Schwächen des Protokolls IPsec wurde bereits eingegangen. Hier soll es nur kurz um mögliche Implementierungsschwächen gehen. Eine Authentifizierung bei IPsec kann bei manchen Implementierungen[5] durch X.509-Zertifikate erfolgen. Der Private Key kann dabei wahlweise in Dateien oder auf einer Chipkarte gespeichert werden. Bei IPsec spielt für die Interoperabilität die Implementierung des Key Management, die verwendeten Verschlüsselungsarten und die Implementierungsart eine entscheidende Rolle. Bei Einbindung von Zertifikaten kommen noch die erwähnten Probleme des letzten Abschnitts hinzu. Aus diesem Grund bestehen IPsec-Produkte heute meistens aus geschlossenen Lösungen.

Eine herstellerneutrale Lösung bietet unter anderem PGPnet. Hier wird eine wechselseitige Authentisierung der Rechner über gemeinsame Passwörter, PGP-Schlüssel oder X.509-Zertifikat ermöglicht. Dabei wird PGPnet innerhalb des IP-Protokollstapels eingesetzt, wodurch alle IP-Verbindungen zu ausgewählten Rechnern oder Teilnetzen geschützt werden. Wenn ein Rechner oder Teilnetz als sicher eingetragen wird, versucht PGPnet mittels IPsec eine gesicherte Verbindung aufzubauen. Der Verbindungsaufbau und die Nutzung der Verbindung geschehen für den Benutzer und die Software auf dem Rechner transparent. Falls der betreffende Rechner jedoch nicht korrekt antwortet,

5 Safeguard VPN und Safeguard PKI von Ultimaco

kommt keine Verbindung zustande. Die zum Schutz der Verbindung verwendeten Algorithmen und Schlüssellängen werden zwischen den beteiligten Rechnern automatisch unter Verwendung des Protokolls IKE[6] ausgehandelt.

PGPnet ist auch in so genannten Mischlösungen mit anderen Produkten, unabhängig von Betriebssystemen einsetzbar. Bei Router-basierten Produkten ist eine Interoperabilität nicht ohne weiteres hinzubekommen, da die IPsec-Implementierung unterschiedlich realisiert wurden. Hinzu kommen die noch bestehenden Lücken des Standards, die zuerst beseitigt werden müssen.

8.1.3 Key Management

Die Internet Engineering Task Force (IETF) hat ein Position Statement zur IKE-Entwicklung herausgegeben. Gegenstand des Papiers sind ernsthafte Sicherheitsrisiken des IKE-Protokolls aufgrund von dessen Komplexität. Wegen der aufgetretenen Probleme haben die Security-Area-Direktoren in der IETF nach Beratung mit den Fachleuten der IESG und der IAB beschlossen, die Weiterentwicklung von IKE erst einmal zu stoppen. Zielsetzung ist, eine sicherere, einfachere und robustere IKE-Version.

Grund für die Komplexität sind die Erweiterungen für ISAKMP/IKE, die auf unterschiedliche Art und Weise durchgeführt wurden. Vorschläge wie unter anderem IKE-CFG, XAUTH, Hybrid-AUTH und CRACK verringern nicht die Komplexität, da sie auf IKE als Ganzes verweisen und nicht einzelne Spezifikationen verbessern. Deshalb soll erst einmal der Schwerpunkt auf die Verbesserung bestehender Protokollschwächen gelegt werden, anstatt Erweiterungen zu spezifizieren. Die IETF sieht es heute als Problem an, wenn zu viel vom IKE-Code in neuen Protokollen wie PIC und GDOI verwendet wird, da eventuell implementierte Schwächen vorhanden sind.

Die IETF schlägt vor, dass Entwickler keine gemeinsamen Bibliotheksfunktionen nutzen, die Funktionen verwenden, die sich im Fehlerfall potenzieren würden. Versuche, die Stati zu teilen oder den Nachrichtentausch zu optimieren, führen sonst wahrscheinlich zu größeren Sicherheitslücken. Die Security-Area-Direktoren haben die IPsec-Arbeitsgruppe darum gebeten, nach einem Ersatz für IKE zu suchen. Diese Arbeit ist bereits auf den Weg gebracht worden und ist bekannt unter dem Namen „Son of IKE“. Diese Entwicklung hat zu heftigen Auseinandersetzungen innerhalb der IP Society geführt. Trotzdem sind sich alle einig, wenn IKE verwundbar ist, müssen alle an Verbesserungen bzw. Erweiterungen gemeinsam arbeiten. Die IPsec-Gemeinschaft muss sich auch überlegen wie, man mit Gruppen zusammen arbeitet, die ähnliche Ziele verfolgen, um das Internet sicher zu machen. Diese Anstrengungen müssen gebündelt werden und in einem gemeinsamen Schlüsselvereinbarungsprotokoll für IPsec münden, um IKE zu ersetzen oder zu vereinfachen.

6 Internet Key Exchange

8.1.4 Sicherung von Webzugriffen durch SSL

Bei der Sicherung von Webzugriffen hat sich SSL[7] als Standard durchgesetzt und wird von allen gängigen Browsern sowie Webservern unterstützt. Die Authentisierung erfolgt dabei über das X.509-Zertifikat. Dieses wird in der gleichen Infrastruktur verwaltet und verfügbar gemacht und kann auch für den Schutz von E-Mails und VPNs verwendet werden. Diese Anwendung ist technisch relativ unproblematisch. Allerdings führt die Tatsache, dass alle kryptographischen Operationen für den Benutzer verborgen ablaufen, zu einer mangelnden Transparenz der Sicherungsvorgänge.

Die Verwendung von SSL wird in der Regel vom Webserver aus gesteuert. SSL wird für einzelne Webseiten gefordert oder optional vom Server angeboten. Die entsprechenden Einstellungen können durch geeignete Konfigurierung des Webservers und der angebotenen Webstruktur umgesetzt werden. Auf der Seite des Clients sind keine besonderen Aktionen nötig, um SSL zu nutzen, da die heutigen Browser automatisch auf SSL umschalten, wenn dies vom Server gefordert wird.

Auch ist der Schutz auf diejenigen Datentransfers limitiert, die diesen Schutz aus der Sicht des Servers erfordern. Ein Zugriff auf nicht geschützte Seiten oder auf andere, ungeschützte Webserver ist möglich, ohne dass dazu ein Eingreifen des Benutzers notwendig ist. Dieser gezielte Schutz einzelner Datentransfers hat allerdings zur Folge, dass sowohl Client als auch Server potenziell Angriffen über andere, ungeschützte Verbindungen ausgesetzt sind, wobei diese Angriffe sowohl über HTTP als auch über andere Protokolle des IP-Stacks erfolgen können. Während Server in der Regel durch Firewalls geschützt sind, kann die Bedrohung der Clients für bestimmte Anwendungen ein untragbares Risiko beinhalten.

8.1.5 Verschlüsselung und digitale Signatur

Bei Einsatz von software-basierten Produkten zur sicheren Kommunikation mit vertrauenswürdigen Schlüsseln muss eine hohe Sorgfalt bei der Softwareinstallation und der Erzeugung und Verteilung der Schlüssel gelegt werden. Der Einsatz einer zentralen Stelle zur Überprüfung und Signatur öffentlicher Schlüssel erlaubt jedoch auch den Aufbau eines Verschlüsselungssystems hoher Güte, welches bereits mit Softwarelösungen wie PGP ermöglicht werden kann. Beispielsweise kann die Bereitstellung einer vorkonfigurierten Client-Software auf einem oder mehreren Servern zum Abruf durch die Benutzer und durch Festlegung von Regeln[8] erzwungen werden, sodass nur Schlüssel verwendet werden, die eindeutig ihren Eigentümern zugeordnet werden können. Dadurch lassen sich sicherheitstechnische Schwachstellen einer zentralen Schlüsselgenerierung vermeiden.

---

7 Secure Socket Layer

8 Welche Bedingung muss erfüllt sein, damit ein Schlüssel als gültig akzeptiert wird?

Nachdem ein Benutzer sein Schlüsselpaar erzeugt hat, meldet er die Schlüsselerzeugung unter Angabe des so genannten Fingerprints[9] an die Schlüsselzentrale, die den Schlüsselserver betreibt. Diese Meldung geschieht auf dem Papierweg, um jede Manipulation der elektronischen Vorgänge auszuschließen und später eine rechtlich verwertbare Dokumentation der erzeugten Schlüssel zu haben. Zur Verstärkung der Kontrolle kann zusätzlich ein Formblatt vorgesehen werden, welches von der jeweiligen Fachabteilung gegengezeichnet werden muss. Parallel dazu wird der öffentliche Schlüssel in elektronischer Form an die Schlüsselzentrale gesendet.

Die Schlüsselzentrale vergleicht den Fingerprint des erhaltenen Schlüssels mit der Angabe auf dem Formblatt. Wenn beides übereinstimmt, ist der Schlüssel authentisch und kann von der Schlüsselzentrale mit einem unternehmensweiten Schlüssel digital signiert werden, sodass seine Authentizität für die anderen Teilnehmer des Verschlüsselungssystems nachprüfbar ist. Der signierte Schlüssel wird zum Abruf durch andere Teilnehmer in den Schlüsselserver eingetragen und optional an den Benutzer als E-Mail zurückgesandt, sodass er den signierten Schlüssel auch in seine eigenen Schlüsselring-Datei eintragen kann. Der Benutzer kann die Echtheit der Signatur durch den Fingerprint des unternehmensweiten Schlüssels überprüfen. Dieser Fingerprint muss dafür allgemein bekannt gemacht werden.

Bei der Kommunikation mittels E-Mail kommen hingegen nur Lösungen in Betracht, die weitestgehend eine Kompatibilität mit herkömmlichen Systemen besitzen. Dabei muss man auch wieder geschlossene und offene Systeme betrachten. Im letzteren Fall muss man ebenfalls wieder auf PGP zurückgreifen, da dieses System weit verbreitet ist und sich leicht in bestehende Clients integrieren lässt. Für geschlossene Strukturen kann auf Verfahren wie X.509 und S/MIME[10] [DHRL+98] zurückgegriffen werden. Von den bisher S/MIME-kompatiblen Produkten lassen sich die meisten heute aber nicht stabil betreiben. Microsoft Office im Zusammenspiel mit Outlook scheidet im Vergleich noch am besten ab. Hier ist es möglich, die Verschlüsselungs- und Signaturfunktionen von Outlook mit S/MIME und das E-Mail-Plug-in von PGP parallel zu nutzen, sodass je nach Kommunikationspartner das geeignete Verfahren manuell ausgewählt werden kann. Andere Produkte zeigen heute jedoch erhebliche funktionale Mängel auf, die eine Austauschbarkeit von Zertifikaten nicht ausreichend stabil ermöglichen. Die Nutzung einer X.509-konformen Zertifizierungshierarchie stellt daher heute ein nicht zu unterschätzendes technisches Risiko dar. Auch ist eine Vereinheitlichung der Strukturen von PGP und X.509 erst in Ansätzen zu erkennen. Aus diesem Grund kann man momentan nur

9 D.h. des Hashwerts über den Schlüssel.

10 Secure/Multipurpose Internet Mail Extensions

empfehlen, ausgiebige Tests hinsichtlich Konformität, Interoperabilität, Nutzung von Schlüsselverzeichnissen und Anwendbarkeit vor Auswahl eines Systems durchzuführen. [WECK01]

8.1.6 Fazit

Eine sichere und geschützte Kommunikation muss die Möglichkeit für einen spontanen Datenaustausch bieten. Diese Zielsetzung lässt sich allerdings nur in Verbindung mit Public-Key-Verfahren erreichen. Doch der effiziente Einsatz von Sicherheitsmechanismen ist nur mit der Kombination von symmetrischen und asymmetrischen Verfahren möglich. Um Public-Key-Verfahren sicher einsetzen zu können, bedarf es einer entsprechenden Infrastruktur. Denn erst durch Zertifikate und Certification Authorities (CA) bzw. Trust Center lässt sich der Einsatz von öffentlichen Schlüsseln sicher gestalten, da sich so die Authentizität der öffentlichen Schlüssel überprüfen lässt. Voraussetzung dabei ist, dass die CA vertrauenswürdig ist. Diese Voraussetzung ist auch für den Fall unablässig, dass die Generierung der Schlüssel von der CA übernommen wird, da die prinzipielle Möglichkeit besteht, dass dort Unbefugte Zugriff auf den privaten Schlüssel erhalten können.

Bei der Auswahl der zu verwendenden Verschlüsselungsverfahren und Hash-Funktionen muss eine Bewertung der Sicherheit der Verfahren erfolgen. Dabei ist für die Verschlüsselungsverfahren die Länge der Schlüssel maßgebend. Denn auch wenn der zugrunde liegende Algorithmus frei von Fehlern ist, sind Verfahren mit einer geringen Schlüssellänge anfällig für so genannte Bruteforce-Attacken. Aus diesem Grund muss beispielsweise für DES nach Alternativen gesucht werden. Weiterhin müssen die Erfolge der Kryptoanalyse ständig verfolgt werden, um festzustellen, ob die verwendeten Verfahren Schwächen gezeigt haben.

Abgesehen von der Güte der einzelnen Verfahren lässt sich erst mit deren Einbindung in ein fehlerfreies Protokoll eine geschützte Kommunikation realisieren. Die im Rahmen dieses Buchs betrachteten Protokolle haben dabei unterschiedliche Ansatzpunkte. Mit SSL und den Sicherheitsmechanismen von IP werden sichere Kanäle geschaffen, über die ungesicherte Daten versendet werden können. S-HTTP dagegen wählt den umgekehrten Ansatz. Mit S-HTTP werden gesicherte Daten über einen unsicheren Kanal gesendet. Beides bietet Vor- und Nachteile. Über einen sicheren Kanal können jegliche Daten gesendet werden. Voraussetzung dabei ist, dass die beteiligten Protokolle direkt oder indirekt auf SSL oder IP inklusive Sicherheitsmechanismen aufsetzen. S-HTTP hingegen arbeitet dokumentenbezogen und ist im Allgemeinen auf HTTP-Nachrichten beschränkt. Im Gegensatz zu SSL und den Sicherheitsmechanismen von IP bietet es aber die Möglichkeit der digitalen Signatur. Dies ist besonders für kommerzielle Anwendungen interessant, da sich so beispielsweise

Unterschriften für Verträge erstellen lassen. SSL und die Sicherheitsmechanismen von IP erfüllen beide annähernd die gleiche Aufgabe, nur auf unterschiedlichen Schichten des OSI-Referenzmodells. Allen Protokollen ist gemeinsam, dass sie in ihrer Konzeption nicht auf spezielle Verschlüsselungsverfahren beschränkt sind. Damit können die Protokolle unabhängig von den verwendeten Verfahren eingesetzt werden.

Auch ohne eine verbreitete Infrastruktur ist eine geschützte Datenkommunikation möglich. Programme wie PGP und SSH bieten zum Beispiel diese Möglichkeit. Ein Senden von Passwörtern im Klartext ließe sich mit einer Umstellung von telnet und ftp auf SSL-fähige Varianten vermeiden. Es genügen selbstgenerierte Zertifikate, da auf eine eindeutige Authentifizierung des öffentlichen Schlüssels des Servers verzichtet werden kann. Die Client-Authentifizierung erfolgt wie immer durch die Kombination von Nutzerkennung und Passwort. Da die SSL-fähigen Applikationen auch den Einsatz ohne SSL bieten, ist ein gleitender Übergang möglich.

Alle behandelten Protokolle sind gegen so genannte Verkehrsanalysen anfällig. Erst eine Verschlüsselung auf der Link-Level-Ebene würde dieses Problem beseitigen. Das wäre jedoch mit einer Umstellung der Hardware und dem Erstellen eines entsprechenden Key-Managements verbunden. In den Verbindungsknoten des Netzes müssen die Daten jedoch zu Routing-Zwecken im Klartext vorliegen. Dies bietet wiederum Möglichkeiten zum Angriff. Für die End-to-End-Verschlüsselung gilt das nicht. Der Einsatz von Verschlüsselungsverfahren führt auf jeden Fall zu Performance-Einbußen, da zusätzliche Verarbeitungsschritte durchgeführt werden müssen. Entscheidend dabei ist aber, in welchem Ausmaß die Performance beeinflusst wird und welche Performance-Einbußen man für eine geschützte Kommunikation in Kauf zu nehmen bereit ist. Die exemplarischen Messungen haben gezeigt, dass die Verzögerungen akzeptabel bleiben, aber dennoch durchaus auf 1/10 der ursprünglichen Performance absacken können.

Besonders für kommerzielle Anwendungen ist die rechtliche Seite der Kryptographie von Bedeutung. Erst wenn die digitale Signatur auch rechtlich abgesichert ist, leistet sie auch wirklich Authentizität und Verbindlichkeit. Dies gilt heute zwar bereits für Deutschland, muss sich aber weltweit einheitlich durchsetzen. Aber auch speziell beim Einsatz von Verschlüsselungsverfahren sind die rechtlichen Aspekte interessant. So sind unterschiedliche Schlüssellängen teilweise in verschiedenen Ländern vorgeschrieben.

Abschließend kann man festhalten, dass die notwendigen Sicherheitsmechanismen für eine sichere, geschützte Kommunikation im Allgemeinen bestehen. Es bedarf nur noch der richtigen Umsetzung und Verbreitung. Dabei ist entscheidend, wie komfortabel und handhabbar die Nutzung der einzelnen Sicherheitsmechanismen ist. Wenn sich die einzelnen Sicherheitsmechanismen

leicht in Anwendungen und in bestehende Systeme integrieren lassen, werden die kryptographischen Verfahren eine weite Verbreitung finden.

8.2 Quality-of-Service (QoS)

8.2.1 Warteschlangenmechanismen

Die heutigen QoS-Mechanismen basieren bei IP hauptsächlich auf unterschiedlichen Warteschlangenverfahren, die unterschiedliche Funktionsweisen und Vor- und Nachteile haben:

1. **Weighted Round Robin (WRR):** Es wird der gesamte für das Queueing vorgesehene Speicherplatz in mehrere Warteschlangen[11] eingeteilt. Dabei wird für jede Queue eine Gewichtung festgelegt. Sind in jeder Queue ausreichend Datenpakete vorhanden, so wird eine Anzahl an Bytes proportional zur Gewichtung aus der Queue verarbeitet. Diese Scheduling-Strategie ist in den Catalysts6xxx auf den Ausgangs-Ports realisiert. Der Proportionalitätsfaktor zur Gewichtung ist dabei gleich 255.
2. **Custom Queueing (CQ):** Läuft ähnlich dem WRR ab. Diese Scheduling-Strategie wird in Cisco-Routern auf der Schicht 3 eingesetzt. Lediglich in zwei Punkten unterscheidet sich CQ vom WRR-Queueing: Es ist eine variable Anzahl an Queues zugelassen[12]. Die Festlegung der Gewichtung wird nicht in der Zahl der vermittelten Bytes, sondern explizit in der Zahl der zu verarbeitenden Pakete festgehalten.
3. **Strikte Priorisierung:** Auch hier kann die maximale Burst Size durch die Konfiguration der Queue Size implizit beeinflusst werden. Bei dem Catalyst6xxx kann sowohl auf einem Input- als auch auf einem Output-Interface eine Queue mit strikter Priorisierung eingerichtet werden. In einem solchen Fall stehen auf dem Output-Interface insgesamt drei Warteschlangen zur Verfügung. Zwei davon werden mit WRR gescheduled und eine strikt gegenüber dieser priorisiert.
4. **Priority Queueing (PQ):** In den Cisco-Routern existiert seit IOS 12.0(x) eine strikte Priorisierung auf der Schicht 3. Dabei können Aggregationen von Datenströmen einer der vier zur Verfügung stehenden Queues zugeordnet werden. Somit könnten bis zu vier unterschiedliche Dienste inklusive Best-effort in einer reinen Schicht-3-Umgebung realisiert werden. Da das Verhalten des PQ bei ausreichend großen Datenraten (ab 10 Mbit/s) im Wesentlichen unabhängig von der Paketgröße ist, können prinzipiell unterschiedliche Anwendungen (z.B IP-Telefonie und Videokonferenzen) über eine Queue laufen gelassen werden.

11 Die Catalyst6xxx-Serie besitzt nur zwei Queues.

12 Im IOS 12.x sind bis zu 16 Queues möglich.

5. **Weighted Fair Queueing (WFQ):** Unterschiedliche Datenströme werden bei gleicher mittlerer Datenrate am Eingang der Warteschlange sehr unterschiedlich behandelt. Um diesen Effekt zu vermeiden, hat Cisco in ihren Routern WFQ implementiert. Dabei können bis zu 255 Datenströme in getrennten Warteschlangen abgearbeitet werden. Dabei wird für jeden Datenstrom[13] eine Gewichtung während der Verarbeitung dynamisch gebildet.
6. **Class Based Weighted Fair Queueing (CB-WFQ): Ab** IOS 12.1(T) hat Cisco das WFQ-Verfahren um CB-WFQ erweitert. Dabei können bis zu 16 Verkehrsklassen definiert werden. Einer Klasse kann eine feste Datenrate zugeordnet werden, was wahlweise im internen WFQ-Algorithmus oder strikt erfolgen kann. Wird sie strikt einer Klasse zugeordnet, so verhält sich diese gegenüber anderen Klassen wie beim PQ. Der Vorteil einer strikten Priorisierung innerhalb des CB-WFQ gegenüber einer PQ liegt darin, dass die Datenströme anderer Klassen weiterhin mit dem WFQ abgearbeitet werden. Die Schätzung der Verzögerung bei einer strikten Priorisierung innerhalb des CB-WFQ ist dieselbe wie beim PQ. Bei einer Zuordnung der Datenrate im Algorithmus kann man keine Abschätzung der Verzögerung machen, da der interne Algorithmus nicht offen ist.

Die Dimensionierung der Queue-Größen beim Einsatz von WRR ist sehr wichtig. Es ist dabei folgende Bedingung zu erfüllen: Hält der priorisierte Datenstrom die vereinbarte Datenrate und Burst Size ein, so darf in der zugehörigen Warteschlange kein Paketverlust auftreten. Hier muss also eine minimale Queue Size bestimmt werden, bei der im ungünstigsten Betriebsfall kein Datenverlust in der priorisierten Queue stattfindet. Dabei ist die Betrachtungen der einzelnen Queues gleichwertig. Da die Summe von beiden Warteschlangen vom System fest vorgegeben ist, kann die Forderung an Verlustfreiheit in der Regel nur für eine Queue erfüllt werden. Da eine Warteschlange für den Best-effort-Verkehr benötigt wird, kann man für die entsprechende Queue keine Annahmen bezüglich maximaler Datenrate und Burstiness machen. Es kann aber eine minimale Dienstgüte für den Best-effort-Verkehr aus dem Wert der zweiten Warteschlange gebildet werden. Werden die Ressourcen von einem Datenstrom nicht vollständig ausgenutzt, so können diese vom anderen Datenstrom benutzt werden. Das Bandwidth Borrowing ist aber nicht gedächtnisbehaftet. Das bedeutet, dass man durch eine nicht volle Nutzung der Ressourcen zum gegenwärtigen Zeitpunkt keinen Anspruch auf eine Überbelegung der Ressourcen in der Zukunft bekommen kann.

13 Ein Datenstrom wird dabei durch die Sende-/Empfangs-IP-Adressen- und Port-Nummernpaare gekennzeichnet.

Die Berechnung der Queue Size wird wie folgt vorgenommen:

$$Q_1[Byte] = W_2[1] \cdot K[Byte] \cdot \frac{r_1[Bit/s]}{r[Bit/s]} + B_1$$

mit Q1 = Queue 1; W2 = Gewichtung von Queue 2; K = fiktive Paketgröße; r = verfügbare Datenrate am Ausgang der Queue; r1 = maximal zulässige (garantierte) Datenrate an der Queue 1

Beim CQ-Verfahren muss man die Paketgröße aller beteiligten Datenströme genau kennen bzw. abschätzen. Gerade bei dem Best-effort-Datenstrom kann man hier lediglich vage Annahmen machen. In den Routern stehen pro Port viel größere Warteschlangen zur Verfügung. Diese kann man explizit auf eine bestimmte Anzahl von IP-Paketen beschränken. Die Berechnung der Queue Size findet hier wie folgt statt:

$$Q_n = \frac{r_n}{r} \cdot \sum_{i \neq n} w_i \cdot b_i$$

mit Qn = N-Warteschlangen; wn = Gewichtung (die Anzahl der, in der Zeitscheibe tatsächlich zu verarbeitenden IP-Pakete); bn = angenommene Paketgröße in der n-ten Queue in Byte

Man kann durch den Einsatz von WRR das Verhalten der Datenströme zwar sehr fein tunen, allerdings ist dies mit erheblichen Aufwand verbunden. Dieser ist bei einer oft veränderlichen Konfiguration und einer großen Anzahl an Queues oft nicht tragbar. Außerdem steigt die Burstiness in jeder Queue stark an, wenn diese mit WRR vorgenommen wird. Falls die Systeme es hergeben, ist es oft wesentlich einfacher Warteschlangen mit strikter Priorisierung zu nehmen. Liegen Datenpakete in einer höher priorisierten Warteschlange zur Verarbeitung vor, so werden diese immer zuerst abgearbeitet. Liegen keine Datenpakete in der höher priorisierten Warteschlange vor, so wird die nächste Queue abgearbeitet. Bei dieser Beschreibung der Funktionsweise wird auch die Gefahr dieses Verfahrens klar: wird mit voller Datenrate über eine höher priorisierte Warteschlange gesendet, so werden die niedriger priorisierten Queues überhaupt nicht mehr abgearbeitet. Man sollte also immer die Datenpakete filtern, bevor sie in eine hoch priorisierte Queue mit strikter Priorisierung gestellt werden. Falls ein Paket in die hoch priorisierte Queue eingelesen und zu diesem Zeitpunkt gerade ein Datenpaket aus der anderen Queue verarbeitet wird, ist zu beachten, dass die aktuelle Verarbeitung nicht unterbrochen wird. Diese Zeit ist vor allem auf langsamen Verbindungen nicht vernachlässigbar.

Die Gewichtung bei WFQ wird hingegen nach folgenden Kriterien vorgenommen:

- Datenströme mit geringerer Datenrate werden höher als die mit einer höheren Datenrate gewichtet.
- Datenströme mit kürzeren Paketen werden höher als die mit langen Rahmen gewichtet.
- In bestimmten Implementierungen kann die IP Precedence in die Berechnung der Gewichtung einbezogen werden.

Die Queues werden dann nach dem WRR-Prinzip abgearbeitet. Dabei wird die jeweils dynamisch berechnete Gewichtung als WRR-Parameter der entsprechenden Queue genommen. [SIEM00]

8.2.2 Congestion Avoidance

Bislang wurde bei allen Betrachtungen der schlechteste Fall angenommen. Diese Betrachtungsweise ist auch gerade bei der Realisierung zeitkritischer Dienste im Internet legitim. Man möchte ähnlich zum Telefonnetz einen laufenden Dienst nicht wegen mangelnder Ressourcen im Netz unterbrechen. Doch der Worst Case tritt im richtig dimensionierten Netz nicht oft und nur kurzzeitig auf. Es ist also naheliegend, im Internet eine gewisse Flusskontrolle zusätzlich zu den aufwendigen Queueing-Mechanismen einzuführen. Würde das gelingen, so könnte in bestimmten Fällen eine effiziente Flusskontrolle ein komplexes Queueing-Verfahren komplett ersetzen. Diese hätte den Vorteil, dass nur in Zeiten der Überlast bestimmte Datenströme durch explizite oder implizite[14] Signalisierung abgebremst wären. Solche Möglichkeiten zur Flusskontrolle sind beispielsweise im X.25-Standard sowie im ATM zu finden. Im Internetprotokoll ist aber ursprünglich kein Mechanismus zur Flusskontrolle vorgesehen worden. Nur durch die Erweiterung von TCP mit zusätzlichen Algorithmen[15] wird die Datenrate einer TCP-Übertragung an die aktuelle Belastungssituation im Netz dynamisch angepasst. Da diese Methoden zur Vermeidung von Verstopfungen im Netz eingesetzt werden, werden sie im IP-Umfeld als Congestion Avoidance bezeichnet. Dies wird aber nur bei TCP vorgenommen. UDP zur Unterstützung von Echtzeitprozessen besitzt keinerlei Mechanismen.

Der große Nachteil der Flusskontrolle bei IP ist, dass sie nur TCP-Datenströme beeinflusst. Das UDP-Protokoll war ursprünglich für kurze Transaktionen gedacht. Somit sind in das Protokoll keine Mechanismen zur Flusskontrolle eingeflossen. Doch heute werden die meisten multimedialen Anwendungen gerade wegen der hohen Effizienz über das UDP-Protokoll übertragen und die Methoden zur Congestion Avoidance im Rahmen von Echtzeitapplikationen spielen eine

14 Z.B. durch Verwerfen eines Datenpakets
15 [STEV97]

untergeordnete Rolle. Das Verfahren Drop Threshold ist beispielsweise ein Congestion-Avoidance-Verfahren und wird von Cisco als ein zusätzliches Queue Management dargestellt. Dabei werden für unterschiedliche Verkehrsklassen innerhalb einer Queue so genannte Drop Thresholds definiert. Eine Drop Threshold von 50% beispielsweise bedeutet, dass Datenpakete bei einem Füllgrad von mehr als 50% verworfen werden.

Es wird auf je zwei Verkehrsklassen, gekennzeichnet durch unterschiedliche Prioritäten im TOS-Feld, eine Drop Threshold angewendet. Ist die Queue zu 20% gefüllt, so werden alle ankommende Pakete, die mit der Priorität 0 bzw.1 gekennzeichnet sind, verworfen. Lediglich Datenpakete mit der Priorität 6 und 7 kommen durch die fast volle Queue durch. Es stellt sich die Frage, wie die Drop Threshold für Echtzeitanwendungen eingestellt werden soll. Möchte man die maximale Verzögerung minimieren, so sollte man den geringsten Drop Threshold für Echtzeitanwendungen vorsehen. Das hat aber die Konsequenz, dass bei einer steigenden Belastung des Netzes alle Datenpakete dieses Stroms trotz verfügbarer Ressourcen verworfen werden. Man stößt somit beim Versuch, Echtzeitanwendungen mit Drop Threshold zu priorisieren, auf einen Widerspruch bei der Konfiguration! Es ist außerdem zu beachten, dass die Funktionsweise dieser Drop Threshold nicht mit der im RFC-2597 empfohlenen Drop Precedences übereinstimmt. Unterschiedliche Drop Precedences werden auf Datenpakete innerhalb einer Queue angewendet, wenn ein Datenpaket wegen mangelnder Ressourcen verworfen werden muss. Die Anwendung von Drop Threshold dagegen bewirkt, dass Datenpakete verworfen werden, wenn noch Ressourcen im Netz verfügbar sind.

8.2.3 Policy Management (PM)

Anfangs sind einzelne Queueing-Strategien dargestellt worden. Dabei war es unerheblich, nach welchen Kriterien die Datenströme in einzelne Klassen eingeteilt worden sind. Es ist aber klar, dass für einen QoS-Dienst die Kriterien zur Einordnung unterschiedlicher Datenströme in einzelne Queues[16] eine sehr wichtige Rolle spielen. Außer der Zuordnung zu einzelnen Queues ist oft zusätzlich eine Datenratenbeschränkung bzw. eine komplette Abweisung bestimmter Datenströme notwendig. Die Gesamtheit dieser Aufgaben wird als Policy Management bezeichnet. Da Switches ohne Schicht-3-Funktionalität keine Informationen des IP-Headers auswerten, wird in diesen lediglich der Schicht-2-Header für Policy-Entscheidungen benutzt. Prinzipiell sind folgende Policy-Kriterien denkbar:

- Source-MAC-Adressen
- Destination-MAC-Adressen

16 In diesem Fall auch als Filtering definiert.

- Inhalt des TOS-Feldes bei IPv4
- Per Port-Policy

Davon ist beispielsweise in der Catalyst6xxx-Serie ohne Schicht-3-Funktionalität folgendes realisiert worden:

1. **Einfache Übernahme des empfangenen TOS-Wertes:** Dies ist nur dann möglich, wenn die Gegenstelle Rahmen im IEEE 802.1D-Format sendet.
2. **Setzen des TOS-Feldes anhand der Destination-MAC-Adresse:** Wird diese Methode angewandt, so kann ein QoS-Queueing nur am Ausgangs-Port stattfinden.
3. **Zuordnung eines festen CoS-Werts zu einem bestimmten Port:** Hierbei werden alle auf dem Port ankommenden Rahmen mit dem entsprechenden CoS-Wert markiert.

Als VoIP-Beispiel wird an dieser Stelle kurz der Fall einer IP-Telefonie-Implementierung betrachtet. Wie bereits im VoIP-Kapitel festgestellt, benötigt ein IP-Telefon eine Datenrate von 19-80 Kbit/s. Um die Verzögerungen in den Warteschlangen zu minimieren, müssen die IP-Telefone an 100-Mbit/s-Ports angeschlossen werden. Es stellt sich dabei die Frage, welche der drei Policy-Strategien, die Cisco anbietet, hier am sinnvollsten angewendet werden kann. Bei Einsatz von IP-Telefonen mit integriertem Switch können die implementierten Policy-Methoden als unbefriedigend bezeichnet werden. Das pauschale Setzen des Ports auf „trusted-COS“ macht den Weg zum Missbrauch frei. Man kann auf einer Workstation bzw. auf einem PC mit aktuellen Netzwerkkartentreibern geschickt alle Rahmen mit der höchsten Priorität markieren. Auch die Priorisierung per Port scheidet aus, sobald man auf einem Port sowohl einen PC als auch ein IP-Telefon anschließt. Für die Priorisierung anhand der Destination-MAC-Adresse gibt es ebenfalls keinen Anwendungsfall. Sinnvoll wäre hier eine Priorisierung anhand der Source-MAC-Adresse, die allerdings bei Cisco nicht implementiert ist.

Zusätzlich muss man sich den Mangel an Flexibilität auf Schicht 2 verdeutlichen. Man kann die Queue-Sizes, Drop-Thresholds und die WRR-Parameter nur global für einen ganzen Switch festlegen. Hat man aber mehrere Trunks auf einem Switch, auf dem der Datenaustausch priorisiert läuft, so kann die allozierte Datenrate und die zugelassene Verzögerung für diese Trunks nicht voneinander entkoppelt werden. Einige zusätzliche Möglichkeiten bieten die Switches der Catalyst6000-Serie mit einer Policy Feature Card (PFC). Es ist hierbei zu beachten, dass Switches mit einer Schicht-3-Funktionalität normalerweise ein VLAN als logisches Interface betrachten. Was speziell Policy Management betrifft, bilden die Switches hier eine Ausnahme. Zusätzlich können Access-Listen sowohl auf ein VLAN als auch auf einzelne Ports angewandt werden.

In den Routern dagegen bietet Cisco vielfältige Möglichkeiten für das PM. Ein Hauptelement dafür sind Access Control Lists (ACLs). Eine ACL kann

mehrere solche Einträge enthalten. Man kann weiter explizit bestimmte Adressen zulassen oder sperren. Weiterhin kommt die IP-Adresse mit den so genannten Wildcard Bits. Diese haben genau die inverse Bedeutung der Netzmaske. Man kann ACL auf bestimmte TCP- bzw. UDP-Portnummern bzw. ganze Portnummern-Bereiche beziehen. Man kann aber nicht in einer Regel Adressen und Portnummern gleichzeitig angeben. Kommt ein Datenpaket an, so wird es der Reihe nach gegen jede Regel in der ACL geprüft, bis eine Entsprechung gefunden wird. Passt eine Regel zum Datenpaket, so wird die Aktion *permit* bzw. *deny* angewandt. Doch interessant für die QoS-Realisierung sind die Access-Listen dadurch, dass anhand dieser dann bestimmte Regeln zur Einordnung in eine bestimmte Queue abgeleitet werden können. Mit einer so genannten Policy Map verbindet man die Access-Listen mit den entsprechenden Warteschlangen. Es können mehrere solche Policy Maps definiert werden. Danach kann man jeweils eine Policy Map einem bestimmten Netz-Interface zuweisen.

Bei Cisco ist das RTP-Queueing explizit eingeführt worden, um RTP-Daten zu priorisieren. Dabei werden alle UDP-Datenpakete, die auf einem ungeraden Port im Bereich von 16 384 bis 32 768 gesendet werden, strikt priorisiert. Diese Ports werden gewählt, weil typische Sprachverbindungen auf diesen Ports abgehandelt werden. Dieses Queueing wird zwar in Verbindung mit CB-WFQ eingesetzt, die Priorisierung wird aber im Gegensatz zum Low Latency Queueing außerhalb des CB-WFQ-Algorithmus abgearbeitet. Das hat die Konsequenz, dass man das RTP-Queueing nicht vorher mit Hilfe einer Access-Liste filtern kann, wodurch sich eine Lücke im Policy-Konzept auftut. Der Einsatz von RTP-Queueing würde also alle über das konfigurierte Interface laufenden Verbindungen, die auf den genannten Ports ablaufen, unkontrolliert priorisieren.

Eine der wichtigsten Aufgaben des Policy Managements ist die Kontrolle der Zuteilung von Datenraten für priorisierte Verbindungen. Die Gewährleistung einer Priorität ist fast immer mit der Einhaltung einer bestimmten vereinbarten Datenrate und einer maximalen Burstiness verbunden. Diese Kontrolle wird durch Traffic Shaping und Committed Access Rate (CAR) ermöglicht. Sowohl das Policing[17] als auch das Shaping wird wiederum mit Hilfe von ACL realisiert. Man wendet also einen Traffic Shaper bzw. CAR auf eine definierte ACL an.

Bei der Traffic-Shaper-Implementierung von Cisco definiert man eine zulässige maximale Datenrate und zusätzlich dazu zwei Werte für die Burst Size. Die Festlegung zweier Werte für die Burst Size ist ähnlich der der Schwellwerte des RED. Wird der erste Schwellwert überschritten, so werden lediglich einzelne Pakete verzögert, wird dagegen der zweite Wert überschritten, so werden alle Pakete verzögert. Sind die beiden Werte für die Burst Size identisch, so verwendet der Traffic Shaper für die Verzögerung den dafür maximal verfügbaren

17 An dieser Stelle das Anwenden eines CAR-Policers

Speicherplatz. Nur wenn der Pufferplatz nicht ausreicht, werden die Datenpakete verworfen. Bei Verwendung von CAR werden wiederum drei Parameter wie bei einem Traffic Shaper angegeben. Dabei wird aber nicht versucht, Pakete zu verzögern. Wird die Datenrate bzw. die Burst Size überschritten, so werden diese anhand der für diesen Fall definierten Regeln weiterverarbeitet. Dabei kann die Regel das Verwerfen der Pakete, eine Re-Markierung oder einfaches Weiterleiten beinhalten. Beim Weiterleiten von konformen Datenpaketen kann zusätzlich zum Weiterleiten eine Markierung stattfinden. Dabei ist auf ein grundsätzliches Problem beim Traffic Shaping und CAR hinzuweisen: Man kann nicht beliebig kleine Werte für die Burst-Größe angeben, da diese über eine Zeitkonstante von 10 ms fest an die zulässige Datenrate gekoppelt ist. Beschränkt man also die Datenrate auf 5 Mbit/s, so kann die Burst Size nicht geringer als 50.000 Bits = 6.250 Byte eingestellt werden. Das kann man aber nur als grobes Shaping betrachten, da die vom Betriebssystem verursachte Burstiness im Bereich einiger Millisekunden liegt.

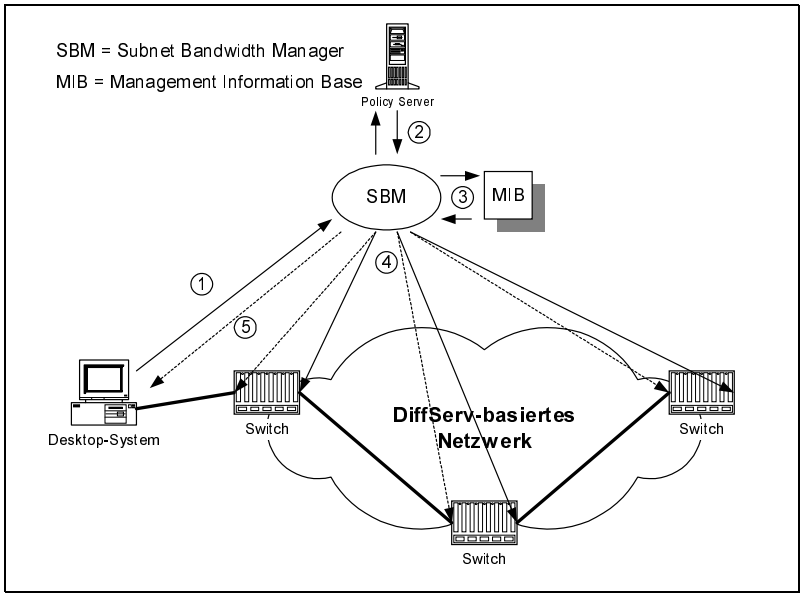


Abb. 8.1 Schematische Darstellung eines DiffServ-Netzes mit Policy-Request

Nutzt man die bisher beschriebenen Möglichkeiten zum Policy Management, so müssen alle Konfigurationsmaßnahmen manuell auf jedem Switch bzw. Router eingetragen werden. Dabei zeichnen sich drei grundsätzliche Probleme ab:

1. Das **Policy Management** sollte an einer einzigen Stelle im Netz stattfinden und danach automatisch auf einzelne Systeme verteilt werden.

2. Die **QoS-Anforderungen** müssen in anwendungsbezogenen Größen stattfinden (Datenrate, Verzögerung etc.). Die einzelnen Queueing-Strategien und deren Parameter sollen für den Administrator im Wesentlichen verdeckt bleiben. Gegebenfalls müssen Schablonen für häufige Anwendungsfälle zu erzeugen sein, die dann auszuwählen und anzuwenden wären.
3. Man benötigt Instanzen, die die Policies personenbezogen speichern und verteilen. Solche Instanzen werden als **Policy Server** bezeichnet.

In Abb. 8.1 ist exemplarisch eine DiffServ-Wolke dargestellt, die eine Policy-Anfrage verarbeiten muss. Folgende Ressourcenanforderungen müssen dabei beachtet werden:

1. Eine Ressourcenanforderung wird an einen Bandwidth Broker bzw. Subnet Bandwidth Manager (SBM) gesendet.
2. Der SBM sendet eine Anfrage an einen Policy Server. Der Policy Server verfügt über alle Benutzerinformationen. Dieser kann prinzipiell komplett losgelöst von einer Netzarchitektur implementiert werden, z.B. durch einen LDAP-Server. Seine Aufgabe ist es, den SBM über aktuelle Berechtigungen bzw. Restriktionen für einzelne Benutzer zu informieren.
3. Ist die Policy-Abfrage positiv abgeschlossen, so wird geprüft, ob im Netz ausreichend Ressourcen zur Verfügung stehen. Der SBM muss also mit einem ständig aktuellen Abbild des Netzes arbeiten.
4. Sind ausreichend Ressourcen vorhanden, so werden Access-Listen angepasst; d.h. zusätzliche IP-Adressen, Port-Nummern usw. werden zugelassen.
5. Optional kann dem Endsystem signalisiert werden, mit welchem CoS- bzw. TOS-Wert die Datenpakete zu markieren sind.

Um das oben beschriebene Policy-Modell zu realisieren, wurde von der IETF das Protokoll Common Open Policy Service (COPS) nach RFC-2748 entwickelt. Dieses arbeitet somit herstellerneutral. Außerdem kann für den Nachrichtenaustausch der Einsatz von RSVP eingesetzt werden.

8.2.4 QoS in den Endgeräten

Das Endsystem ist jeweils das erste und das letzte Glied in der Kette einer QoS-Realisierung. Als letztes Glied muss es in der Regel lediglich Daten korrekt empfangen, doch in einigen Fällen nimmt es an der Flusskontrolle und an der Signalisierung teil[18]. Der Sender kennt meist am besten die Anforderungen an das Netz und kann sich evtl. an die Gegebenheiten das Netzes z.B. durch Veränderung der Kodierung anpassen. Es ist am Endsystem am einfachsten, einzelne Datenströme nach ihrer Wichtigkeit bzw. Priorität zu unterscheiden. Es ist aber auch zumindest ein administrativer Aspekt von besonderer Bedeutung. Zwi-

18 Z.B. kann keine RSVP-Session zustande kommen, wenn der Empfänger keine RSVP-Fähigkeit besitzt.

schen dem Endsystem und der ersten aktiven Komponente im Netz befindet sich eine Grenze der Zuständigkeiten. Es besteht dort in der Regel eine Kunde-Lieferant-Beziehung. Diese Service Level Agreements (SLA) muss das Endsystem einhalten.

Somit stellen sich folgende Anforderungen an die Endsysteme:

1. Endsysteme müssen an einer QoS-Signalisierung teilnehmen, z.B. mit Hilfe von RSVP, COPS oder durch implizite Signalisierung, wie das Pre-Marking der Datenpakete.
2. Ein Endsystem muss unter Umständen die SLA in Form einzelner Übertragungsparameter kennen.
3. Es muss unter Umständen Datenpakete selektieren können. Das beinhaltet z.B. die Entscheidung, welche UDP-Datenströme priorisiert werden. Oftmals stellt sich die Frage auch allgemeiner: welcher Sender einer Multicast-Sitzung wird überhaupt empfangen?
4. Durch das beschriebene Problem mit der Burstiness kann ein System schon allein durch Traffic Shaping bzw. durch geschicktes Packet Scheduling bei einer Konkurrenz um Netzressourcen gewinnen.

Zur RSVP-Signalisierung stehen einige Implementierungen für unterschiedliche Linux-Systeme zur Verfügung. Außerdem besteht die Möglichkeit, über das Setzen bestimmter Socket-Optionen den DSCP-Codepoint unter Windows 2000 auf den gewünschten Wert zu setzen. Diese Möglichkeiten setzen eine Anpassung bestehender Anwendungsprogramme und Neuübersetzung derer voraus. Eine andere interessante Möglichkeit ist das Setzen bestimmter Filter im Netzwerk-Kartentreiber oder im Betriebssystem bzw. seinen Erweiterungen. Anhand dieser Filter werden dann Datenpakete markiert. Es wäre weiterhin eine RSVP-Signalisierung als Betriebssystemerweiterung von Interesse. Da aber diese Signalisierung mit größeren Verzögerungen während des Verbindungsaufbaus verbunden ist und eine gewisse Interaktion mit der initiierenden Anwendung voraussetzt, ist eine solche Realisierung nicht unproblematisch.

Class-of-Service	IP Precedence	IP TOS	IEEE802.1D
Network Critical	7	14	7
Interactive Voice	6	10	6
Interactive Multimedia	5	14	5
Streaming Multimedia	4	4	4
Business Critical	3	6	3
Background	2	1	2
Best-effort	1	1	1
Standard	0	1	0

Tab. 8.1 Zuordnung von unterschiedlichen QoS-Feldern nach 3Com

Dynamic Access ist beispielsweise eine Betriebssystemerweiterung für Windows. Diese ist von 3Com entwickelt worden und funktioniert mit den eigenen Netzwerkkarten. Dynamic Access filtert Anwendungen anhand des Protokolls (TCP/UDP) und der Port-Nummer. Es können für jedes Protokoll bis zu 32 Port-Nummern bzw. Bereiche von Port-Nummern definiert werden, die einer QoS-Klasse zugeordnet werden. Es können Aliase sowohl für die Port-Nummern als auch für die QoS-Klassen vergeben werden. Man kann aber keine IP-Adressen zur Filterung verwenden. Dynamic Access kann Datenpakete sowohl im Ethernet-II-Rahmen[19] als auch im IEEE-802.1D-Rahmen senden. Werden die Daten im IEEE-802.1D-Rahmen gesendet, so werden das CoS-Feld und die DSCP- bzw. IP-Precedence-Bits gesetzt. Wird der Ethernet-II-Rahmen verwendet, so wird lediglich das TOS-Feld bzw. der DSCP gesetzt. Tab. 8.1 zeigt die Zuordnung der Markierung in unterschiedlichen Protokollrahmen zueinander. Das Scheduling auf dem Interface wird in einem abgewandelten 1-Parameter-WRR durchgeführt. Es wird eine Gewichtung G im Wertebereich $1 \leq G \leq 255$ angegeben. Dabei hat G folgende Bedeutung: Es werden immer G-Pakete der hoch priorisierten Warteschlange abgearbeitet, bevor ein Datenpaket der geringer priorisierten Queue abgearbeitet wird.

Dynamic Access von 3Com versetzt das Betriebssystem in die Lage, einzelne Pakete zu markieren. Es ist auch von Vorteil, dass eine primäre Priorisierung schon auf dem Netzinterface und nicht erst im Netz stattfinden kann. Dynamic Access ist heute eine der wenigen Implementierungen von IEEE 802.1D in den Endgeräten. Dieses ermöglicht prinzipiell eine QoS-Realisierung im LAN. Es fehlt aber vor allem ein brauchbarer Paketfilter. Zum einen reicht es nicht, Datenpakete allein anhand von Port-Nummern zu priorisieren. Mindestens eine Angabe der Empfangsadresse ist notwendig. Äußerst problematisch ist auch die Priorisierung bei dynamischen Ports. Da Dynamic Access Pakete lediglich anhand der Sender-Port-Nummer filtert, ist sogar eine Markierung von Datenpaketen mit so genannten Well-Known-Ports nicht möglich[20]; zum Senden werden fast immer dynamische Port-Nummern verwendet. Es ist weiterhin unklar, wie das Scheduling bei Dynamic Access funktioniert, wenn mehr als zwei QoS-Klassen definiert werden. Standardmäßig kann Dynamic Access bis zu 16[21] UDP-Datenströme verarbeiten. Damit ist der Einsatz von Dynamic Access im Server-Bereich eher fraglich. Möglichkeiten zur expliziten QoS-Signalisierung mittels RSVP und/oder COPS stehen zur Zeit nicht zur Verfügung.

19 Kennung 0x800

20 Diese Festlegungen betreffen die Empfänger-Ports.

21 Maximal bis zu 64

8.2.5 Fazit

Mit den geschaffenen Erkenntnissen ist man nun in der Lage, ein QoS-fähiges Netz für Videoübertragung sowie Voice-over-IP (VoIP) zu implementieren. Trotzdem ergaben sich durch die Untersuchungen einige Schwachstellen, die weiterbearbeitet werden müssen:

1. **QoS-Management:** Es sollten Implementierungen der Hersteller bzgl. des QoS-Management getestet werden (z.B. der Cisco Policy Manager).
2. **Bandwidth Broker bzw. Subnet Bandwidth Manager:** Es sind inzwischen einige Spezifikationen zur Realisierung von Bandwidth Broker entstanden. Diese jungen Standards müssen ebenfalls in einer Testumgebung erst einmal evaluiert werden.
3. **Einsatz von COPS bzw. RSVP:** Es ist eine Schnittstelle zwischen dem Endsystem und der Netzmanagement-Instanz[22] zu implementieren, damit die Konfigurationen dynamisch auf Anforderung des Endsystems angepasst werden können.
4. **Policy:** Es ist eine Policy-Architektur bzw. ein Policy-Server zu implementieren und in die QoS-Struktur zu integrieren.
5. **Accounting:** Es sind jegliche Accounting-Aspekte zur Zeit offen. An dieser Stelle fehlen nach wie vor realisierbare Ansätze

Zur praktischen Realisierung der QoS-Ansätze sind folgende Voraussetzungen zu erfüllen:

1. Netzknoten müssen Signalisierungs-, Paket-Scheduling- und Policing-Mechanismen unterstützen. Außerdem müssen funktionsfähige Authentisierungs- und Policing-Mechanismen zur Verfügung stehen.
2. Endgeräte müssen die Signalisierungsmechanismen wie RSVP bzw. Paketmarkierung (Labeling) unterstützen.
3. In den Hosts müssen Traffic-Control-Einheiten realisiert sein und auch vom Administrator steuerbar sein.
4. Es muss Anwendungen geben, die eine Ressourcenreservierung benötigen und diese über definierte Schnittstellen auch realisieren können.

Um die Wechselwirkung zwischen unterschiedlichen Verkehrsströmen zu eliminieren, muss man unterschiedliche Wege beschreiten. Die erste Möglichkeit ist, für bestimmte Dienstarten[23] Ressourcen in jedem Router sowie auf jeder Übertragungsstrecke so zu reservieren, dass bestimmte Dienstgüteparameter eingehalten werden. Das kann beispielsweise dadurch realisiert werden, dass für jeden Datenstrom in jedem Router eine einzelne Warteschlange verwaltet wird. Durch Setzen unterschiedlicher Prioritäten für einzelne Queues kann die angebotene Dienstgüte variiert werden (z.B. mittels IntServ). Man könnte aber auch

22 Bandwidth Broker

23 Im Wesentlichem für multimediale Anwendungen

den gesamten Datenverkehr anhand seiner Anforderungen in einige wenige Dienstgüte-Klassen unterteilen und danach den Routern unterschiedliche Verarbeitungsmuster für die jeweiligen Klassen mitteilen (mittels DiffServ).

Nachdem die QoS-Realisierung verifiziert wurde, kann man dieses Modell auch auf andere Netze ausweiten. Sind bestimmte Erfahrungen mit einer QoS-Realisierung im Campus-Netz vorhanden, so sollten sie zusammengefasst und verallgemeinert werden, damit bei weiteren Konfigurationen bzw. bei der Bildung von Templates für die Managementinstanz auf empirische Werte zurückgegriffen werden kann. Im Rahmen dieser Arbeit sind Performance-Messungen von QoS-Implementierungen in den Endsystemen ausführlich beschrieben worden. Die Randbedingungen ändern sich aber sehr schnell wieder mit jeder neuen Softwareversion. Deshalb müssen implementierte Merkmale auf unterschiedlichen Rechnerplattformen immer wieder neu untersucht werden, wenn man ein QoS-basiertes Netz schaffen möchte. Dies zeichnet ein breites Spektrum an weiteren Untersuchungen und Implementierungen auf.

8.3 Traffic Engineering (TE)

Die ständig zunehmenden Teilnehmerzahlen und die Vielzahl unterschiedlicher Applikationen stellen hohe Anforderungen an die Internettechnologie. Eine wichtige Rolle spielt hierbei die technische Realisierung der WAN-Netze. WAN-Netze auf Basis der ATM- oder der PoS-Technologie bieten Vor-, aber auch Nachteile. Ein ATM-WAN stellt ein Hochgeschwindigkeitsnetz dar, über das in den meisten Fällen IP-Daten übertragen werden. Aufgrund seiner verbindungsorientierten Struktur ermöglicht das ATM die Realisierung eines effektiven Traffic Engineerings. Nachteile dieser Lösung bestehen in der komplexen Integration von IP und ATM. Diese Integration muss über relativ aufwendige Abbildungsmechanismen realisiert werden. Für die Adressenauflösung werden oftmals zusätzliche Server benötigt. Auch eine Realisierung von QoS ist ohne Abbildungsmechanismus nicht möglich. Zusätzlich sind die Fähigkeiten zur Unterstützung von VPNs in ATM-Netzen gegeben. Diese sind für das Bedienen großer Kundenzahlen aber unzureichend.

Die Verwendung der PoS-Technologie im WAN-Bereich stellt eine interessante Alternative zur ATM-Lösung dar. Da PoS ein IP-Netz realisieren kann, entfällt eine aufwendige Integration. Aus diesem Grund weist ein PoS-basiertes WAN eine vergleichsweise geringe Komplexität auf. Ein weiterer Vorteil liegt in der möglichen Realisierung eines IP-QoS über die Modelle IntServ und DiffServ. Ein Nachteil dieser WAN-Lösung liegt in den schlechten Fähigkeiten des Traffic Engineering. Diese beruhen auf der Nutzung von konventionellem IP-Routing. Auch die Realisierung von VPNs ist nur mit hohem Aufwand, wie beispielsweise durch die Einführung von IPsec, durchzuführen.

Das in diesem Buch vorgestellte neue Vermittlungskonzept MPLS kombiniert ATM und IP. Bei dieser Integration stehen dem Anwender die Vorteile beider Protokolle zur Verfügung, was für einen Einsatz von MPLS im WAN-Bereich spricht. MPLS vereinfacht die Integration von ATM und IP in einer Form, wie es Overlay-Lösungen wie Classical IP und LAN-Emulation nicht ermöglichen können. Hierbei verzichtet dieses neue Konzept vollständig auf den Einsatz zusätzlicher Server. Des Weiteren vereinfacht es die Implementierung eines IP-QoS in den ATM-Netzen. Im Vergleich zu reinen IP-Netzen ermöglicht das MPLS das Betreiben des Traffic Engineerings. Hierfür verwendet es das Constraint-based Routing und steigert auf diese Weise die Effizienz eines WAN.

Mit der zunehmenden Internationalisierung und Globalisierung sind Unternehmen mit verschiedenen Standorten auf ein Firmennetz angewiesen, das einen reibungslosen Kommunikationsfluss erlaubt und dabei auch notwendige Sicherheitsaspekte einbezieht. Die effektive Realisierung eines VPN, das diese Anforderungen erfüllt, ist ebenfalls ein maßgeblicher Vorteil von MPLS. MPLS wird in naher Zukunft als eine weitere Lösung für einen Anwendungsbereich zur Verfügung stehen, dessen Realisierung derzeit auf Technologien wie ATM oder PoS basiert.

8.3.1 ATM und MPLS

Durch eine der Hauptanwendungen von MPLS, welche in der Integration von IP und ATM besteht, kann die Effizienz eines WAN auf ATM-Basis erhöht werden. Auf diese Weise erhöht sich auch die Lebensdauer bestehender ATM-Netze. Da durch eine Implementierung von MPLS in IP-Netzen die Vorteile von ATM fast auch ohne konventionelle ATM-Technologie realisiert werden können, bleibt es aber abzuwarten, welche langfristigen Auswirkungen die Entwicklung und Einführung des MPLS auf die ATM-Technologie haben wird. Aus heutiger Sicht nutzt MPLS bestimmte ATM-Eigenschaften aus, um beispielsweise einen Hard-State QoS garantieren zu können. Langfristig könnte MPLS sich aber von ATM vollständig lösen, da aufgrund der expliziten Pfade durch das Netzwerk auch mit IP-QoS-Ansätzen ähnliche Resultate zu schaffen sind. Allerdings ist dies noch nicht in einer größeren Testumgebung nachgewiesen worden.

8.3.2 IPv6 und MPLS

Die Festlegungen, die bisher für MPLS getroffen wurden, fokussieren seinen Einsatz auf den Transport von IPv4-Daten. Dennoch wird auch an der Unterstützung des IPv6 gearbeitet. Diese Unterstützung könnte die nur langsam vorangehende Umstellung auf IPv6 beschleunigen. IPv6 fordert, dass alle Systeme in einem solchen Netz IPv6 sprechen und auch die Applikationen angepasst werden müssen. Dabei sind die Kernrouter eines der Hauptprobleme, die einer

Umstellung auf IPv6 entgegenstehen. Mit Einführung von MPLS wäre das Forwarding in den Kernroutern unabhängig von dem verwendeten Schicht-3-Protokoll, wodurch sich der Umstellungsaufwand von IPv4 auf IPv6 ausschließlich auf die Randgeräte beschränken würde.

8.3.3 IP-over-Optical

Heutige High-Speed-Netze bestehen nahezu vollständig aus Glasfaserkabeln. Die Daten, die zwischen zwei Netzelementen ausgetauscht werden sollen, werden optisch übertragen. Für die Vermittlung im WAN müssen die optischen Signale jedoch in jedem Netzelement auf die elektrische Ebene zurückgeführt werden. Die notwendigen Umwandlungen von elektrisch in optisch und umgekehrt erfordert kostenintensive Hardware und wirkt sich darüber hinaus negativ auf die Übertragungsrate aus. Aus diesem Grund wurde von der IETF die Working Group IP-over-Optical gebildet, welche sich mit den Möglichkeiten befasst, das IP relativ direkt auf das optische Übertragungsmedium aufzusetzen. Die Vermittlung und das Routing könnte hierbei beispielsweise über die Wellenlänge durchgeführt werden. Auf diese Weise würde ein vollständiger Protokollstack eingespart, da die Schicht 2 des OSI-Referenzmodells entfällt.

Während das MPLS vergleichsweise kurz vor einer vollständigen Standardisierung steht, wird die Entwicklung der IP-over-Optical-Technologie noch einige Zeit in Anspruch nehmen. Dennoch könnte in dieser Technologie, wenn auch erst in ferner Zukunft, eine weitere Lösung bestehen, welche die Performance des Internets erneut optimiert. Dabei ist für die Realisierung des IP-over-Optical die Nutzung von MPLS in einer erweiterten Form vorgesehen. [FROM01]

8.3.4 Signalisierung

RSVP-TE ist ein konkurrierendes Signalisierungsprotokoll zu MPLS-LDP und CR-LDP, welches die gleiche Funktionalität bietet. Der größte Unterschied besteht darin, dass RSVP-TE zustandslos arbeitet. Das hat den Vorteil, dass anders als bei LDP-MPLS keine komplexe Zustandsmaschine verwaltet werden muss. Deshalb lässt sich RSVP-TE einfacher implementieren und kann sich in Fehlersituationen durch Re-Routing schnell anpassen. Andererseits ist aufgrund der fehlenden Zustandsinformationen ein periodisches Auffrischen der aufgebauten Pfade und der Reservierung notwendig. Dies zeigt der Skalierbarkeit Grenzen auf. RSVP-TE wird deshalb hauptsächlich in Netztopologien eingesetzt werden, die eine verhältnismäßig kleine Anzahl von Pfaden besitzt. Für VPN-Szenarien, die meistens eine große Anzahl von Pfaden benötigen, erhält in der Regel CR-LDP den Vorzug. RSVP-TE eignet sich sehr gut zur Bandbreitenverwaltung. Da es die LSP fortlaufend erneuert, passt sich die Bandbreitenreservierung dynamisch an die aktuelle Auslastung des Routers an. [SCHR01]

Tab. 8.2 zeigt die verschiedenen Eigenschaften beider Signalisierungsarten auf. Dabei ist zu erkennen, dass die Hauptunterschiede in der Zuverlässigkeit des darunter liegenden Transportprotokolls bestehen und in welcher Richtung die Ressourcenreservierung vorgenommen wird. Von diesem Standpunkt aus betrachtet, gibt es eine Reihe weiterer Unterschiede in der Funktionalität:

- **Verfügbarkeit:** Die Verbreitung des Label Request wird bei RSVP verbindungslos über IP oder UDP durchgeführt. Dabei ist zu beachten, dass eine Anforderung von RSVP ist, dass alle empfangenen IP-Pakete, die RSVP-Nachrichten enthalten, an den RSVP-Protokollcode ohne Referenz zur aktuellen Empfangsadresse in dem Paket weitergeleitet werden. Dieses Merkmal erfordert eventuell eine geringfügige Änderung in der IP-Implementierung. CR-LDP hingegen nutzt UDP, um MPLS-Peers zu erkennen, aber TCP zur verbindungsorientierten Informationsweiterleitung. Allerdings ist TCP nicht in allen IP-Stacks enthalten. Weiterhin können ATM-Switches manchmal den IP-Stack nicht verarbeiten und die Verfügbarkeit und Erreichbarkeit des Transportprotokolls ist ausschlaggebend für die Wahl des LDP.
- **Security:** TCP ist verletzlich gegenüber Denial-of-Service (DoS)-Attacken, wodurch die Performance einer TCP-Sitzung ernstlich leiden kann. Dies kann CR-LDP beeinflussen. Authentifizierung und Policy Control sind hingegen für RSVP spezifiziert worden. Dadurch kann der Urheber einer Nachricht überprüft[24] und unautorisierte Reservierungen können untersagt werden. Ähnliche Merkmale könnten auch bei CR-LDP spezifiziert werden, was aber durch die Verbindungsorientiertheit von TCP als Anforderung entfiel. TCP selbst ist in der Lage, MD5 zu nutzen. IPsec kann ebenfalls im Zusammenhang mit CR-LDP eingesetzt werden. Bei RSVP ist der Einsatz von IPsec nicht möglich, da eine PATH-Meldung zu einem Egress LSR geschickt wird und nicht zum nächstmöglichen LSR, wodurch dieser LSR nicht in der Lage ist, die Informationen der PATH-Nachricht auszuwerten.
- **Skalierbarkeit:** RSVP ist ein Soft-State-Protokoll. Das bedeutet, dass periodisch der Status jedes LSP zwischen den kommunizierenden Randknoten überprüft werden muss. Dadurch ist RSVP in der Lage, Änderungen in den Routing-Baum automatisch zu übertragen. Eine PATH-Nachricht wird 128 Byte betragen, die jeweils um 16 Byte pro Knoten ansteigt, wenn eine explizite Route verwendet wird. Eine RESV-Nachricht wird 100 Byte betragen. Mit 10.000 LSPs auf einer Verbindung und einer Refresh Time von 30 s würden 600 Kbit/s als zusätzliche Datenrate auf dem Link entstehen. CR-LDP benötigt keine periodische Überwachung der Verbindung, da TCP als

24 Z.B. durch Verwendung von MD5

Transportprotokoll für Kontrollinformationen angewandt wird. Dadurch wird erreicht, dass Nachrichten wie Label_Request und Label_Mapping zuverlässig übertragen werden. Zusätzlich entsteht kein weiterer Overhead auf dem Datenpfad, da TCP auf einem anderen Kontrollpfad verwendet wird. Um benachbarten Knoten eine Verbindung zu ermöglichen, sendet CR-LDP Hello-Mitteilungen zur Überwachung, ob die Knoten vorhanden sind, und Keepalive-Nachrichten, um die TCP-Verbindung zu kontrollieren. Diese Nachrichten erscheinen zwar periodisch, sind aber so klein, dass sie keine relevanten Durchsatzverzögerungen hervorrufen.

- **Speicheranforderungen:** Jedes verbindungsorientierte Protokoll muss eine bestimmte Menge sicher zwischenspeichern, bevor der Empfänger erreicht wird. RSVP muss dies ebenfalls beherrschen, da alle Statusinformationen an jedem LSR ankommen müssen, um periodisch die neuesten Mitteilungen zu bekommen. Diese Daten beinhalten Verkehrsparameter, Ressourcenreservierung und explizite Routen. CR-LDP benötigt nur den Ingress und Egress LSR, um dieselben Informationen zu verteilen.
- **Hohe Verfügbarkeit:** Mit dieser Eigenschaft ist die Netzverfügbarkeit gemeint, die im Bereich von 99,999% liegen sollte. RSVP ist dafür ausgelegt worden, im Fehlerfall sofort auf andere redundante Pfade umzuschwenken, da die Verbindung ja periodisch kontrolliert wird. CR-LDP besitzt keinen solchen Mechanismus und ist daher eher einem solchen Fehler ausgeliefert. Es ist ebenfalls besonders schwierig, TCP fehlertolerant zu machen, sodass man bei Abbruch einer Verbindung auf einen redundanten Link ausweichen könnte. Deshalb wird eine Verbindung bei CR-LDP neu aufgebaut.
- **Policy Control:** RSVP ist so spezifiziert, dass PATH- und RESV-Nachrichten in der Lage sind, Policy-Objekte mit undurchsichtigem Inhalt zu versenden. Diese Daten können für verarbeitende Nachrichten zum Ausführen von Policy-basierter Verwaltungssteuerung genutzt werden. Das ermöglicht Labels, RSVP sehr eng an Policy-Protokolle wie COPS zu binden. Auf der anderen Seite bietet CR-LDP nur die Möglichkeit, implizit Policy-Daten in Form einer Empfängeradresse und administrative Ressourcen in den Verkehrsparametern zu transportieren.
- **Layer-3-Protokoll:** Obwohl ein LSP jede Art von Daten transportieren kann, gibt es doch Anlässe, wo das Wissen über das jeweilige Layer-3-Protokoll von dem jeweiligen Ingress oder Egress LSR benötigt wird. So könnte ein LSR nicht in der Lage sein, Pakete weiterzuleiten, weil ein Ressourcenfehler vorliegt. In diesem Fall würde er ein Fehlerpaket zurücksenden, wenn er wüsste, um welches Layer-3-Paket es sich handelt. Bei IP wäre dies ein ICMP-Paket, um dem Sender das Problem mitzuteilen. RSVP identifiziert ein einzelnes Payload-Protokoll während eines LSP-Setup. Für CR-LDP gibt es kein Feld für solche Informationen.

Tab. 8.2 Vergleich beider Signalisierungsansätze [BRFA01]

Eigenschaften	CR-LDP	RSVP-TE
Transport	TCP	IP
Security	Ja[a]	Ja[24]
Multipoint-to-point	Ja	Ja
Multicast	Nein[b]	Nein[25]
LSP Merging	Ja	Ja
LSP State	Hard	Soft
LSP Refresh	Nicht notwendig	Periodisch, hop-by-hop
Hohe Verfügbarkeit	Nein	Ja
Re-Routing	Ja	Ja
Explicit Routing	Strict and loose	Strict and loose
Route Pinning	Ja	Ja, durch aufgezeichneten Pfad
LSP Pre-emption	Ja, prioritätsbasierend	Ja, prioritätsbasierend
LSP Protection	Ja	Ja
Shared Reservations	Nein[c]	Ja
Traffic Parm Exchange	Ja	Ja
Traffic Control	Forward Path	Reverse Path
Policy Control	Implizit	Explizit
Erkennen des Layer-3-Protokolls	Nein	Ja
Resource Class Constraint	Ja	Nein

a. CR-LDP erhält seine gesamte Transportsicherheit durch das TCP-Protokoll; RSVP-TE kann IPsec nicht nutzen, besitzt aber seine eigenen Authentisierungsmechanismen.
b. Multicast ist momentan für kein existierendes Label Distribution Protocol definiert.
c. CR-LDP erlaubt nicht das explizite Teilen der Ressourcen.

CR-LDP und RSVP-TE sind beides gute technische Lösungen, um ein Traffic Engineering (TE) zu ermöglichen. Das beweisen bereits vorhandene Testinstallationen. Die angesprochenen Unterschiede werden aber dazu führen, dass es nie zu einer Interoperabilität zwischen beiden Verfahren kommen kann. Für welches Verfahren zur Verkehrssteuerung man sich letztendlich entscheidet, hängt von den jeweiligen Einsatzgebieten ab, vor allem aber von der Wahl des Herstellers und dessen Unterstützung von Signalisierungsprotokollen.

8.4 Voice-over-IP (VoIP)

Mit VoIP steht eine Technik zur Verfügung, die zur Zeit vor allem im Unternehmensbereich sehr gute Möglichkeiten für die Integration von Audio, Video und Daten in einem IP-basierten Netz bietet. In diesem Bereich lässt sich vor allem eine höhere Produktivität der Mitarbeiter erreichen, die durch eine einheitliche Kommunikationsplattform für Telefonie, E-Mail und Fax schneller Informationen abrufen und verarbeiten können. Durch den Einsatz der IP-Telefonie lassen sich auch Infrastrukturkosten senken, da durch eine gemeinsame Netzstruktur für Sprache und Daten sämtliche Kosten für ein separates Telefonnetz entfallen. In diesem Zusammenhang kann durch eine dynamische IP-Adressenvergabe über DNS/DHCP an die IP-Telefone/Soft-Clients der administrative Aufwand verringert werden. Das bedeutet unter anderem höhere Mobilität und Senkung von Umzugskosten, da der Mitarbeiter unter seiner Rufnummer und seinen eingerichteten persönlichen Profilen ohne aufwendige Umstellungen wie bei einer klassischen TK-Lösung erreichbar bleibt. Mit Hilfe der Sprach- und Datenkonvergenz können Leistungsmerkmale leichter und kostengünstiger implementiert werden, durch die völlig neue Anwendungen etabliert werden können, wie beispielsweise Voice-Mail (Unified Messaging), Virtual Call Center bzw. Web Call Center (eCommerce). Dadurch bieten sich Wettbewerbsvorteile, die eine steigende Kundenzufriedenheit und Kundenbindung zur Folge haben können.

Wie in diesem Buch beschrieben, ist die Verbindungsqualität das Maß für die Akzeptanz der IP-Telefonie. Durch diese Kriterien werden derzeit Grenzen für einen sinnvollen Einsatz der VoIP-Technologie in Netzen ohne QoS gesetzt. In Unternehmensnetzen sind die zulässigen und akzeptablen Werte für Laufzeit (Echo), Paketverlust und Jitter auf Grund administrativer Möglichkeiten leichter zu erreichen. Bei der IP-Telefonie über das öffentliche Internet sind vor allem hohe Anforderungen an die ISPs und an die verschiedenen Netzbetreiber[25] gestellt, die VoIP-Technologie einsetzen. Hier sind vor allem noch Fragen zu einer zentralen Netzmanagement-Plattform und zur unterschiedlichen Lastsituation des Netzes zu klären. Als problematisch kann sich darüber hinaus die Interoperabilität bei dem Einsatz von standardkonformen VoIP-Produkten unterschiedlicher Hersteller erweisen.

8.4.1 Technische Umsetzung

Die technischen Lösungen für VoIP gerade im Telefonbereich sind sehr aufwendig, ohne heute einen höheren Nutzen[26] erfüllen zu können, wie sie die traditionelle Telefonie bietet. Zum Vergleich: In einem heutigen ISDN-Telefon der

25 Netzübergänge bzw. Peering Points
26 Bessere oder einfachere Bedienung

obersten Komfortklasse mit Grafikdisplay findet man typischerweise einen 16-Bit-Microprozessor mit max. 1 Mbyte Flash-EEPROM-Speicher und 128 Kbyte SRAM-Speicher. Diese ISDN-Telefone konnten sich im Laufe der letzten 10 Jahren als Geschäftstelefone durchsetzen, weil die Preise für Flash-EEPROM-Speicher auf etwa 10% ihres ursprünglichen Preises gefallen sind. Außerdem konnte durch den Einsatz von 16-Bit-Mikroprozessoren eine deutlich bessere Bedienoberfläche mit Grafikdisplay bei gleichzeitig komplexeren Funktionen realisiert werden. Dies stellt für den Kunden ein höherwertiges Gerät dar, wofür er auch bereit ist, mehr Geld auszugeben. Außerdem hielt mit den 16-Bit-Mikroprozessorsystemen auch die Programmierung in einer Hochsprache Einzug, was den Aufwand für eine Software-Portierung auf preisoptimierte Prozessoren zusätzlich verringert hat.

Die heutigen VoIP-Telefone treiben allerdings den technischen Aufwand nochmals weiter stark nach oben: Es ist für eine VoIP-Telefon ein Digital Signal Processor (DSP) mit mindestens 80MHz-Taktfrequenz für die komprimierenden Sprachkoder erforderlich. Außerdem ist ein sehr schneller zweiter Prozessor für die IP-Protokoll-Verarbeitung erforderlich, wie beispielsweise ein 32-Bit-RISC-Prozessor mit 100MHz-Taktfrequenz. Der Speicheraufwand beträgt 4 MByte Flash-EEPROM-Speicher und zusätzlich 8 MByte SDRAM-Speicher. Wegen den hohen Taktfrequenzen ist grundsätzlich eine sechslagige Multilayer-Leiterplatte erforderlich. Allein durch dieses Voraussetzungen ist eine deutliche Verteuerung gegenüber einem ISDN-Telefon unumgänglich, welches lediglich mit einer zweilagigen Leiterplatte auskommt. Den Ausweg aus diesen hohen Kosten für ein VoIP-Telefon ist in einer PC-gestützten Telefonlösung[27] zu suchen, die allerdings heute aufgrund unterschiedlicher Sprachqualitäten der Soundkarten bzw. Performance der PCs auch als fragwürdig bezeichnet werden kann.

Was VoIP im WAN angeht, sind sogar noch größere Defizite gegenüber der herkömmlichen Telefonie hinsichtlich der Wirtschaftlichkeit zu verzeichnen. Die angeblich größten Anbieter, deren Bilanzzahlen von VoIP-Telefondiensten von Frost&Sullivan untersucht worden, sind:

- Net2Phone (http://www.net2phone.com)
- ITXC (http://www.itxc.com)
- Delta-Three (http://www.deltathree.com)

Diese Firmen versuchen Privat- aber auch Geschäftskunden mit angeblich preiswerten oder sogar kostenlosen Ferngesprächen über das Internet anzubieten und machen lediglich exorbitante Verluste. Zum Beispiel machte Net2Phone im Jahr 1999/2000 mit VoIP-Telefonie lediglich ca. 72 Mio. Dollar Umsatz und einen Verlust von 128 Mio. Dollar. Bei den anderen Firmen sieht die Bilanz

27 Soundkarte und Headseat o.Ä

grundsätzlich nicht anders aus. Zur Zeit vernichten alle genannten Internet-VoIP-Firmen oder besser Internet Telephony Service Provider (ITSP) das Geld ihrer Aktienanleger in Höhe von jährlich mehreren 100 Millionen Dollar. Hierzu muss man sich lediglich die Bilanzberichte ansehen, die diese Firmen gegenüber der amerikanischen Börsenaufsicht abgeben. Im Internet findet man diese Bilanzberichte und die entsprechenden Aktienkurse unter http://www.nasdaq.com. Die Aktienkurse dieser ITSP-Firmen haben aufgrund der nicht vorhandenen Wirtschaftlichkeit gerade im letzten Jahr einen massiven Wertverlust ausgewiesen. Aus den aus Aktienemissionen und mit Verlusten subventionierten billigen Ferngesprächen für das Telefonieren im Internet auf eine überlegene wirtschaftliche und technische Alternative durch VoIP zu schließen, kann aufgrund der Bilanzdaten dieser Firmen wohl nicht geschlossen werden[28].

Weiterhin wird bei der ITU-T und der amerikanischen Telecommunications Industry Association (TIA) diskutiert, einen nichtkomprimierenden G.711-Codec (64 Kbit/s) als Standard-Codec für VoIP zu empfehlen[29]. Andere zurzeit noch nicht veröffentlichte Gremienpapiere der TIA und von ETSI[30] gehen in dieselbe Richtung. Der Grund hierfür ist, dass komprimierende Codecs, so genannte Non-Waveform-Codecs, die Sprache nicht digitalisieren wie bei einem Waveform-Codec (PCM- oder AD-PCM): Die Sprache wird aufgrund ihrer statistischen Kennwerte bewertet und entsprechend kodiert. Durch diese Art der Kodierung leidet die Qualität des Sprachsignals unter Umständen erheblich. Bei einer Kodierung mit G.723.1 (6,4 Kbit/s) ist beispielsweise schon fast keine Erkennbarkeit des Sprechers mehr möglich. Das heißt, wenn man nicht vorher schon weiß, mit wem man gerade über eine entsprechend kodierte Übertragungsstrecke sprechen wird, ist aufgrund des Klangbilds nicht möglich, eine Person zweifelsfrei zu erkennen[31]. Außerdem wird bei entsprechenden Non-Waveform-Codecs auch das Hintergrundgeräusch mit in die statistische Bewertung aufgenommen. Wird in einem Raum mit Hintergrundgeräuschen über ein Non-Waveform-Codec telefoniert, so ist die Sprachqualität deutlich verschlechtert und die Verständlichkeit unter Umständen erheblich reduziert. Dies gilt besonders bei den hochkomprimierenden Codecs G.723.1 und G.729a.

Zusätzlich wird die Wirkung der Kaskadierung entsprechender Codecs[32] häufig unterschätzt. Als Beispiel sei das mehrfache Transcoding innerhalb eines heterogenen Sprachnetzwerks genannt. Nimmt man beispielsweise an, dass aus

28 Dies ist auch am MIT schon vor Jahren durch ein Thesenpapier belegt worden: „Internet Telephony: Costs, Pricing, and Policy“ von Dr. McKnight, Lee und Brett Leida.

29 Siehe z.B. TIA/EIA-IS-810, "Transmission Requirements for Narrowband Voice over IP and Voice over PCM Digital Wireline Telephones"

30 European Telecommunications Standard Institute

31 Das beweisen auch praktische Erfahrungen mit VoIP.

32 Auch Transcoding genannt

einer VoIP-Insel (z.B einer Filialstelle) über VoIP-Gateways in die Hauptnebenstellen-Anlage eingewählt und von dort aus ein Ferngespräch über das konventionelle Fernsprechnetz (PSTN) weitervermittelt wird. Es soll zusätzlich angenommen werden, dass in einem zukünftigen PSTN durchaus auch ein Netzübergang zu einem ITSP gemacht wird. Dadurch wird bereits ein zweimaliges Transcoding erforderlich[33]. Trifft diese Verbindung auf der Zielseite auf eine weitere VoIP-Insel, die nur über ein VoIP-Gateway am PSTN erreicht werden kann, so muss aus dem ITSP-Netz wieder in das PSTN vermittelt werden. Am Ende dieser Verbindung ist ein dreimaliges Transcoding erforderlich[34]. Dabei leidet natürlich die Sprachqualität erheblich. Werden dann auch noch die Laufzeiten durch eine Weitverkehrsverbindung oder durch die ITSP-Netze mit in die Bewertung hinzugerechnet, wird man höchstwahrscheinlich auf einen MOS-Wert kommen, der unterhalb von 1 liegt. Das heißt, dass beim Einsatz von komprimierenden Codecs grundsätzlich mit äußerster Vorsicht vorgegangen werden muss und dass die Netzstruktur genau bekannt sein muss. Dies ist wohl aber in den seltensten Fällen im Detail und im Voraus bekannt.

Weiterhin wird auch die Wirkung auf die Sprachverständlichkeit durch die Methode der Silence Suppression häufig unterschätzt. Dieser Begriff wird in der Fachliteratur häufiger auch als Voice Activity Detection bzw. VAD bezeichnet. Durch VAD wird durch einen Lautstärkeschwellwert bestimmt, ob ein Signal gesendet werden soll oder nicht. Dies führt in der Praxis dazu, das durch VAD Silben verstümmelt werden. Das heißt, dass beispielsweise der Wortanfang abgeschnitten wird oder dass es zu einem Zerhacken des Sprachflusses kommt. Außerdem ist VAD ebenfalls abhängig vom Hintergrundgeräusch auf der Senderseite und führt zu entsprechend unangenehmen Effekten.

Die ITU-T G.114 besagt, dass eine Laufzeit[35] von 150ms ein akzeptabler Wert für die meisten Anwendungen darstellt, unter der Vorraussetzung, dass entsprechend wirksame Maßnahmen für eine Echo-Kompensation getroffen wurden. Weiter heißt es in dem Standard: „Bei Kommunikationsaufgaben mit einem hohen Anteil an Interaktionen wird bereits bei Laufzeiten unter 150 ms eine Verschlechterung der Kommunikation empfunden." Darüber hinaus besagt die ITU-T G.114, dass Laufzeiten zwischen 150 ms und 400 ms noch als akzeptabel gelten können, beispielsweise für internationale Weitverkehrsverbindungen unter Einsatz eines Satelliten. Laufzeiten von 150 ms bis 400 ms sind somit nur bei seltenen Gesprächsverbindungen als zulässig zu betrachten. Die Qualitätsgrenze sollte man demnach nicht erst bei 400 ms ziehen, sondern bereits bei 150 ms. Nur in Ausnahmen ist mehr als 150 ms akzeptabel. Dies gilt in besonderer Form für VoIP, da durch den Einsatz von stark komprimierenden

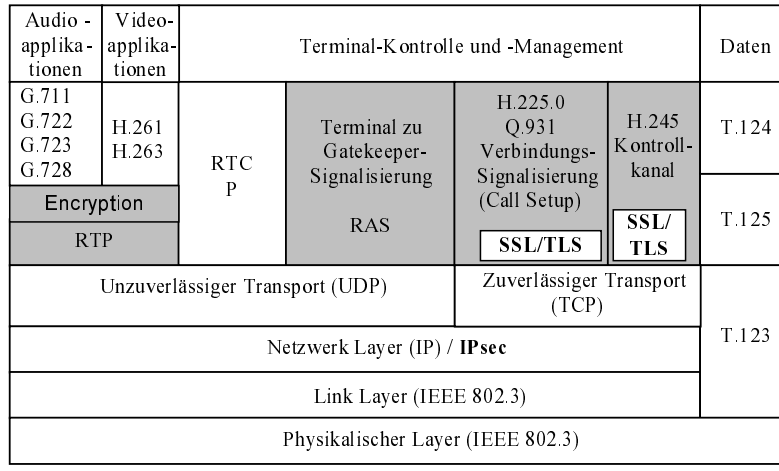
33 G.72x nach G.711 nach G.72x
34 G.72x nach G.711 nach G.72x nach G.711 nach G.72x
35 One-way-delay

Codecs bereits ein erheblicher Anteil an diesem Laufzeit-Budget allein durch den Codec verbraucht wird. Der G.723.1-Codec allein bringt es bereits bei einem Sprachrahmen je IP-Paket auf eine Laufzeit von ca. 67 ms[36].

8.4.2 Security

VoIP ist zwar innerhalb eines gesicherten Intranets oder Extranets gegen Angriffe gesichert. Das hilft in heutigen Einsatzgebieten weiter, kann aber zukünftig bei Einsatz in Weitverkehrsnetzen zu Problemen führen, wenn Verbindungen völlig unabhängig von der herkömmlichen Telefonieinfrastruktur aufgebaut werden. Zum einen muss sichergestellt werden können, dass einzelne Gespräche verschlüsselt übertragen werden, und zum anderen müssen auch die Carrier so genannte Abhöreinrichtungen anbieten können, die es dem Sicherheitsdienst erlauben, bei kriminellen Absichten einzelne Gespräche zu belauschen. Das ist allerdings bei der verbindungslos arbeitenden VoIP-Technik zur Zeit nicht möglich.

Abb. 8.2 Sicherheitsmaßnahmen bei H.323



Bei H.235 liegen Authentifizierung, Integrität und Vertraulichkeit im Rahmen der H-Serie zugrunde. H.235 beschreibt Implementierungsdetails in Verbindung mit H.323. Sicherheit soll dabei für alle Kommunikationsarten gewährleistet werden, die auf H.3xx basieren. Dies beinhaltet Aspekte des Verbindungsaufbaus, der Verbindungskontrolle und des Datenaustauschs zwischen den beteiligten Endgeräten. Im Wesentlichen werden die Datenfelder im Call Connection Channel[37] und Call Control Channel[38] für die Aushandlung und

36 Speechframe-Coding und Jitter-Buffer für einen IP-Rahmen
37 H.225.0 für H.323
38 H.245

Übertragung sicherheitsrelevanter Parameter erweitert. Die Sicherheitsparameter von H.235 sind optional und können im Protokoll ausgehandelt werden. Alternativ zur Verwendung verschiedener Algorithmen und Hash-Funktionen können andere existierende Sicherheitsprotokolle wie SSL/TLS oder IPsec standardkonform eingebunden werden. Abb. 8.2 zeigt die erweiterten Bereiche des H.323-Protokollstapels, um eine sichere Übertragung von Sprache und Daten umzusetzen. Durch den Einsatz von IPsec, wird hier eine Verschlüsselung der Sprachsignale erreicht.

Bei einer Kommunikationsanfrage gibt es zwei Möglichkeiten, um die Daten zu verschlüsseln. Zum einen kann man mittels Symmetric Encryption-based Authentication und des Diffie-Hellmann-Verfahrens den Schlüsselaustausch sicher über unsichere Verbindungen hinweg vornehmen. Anschließend wird ein Verschlüsselungsalgorithmus bestimmt, der mit diesem Schlüssel verwendet werden soll. Als zusätzliches Element kann die digitale Signatur genommen werden. Die zweite Variante verwendet ein Challenge/Response-Verfahren, für das drei mögliche Implementierungen spezifiziert sind. Voraussetzung für den Einsatz ist, dass beide Endteilnehmer bereits über einen Shared Secret verfügen. Der Gatekeeper wird dabei kontinuierlich in die Kommunikation einbezogen, damit ein unerlaubter Zugriff auf die angebotenen Dienste entfällt.

Nachdem die logischen Kanäle und die notwendigen UDP-Ports aufgebaut wurden, kann die eigentliche Übertragung von Audio und Video stattfinden. Diese können mit dem ausgehandelten Kryptoalgorithmus in Verbindung mit dem im Vorfeld übermittelten Sitzungsschlüssel verschlüsselt werden. Die H.235-Sicherheitsnachrichten werden in ASN.1 kodiert. Der Mechanismus, der zum Schutz des Schlüssels verwendet werden soll, wird wie folgt signalisiert:

- **SecureChannel:** Bei Verwendung eines sicheren H.245-Kanals wird der Schlüssel nicht mehr zusätzlich geschützt, sondern im Klartext übertragen.
- **SharedSecret:** Wurden der geheime Schlüssel und der Kryptoalgorithmus bereits im Vorfeld etabliert, wird das Shared Secret verwendet, um den Sitzungsschlüssel zu verschlüsseln.
- **CertProtectedKey:** Zertifikate können eingesetzt werden, wenn der H.245-Kanal nicht sicher ist, oder als zusätzliche Maßnahme bei einem sicheren H.245-Kanal.

Während einer Sitzung kann von einem Sender oder Empfänger über einen encryption_update_request jederzeit ein neuer Session Key angefordert werden. In diesem Fall wird vom Multipoint Controller (MC) ein neuer Schlüssel generiert und die Mitteilung encryption_update an die übrigen Teilnehmer gesendet. Wird ein neuer Schlüssel zum Schutz der Audio- und Videodaten übertragen, so wird dessen Verwendung im RTP-Header durch Änderung des Payload Types angezeigt.

H.245 besitzt keinerlei Methoden, um Sicherheitsverletzungen zu erkennen. Wenn ein Kommunikationsendpunkt den Verlust des Datenschutzes auf einem Logical Channel bemerkt, sollte dieser auf jeden Fall sofort wieder einen neuen Sitzungsschlüssel beantragen und/oder den betreffenden Kanal schließen. Es liegt nach dem Standard im Ermessen der Multipoint Control Unit (MCU), wie auf Sicherheitsverletzungen reagiert wird.

Alle Grundbausteine, die für eine sichere Kommunikation benötigt werden, enthalten kryptographische Komponenten. Für diese Komponenten werden geheime Schlüssel benötigt, die sicher und vor nicht autorisiertem Zugriff geschützt aufbewahrt werden müssen. Hierfür bieten sich Prozessorchipkarten an, so genannte SmartCards. Dabei geht der Funktionsumfang weit über das bloße Aufbewahren von Schlüsseln hinaus. Eine Prozessorchipkarte ist in der Lage, kryptographische Berechnungen selbst durchzuführen. Weiterhin kann sie einen Benutzer identifizieren und digitale Signaturen und Zufallszahlen erzeugen. Daher sollte eine SmartCard ein fester Bestandteil einer Sicherheitslösung sein. [GORE99]

H.323 bietet an, neue Konferenzen einzuberufen. Bei den so genannten Ad-hoc-Konferenzen wird ein weiterer Teilnehmer spontan zugeschaltet. In diesem Fall muss man durch eine Authentifizierung für die Identität aller Konferenzteilnehmer sorgen. Das Konferenzmanagement wird von einem Key Server Module (KSM) übernommen, welches Bestandteil des Gatekeepers wird und für die Erzeugung und Verteilung von Schlüsseln zuständig ist. Dem Gatekeeper wird der Wunsch mitgeteilt, dass eine weitere Person an der Konferenz teilnehmen möchte. Das KSM schickt eine Zufallszahl an den neuen Teilnehmer. Dieser signiert diese Zufallszahl mit seinem privaten Schlüssel auf der SmartCard und verschlüsselt Zufallszahl und Signatur mit einem Shared Secret. Das KSM kann nun prüfen, ob der Teilnehmer über das gemeinsame Geheimnis verfügt. Anhand der Signatur kann die Integrität der Antwort sowie die Authentizität des Absenders festgestellt werden. Ist der Absender authentifiziert, wird ihm der Konferenzschlüssel chiffriert zugesendet. Anschließend ergibt sich der folgende Ablauf:

1. Der Initiator meldet die Konferenz beim Gatekeeper/KSM an, nachdem er sich authentifiziert hat. Das KSM erzeugt einen Konferenzschlüssel, der an den Initiator übermittelt wird.
2. Wenn ein neuer Konferenzteilnehmer hinzukommen soll, muss sich dieser wie unter 1. ebenfalls am KSM authentifizieren. Anschließend wird ihm der Konferenzschlüssel zugesendet. Der Schlüssel wird vor der Übermittlung mit dem öffentlichen Schlüssel des neuen Teilnehmers verschlüsselt. Durch Entschlüsselung mit seinem privatem Schlüssel verfügt der neue Teilnehmer nun ebenfalls über den Konferenzschlüssel.

3. Die eigentliche Kommunikation mittels Daten, Audio und Video wird über symmetrische Verschlüsselungsverfahren abgewickelt. Nach der gesicherten Verteilung des Konferenzschlüssels kann der Informationsfluss zustande kommen:
   - Der Initiator verschlüsselt seine Daten mit dem Konferenzschlüssel
   - Senden von Daten, Audio, Video an alle Teilnehmer über TCP/IP
   - Der/die Empfänger entschlüsseln die Daten mit dem Konferenzschlüssel

Dementsprechend sind Möglichkeiten einer sicheren Kommunikation bei Echtzeitapplikationen auch bereits durch vorhandene Standards vorhanden. Nur werden diese bislang bei Herstellerlösungen nicht eingesetzt.

8.4.3 Zukunft von H.323 und SIP

Die beiden unterschiedlichen Ansätze H.323 und deren Protokolle inklusive H.245 und H.225.0 sowie das Session Initiation Protocol (SIP) sind im Grunde nicht direkt miteinander vergleichbar, da H.323 eine architektonische Beschreibung der ITU und SIP ein Protokoll zur Verbindungssteuerung von der IETF ist. Beide Ansätze verfolgen jedoch vergleichbare Ziele und können daher in bestimmten Bereichen durchaus miteinander verglichen werden. Die meisten VoIP-Lösungen oder Videokonferenzsysteme verwenden heute H.323. Der Trend weist allerdings in Richtung SIP, welches anders als bei H.323 eine klare Zielvorstellung für alle Arten der Kommunikation über IP vorweist. Die Protokolle von H.323 sind deutlich auf ISDN ausgerichtet. Die gesamte Verbindungssteuerung über H.225.0 hat sehr große Ähnlichkeit mit dem D-Kanal-Protokoll.

SIP ist nur das Protokoll zur Verbindungssteuerung; eine Architektur wird nicht definiert. Die derzeit vorgeschlagene Architektur orientiert sich grundlegend an den Spezifikationen von H.323 und den TIPHON-Festlegungen der ETSI, welche durch die Verwendung von Proxy-Servern noch modularer geworden sind. In der ITU-Umgebung werden immer zuerst die Architekturen und Funktionen beschrieben und anschließend die Protokolle definiert. In der ITU sind daher die Protokolle austauschbar, während dies in der IETF die Architekturen betrifft. SIP ist daher für viele Grundarchitekturen anwendbar; so beispielsweise auch für TIPHON.

Die Unterschiede zwischen den Steuerungsprotokollen SIP und H.225.0 von H.323 lassen sich folgendermaßen zusammenfassen:

- Die Implementierungsstruktur ist bei SIP modularer (SIP-Proxy), während H.323 als monolithischer angesehen werden kann.
- Das Grundmodell ist bei SIP das Internet und WWW, während H.323 sich am ISDN-Telefonnetz orientiert.

- Die Kodierung der Nachrichten und Protokollelemente erfolgt bei SIP nach ASCII und bei H.323 binär, Q.931 in ASN.1.
- SIP ähnelt eher dem Protokoll HTTP, während die Steuerungsprotokolle nach H.323 eher Q.931 oder Q.SIG sind.
- SIP ist, gemäß der Tradition des Internets, leicht erweiterbar. H.323 hingegen ist sehr umfangreich und Erweiterungen lassen sich deshalb auch schwer einbringen.
- Die direkte Unterstützung von Mehrpunkt-Signalisierung ist nur bei SIP vorgesehen.

Zusammenfassend lässt sich festhalten, dass der H.323-Ansatz stärker in der klassischen Telefonwelt verankert ist. Dieser Ansatz beschreibt ein heterogenes Modell, in dem beide Netzansätze unterstützt werden, ohne dabei ein möglichst einfaches Konzept zu haben. Genau dieses wird aber durch den SIP-Ansatz definiert. Während H.323 weder auf das Internet noch auf das klassische Fernsprechnetz ideal ausgerichtet ist, stützt sich SIP komplett auf die Internettechnologie. Die HTTP-Eigenschaften ermöglichen in Zukunft eine große Anzahl übergreifender Dienste. H.323 besitzt hingegen keine Schnittstelle zum WWW. Zusätzlich kann aber wiederum SIP ein Interworking zwischen dem Internet und den Telekommunikationsnetzen herstellen. Sogar die Bearbeitung des Signalisierungsprotokolls Nr. 7 ist hierbei eingeschlossen. Das ist auch der Grund, weshalb SIP das bevorzugte Protokoll der 3GPP für die Steuerung von UMTS-Endgeräten geworden ist.

8.4.4 Gateway-Protokolle

H.323 führte unter anderem die Begriffe Gatekeeper, Gateway und Terminal ein, wobei der Gatekeeper die Steuereinheit für die anderen beiden Netzelemente darstellt, die kollektiv als Endpunkte bezeichnet werden. Er führt die Adressumwertung[39], Verbindungssteuerung und Ressourcenverwaltung durch und stellt das Zusammenspiel mit anderen Netzen sicher. Die Gateways sorgen für die Umsetzung der Nutzdatenströme, wie beispielsweise von PCM auf IP. In einer H.323-Zone kann ein zugeordneter Gatekeeper alle Verbindungen einer Multimediakommunikation steuern. Falls sich die Terminals kennen, können sie auch ohne den Gatekeeper eine Verbindung zueinander aufbauen. Zur Übergabe der Signalisierungsinformation in das SS7-Netz der öffentlichen leitungsvermittelnden Netze wird ein spezielles Signalling Gateway eingeführt.

SIP als so genanntes Client-Server-Protokoll beinhaltet zwei grundlegende Nachrichtentypen: Request als Anforderung vom Client an den Server und Response als Antwort des Servers an den Client. Für die Nutzer einer Multimediakommunikation handeln Agenten. So bildet der Media Gateway Controller

39 Von IP zum öffentlichen TK-Netz und umgekehrt

einen Agenten im Sinne des SIP-Protokolls. Dieser fordert vom Proxy-Server[40] den Aufbau einer Verbindung an. Der Proxy-Server stellt beispielsweise unter Zuhilfenahme eines Location Servers fest, wie der Kommunikationspartner zu erreichen ist, und gibt die Aufbauanforderung über ein Request an den entsprechenden Agenten weiter. Auch in SIP können die intelligenten Terminals direkt Verbindung miteinander aufnehmen. In H.323 wurden keine Protokolle definiert, mit denen sich Gatekeeper unterschiedlicher Zonen miteinander verständigen können. Im Gegensatz dazu kann SIP auch zur Kommunikation zwischen mehreren Proxy-Servern eingesetzt werden.

Gehen die Protokolle nach H.323 und SIP von intelligenten Endgeräten aus, so ist die Philosophie der Protokolle H.248 und MGCP die, dass die Endgeräte relativ wenig Funktionalität bieten und die Intelligenz stattdessen im Netz sitzt. Unterscheiden muss man hier zwischen dem Media Gateway Control Protocol (MGCP), welches von der IETF spezifiziert wurde und der Spezifikation der ITU und IETF H.248/MEGACO (Media Gateway Control). Das MGCP wurde als Protokoll zur Steuerung der Media Gateways entwickelt. Es impliziert, dass die Verbindungssteuerlogik im Media Gateway Controller außerhalb des Gateways angesiedelt ist. MGCP geht von einem Netz, bestehend aus Endpunkten und Verbindungen, aus. Über ein einfaches Master-Slave-Verfahren sendet der Media Gateway Controller Kommandos zum Gateway, welches damit angewiesen werden kann, beispielsweise auf Abheben oder Warten auf Wahlinformationen zu warten und diese Informationen zum Media Gateway Controller zu transportieren. Weiterhin wird das Gateway durch den Gateway-Controller aufgefordert, Nutzdaten von Verbindungen zu bestimmten Zieladressen zu senden, um so bei einer aufgebauten Verbindung die Nutzdaten übertragen zu können.

MEGACO ist eine Weiterentwicklung von MGCP. Die ITU hat MEGACO in den Standard H.245 übernommen. Dieses Protokoll lässt aber die weitere Systemarchitektur wie beispielsweise Beziehungen zwischen Media Gateway Controllern untereinander und die Definition der Schnittstellen zum Signalling Gateway offen. Über den Packet-Cable-Standard, welcher in den Vereinigten Staaten für Breitbandkabelnetze definiert wurde, ist hingegen MGCP als Ansteuerungsprotokoll der Gateways für die Telefonanschlüsse festgelegt worden. Aus diesem Grund kann man davon ausgehen, dass sich dieses Protokoll zumindest auf Teilsegmenten etablieren wird. H.248/MEGACO ist als Protokoll für Multimediaendgeräte in den öffentlichen TK-Netzen angetreten und muss sich hier erst einmal gegen SIP durchsetzen.

Eine überaus große Anzahl von Dienstanbietern und Herstellern wird an dem VoIP-Markt partizipieren. Dabei sind die Protokolle der Zielnetzkonfigu-

40 In H.323-Gatekeeper

ration noch nicht einmal eindeutig festgelegt. In gewissen Teilsegmenten scheint die Entwicklung jedoch vorgezeichnet. Bei der Kabeltelefonie wird MGCP für VoIP eingeführt; später wird für Multimediaendgeräte auf SIP umgestiegen. Im Corporate-Bereich haben sich H.323-Lösungen und systemspezifische VoIP-Lösungen durchgesetzt. Auch hier könnte sich langfristig SIP behaupten. Beim Anschluss von Trunk Gateways[41] an das leitungsvermittelnde Festnetz könnte MEGACO eingesetzt werden. Offen ist allerdings, ob SIP sich nicht auch hier gegen seine Konkurrenz durchsetzen kann.

8.4.5 Fazit

Voice-over-IP ist mit Sicherheit ein Thema, welches alle Unternehmen und Hersteller in den letzten Jahren gleichermaßen beschäftigt und in seinen Bann zieht: kann man doch Betriebs- und Personalkosten gleichermaßen einsparen. Dass man neben der Konzeption und Integration von VoIP auch noch das Netzwerk erneuern bzw. mit einem neuen Design ausstatten muss, kommt dabei den Anbietern bzw. Herstellern solcher Lösungen ganz recht! Diese Kosten müssen natürlich den Einsparungspotenzialen entgegengesetzt werden, was oft genug aber nicht passiert oder geschönt dargestellt wird. Hinzu kommt, dass heute oftmals vor der Einführung einer ausgereiften Lösung bereits von allen Seiten darüber diskutiert wird. Ziel ist es, sich einen Markt zu schaffen und erst anschließend mit der Produktentwicklung bzw. Produktion zu starten. So haben vor mehreren Jahren bereits Firmen wie 3Com, Cisco Systems und Siemens über fertige VoIP-Lösungen gesprochen, ohne sie wirklich zu besitzen oder gar anbieten zu können! Noch heute besitzen die vorhandenen Systeme unterschiedliche Leistungsengpässe oder gar Software-Bugs, die konsequent verschwiegen werden. Hier muss man neben einer starken konzeptionellen Phase auch die unterschiedlichen Produkte berücksichtigen, um anschließend eine Kundenlösung zu generieren, die allen Kundenanforderungen gerecht wird – und zwar unabhängig von der Herstellerberatung!

Neben den Implementierungsschwächen gilt es aber auch, die grundsätzlichen Leistungsengpässe des Internets zu beseitigen, die aber meistens von allen Beteiligten ignoriert werden. Laufzeitschwankungen lassen sich bislang im Internet nicht vorhersehbar beherrschen. Aufgrund der chaotischen Struktur sind Überbelastungen einzelner Netzknoten an der Tagesordnung. In diesem Fall würden die Datenpakete verworfen und später noch einmal angefordert werden. Das ist bei Sprachverkehr nicht tragbar. Höhere Bandbreiten lösen zwar kurzfristig dieses Problem, können aber auch nicht als alleinige Lösung Bestand haben. Da das Internet aber auch einem enormen Wachstum unterworfen ist, sind neue Standleitungen und Backbones nach relativ kurzer Zeit

41 Gateway zwischen Paketnetz und leitungsvermitteltem Netz über Verbindungsleitungen mit SS7-Signalisierung

wieder genauso belastet wie zuvor. Aus diesem Grund müssen andere Wege gefunden werden, um das Netz besser ausnutzen und Ressourcen sowie Laufzeiten garantieren zu können. Hierzu gibt es unterschiedliche Ansätze (IntServ, DiffServ, MPLS), die alle eines gemeinsam haben: sie sind bislang nicht verfügbar! Für einen endgültigen Erfolg von VoIP sowie das wirkliche Zusammenwachsen von Telekommunikation und Datennetzen sind Dienstgarantien aber unabdingbar. Erst wenn das Internet die Qualität bietet, die man für das Telefonieren benötigt, wird es eine echte Alternative darstellen. Solange das nicht der Fall ist, wird VoIP auf die Unternehmensnetze beschränkt bleiben.

Anhang

A.1 Literatur

[ABBB+01a] ASHWOOD-SMITH, P.; BANERJEE, A.; BERGER, L.; BERNSTEIN, G.; DRAKE, J.; FAN, Y.; FEDYK, D.; KOMPELLA, K.; MANNIE, E.; LANG, J.P.; RAJAGOPALAN, B.; REKHTER, Y.; SAHA, D.; SHARMA, V.; SWALLOW, G.; TANG, Z.B.: *Generalized MPLS Signaling – CR-LDP Extensions*; Network Working Group; Internet Draft; Expiration Date: January 2002; IETF 2001

[ABBB+01b] ASHWOOD-SMITH, P.; BANERJEE, A.; BERGER, L.; BERNSTEIN, G.; DRAKE, J.; FAN, Y.; KOMPELLA, K.; LANG, J.P.; LIAW, F.; MANNIE, E.; PAN, P.; RAJAGOPALAN, B.; REKHTER, Y.; SAHA, D.; SHARMA, V.; SWALLOW, G.; TANG, Z.B.: *Generalized MPLS Signaling – RSVP-TE Extensions*; Network Working Group; Internet Draft; Expiration Date: January 2002; IETF 2001

[ABSI99] ABOBA, B.; SIMON, D.: *PPP EAP TLS Authentication Protocol*; Network Working Group; Requests for Commments: 2716; Category: Experimental; IETF 1999

[ADFA99] ADAMS, C.; FARELL, S.: *Internet X.509 Public Key Infrastructure – Certificate Management Protocols*; Network Working Group; Request for Comments: 2510; Category: Standards Track; IETF 1999

[ADFF+01] ANDERSSON, L.; DOOLAN, P.; FELDMAN, N.; FREDETTE, A.; THOMAS, A.: *LDP Specification*; Network Working Group; Request for Comments: 3036; Category: Standards Track; IETF 2001

[ATMF95] ATM-Forum specification: *LAN Emulation Over ATM – Version 1.0*; af-lane-0021.000, ATM-Forum 1995

[ATMF97a] ATM-Forum specification: *LAN Emulation Over ATM Version 2.0 – LUNI Specification*; af-lane-0084.000; ATM-Forum 1997

[ATMF97b] ATM-Forum specification: *Multi-Protocol Over ATM Specification v1.0*; af-mpoa-0087.000; ATM-Forum 1997

[ATMF97c] ATM-Forum specification: *Circuit Emulation Service 2.0*; af-vtoa-0078.000; ATM-Forum 1997

[ATMF99a] ATM-Forum specification: *LAN Emulation over ATM Version 2 – LNNI Specification*; af-lane-0112.000; ATM-Forum 1999

[ATMF99b] ATM-Forum specification: *Multi-protocol Over ATM Specification, Version 1.1*; af-mpoa-0114.000; ATM-Forum 1999

[BBCD98] BLAKE, S.; BLACK, D.; CARLSON, M.; DAVIES, E.; WANG, Z.; WEISS, W.: *An Architecture for Differentiated Services;* Network Working Group; Request for Comments: 2475; Category: Informational; IETF 1998

[BCKR98] BATES, T.; CHANDRA, R.; KATZ, D.; REKHTER, Y.: *Multiprotocol Extensions for BGP-4*; Network Working Group; Request for Comments: 2283; Category: Standards Track; IETF 1998

[BEOM97] BERGER, L.; O'MALLEY, T.: *RSVP Extensions for IPSEC Data Flows*; Network Working Group; Request for Comments: 2207; Category: Standards Track; IETF 1997

[BERG98a] BERGER, L.: *RSVP over ATM Implementation Guidelines*; Network Working Group; Request for Comments: 2379; BCP: 24; Category: Best Current Practice; IETF 1998

[BERG98b] BERGER, L.: *RSVP over ATM Implementation Requirements*; Network Working Group; Request for Comments: 2380; Category: Standards Track; IETF 1998

[BERN00] BERNET, Y.: *Format of the RSVP DCLASS Object*; Network Working Group; Request for Comments: 2996; Category: Standards Track; IETF 2000

[BFYB+00] BERNET, Y.; FORD, P.; YAVATKAR, R.; BAKER, F.; ZHANG, L; SPEER, M.; BRADEN, R.; DAVIE, B.; WROCLAWSKI, J.; FELSTAINE, E.: *A Framework for Integrated Services Operation over Diffserv Networks;* Network Working Group; Request for Comments: 2998; Category: Informational; IETF 2000

[BRAD94] BRADEN, R.; CLARK, D.; SHENKER, S.: *Integrated Services in the Internet architecture: an overview*; RFC-1633; Status: Informational; IETF 1994

[BRAD97] BRADEN, R.; ZHANG, L.; BERSON, S.; HERZOG, S.; JAMIN, S.: *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification;* Category: Standard Track; Network Working Group; RFC-2205; Updated by RFC-2750; IETF 1997

[BRAD98] BRADEN, R.; HOFFMANN, D.: *Resource Reservation Protocol API (RAPI);* Open Group Technical Standard; C809 ISBN 1-85912-226-4; The Open Group; Dezember 1998

[BRAD01] BRADEN, R.; ZHANG, L.: *RSVP Cryptographic Authentication – Updated Message Type Value;* Category: Proposed Standard; RFC-3097; Network Working Group; IETF 2001

[BRFA01] BRITTAIN, P.; FARREL, A.: *MPLS Traffic Engineering: A choice of signaling protocols*; Analysis of the similarities and differences between the two primary MPLS label distribution protocols: RSVP and CR-LDP; Data Connection Limited; Enfied 2000

[CBBB+98] CRAWLEY, E. (Editor); BERGER, L.; BERSON, S.; BAKER, F.; BORDEN, M.; KRAWCZYK, J.: *A Framework for Integrated Services and RSVP over ATM*; Network Working Group; Request for Comments: 2382; Category: Informational; IETF 1998

[CCITT87] CCITT, Recommendation X.509: *The Directory-Authentication Framework*; Consultation Committee; International Telephone and Telegraph; International Telecommunication Union; Geneva 1987

[CDFG98] CALLAS, J.; DONNERHACKE, L.; FINNEY, H.; THAYER, R.: *OpenPGP Message Format*; Network Working Group; Request for Comments: 2440; Category: Standards Track; IETF 1998

[CDM01] CONTA, A.; DOOLAN, P.; MALIS, A.: *Use of Label Switching on Frame Relay Networks Specification*; Network Working Group; Request for Comments: 3034; Category: Standards Track; IETF 2001

[DARE00] DAVIE, BRUCE; REKHTER, YAKOV: *MPLS – Technology and Applications*; Morgan Kaufmann 2000

[DBCH00] D. DURHAM, Ed.; BOYLE, J.; COHEN, R.; HERZOG, S.; RAJAN, R.; SASTRY, A.: *The COPS (Common Open Policy Service) Protocol;* Network Working Group; Request for Comments: 2748; Category: Standards Track; IETF 2000

[DEER01] DETKEN, KAI-OLIVER; EREN, EVREN: *Extranet – VPN-Technik zum Aufbau sicherer Unternehmensnetze;* Addision-Wesley; München 2001

[DEHI98] DEERING, S.; HINDEN, R.: *Internet Protocol, Version 6 (IPv6) Specification;* Network Working Group; Request for Comments: 2460; Obsoletes: 1883; Category: Standards Track; IETF 1998

[DETK98] DETKEN, KAI-OLIVER: ATM in TCP/IP-Netzen: Grundlagen und Migration zu High Speed Networks; Hüthig; Heidelberg 1998

[DETK99a] DETKEN, KAI-OLIVER: *Local Area Networks – Grundlagen, Internetworking und Migration;* Hüthig; Heidelberg 1999

[DETK99b] DETKEN, KAI-OLIVER: *ATM-Messungen: Test der Dienstgüte undStreßtests*; ATM-Handbuch; Kapitel 11.500; Hüthig; Heidelberg 1999

[DETK00a] DETKEN, KAI-OLIVER: *Quality-of-Service (QoS) versus Class-of-Services (CoS): Garantierte Dienstgüte in IP- und ATM-Netzen*; ONLINE 2000; Congressband II: Fixed, Mobile & High End Networking; 23. Europäische Kongressmesse für Technische Kommunikation; Düsseldorf 2000

[DETK00b] DETKEN, KAI-OLIVER: *Layer-3-Switching*; Handbuch Telekommunikation – Dienste und Netze wirtschaftlich planen, einsetzen und organisieren, INTEREST, Augsburg 2000

[DETK01] DETKEN, KAI-OLIVER: *Echtzeitplattformen für eBusiness: Forschungsergebnisse aus dem EU-Projekt INTELLECT*; Elektronische Geschäftsprozesse: Grundlagen, Sicherheitsaspekte, Realisierungen, Anwendungen; Herausgeber: Patrick Horster; IT-Verlag für Informationstechnik; Höhenkirchen 2001

[DLRS+01] DAVIE, B.; LAWRENCE, J.; ROSEN, E.; SWALLOW, G.; REKHTER, Y.; DOOLAN, P.: *MPLS using LDP and ATM VC Switching*; Request for Comments 3035; IETF 2001

[DHRL+98] DUSSE, S.; HOFFMAN, P.; RAMSDELL, B.; LUNDBLADE, L.; REPKA, L.: *S/MIME Version 2 Message Specification*; Network Working Group; Request for Comments: 2311; Category: Informational; IETF 1998

[DOHA00] DORASWAMY, N.; HARKINS, D.: *IPsec – der neue Sicherheitsstandard für das Internet, Intranets und virtuelle private Netze*; Addison-Wesley; München 2000

[DOMM98] DOMMEL, CLAUDIUS: *Evaluierung von IP-Telephonie-Standards*; Diplomarbeit an der Technischen Universität Darmstadt; Institut für Datentechnik; Darmstadt 1998

[EPPI93] EPPINGER, B.: *Sprachverarbeitung*; Hanser, München 1993

[ERDE01] EREN, E.; DETKEN, K.-O.: *Mobiles Internet: Planung, Konzeption und Umsetzung mit WAP*; Addision-Wesley; München 2001

[FEHU98] FERGUSON, Paul; HUSTON, GEOFF: *Quality of Service: Delivering QoS on the Internet and in Corporate Networks;* published by John Wiley & Sons, January 1998

[FELLB84] FELLBAUM, K.: *Sprachverarbeitung und Sprachübertragung*; Springer-Verlag; Berlin 1984

[FROM01] FROMMHERZ, DANIEL: *Multi Protocol Label Switching – Technische Betrachtung des MPLS und Bewertung seines Einsatzes im WAN-Bereich*; Diplomarbeit an der Fachhochschule Darmstadt; Darmstadt 2001

[GABO98] GARRETT, M.; BORDEN, M.: *Interoperation of Controlled-Load Service and Guaranteed Service with ATM*; Network Working Group; Request for Comments: 2381; Category: Standards Track; IETF 1998

[GORE99] GORECKI, CHRISTIAN A.: *Sichere Sprachübertragung über das Internet;* Diplomarbeit an der Universität Bremen; Fachbereich Physik/Elektrotechnik; Prof. Laur; Bremen 1999

[GRHE99] GROSSMAN, D.; HEINANEN, J.: *Multiprotocol Encapsulation over ATM Adaptation Layer 5*; Network Working Group; Request for Comments: 2684; Obsoletes: 1483; Category: Standards Track; IETF 1999

[H.225.0(99)] Recommendation H.225.0 (09/99): *Call signalling protocols and media stream packetization for packet-based multimedia communication systems; ITU-T 1999*

[H.245(99)] Recommendation H.245 (05/99): *Control protocol for multimedia communication; ITU-T 1999*

[H.246(98)] Recommendation H.246 (02/98): *Interworking of H-Series multimedia terminals with H-Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN; ITU-T 1998*

[H.323(99)] Recommendation H.323 (09/99): *Packet-based multimedia communications systems; ITU-T 1999*

[H.332(98)] Recommendation H.332 (09/98): *H.323 extended for loosely coupled conferences*; ITU-T 1998

[H.235(98)] Recommendation H.235 (02/98): *Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals; ITU-T 1998*

[H.245(99)] Recommendation H.245 (05/99): *Control protocol for multimedia communication*; ITU-T 1999

[H.450.1(98)] Recommendation H. 450 (02/98): *Generic functional protocol for the support of supplementary services in H.323*; ITU-T 1998

[HACA98] HARKINS, D.; CARREL, D.: *The Internet Key Exchange (IKE)*; Network Working Group; Request for Comments: 2409; Category: Standards Track; IETF 1998

[HAJA98] HANDLEY, M.; JACOBSON, V.: *SDP: Session Description Protocol*; Network Working Group; Request for Comments: 2327; Category: Standards Track; IETF 1998

[HBWW99] HEINANEN, J.; BAKER, F.; WEISS, W.; WROCLAWSKI, J.: *Assured Forwarding PHB Group;* Network Working Group; Request for Comments: 2597; Category: Standards Track; IETF 1999

[HCBO97] HANDLEY, M.; CROWCROFT, J.; BORMANN, C.; OTT, J.: *The Internet Multimedia Conferencing Architecture*; Internet Draft, IETF 1997

[HEIL92] HEILMANN, O.: *Digitale Übertragungstechnik: PCM-Grundlagen und Messverfahren*; Expert; Böblingen 1992

[HELO90] HERTER, E.; LÖRCHER, W.: *Nachrichtentechnik – Übertragung, Vermittlung, Verarbeitung*; Hanser; München, Wien 1990

[HERT94] HERTER, E.: *Nachrichtentechnik: Übertragung, Vermittlung und Verarbeitung*; Hanser; München 1994

[HLFT94a] HANKS, S.; LI, T.; FARINACCI, D; TRAINA, P.: *Generic Routing Encapsulation (GRE)*; Network Working Group; Request for Comments: 1701; Category: Informational; IETF 1994

[HLFT94b] HANKS, S.; LI, T.; FARINACCI, D; TRAINA, P.: *Generic Routing Encapsulation over IPv4 networks*; Network Working Group; Request for Comments: 1702; Category: Informational; IETF 1994

[HFPS99] HOUSLEY, R.; FORD, W.; POLK, W.; SOLO, D.: *Internet X.509 Public Key Infrastructure – Certificate and CRL Profile;* Network Working Group; Request for Comments: 2459; Category: Standards Track; IETF 1999

[HPVT+99] HAMZEH, K.; PALL, G; VERTHEIN, W.; TAARUD, J.; LITTLE, W; ZORN, G.: *Point-to-Point Tunneling Protocol (PPTP)*; Network Working Group; Request for Comments: 2637; Category: Informational; IETF 1999

[HPW00] HANDLEY, M.; PERKINS, C.; WHELAN, E.: *Session Announcement Protocol*; Network Working Group; Request for Comments: 2974; Category: Experimental; IETF 2000

[HSSR99] HANDLEY, M.; SCHULZRINNE, H.; SCHOOLER, E.; ROSENBERG, J.: *SIP: Session Initiation Protocol*; Network Working Group; Request for Comments: 2543; Category: Standards Track; IETF 1999

[IEEE802.1D98] ANSI/IEEE: *Standard 802.1D Part 3, Media Acces Control (MAC) Bridges*; Status: International Standard; IEEE 1998

[JACH97] JACHALSKY, JÖRG: *Untersuchung kryptographischer Verfahren in der TCP/IP-Protokollarchitektur*; Studienarbeit an der Universität Hannover; Fachbereich Rechnernetze und Verteilte Systeme; Prof. Dr.-Ing. Pralle; Hannover 1997

[JNP99] JACOBSON, V.; NICHOLS, K.; PODURI, K.: *An Expedited Forwarding PHB*; Network Working Group; Request for Comments: 2598; Category: Standards Track; IETF 1999

[KAEO98] KAEO, M.: *Sicherheit Netzwerkarchitektur*; Public Document Cisco Systems; 1998

[KBC97] KRAWCZYK, H.; BELLARE, M.; CANETTI, R.: *HMAC: Keyed-Hashing for Message Authentication*; Network Working Group; Request for Comments: 2104; Category: Informational; IETF 1997

[KEAT98a] KENT, S.; ATKINSON, R.: *Security Architecture for the Internet Protocol*; Network Working Group; Request for Comments: 2401; Obsoletes: 1825; Category: Standards Track; IETF 1998

[KEAT98b] KENT, S.; ATKINSON, R.: *IP Encapsulating Security Payload (ESP)*; Network Working Group; Request for Comments: 2406; Obsoletes: 1827; Category: Standards Track; IETF 1998

[KEAT98c] KENT, S.; ATKINSON, R: *IP Authentication Header*; Network Working Group; Request for Comments: 2402; Obsoletes: 1826; Category: Standards Track; IETF 1998

[KENT93] KENT, S.: *Privacy Enhancement for Internet Electronic Mail – Part II: Certificate-Based Key Management*; Network Working Group; RRequest for Comments: 1422; Obsoletes: 1114; IETF 1993

[KNE97] KATSUBE, Y.; NAGAMI, K.; ESAKI, H.: *Toshiba's Router Architecture Extensions for ATM : Overview*; Network Working Group; Request for Comments: 2098; Category: Informational; IETF 1997

[KRAU98] KRAUSE, J.: *Electronic Commerce-Geschäftsfelder der Zukunft heute nutzen*; Hanser; München 1998

[LAHA98] LAUBACH, M.; HALPERN, J.: *Classical IP and ARP over ATM*; Network Working Group; Request for Comments: 2225; Category: Standards Track; Obsoletes: 1626, 1577; IETF 1998

[LESC00] LENNOX, J.; SCHULZRINNE, H.: *Call Processing Language Framework and Requirements*; Network Working Group; Request for Comments: 2824; Category: Informational; IETF 2000

[LKPC+98] LUCIANI, J.; KATZ, D.; PISCITELLO, D.; COLE, B.; DORASWAMY, N.: *NBMA Next Hop Resolution Protocol (NHRP)*; Network Working Group; Request for Comments: 2332; Category: Standards Track; IETF 1998

[MASI99] MALIS, A.; SIMPSON, W.: *PPP over SONET/SDH*; Network Working Group; Request for Comments: 2615; Obsoletes: 1619; Category: Standards Track; IETF 1999

[MCGR92] MC GREGOR, G.: *The PPP Internet Protocol Control Protocol (IPCP)*; Network Working Group; Request for Comments: 1332; Obsoletes: RFC 1172; IETF 1992

[MOCK87] MOCKAPETRIS, P. V.: *Domain Names – Implementation and Specification*; Network Working Group; Request for Comments: 1035; Obsoletes: RFCs 882, 883, 973; IETF 1987

[MODE90] MOGUL, J.; DEERING, S.: *Path MTU Discovery*; Network Working Group; Request for Comments: 1191; Obsoletes: RFC 1063; IETF 1990

[MSST98] MAUGHAN, D.; SCHERTLER, M.; SCHNEIDER, M.; TURNER, J.: *Internet Security Association and Key Management Protocol (ISAKMP)*; Network Working Group; Request for Comments: 2408; Category: Standards Track; IETF 1998

[NBBB98] NICHOLS, K.; BLAKE, S.; BAKER, F.; BLACK, D.: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*; Network Working Group; Request for Comments: 2474; Obsoletes: 1455, 1349; Category: Standards Track; IETF 1998

[NEHH+96a] NEWMAN, P.; EDWARDS, W.L.; HINDEN, R.; HOFFMAN, E.; LIAW, F. CHING; LYON, T.; MINSHALL, G.: *Transmission of Flow Labelled IPv4 on ATM Data Links Ipsilon Version 1.0*; Network Working Group; Request for Comments: 1954; Category: Informational; IETF 1996

[NEHH+96b] NEWMAN, P.; EDWARDS, W.L.; HINDEN, R.; HOFFMAN, E.; LIAW, F. CHING; LYON, T.; MINSHALL, G.: *Ipsilon Flow Management Protocol Specification for IPv4 Version 1.0*; Network Working Group; Request for Comments: 1953; Category: Informational; IETF 1996

[NEHH+98] NEWMAN, P.; EDWARDS, W.L.; HINDEN, R.; HOFFMAN, E.; LIAW, F. CHING; LYON, T.; MINSHALL, G.: *Ipsilon's General Switch Management Protocol Specification Version 2.0*; Network Working Group; Request for Comments: 2297; Updates: 1987; Category: Informational; IETF 1998

[NEWM01] NEWMAN, DAVID: *Internet Core Router Test – Juniper Networks, Inc. Wins!*; Test Report Network Test Inc.; Light Reading; March 2001

[NICA01] NICHOLS, K.; CARPENTER, B.: *Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification;* Network Working Group; Request for Comments: 3086; Category: Informational; IETF 2001

[NKSM+97] NAGAMI, K.; KATSUBE, Y.; SHOBATAKE, Y.; MOGI, A.; MATSUZAWA, S.; JINMEI, T.; ESAKI, H.: *Toshiba's Flow Attribute Notification Protocol (FANP) Specification*; Network Working Group; Request for Comments: 2129; Category: Informational; IETF 1997

[NJZ99] NICHOLS, K.; JACOBSON, V.; ZHANG, L.: *A Two-bit Differentiated Services Architecture for the Internet;* Network Working Group; Request for Comments: 2638; Category: Informational; IETF 1999

[PAKO99] PAPE, O.; KOELLER, P.: *Sprach-Datenintegration;* Diplomarbeit an der Hochschule Bremen; Fachbereich Elektrotechnik und Informatik; Bremen 1999

[PAND95] PAND, R.: *Emerging mobile and personal communication systems*; IEEE Communications Magazine*;* Heft 33; June 1995

[PIPE98] PIPER, D.: *The Internet IP Security Domain of Interpretation for ISAKMP*; Network Working Group; Request for Comments: 2407; Category: Standards Track; IETF 1998

[POST81] POSTEL, J.: *Internet Protocol;* DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION; RFC: 791; IETF 1981

[Q.931(98)] Recommendation Q.931 (05/98): *ISDN user-network interface layer 3 specification for basic call control*; ITU-T 1998

[RDKR+97] REKHTER, Y., DAVIE, B.; KATZ, D.; ROSEN, E.; SWALLOW, G.: *Cisco Systems' Tag Switching Architecture Overview*; Network Working Group; Request for Comments: 2105; Category: Informational; IETF 1997

[REDE99] REDER, B.: *Zwischen Euphorie und Skepsis – elektronischer Handel in Deutschland*; NetworkWorld 01/99; Computerwoche; München 1999

[REPO92] REYNOLDS, J.; POSTEL, J.: *ASSIGNED NUMBERS*; Network Working Group; Requestfor Comments: 1340; Obsoletes RFCs: 1060, 1010, 990, 960, 943, 923, 900, 870, 820, 790, 776, 770, 762, 758,755, 750, 739, 604, 503, 433, 349; Obsoletes IENs: 127, 117, 93; IETF 1340

[RESC99] RESCORLA, E.; SCHIFFMAN, A.: *The Secure HyperText Transfer Protocol*; Network Working Group; Request for Comments: 2660; Category: Experimental; IETF 1999

[RMKG+96] REKHTER, Y.; MOSKOWITZ, B.; KARRENBERG, D.; DE GROOT, G.J.; LEAR, E.: *Address Allocation for Private Internets*; Network Working Group; Request for Comments: 1918; Obsoletes: 1627, 1597; BCP: 5; Category: Best Current Practice; IETF 1996

[ROSS01] ROSSENHÖVEL, C.: *Der Eindruck zählt – Sprachqualität in Voice-over-IP-Anlagen*; NetworkWorld 03-01; Computerwoche Verlag GmbH; München 2001

[RTFR+01] ROSEN, E.; TAPPAN, D.; FEDORKOV, G.; REKHTER, Y.; FARINACCI, D.; LI, T.; CONTA, A.: *MPLS Label Stack Encoding*; Network Working Group; Request for Comments: 3032; IETF 2001

[RVC01] ROSEN, E; VISWANATHAN, A.; CALLON; R: *Multiprotocol LabelSwitching Architecture*; Network Working Group; Request for Comments: 3031; IETF 2001

[SCFJ96] SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON, V.: *RTP: A Transport Protocol for Real-Time Applications*; Network Working Group; Audio-Video Transport Working Group; Request for Comments: 1889; Category: Standards Track; IETF 1996

[SCHN96] SCHNEIER, B.: *Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C*; Addison-Wesley; Bonn 1996

[SCHR01] SCHRENK, G.: *Multiprotocol Label Switching (MPLS) mit RSVP-TE: Traffic-Management in IP-Netzen*; NetworkWorld 14/01; München 2001

[SCHU96] SCHULZRINNE, H.: *RTP Profile for Audio and Video Conferences with Minimal Control*; Network Working Group; Audio-Video Transport Working Group; Request for Comments: 1890; Category: Standards Track; IETF 1996

[SCRO00] SCHULZRINNE, H.; ROSENBERG, J.: *A Framework for Telephony Routing over IP*; Network Working Group; Request for Comments: 2871; Category: Informational; IETF 2000

[SEUM01] SEUM, A.: *Priorisierungsmechanismen in Datennetzen*; Addision-Wesley; München 2001

[SHEN97] SHENKER, S.; PARTRIDGE, C.; GUERIN, R.: *Specification of Guaranteed Quality of Service*, Network Working Group; RFC-2212; Category: Standards Track; IETF 1997

[SIEM99] SIEMENS, E.: *Quality of Service in Rechnernetzen; Untersuchung von Verfahren zur QoS-Realisierung in IP-basierten Netzen*; Studienarbeit Universität Hannover; Prof. Dr.-Ing. H. Pralle; Hannover 1999

[SIEM00] SIEMENS, E.: *Realisierung und Bewertung eines QoS-Dienstes im Campus-Netz*; Diplomarbeit Universität Hannover; Lehrgebiet Rechnernetze und Verteilte Systeme; Prof. Dr.-Ing. H. Pralle; Hannover 2000

[SIMP94a] SIMPSON, W. (Editor): *The Point-to-Point Protocol (PPP);* Network Working Group; Request for Comments: 1661; STD: 51; Obsoletes: 1548; Category: Standards Track; IETF 1994

[SIMP94b] SIMPSON, W. (Editor): *PPP in HDLC-like Framing*; Network Working Group; Request for Comments: 1662; STD: 51; Obsoletes: 1549; Category: Standards Track; IETF 1994

[SPG97] SHENKER, S.; PARTRIDGE, C.; GUERIN, R.: *Specification of Guaranteed Quality of Service;* RFC-2212; Network Working Group; Category: Standards Track; IETF 1997

[SRL98] SCHULZRINNE, H.; RAO, A.; LANPHIER, R.: *Real Time Streaming Protocol (RTSP)*; Network Working Group; Request for Comments: 2326; Category: Standards Track; IETF 1998

[SRIS00] SRISURESH, P.: *Secure Remote Access with L2TP*; Network Working Group; Request for Comments: 2888; Category: Informational; IETF 2000

[SSCW00] SEAMAN, M.; SMITH, A.; CRAWLEY, E.; WROCLAWSKI, J.: *Integrated Service Mappings on IEEE 802 Networks*; Network Working Group; Request for Comments: 2815; Category: Standards Track; IETF 2000

[STEV97] STEVENS, W.: *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*; Network Working Group; Request for Comments: 2001; Category: Standards Track; IETF 1997

[T.120(98)] Recommendation T.120 Annex C (02/98): *Lightweight profiles for the T.120 architecture*; ITU-T 1998

[TVRP+99] TOWNSLEY, W.; VALENCIA, A.; RUBENS, A.; PALL, G.; ZORN, G.; PALTER, B.: *Layer Two Tunneling Protocol "L2TP"*; Network Working Group; Request for Comments: 2661; Category: Standards Track; IETF 1999

[VLK98] VALENCIA, A.; LITTLEWOOD, M.; KOLAR, T.: *Cisco Layer Two Forwarding (Protocol) "L2F"*; Network Working Group; Request for Comments: 2341; Category: Historic

[WECK01] WECK, G.: *Zertifikate, Protokolle und Normen: Stolpersteine auf dem Weg zu einer PKI*; aus: Elektronische Geschäftsprozesse: Grundlagen, Sicherheitsaspekte, Realisierungen, Anwendungen; Patrick Horster; IT Verlag für Informationstechnik; Höhenkirchen 2001

[WILD99] WILDE, A.: *SDH in der Praxis*, VDE Verlag; 1999

[WROC97a] WROCLAWSKI, J.: *Specification of the Controlled-Load Network Element Service;* RFC-2211; Network Working Group; Category: Standards Track; IETF 1997

[WROC97b] WROCLAWSKI, J.: *The Use of RSVP with IETF Integrated Services*; RFC-2210; Network Working Group; Category: Standards Track; IETF 1997

[X.209(88)] Recommendation X.209: *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)*; ITU-T 1988

[X.509(00)] Recommendation X.509: Information technology – Open Systems Interconnection – The directory: *Public-key and attribute certificate frameworks*; ITU-T 200

[X.500(01)] Recommendation X.500: *Information Technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*; Available as a prepublished version; ITU-T 2001

[XIAO00] XIAO, X.: *Providing Quality-of-Service in the Internet*; Dissertation; Department of Computer Science and Engineering; USA 2000

[YHBB+00] YAVATKAR, R.; HOFFMAN, D.; BERNET, Y.; BAKER, F.; SPEER, M.: *SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks*; Network Working Group; Request for Comments: 2814; Category: Standards Track; IETF 2000

A.2 Glossar

ATM Adaptation Layer: Anpassungsschicht von ATM für die Schnittstelle zwischen den höheren Anwendungsschichten und der ATM-Schicht	*AAL*
ATM Adaptation Layer Typ 5: Anpassungsschicht zur Übertragung von Daten der Dienstklasse D	*AAL-5*
Available Bit Rate: ATM-Dienstklasse, die sich immer an der Netzauslastung orientiert und so dem Benutzer nur die freien Ressourcen zur Verfügung stellt. Dabei ist der Benutzer auch in der Lage, die gewünschte Datenmenge skalierbar einzustellen.	*ABR*
Algebraic CELP	*ACELP*
Admission_Confirm-Nachricht: der Zugang zum H.323-Endpunkt wird bestätigt.	*ACF*
Access Control List: Zugriffslisten, welche auf einem Router als Filtermöglichkeit implementiert werden können.	*ACL*
Adaptive Difference Pulse Code Modulation	*ADPCM*
Advertising Specification: diese Struktur wird in der PATH-Message von RSVP Hop-by-hop durchgereicht und dient zur Bestimmung der gesamten Dienstgüte vom Sender bis zum Empfänger.	AD_{SPEC}
Aggressive Exchange: vollständiger Austausch bei IPsec im Aggressive-Modus	*AE*
Assured Forwarding: DiffServ-Servicelevel	*AF*
Assured Forwarding PHB Group: bietet eine stärkere Differenzierung der Datenströme und eine höhere Dienstgüte als EFP.	*AFPG*
Eine Zusammensetzung mehrerer Datenströme anhand bestimmter Kriterien	*Aggregate*
Authentication Header (IPsec): authentifiziert Paketdaten und Teile des IP-Headers bei IPsec.	*AH*
In der Literatur über Verschlüsselung bzw. Kryptographie werden den an Protokollen beteiligten Personen oft Namen gegeben. So kommt jeder Person eine bestimmte Aufgabe zu, die sich meist schon in dem jeweiligen Namen spiegelt: Alice ist die erste Person in allen Protokollen.	*Alice*
American National Standards Institute	*ANSI*
Nachträgliches Einführen von Datenpaketen in einen vorhandenen Datenstrom, um unerkannt Attacken ausführen zu können.	*Anti-Replay Attacke*

API Application Programming Interfaces: Schnittstelle zwischen der OSI-Schicht 7 und den Anwendungen/Diensten

ARIS Aggregate Route IP Switching: Label-Switching-Verfahren von IBM, welches in die MPLS-Standardisierung mit eingeflossen ist.

ARJ Admission_Reject-Nachricht: der Zugang zum H.323-Endpunkt wird abgelehnt.

ARP Address Resolution Protocol: Umwandlung von IP-Adressen in MAC-Adressen

ARQ Admission_Request-Nachricht: ermöglicht den Zugang zu einem H.323-Endpunkt in einem LAN über einen Gatekeeper.

AS Autonomous Systems: in der IP-Terminologie ist ein autonomes System ein Verbund von Routern und Netzwerken, die einer einzigen administrativen Instanz unterstehen. Das bedeutet, dass sie alle zu einer Organisation oder zu einem Unternehmen gehören.

ASN.1 Abstract Syntax Notation One

ATM Asynchronous Transfer Mode: zellbasiertes Hochgeschwindigkeitsnetz mit eigener Signalisierung und eigenem Routing; bildet die Basistechnologie für das B-ISDN.

ATMARP Asynchronous Transfer Mode Address Resolution Protocol: Umwandlung von IP- in ATM-Adressen

AVP Attribute Value Pairs: Parameter, die in den Steuernachrichten des L2TP-Protokolls enthalten sind.

B2B Business-to-Business

B2C Business-to-Consumer

BA Basic Authentication: Authentifizierungsmöglichkeit des Clients bei HTTP; Authentifizierung über User-ID und dazugehöriges Passwort, welches in Klartext übermittelt wird.

BA Behavior Aggregate: ein Bündel von Verkehrsflüssen bei DiffServ, die vom Netz auf gleiche Weise behandelt werden.

BB Bandwidth Broker

BCF Bandwidth_Confirm-Nachricht: Bestätigung der Anfrage nach Änderung des Bandbreitenbedarfs bei H.323 innerhalb eines Gatekeeper

Basic Encoding Rules: definiert nach X.209; schreibt vor, dass die Kodierung vier Komponenten in einer bestimmten Reihenfolge enthalten muss.	*BER*
Bürgerliches Gesetzbuch	*BGB*
Border Gateway Protocol: dynamisches Routing-Protokoll, welches sowohl zwischen zwei als auch innerhalb eines autonomen Systems eingesetzt werden kann.	*BGP*
Broadband-Integrated Services Digital Network: Definition eines dreidimensionalen Referenzmodells, welches das herkömmliche ISDN auf breitbandiger Basis mittels ATM erweitert.	*B-ISDN*
Bump-in-the-Stack: Implementierung im Protokollstapel (siehe IPsec)	*BITS*
Bump-in-the-Wire: Implementierung in einem unabhängigen Zusatzgerät (siehe IPsec)	*BITW*
In der Literatur über Verschlüsselung bzw. Kryptographie werden den an Protokollen beteiligten Personen oft Namen gegeben. So kommt jeder Person eine bestimmte Aufgabe zu, die sich meist in dem jeweiligen Namen spiegelt: Bob ist die zweite beteiligte Person in allen Protokollen.	*Bob*
Ein Link bei z.B. DiffServ, der zwei angrenzende Knoten aus benachbarten Domänen verbindet.	*Boundary Link*
Bandwidth_Reject-Nachricht: Ablehnung der Anfrage nach Änderung des Bandbreitenbedarfs bei H.323 innerhalb eines Gatekeeper	*BRJ*
Bandwidth_Request-Nachricht: Anfrage nach Änderung des Bandbreitenbedarfs bei H.323 innerhalb eines Gatekeeper	*BRQ*
Blockade State Block; beinhaltet die höchsten zulässigen Grenzwerte für neue Reservierungsanforderungen im Blockadezustand bei RSVP	*BSB*
Berkeley Software Distribution: frei erhältliches Unix-System von der Universität Berkeley	*BSD*
Eine Menge von Datenpaketen, die direkt aufeinanderfolgend (mit der maximalen bzw. beinahe maximalen Datenrate auf dem Netzinterface) in einem Knoten ankommen bzw. vom Knoten gesendet werden.	*Burst*
Unterschiedliche Datenaufkommen, die stochastisch – also nicht voraussagbar – auftreten: das Verhältnis der maximalen zur mittleren Datenrate.	*Burstiness*
Broadcast and Unknown Server: LANE-Implementierung für das Weiterleiten von Broadcasts und dem Etablieren von VCCs	*BUS*

C Container: Payload-Container im SDH-Multiplexschema

CA Certification Authority: erstellt Zertifikate für öffentliche Schlüssel und bürgt damit für deren Echtheit.

CAC Call Admission Control

CAS Channel Associated Signalling: bestimmte Bits können bei ATM fest einzelnen Kanälen zur Übermittlung ihrer Zeichengabeinformation zugeordnet werden.

CAST CAST-Algorithmus stammt aus Kanada und wurde von Carlisle Adams und Stafford Tavares entwickelt. Der Algorithmus verschlüsselt 64-Bit-Klartextblöcke mit 64-Bit-Schlüsseln bzw. 128-Bit-Schlüsseln bei CAST-128.

CBC Ciper Block Chaining: eine Betriebsart der Blockchiffren, bei der das Ergebnis der Verschlüsselung eines Klartextblocks mittels XOR-Funktion mit dem nachfolgenden Klartextblock verknüpft wird, bevor die nächste Verschlüsselung stattfindet. Der erste Block wird dabei mit einem Initialisierungsvektor (IV) verknüpft.

CBR Constant Bit Rate: ATM-Dienstklasse, die eine ganz bestimmte Bandbreite und Dienstgüte benötigt, die weder signifikante Verzögerungen noch Jitter oder Zellverluste verträgt.

CB-WFQ Class Based Weighted Fair Queueing: Einer Klasse kann hierbei eine feste Datenrate zugeordnet werden. Diese kann wahlweise im internen WFQ-Algorithmus oder strikt zugeordnet werden. Wird sie einer Klasse strikt zugeordnet, so verhält sich diese gegenüber anderen Klassen wie beim PQ. Der Vorteil einer strikten Priorisierung innerhalb des CB-WFQ gegenüber einer PQ besteht darin, dass die Datenströme anderer Klassen weiterhin mit dem WFQ abgearbeitet werden.

CC Cross Certificates: zwei Certification Authorities zertifizieren sich gegenseitig.

CCE Collaborative Computing Environments: unternehmensweite digitale Bibliotheken für die Zusammenarbeit und das Wissensmanagement im Unternehmen

CCP Compression Control Protocol

CDV Cell Delay Variation: Zellverzögerungsschwankung; die Laufzeitschwankungen sind für Echtzeitanwendungen möglichst gering zu halten. Dabei können sowohl Zellverluste als auch Änderungen der Laufzeit, bedingt durch Zwischenspeicherung und Vermittlung, zu Störungen führen.

CDVT Cell Delay Variation Tolerance: ATM-Dienstparameter, welches den Toleranzwert von CDV zur Vermeidung von Jitter-Effekten festlegt.

Empfang von ATM-Zellen in unbestimmter Reihenfolge, wodurch eine Zuordnung von Zellen zu Paketen nicht mehr gewährleistet werden kann.	*Cell Interleave*
Codebook Excited Linear Predictive Coding: hybrides Kodierungsverfahren, welches die Vorteile der Signalformkodierung und der parametrischen Verfahren ausnutzt, ohne die schlechten Eigenschaften beider Verfahren übernehmen zu müssen.	*CELP*
Cell Error Ratio; Zellfehlerverhältnis: Der CER-Wert gibt den Anteil aller Zellen an, die fehlerhaft übertragen wurden. Bei reiner Datenübertragung bedeutet der Verlust eines Pakets die Wiederholung der Übertragung und damit eine Verringerung des Durchsatzes.	*CER*
ConFirm: Bestätigungsnachricht	*CF*
Cipher Feedback: eine Betriebsart der Blockchiffren, bei dem der vorangehende chiffrierte Block verschlüsselt und mit dem aktuellen Textblock mittels XOR verknüpft wird.	*CFB*
Conference_ID-Nachricht	*CFI*
Challenge Handshake Authentication Protocol: überprüft periodisch die Identität des Kommunikationspartners.	*CHAP*
Common Gateway Interface	*CGI*
Classifier Klassifizierer: eine Einheit, die Datenpakete anhand des IP-Headers entsprechend der definierten Regeln sortiert (klassifiziert).	
Client ID	*CLID*
Controlled Load Network Element; ermöglicht die dedizierte Zuweisung von Bandbreite nach RFC-2211. Dadurch wird die Kontrolle der von den Applikationen angeforderten Übertragungsraten ermöglicht. Alle anderen QoS-Parameter bleiben jedoch unberücksichtigt, das heißt, Netzparameter können nicht mit einbezogen werden.	*CLNE*
Cell Loss Priority: gibt die Zellpriorität bei ATM an.	*CLP*
Cell Loss Ratio: Zellverlustverhältnis; der Zellverlust ist das Verhältnis der Anzahl der verlorenen Zellen zu der Gesamtzahl der gesendeten Zellen innerhalb eines bestimmten virtuellen Kanals. Eine genauere Erfassung des CLR-Werts kann durch die Auskopplung von langen Fehler-Bursts erfolgen.	*CLR*
Conservative Label Retention Mode: reduziert die Anzahl der in den LSR zu verwaltenden Labels bei MPLS.	*CLRM*

CMR Cell Misinsertion Rate; Zelleinfügungsfehlerrate: die Zellen, welche fehlerhaft in den aktuellen Zellstrom eingefügt werden, kann man durch diesen Parameter feststellen. Fehlerhaft eingefügte Zelle heißt, die Zelle ist über den falschen virtuellen Kanal und/oder Pfad empfangen worden, wodurch Zellen verworfen werden müssen.

CN Corporate Network

COPS Common Open Policy Server: Signalisierungsprotokoll im Internet, welches zur Steuerung einzelner Netzwerkelemente beim Bandwidth Broker entwickelt wurde.

CP Certification Path: Kette von Zertifikaten, die so lange verfolgt werden muss, bis man auf eine Zertifizierungsinstanz trifft, der man vertraut.

CPCS Common Part Convergence Sublayer: gemeinsame Konvergenzteilschicht (AAL-Typ 3/4 und 5)

CPL Call Processing Language: Eine Sprache für die Benutzerkontrolle von IP-Telefoniediensten

CQ Customer Queueing: läuft ähnlich dem WRR ab; Unterschiede bestehen darin, dass eine variable Anzahl an Queues zugelassen ist und die Festlegung der Gewichtung nicht in der Zahl der vermittelten Bytes, sondern explizit in der Zahl der zu verarbeitenden Pakete festgehalten wird.

CR Constraint-based Routing: Signalisierungsverfahren, welches zum Traffic Engineering bei MPLS eingesetzt werden kann.

CRC Cyclic Redundancy Check: Prüfsumme zur Erkennung und ggf. Korrektur von Übertragungsfehlern

CRL Certificate Revocation List: Liste mit Zertifikaten, die widerrufen wurden und noch nicht abgelaufen sind, da die Echtheit nicht mehr sichergestellt werden kann.

CRV Call Reference Value: Kanalbezugnahme bei VoIP

CS Convergence Sublayer: Konvergenzteilschicht ist für die Bereitstellung der dienstspezifischen Dienstparameter zuständig. Hierfür wird sie je nach Dienstklasse in eine dienstspezifische Teilschicht und eine gemeinsame Teilschicht unterteilt.

CS-ACELP CONJUNGATE STRUCTUR ALGEBRAIC CELP: entspricht dem ITU-T-Standard G.729 und führt vor der Kodierung aufwendige Analysen beim Vergleich des Sprachsignals mit dem Modell durch. CS-ACELP erreicht eine identische Sprachqualität wie LD-CELP, benötigt jedoch nur die halbe Bitrate. Dafür ist

eine wesentlich höhere Rechenleistung bei der Kodierung und Dekodierung der Signale erforderlich. Die verursachte Signalverzögerung durch den Kodierer beträgt mindestens 20 ms.

Class Selector Codepoints: wurden definiert, um eine Abwärtskompatibilität zum IPv4-TOS-Feld zu schaffen; hier wurden deshalb einige Codepoints vordefiniert. *CSC*

Constraint Shortest Path First: es handelt sich hierbei um einen modifizierten Shortest Path First Algorithmus, der bei der Pfadbestimmung bei MPLS auch die zusätzlichen Parameter berücksichtigen kann. *CSPF*

Cell Switch Router: Label Switching Verfahren von Toshiba; Komponente kann mit einem LSR gleichgesetzt werden. *CSR*

Cell Transfer Rate; Zellübertragungsrate: Die Zellmenge, die während eines Datenaustauschs transportiert wird, wird durch diesen Parameter angegeben. Bei isochronen Datenströmen muss dieser Wert sehr klein sein, um die Verständlichkeit zweier Kommunikationspartner gewährleisten zu können. *CTR*

Digest Access Authentication: Authentifizierungsmöglichkeit des Clients bei http; Authentifizierung über User-ID und dazugehöriges Passwort, welches nicht in Klartext übermittelt wird. *DAA*

Disengage_Confirm-Nachricht: Bestätigung der Verbindungsauflösung bei H.323 *DCF*

Data Encryption Standard: symmetrischer Verschlüsselungsalgorithmus, der zur Klasse der Blockchiffren gehört. *DES*

Default Forwarding: DiffServ-Servicelevel *DF*

Dynamic Host Control Protocol: ursprünglich von Microsoft entwickeltes Protokoll für die dynamische Vergabe von IP-Adressen nach RFC-2131 *DHCP*

Differentiated Services *DiffServ*

Eine digitale Signatur wird mit einer Kombination aus Einweg-Hash-Funktion und asymmetrischem Verschlüsselungsverfahren erstellt. Der Hash-Wert der zu signierenden Nachricht wird dabei mit dem privaten Schlüssel chiffriert. Digitale Signaturen dienen der Authentifizierung und dem Schutz der Integrität der Nachricht. Sie können auch Verbindlichkeit gewährleisten. *Digitale Signatur*

Data Link Connection Identifier: Feld im Frame-Relay- Header *DLCI*

Domain Name Service: verteiltes Datenbanksystem, welches IP-Adressen in Domain-Namen umwandelt. *DNS*

DOI Domain-of-Interpretation: Protokollunabhängigkeit wird dadurch geschaffen, dass die Parameter getrennt vom eigentlichen Protokoll vereinbart werden (Beispiele: IKE und IPsec).

DOWN DS Domain Downstream DiffServ Domain: Eine Domäne, die die Daten von der benachbarten Domäne empfängt; also von außen in die Domäne

DP Drop Precedence: Relative Verlustwahrscheinlichkeit bei AFPG, die jedem Paket durch Klassen zum Verwerfen des Pakets zugewiesen wird.

DPCM Difference Pulse Code Modulation

DRJ Disengage_Reject-Nachricht: falls ein Endpunkt bei einem Gatekeeper nicht registriert war und dennoch eine DRQ empfängt, sendet er diese Nachricht. Dropper Eine DiffServ-Einheit, die das Droppen in einem DS-Knoten an einem bestimmten Datenstrom vornimmt.

Dropping Verwerfen der Pakete anhand definierter Regeln

DRQ Disengage_Request-Nachricht: ein Endpunkt oder ein Gatekeeper signalisieren, dass die Verbindung aufgelöst wird.

DSBM Designated Subnet Bandwidth Manager: steuert die Ressourcenanforderungen für jedes Netzsegment.

DS Boundary Node Wird auch als DS-Grenzknoten bezeichnet: Ein DS-Knoten einer Domäne, der entweder mit einem Grenzknoten einer anderen DS-Domäne oder mit einer Nicht-DS-Domäne verbindet.

DS Domain Eine Menge nebeneinander liegender Knoten, die die definierten PHB-Regeln und Dienstrealisierung auf die gleiche Weise unterstützen. In der Regel stellt eine DS-Domäne eine administrative Einheit dar.

DS Egress Node Austrittsknoten: Ein Grenzknoten (Boundary Node), der den Datenverkehr aus der DS-Domäne zur benachbarten DS-Domäne bzw. zur Nicht-DS-Domäne behandelt.

DS Ingress Node Einlaufknoten: ein Grenzknoten (Boundary Node), der den von außen kommenden Verkehr beim Eintritt in die DS-Domäne behandelt.

DS Interior Node Ein Knoten, der kein DS-Grenzknoten ist.

DS Differentiated Services

DSAP Destination Service Access Point

DS-Codepoint Ein bestimmter Wert der ersten 6 Bit des DS-Feldes. Jedem DS-Codepoint wird ein PHB eindeutig zugeordnet.

Ein Knoten bzw. eine Domäne, in den die in DiffServ beschriebene Architektur implementiert ist.	*DS-fähig*
Das TOS-Oktett des IP-Header (IP-Version 4) bzw. Traffic Class Octet (IP-Version 6), wenn es in Bezug auf die Implementierung der DiffServ gemäß Definition in RFC-2474 verwendet wird.	*DS-Feld*
Ein DS-Netzknoten: ein Host oder ein Router, der mindestens einen DiffServ-Dienst unterstützt.	*DS-Node*
Digital Signature Standard: ist ein standardisiertes Signaturverfahren, welches aus lizenzrechtlichen Gründen nicht auf RSA basiert.	*DSS*
Dual Tone Multi-Frequency	*DTMF*
ITU-Empfehlung: The international public telecommunication numbering plan	*E.164*
Explicit Admission Control: jede einzelne Ressourcenanforderung wird explizit angefordert und bestätigt.	*EAC*
Electronic Code-book: eine Betriebsart von Blockchiffren	*ECB*
Encryption Control Protocol	*ECP*
Electronic Data Interchange: in Form des elektronischen Dokumentenaustauschs und Zahlungsverkehrs	*EDI*
Elektronische Datenverarbeitung	*EDV*
Expedited Forwarding: DiffServ-Servicelevel	*EF*
Expedited Forwarding PHB: ist ähnlich dem Controlled Load Service und soll lediglich einen besseren Dienst als den Best-effort anbieten. Er wird oft auch als Premium Service bezeichnet.	*EFP*
Electronic Funds Transfer	*EFT*
Einführungsgesetz zum Bürgerlichen Gesetzbuch	*EGBGB*
Emulated LAN: Nachbildung eines Ethernet- oder Token-Ring-Netzes auf ATM	*ELAN*
Ist eine Instanz, die Daten vom RSVP-Sender erwartet. Da das Aushandeln der QoS-Parameter unabhängig vom Nutzdatenfluss stattfindet, werden IP-Pakete zwischen RSVP-Sender und RSVP-Empfänger in beiden Richtungen ausgetauscht. Somit kann der RSVP-Empfänger auch Sender eines IP-Pakets (z.B. einer RESV-Nachricht) sein. Der RSVP-Empfänger sendet Reservierungsanforderungen ans Netz.	*Empfänger*

EPD Early Packet Discard: bevor eine Überlastsituation auftreten kann, werden Zellen bei ATM über die Anpassungsschicht verworfen.

ER-Hop TLV Explicit Route Hop TLV: entspricht einem Subeintrag im ERO des RSVP-TE bei dem Ansatz CR-LDP von MPLS.

ERO Explicit Route Object: enthält die Adressen der zu durchlaufenden LSR und stellt eine wesentliche Komponente des Explicit Routings bei MPLS dar.

ER-TLV Explicit Route TLV: beinhaltet analog zum ERO im RSVP-TE die Adressen der zu durchlaufenden LSR bei MPLS.

ESP Encapsulating Security Payload (IPsec): Header kapselt die zu schützenden Daten ein und gewährleistet bei Bedarf deren Vertraulichkeit durch Verschlüsselung. Außerdem sind Möglichkeiten zum Schutz der Integrität und zur Authentizität der Datenpakete vorgesehen.

ETSI European Telecommunications Standards Institute

Eve In der Literatur über Verschlüsselung bzw. Kryptographie werden den an Protokollen beteiligten Personen oft Namen gegeben. So kommt jeder Person eine bestimmte Aufgabe zu, die sich meist schon in dem jeweiligen Namen spiegelt: Eve ist ein passiver Gegner (Eavesdropper)

Explicit Routing Source Routing: es wird der Routingpfad bereits an der Quelle festgelegt

FCS Frame Check Sequence: Blocksicherung bzw. Prüfsumme in LANs mittels eines Generatorpolynoms, um die Fehlerfreiheit von übertragenen Paketen feststellen zu können.

FEC Forward Error Correction: Echtzeitfehlerkorrektur in ATM-Netzen

FEC Forwarding Equivalence Class: wird durch bestimmte Metriken bei MPLS festgelegt. Üblicherweise wird das Adressenpräfix der Zieladresse verwendet. Alle Pakete, deren Metriken innerhalb einer FEC liegen, werden vom LSR gleich behandelt.

FIB Forwarding Information Base: Informationsdatenbank im ARIS-Verfahren von IBM (einem Label-Switching-Verfahren)

FIFO First-In-First-Out: Speicherarbeitsprinzip

$FILTER_{SPEC}$ Beinhaltet die Adressen (IP-Adresse und Port-Nummer) der in der RSVP-Sitzung zugelassenen Sender. Dabei können die Sender explizit aufgelistet oder als Wildcard angegeben werden.

Als ein Datenfluss (flow) wird laut [BRAD94] eine Menge von Datenpaketen, die alle vom selben Sender erzeugt werden und zu ein und derselben QoS-Anforderung gehören, bezeichnet.	*Flow*
Die $FLOW_{SPEC}$-Datenstruktur beschreibt die gewünschten QoS-Parameter einer Reservierungsanforderung. Diese Struktur besteht wiederum aus einer T_{SPEC}, die die Parameter des Datenflusses beschreibt, und einer R_{SPEC}, die zusätzliche Reservierungsparameter beschreibt. Das Format und der Inhalt der $FLOW_{SPEC}$ wird in den jeweiligen Dienstspezifikationen des IntServ festgelegt und ist für das RSVP unsichtbar.	*$FLOW_{SPEC}$*
Frame Relay	*FR*
FEC-to-NHLFE: die FTN ist für die LER bei MPLS erforderlich, da diese im Gegensatz zu den LSR nicht nur MPLS-, sondern auch konventionelle Schicht-3-Pakete verarbeiten müssen.	*FTN*
File Transfer Protocol	*FTP*
ITU-Empfehlung: General Recommendations on the transmission quality for an entire international telephone connection: One-way transmission time	*G.114*
ITU-Empfehlung: General – Physical/electrical characteristics of hierarchical digital interfaces	*G.703*
ITU-Empfehlung: General – synchronous frame structures used at 1544, 6312, 2048, 8488 and 44 736 Kbit/s hierarchical levels	*G.704*
ITU-Empfehlung: Coding of analogue signals by pulse code modulation: Pulse code modulation (PCM) of voice frequencies	*G.711*
ITU-Empfehlung: Coding of analogue signals by methods other than PCM	*G.720-G.729*
ITU-Empfehlung: 7 kHz audio-coding within 64 Kbit/s	*G.722*
ITU-Empfehlung: Speech coders	*G.723*
ITU-Empfehlung: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbit/s	*G.723.1*
ITU-Empfehlung: 40, 32, 24, 16 Kbit/s adaptive differential pulse code modulation (ADPCM)	*G.726*
ITU-Empfehlung: 5-, 4-, 3- and 2-bits/sample embedded adaptive differential pulse code modulation (ADPCM)	*G.727*
ITU-Empfehlung: Coding of speech at 16 Kbit/s using low-delay code excited linear prediction	*G.728*

G.729 ITU-Empfehlung: Coding of speech at 8 Kbit/s using conjugate-structure algebraic-code-excited linear-prediction

G.729A Reduced complexity 8 Kbit/s CS-ACELP speech codec

GCF Gatekeeper_Confirm-Nachricht

GCP Generalized Characterization Parameter: allgemeine Charakterisierungsparameter der AD_{SPEC}

GFC Generic Flow Control: Zugriffssteuerung/Sendeberechtigung innerhalb eines ATM-Protokollkopfs

GFR Guaranteed Frame Rate: ATM-Dienstklasse, die Zellen über Pakete überträgt.

GQOS Guaranteed Quality-of-Service; dieser Service nach RFC-2212 kontrolliert die maximale Pufferverzögerung bei IntServ. Er stellt somit sicher, dass Pakete das Ziel mit der angeforderten Verzögerung erreichen.

GRE V2 Generic Routing Encapsulation Protocols der Version 2: es handelt sich um eine Richtlinie, wie Tunnelpakete aufgebaut sein sollen.

GRJ Gatekeeper_Reject-Nachricht

GRQ Gatekeeper_Request-Nachricht

GSM Global System for Mobile Communication: digitales Mobilfunksystem der zweiten Generation (2G)

GSMP General Switch Management Protocol: Es handelt sich hierbei um ein Master-/Slave-Protokoll, das kein fester Bestandteil des Ipsilon IP-Switching ist, sondern eine Systemoptimierung darstellt. Es erlaubt dem IP-Switch-Controller (Master) unter anderem VC-Verbindungen auf- und abzubauen.

GSTN General Switched Telecommunication Network

H.225.0 ITU-Empfehlung: *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*

H.230 ITU-Empfehlung: Frame-synchronous control and indication signals for audiovisual systems

H.231 ITU-Empfehlung: Multipoint control units for audiovisual systems using digital channels up to 1920 Kbit/s

H.235 ITU-Empfehlung: *Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals*

ITU-Empfehlung: System for establishing communication between audiovisual terminals using digital channels up to 2 Mbit/	*H.242*
ITU-Empfehlung: *Control protocol for multimedia communication*	*H.245*
ITU-Empfehlung: *Interworking of H-Series multimedia terminals with H-Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN*	*H.246*
ITU-Empfehlung: Coding of moving video – Video codec for audiovisual services at p x 64 Kbit/s	*H.261*
ITU-Empfehlung: Video coding for low bit rate communication	*H.263*
ITU-Empfehlung: Systems and terminal equipment for audiovisual services: Narrow-band visual telephone systems and terminal equipment	*H.320*
ITU-Empfehlung: Systems and terminal equipment for audiovisual services: Packet based multimedia communications systems	*H.323*
ITU-Empfehlung: Generic functional protocol for the support of supplementary services in H.323	*H.450.1*
Header Error Control: Prüfsequenz für den Zellkopf von ATM	*HEC*
Ist ein verschachtelter Key Hash, welcher in RFC-2104 beschrieben ist und mit jeder beliebigen Hash-Funktion benutzt werden kann.	*HMAC*
Head-of-the-Line-Blocking: Überbelastung eines Ports wirkt sich durch die Backplane-Struktur auch auf unbelastete Ports aus.	*HoLB*
Heterogeneous Source Branch Points: ein Punkt, an dem ein Multicast-Strom auf unterschiedliche Zweige verteilt wird. Dabei sind nicht alle T_{SPEC}-Parameter gleich.	*HSBP*
Hyper-Text Markup Language: Beschreibungssprache; eine einfache Integration von Layoutinformationen, Grafiken und Sound und die Möglichkeit mittels Formularelementen (Schaltflächen, Listen, Auswahlfelder etc.) eine Interaktion mit dem Benutzer herzustellen.	*HTML*
Hyper-Text Transport Protocol: basierend auf dem Client-/Server-Prinzip kann ein Benutzer mit einem speziellen Client-Programm (für alle gängigen Plattformen verfügbar) Dokumente und Informationen von einem Rechner, genannt Server, anfordern.	*HTTP*
ITU-Empfehlung: B-ISDN ATM adaptation layer (AAL) specification	*I.400*
Integrity Check Value: Integritätsüberprüfung bei IPsec	*ICV*

IDEA International Data Encryption Algorithm: ein symmetrischer Verschlüsselungsalgorithmus, der zur Klasse der Blockchiffren gehört.

IFMP Ipsilon Flow Management Protocol: Nach den Festlegungen von Ipsilon ist das IFMP ein Soft-State-Protokoll, welches das Downstream-Binding unterstützt. Zur Übertragung der IFMP-Nachrichten werden die Default VCs verwendet.

IGP Interior Gateway Protocol

IKE Internet Key Exchange: ein in RFC-2409 beschriebenes konkretes Schlüsselaustauschverfahren, das auf die Benutzung mit ISAKMP abgestimmt ist.

ILM Incoming Label Map: eine Komponente einer MPLS-Forwarding-Tabelle, deren Funktion darin besteht, jedes empfangene MPLS-Paket anhand seines Labels auf einen oder mehrere NHLFE abzubilden.

InATMARP Inverse ATMARP: Umgekehrte Vorgehensweise von ATMARP bei der Umwandlung von ATM- in IP-Adressen

IntServ Integrated Services

IP Internet Protocol

IPCP Internet Protocol Control Protocol: NCP-Definition zum Etablieren und Konfigurieren von IP über PPP

IPE Identity Protection Exchange: Austausch bei IPsec, bei dem die Identität der Teilnehmer geschützt ist.

IPRA Internet Policy Registration Authority: zertifiziert nur die auf der Stufe 2 stehenden, so genannten Policy Certification Authorities (PCAs) und legt internetweite Richtlinien fest, an die die PCAs gebunden sind.

IPv4 Internet Protocol Version 4

IPv6 Internet Protocol Version 6

IPX Internetwork Packet Exchange Protocol: herstellerspezifisches Netzprotokoll, welches hauptsächlich in Netware-Netzen eingesetzt wird.

IRQ Information_Request-Nachricht: Statusabfrage bei H.323

IRR Information_Request_Response: Antwortnachricht der IRQ-Nachricht, welche Informationen wie Adressen, Referenzwerte und die Bandbreite für die Verbindung enthält.

ISAKMP Internet Security Association and Key Management Protocol: dient dazu, eine Security Association (SA) zwischen Kommunikationspartnern auszuhandeln,

d.h., es werden Methoden und Formate zur Erzeugung, Modifikation und Vernichtung der SA definiert.	
Internet Server Application Interface	*ISAPI*
Internet Service Provider: Anbieter von Internet Connectivity und IP-Diensten; der Trend geht zum Application/Content Service Provider	*ISP*
Integrated Switch Router: Label-Switching-Router beim IBM-Ansatz; ist mit dem LSR gleichzusetzen.	*ISR*
Initialisierungsvektor: der IV besitzt starke Eigenschaften bzgl. Zufallszahlen, um sicherstellen zu können, dass ein identischer Ursprungstext nicht einen identischen chiffrierten Block ergibt. Er wird in verschiedenen Betriebsmodi von Verschlüsselungsalgorithmen verwendet.	*IV*
Interworking Functions: Anpassung unterschiedlicher Technologien, um eine direkte Kompatibilität herzustellen.	*IWF*
*Jitter sind Kurzzeitabweichungen der Kennzeitpunkte (*die Zeitpunkte, zu denen im Originalsignal die steigenden/fallenden Flanken liegen*) von Digitalsignalen gegenüber ihren idealen (äquidistanten) zeitlichen Positionen*	*Jitter*
Kombination aus Jitterfrequenz und Jitteramplitude, die ein System noch verträgt.	*Jitterverträglichkeit*
Key Distribution Center: sichere Verifikation von Diensten und Nutzern	*KDC*
Key Management Protocol: Schlüsselmanagementprotokoll, z.B. IKE, ISAKMP oder SKIP	*KMP*
Als Knoten wird ein Endgerät (Host) oder ein Router im Netz bezeichnet.	*Knoten*
Layer 2 Forwarding: Protokoll von Cisco nach RFC-2341. Als Home Gateway kann ein Router eingesetzt werden, während Access Server als RAS zum Einsatz kommen.	*L2F*
Layer 2 Tunneling Protocol: Cisco und Microsoft haben sich dabei auf die Zusammenarbeit an der Entwicklung eines einzigen Standards innerhalb der IETF nach RFC-2661 geeinigt; L2TP ist ein hybrides Verfahren aus L2F und PPTP.	*L2TP*
Label-FEC-Zuordnungen bei MPLS bzw. Label-basierten Verfahren	*Label Binding*
L2TP Access Controller	*LAC*
LAN-Emulation: ATM-Forum-Spezifikation für die Anbindung von Legacy LANs (Shared Media LANs wie Ethernet und Token Ring) an ATM	*LANE*

Layer-2-Domäne Layer-2-Netz: so wird ein zusammenhängender Bereich im Netz bezeichnet, innerhalb dessen das Weiterleiten der Datenpakete nur aufgrund des L2-Headers stattfindet. An den Grenzen einer solchen Domäne – also im Eintrittspunkt und im Austrittspunkt – kommen in der Regel Layer-3-Einheiten zum Einsatz.

Layer-2-Gerät Layer-2-Einheit: so wird ein Gerät bzw. eine Einheit im Netz bezeichnet, die lediglich Daten-Header der Schicht 2-Pakete und keine Schicht 3-Informationen auswertet (z.B. ein Switch oder eine Bridge).

Layer-3-Gerät Layer-3-Einheit: so wird ein Gerät bzw. eine Einheit bezeichnet, die die Schicht-3-Daten-Header auswertet. Dieses kann ein Router oder ein Host sein.

LCF Location_Confirm-Nachricht: Bestätigung der Lokalisierung, die der Gatekeeper als Antwort sendet; enthalten sind weitere Kontaktinformationen über den Endpunkt oder den Gatekeeper des Endpunkts.

LCP Link Control Protocol: zum Etablieren, Konfigurieren und Testen einer Data-Link-Verbindung bei PPP

LDAP Lightweight Directory Access Protocol: für den Zugang zu Online-Directory-Diensten. Es ist 1995 von der Universität Michigan spezifiziert worden, um den Zugriff auf X.500-Adressverzeichnisse zu erleichtern.

LD-CELP Low-Delay CELP: Der Standard LD-CELP, entsprechend der ITU-T-Spezifikation G.728, weist eine Bitrate von 16 Kbit/s auf und erreicht einen MOS-Wert von 4,0 für die Sprachqualität. Dabei liegt die Signalverzögerung, die durch das Kodieren und Dekodieren entsteht, nur bei 0,625 ms.

LDP Label Distribution Protocol: zur Informierung andere LSRs über die Bindings bei Label-Switching-Verfahren

LDPV Loop Detection via Path Vectors: zur Verhinderung von Routing-Schleifen in MPLS-Netzen, welches sehr schnell reagiert.

LE-ARP LAN Emulation Address Resolution Protocol: zur Ermittlung der ATM-Adresse in einem LANE-Szenario

LEC LANE Client: Client in einem LANE-Szenario
LECS LANE Configuration Server: Funktionssatz in einem ATM-LANE-Netz, der LAN-Endgeräte über die anderen LANE-Dienste informiert.

LER Label Edge Router: Router im Randbereich eines MPLS-Szenarios; bildet Schnittstelle zu angeschlossenen IP-Netzen.

LES LAN Emulation Server: Funktionssatz in einem ATM-LANE-Netz zur Unterstützung von LAN-LAN-Verbindungen

Last-In, First-Out: Serielles Einleseverfahren, welches nach einem anderen Prinzip als das FIFO-Verfahren arbeitet.	*LIFO*
Logical IP Subnets: LIS ist ein einfaches Netzwerk, bei dem jedes Gerät einen direkten Kommunikationspfad zu allen anderen Geräten hat – entspricht funktional einem VLAN.	*LIS*
Logical Link Control: OSI-Protokoll; identisch für alle LAN-Subsysteme im Standard IEEE 802	*LLC*
Liberal Label Retention Mode: ermöglicht eine schnellere Anpassung an Netzwerkänderungen als CLRM bei MPLS	*LLRM*
LANE Network-to-Network Interface: Passt die NNI-Schnittstelle bei ATM auf LANE an.	*L-NNI*
Verhindert beim MPLS-Verfahren, dass Pakete in einer Schleife die Ressourcen aufbrauchen.	*Loop Mitigation*
Verhindert beim MPLS-Verfahren, dass Pakete in eine Schleife vermittelt werden.	*Loop Prevention*
Linear Predictive Code: parametrisches Verfahren	*LPC*
Location_Reject-Nachricht: Alle Gatekeeper, die eine LRQ über den RAS-Kanal erhalten haben und bei denen der nachgefragte Endpunkt nicht registriert ist, antworten mit der Zurückweisung LRJ.	*LRJ*
Location_Request-Nachricht: ein Endpunkt kann bei H.323 hierdurch eine Lokalisierungsanfrage stellen.	*LRQ*
Label Switching: Kernfunktion ist die Trennung in Forwarding- und Control-Komponente	*LS*
Label Switched Path: Verbindung zwischen zwei Label Switch Routern in einem MPLS-Szenario	*LSP*
Label Switch Router: Label-Router in einem MPLS-Szenario; beschränken sich auf eine Paketvermittlung über das LS	*LSR*
Least Upper Bound: nur Maximalwerte von QoS-Dienstparametern finden Berücksichtigung.	*LUB*
LANE User-to-Network-Interface: Passt die UNI-Schnittstelle bei ATM auf LANE an.	*L-UNI*
Message Authentication Code: Code zur Authentifizierung von Nachrichten	*MAC*

Mallory In der Literatur über Verschlüsselung bzw. Kryptographie werden den an Protokollen beteiligten Personen oft Namen gegeben. So kommt jeder Person eine bestimmte Aufgabe zu, die sich meist in dem jeweiligen Namen spiegelt: Mallory ist ein aktiver Gegner (Malicious Active Attacker)

Marking Wird auch als Markierung bezeichnet: Das Setzen des DS-Codepoints in jedem ankommenden Datenpaket anhand definierter Regeln.

MC Mulipoint Controller: bietet Kontrollfunktionen zur Unterstützung von Konferenzen zwischen drei oder mehr Endpunkten. Er führt den Austausch der technischen Möglichkeiten mit jedem Endpunkt durch und sendet jedem Endpunkt die möglichen Übertragungsmodi innerhalb einer Konferenz.

MCR Mean Cell Rate: durchschnittliche Zellrate

MCTD Mean Cell Transfer Delay; durchschnittliche Zellverzögerung: dieser Wert wird durch die Aufenthaltsdauer der Zelle im ATM-Netz bestimmt und ist somit eine statistisch schwankende Größe. Durch diese Zufallsvariable sind N unabhängige Messungen erforderlich, um den Mittelwert bestimmen zu können. Dadurch können Änderungen, bedingt durch Warteschlangen, Vermittlung usw., berücksichtigt werden.

MCU Multipoint Control Unit: Mehrpunktkontrolleinheit, die als Vermittlungsstelle dient, damit Mehrpunktverbindungen mit Unicast-Paketen durchgeführt werden können. Sie sollte aus einem MC und entweder keinem oder mehreren MPs bestehen.

MD2 Message Digest Version 2: Einweg-Hash-Funktion, die einen 128 Bit langen Hash-Wert erzeugt.

MD5 Message Digest Version 5: Einweg-Hash-Funktion **von Ron Rivest**, die einen 128 Bit langen Hash-Wert erzeugt.

Message Digest Die von einer Hash-Funktion zurückgegebene Bitfolge (dasselbe wie ein Hash) Metering Ein Prozess der Messung der aktuellen Verkehrscharakteristik für einen Datenstrom: Metering beeinflusst entsprechend einen Paket-Classifier und -Dropper; außerdem kann Metering zu Accounting-Zwecken eingesetzt werden.

MF-Klassifizierer Multifield Classifier: ist ein Classifier, der die Pakete anhand mehrerer Felder im IP-Header klassifiziert (im Allgemeinem anhand der Sender- und/oder Empfänger-IP-Adresse und der Port-Nummer).

Microflow Eine einzelne Instanz eines Ende-zu-Ende-Datenstroms, gekennzeichnet durch die Sender-IP-Adresse und Port-Nummer, Empfänger-IP-Adresse und Port-Nummer und die IP-Protokoll-ID

Multiplex ID	*MID*
Multipurpose Internet Mail Extensions: standardisierte Übertragung von Grafiken, Sprache und anderen binären Dateien, die kein Text sind. MIME teilt die verschiedenen Dateitypen in verschiedene Haupt- und Untergruppen und integriert in das Versenden von elektronischen Nachrichten bereits die Verschlüsselung von Binärdateien in ASCII-7-Format; wodurch die Verschlüsselung mit UUENCODE bzw. UUDECODE überflüssig wird.	*MIME*
MIME Object Security Services: kryptographisches Nachrichtenformat	*MOSS*
Multipoint Prozessor: ist Teil einer MCU und empfängt Audio-, Video- und Datenströme von Endpunkten in zentralen und hybriden Multipoint-Konferenzen und sendet diese nach erfolgter Verarbeitung zu den Endpunkten zurück. Zur Unterstützung von Terminals, die an einer Konferenz mit unterschiedlich ausgewählten Konferenzmodi teilnehmen, kann ein MP verschiedene Algorithmen und Formatkonvertierungen anbieten.	*MP*
MPOA-Client: kann sowohl ein Router mit einer speziellen ATM-Schnittstelle als auch ein Layer-2-Switch innerhalb eines MPOA-Szenarios sein.	*MPC*
Multi-Protocol Label Switching: Layer-3-Switching zum Etablieren bestimmter Pfade durch das Internet für Dienstgütegarantien, VPN und Traffic Management	*MPLS*
Multi-Pulse Maximum Likelihood Quantization	*MP-MLQ*
Multi-Protocol-over-ATM: Integrationsansatz von IP auf ATM, welcher Unterstandards enthält, um QoS und direkten Verbindungsaufbau über Shortcuts anzubieten.	*MPOA*
MPOA-Router: besitzt Funktionen innerhalb eines MPOA-Szenarios, um Subnetze auf Netzwerkebene (Layer 3) auf ATM-Netze abbilden zu können.	*MPR*
MPOA-Server: ist die logische Komponente eines MPOA-Routers innerhalb eines MPOA-Szenarios.	*MPS*
Maximum Segment Size: maximale Segmentgröße; es handelt sich dabei um die Rahmenlänge, die eine Schicht 4 (z.B. die TCP-Schicht) senden soll, damit das Paket optimal gefüllt werden kann, ohne segmentiert werden zu müssen.	*MSS*
Maximum Transmission Unit: maximal übertragbare Paketgröße, bei der ein Paket nicht fragmentiert werden muss	*MTU*
NBMA Address Resolution Protocol: Erweiterung des ARP-Protokolls um NHRP-Merkmale	*NARP*

NAS Network Access Server: Zugangs-/Anwähl-Server

NBMA Non-Broadcast Multi-Access: Netzwerktechnologie ohne Broadcast-Mechanismen

NCP Network Control Protocol: ein Steuerungsprotokoll für die Netzwerkschicht beim Protokoll PPP

NE Network Element: Netzwerkelement; als Netzelement wird eine Komponente des Internets bezeichnet (ein Host, Router bzw. ein ganzes Teilnetz), die IP-Pakete verarbeiten kann. Dabei wird zwischen nicht QoS-fähigen und QoS-fähigen Netzelementen unterschieden.

NHLFE Next Hop Label Forwarding Entry: eine Komponente einer MPLS-Forwarding-Tabelle, die für das Vermitteln eines MPLS-Pakets verwendet wird.

NHRP Next-Hop Resolution Protocol: Aufbau von Shortcuts über mehrere IP-Subnetze hinweg, um Router-Engpässe vermeiden zu können.

NHS Next-Hop Server: Server in einer NHRP-Umgebung, der die IP-/ATM-Adressenabbildung für die bekannten IP-Knoten, also die Elemente im eigenen LIS, vornimmt.

NLRI Network Layer Reachability Information: Feld bei BGP-4, welches Label-Binding-Informationen von MPLS transportieren kann.

NNI Network-to-Network Interface: ATM-Schnittstelle innerhalb eines ATM-Netzes zwischen verschiedenen ATM-Vermittlungsstellen oder Netzbereichen

NNTP Network News Transfer Protocol: Internetdienst zur Verteilung von Nachrichten

Nonce Pseudo-Zufallszahl, die nur ein einziges Mal benutzt wird.

Nrt-VBR None-real-time Variable Bit Rate: ATM-Dienstklasse, welche die gesicherte Datenübertragung anbietet

OAM Operation and Maintenance: Managementzelle für ATM-Netze zum Transport von Verwaltungs- und Steuerungsinformationen

OFB Output Feedback: eine Betriebsart der Blockchiffren

OSI Open System Interconnection: Referenzmodell für Rechnerarchitekturen auf Basis von sieben Schichten, welches von der Standardisierungsorganisation ISO 1977 entwickelt wurde und noch heute Gültigkeit besitzt.

Open Shortest Path First: dynamisches Routing-Protokoll, welches auf Basis von Hierarchien arbeitet	*OSPF*
Organizationally Unique Identifier	*OUI*
PPTP Access Concentrator	*PAC*
Pulse Amplitude Modulation	*PAM*
Password Authentication Protocol: Methode, damit die Identität des Anwenders festgestellt werden kann.	*PAP*
Ein Fehler im PATH-Zustand, z.B. wegen einer ungültigen Empfängeradresse	*PATH-Error*
RSVP-Nachricht, die Informationen über die Qualität der Verbindungsstrecke und die möglichen Empfänger-Clients sammelt; mit der PATH-Nachricht meldet ein Sender die Datenübertragung an. Diese Nachricht enthält folgende Datenstrukturen: T_{SPEC}, Sender Template, PHOP-Adresse, AD_{SPEC}.	*PATH-Message*
Private Branch Exchange: Bezeichnung für Telefonnebenstellenanlage	*PBX*
Policy Certification Authority: beziehen ihre Richtlinien über die Vergabe von Zertifikaten von der übergeordneten Instanz IPRA und vergeben ihrerseits Zertifikate an die CAs.	*PCA*
Pulse Code Modulation	*PCM*
Peak Cell Rate (Spitzenzellrate): Zulässige Spitzenzellrate, die nicht überschritten werden darf, es sei denn, es wird einer Überbuchung und damit Übertretung des Traffic Contract stattgegeben.	*PCR*
Plesiochrone Digitale Hierarchie: digitales Multiplexverfahren, welches auf verschiedenen Taktzuständen basiert, die durch so genannten Stopfbits kompensiert werden.	*PDH*
Policy Decision Point: Policy-Server/Interpreter. Es werden Policies in spezielle Konfigurationsinformationen für die Netzkomponenten umgesetzt, die dann auf die Komponenten heruntergeladen werden. Anschließend lässt man einen Policy Client Request entweder zu oder lehnt ihn ab. Zusätzlich könnte er eine Anfrage mittels Zugriff auf einen zentralen Verzeichnisdienst überprüfen.	*PDP*
Packet Data Unit: dient der Kommunikation unter gleichberechtigten Protokollschichten und wird deshalb auch als Kommunikationsprotokoll bezeichnet.	*PDU*
Policy Enforcement Point: Policy Client, der für eine Session oder eine Applikation eine bestimmte Dienstgüte beantragen kann.	*PEP*

PESQ Perceptual Evaluation of Speech Quality; Messverfahren zur Ermittlung der Sprachqualität in VoIP-Netzen (Nachfolger des PSQM-Verfahrens)

PFS Perfect Forwarding Secrety: vorwärts gerichtete Sicherheit bei IPsec, bei der alle Daten geschützt übertragen werden, unter anderem auch die Identität der Teilnehmer.

PGP Pretty Good Privacy: dient der Chiffrierung und Authentifizierung von E-Mail und Daten.

PHB Per-Hop Behavior: Ein wohldefinierter Satz von Verhandlungsregeln. Anhand dieser Regeln muss ein DS-fähiger Knoten ein Behavior Aggregate weiterleiten.

PHOP-Adresse IP-Adresse des nächsten Upstream-RSVP-Knoten. Jeder RSVP-Knoten trägt hier beim Weiterleiten der PATH-Message seine Adresse ein. Beim Empfang der PATH-Message wird diese Adresse im PSB gespeichert.

PI Protocol Identifier

PIM Protocol Independent Multicast

PIN Personal Identifier Number

PK Preshared Keys: Schlüssel, die nicht erzeugt werden müssen, sondern schon vorher bekannt sind.

PKCS Public Key Cryptography Standards: ist eine Sammlung von Standards für Public-Key-Verfahren.

PKI Public Key Infrastructure: Sichere Kommunikation und automatisierter Schlüsselaustausch mit globaler Reichweite durch die Errichtung und Nutzung von öffentlichen Infrastrukturen.

PLCP Physical Layer Convergence Procedure: Übertragungsanpassung auf PDH-Leitungen

PM Physical Medium: Die PM-Teilschicht bei ATM legt die Eigenschaften der Übertragungsmedien fest. Das heißt, hier werden Steckertyp, Kabelcharakteristika und Bitkodierung spezifiziert. Dabei können prinzipiell zwei Arten der Übertragung unterschieden werden: die zellbasierte Übertragung sowie die Nutzung von bestehenden Netzarchitekturen.

P-NNI Private Network-to-Network-Interface: dynamische Signalisierung zwischen ATM-Switches, die alternativ zu Layer-3-Routing-Protokollen in der IP-Umgebung verwendet werden kann.

PNS PPTP Network Server
Policing Das Verwerfen bestimmter Pakete mit Hilfe eines Droppers

Point-of-Presence: Bezeichnung des Hauptkommunikationspunktes eines Internet Service Provider	*POP*
Packet-over-SONet: IP-Pakete werden über die SDH-Schicht übertragen, ohne weitere Transportprotokolle wie ATM zu berücksichtigen.	*PoS*
Partial Packet Discard: zusammengehörige Zellen werden auf der Anpassungsschicht bei ATM aufgrund von Überlast verworfen	*PPD*
Point-to-Point Protocol: Standard nach RFC-1661, um eine Punkt-zu-Punkt-Verbindung im WAN herzustellen.	*PPP*
Point-to-Point Tunneling Protocol: ist nach RFC-2637 von Microsoft, Ascend, 3Com und U.S. Robotics ein Layer-2-Tunneling-Protokoll und setzt einen Client voraus, der PPP-Pakete beherrscht.	*PPTP*
Priority Queueing: strikte Priorisierung auf der Schicht 3. Dabei können Aggregationen von Datenströmen einer der zur Verfügung stehenden Queues zugeordnet werden.	*PQ*
Kann man auch als Vormarkierung bezeichnet: das Setzen des DS-Codepoints der weitergeleiteten Datenpakete vor dem Verlassen einer DS-Domäne.	*Pre-marking*
Protection Suite: Protokollfamilie zum Schutz von Daten	*PS*
Path State Block: beinhaltet die relevanten Daten aller bei einer RSVP-Session angemeldeten Sender	*PSB*
Perceptive Speech Quality Measurement: Messverfahren zur Ermittlung der Sprachqualität in Telefonnetzen	*PSQM*
Public Switched Telephone Network: öffentliches Telefonnetzwerk, welches unterschiedliche Technologien beinhaltet.	*PSTN*
Payload Type: unterscheidet Nutzinfo und Netzinfo	*PT*
Payload Type Identifier: zur Identifikation des Formats der Nutzlast und der damit angewandten Kodierungs- und Kompressionsverfahren beim RTP	*PTI*
Permanent Virtual Circuit: permanent geschaltete ATM-Verbindung	*PVC*
ITU-Empfehlung: Network layer – ISDN user-network interface layer 3 – General aspects	*Q.930*
ITU-Empfehlung: ISDN user-network interface layer 3 specification for basic call control	*Q.931*
ITU-Empfehlung: Switching and signalling	*Q.SIG*

QME Quick Mode Exchange: Austauschmodus von IPsec in der Phase II

QoS Quality-of-Service; garantierte Dienstgüte, die das Netz der Applikation oder dem Anwender zur Verfügung stellt.

QoS-Dienst Als QoS-Service wird eine Sammlung von Regeln, die ein Netzelement zur QoS-Realisierung ausüben soll, bezeichnet. Eine Service-Definition beinhaltet Spezifikationen der notwendigen Funktionen, beschreibt Module, die dafür implementiert werden müssen, sowie Nachrichten, die zur Realisierung dieses Dienstes zwischen einzelnen Netzementen ausgetauscht werden.

RA Registration Authority: nach einer Identitätsprüfung wird dem Anwender ein eindeutiger Name zugewiesen, was durch die RA im Zusammenspiel mit einer Certification Authority (CA) vorgenommen wird.

RADIUS Remote Access Dial-In User Service: Authentifizierungs- und Autorisierungsprotokoll für Zugangsserver (Remote Access Server)

RAS Remote Access Server: Zugangsserver

RAS Registration Admission Status: Registrierungs-Statusabfrage bei VoIP und dem Standard H.225.0

RC2/4 RC2 und RC4 sind symmetrische Verschlüsselungsverfahren. RC2 gehört zu den Blockchiffren und RC4 zu den Stromchiffren.

RC5 Symmetrisches Verschlüsselungsverfahren, welches zu den Blockchiffren gehört, aber mit variablen Blockgrößen arbeitet.

RCF Registration_Confirm-Nachricht: Registrierungsbestätigung der Endpunkt-Registrierungsanfrage bei H.323

RED Random Early Discard: Verfahren, mit dem man den Durchsatz unter Überlast minimieren kann, indem Pakete verworfen werden.

RESV-CONFIRM Ist in der RESV-Nachricht das RESV-CONFIRM-Bit gesetzt, so wird die RESV-CONFIRM-Nachricht als Antwort auf eine erfolgreiche Reservierung gesendet. Es ist zu beachten, dass der Empfang dieser Nachricht keine fehlerfreie Reservierung garantiert, sondern vielmehr auf eine hohe Wahrscheinlichkeit, dass eine Reservierung erfolgreich zustande gekommen ist, hinweist.

RESV-Error Meldung einer fehlgeschlagenen Reservierung, z.B. wegen Ablehnung der Anforderung durch eine Admission-Control-Einheit

RESV-Message RSVP-Nachricht, die Informationen über die Ressourcen enthält: unterzieht die vom Sender unterstützten Dienstklassen einer kritischen lokalen Prüfung. Mit der RESV-Nachricht leitet der Empfänger eine Reservierung ein. Sie bein-

haltet die Parameter der vom Netz geforderten Dienstgüte in Form einer $FLOW_{SPEC}$ sowie eine $FILTER_{SPEC}$.	
Request-for-Comments: Spezifikationen der IETF, die unterschiedlich gewichtet sind, um Drafts und Standards voneinander abzugrenzen.	*RFC*
Routing Information Base: zweite Informationsdatenbank im ARIS-Verfahren von IBM	*RIB*
Reject-Nachricht	*RJ*
Routing Over Large Clouds: Arbeitsgruppe der IETF, die u.a. das NHRP entwickelt hat.	*ROLC*
Request-Nachricht	*RQ*
Registration_Reject-Nachricht: Ablehnung der Endpunktregistrierungsanfrage RRQ bei H.323	*RRJ*
Record Route Object: hilft bei der Wegefindung durch ein MPLS-Netz und vermeidet Routing-Schleifen	*RRO*
Registration_Request-Nachricht: Endpunktregistrierungsanfrage bei H.323 an die Transportadresse des RAS-Kanals eines Gatekeepers	*RRQ*
Rivest, Adelman, Shamir; Public-Key-Verfahren, das auch für digitale Signaturen geeignet ist; benannt nach seinen Erfindern.	*RSA*
Reservation State Block; beinhaltet die Daten aller zustande gekommener bzw. fehlgeschlagener, aber noch nicht aufgelöster RSVP-Reservierungen.	*RSB*
Service Request Spezification: R_{SPEC} beschreibt QoS-Parameter, deren Einhaltung vom Netzelement erwartet wird.	R_{SPEC}
Resource Reservation Protocol: Signalisierungsprotokoll im Internet, um Ressourcen zwischen Sender und Empfänger auszuhandeln und festzulegen,	*RSVP*
Der RSVP-Sender ist eine Instanz, die Nutzdaten an den RSVP-Empfänger sendet. Beim Reservierungsvorgang kann der RSVP-Sender Signalisierungspakete (aber keine Nutzdaten) vom RSVP-Empfänger empfangen. Es können gleichzeitig mehrere RSVP-Sender mit demselben RSVP-Empfänger kommunizieren.	*RSVP-Sender*
Resource Reservation Protocol-Traffic Engineering: Erweiterung des RSVP um Traffic Engineering für MPLS; ist zuständig für die Zuweisung der Labels.	*RSVP-TE*
Real-Time Transport Control Protocol: sorgt für die Kontrolle der Signallaufzeiten und misst die Paketverlustraten. Dadurch kann die Übertragung an die jeweiligen Verbindungseigenschaften angepasst werden.	*RTCP*

RTP Real-Time Transport Protocol: arbeitet auf der Grundlage des Internetprotokolls und hat die Aufgabe, Datenströme in Echtzeit (z.B. Audio oder Video) zu transportieren. In erster Linie wurde RTP für Multicast von Daten entwickelt, eine Verwendung für Unicast ist jedoch auch möglich. RTP definiert keine Mechanismen für eine garantierte Dienstgüte und für die Verwaltung von Bandbreite.

RTT Round Trip Time: die Zeit, die ein Paket zurücklegt, um vom Sender zum Empfänger und wieder zurück zu gelangen; wird für Verzögerungsmessungen eingesetzt.

Rt-VBR Real-time Variable Bit Rate: ATM-Dienstklasse, die sich ähnlich zu CBR verhält, wobei geringfügige Veränderungen der Bandbreite (z.B. Video) sowie kleinere Zellverluste toleriert werden, da keine Sicherung der Übertragung gewährleistet wird.

SA Security Association: Vertrag über Sicherheitsparameter einer Kommunikationsbeziehung wie Verschlüsselungs- und Authentifizierungsverfahren, Schlüsselmaterial, -gültigkeitsdauer etc.

SAAL Signalling ATM Adaptation Layer: Signalisierungsanpassungsschicht von ATM auf AAL-Typ5

SAFER Secure and Fast Encryption Routine: ist von James Massey 1994 für die Cylink Corporation entworfen worden. Die Benutzung kann kostenfrei erfolgen, da es keine Copyright-Rechte oder Patente gibt. Der Algorithmus gehört zur Familie der Blockchiffren mit einer Blockgröße von 64 Bit.

SAFI Subsequent Address Family Identifier: Feld bei BGP-4, welches den Inhalt des NLRI-Felds anzeigt und für den Transport von Label-Binding-Informationen bei MPLS auf den Wert 4 gesetzt wird.

SAP Session Announcement Protocol: dient zur Veröffentlichung von Multicast-Adressen bei der Teilnahme an Multimedia-Konferenzen beim Session Initiation Protocol.

SAR Segmentation and Reassembly: ist das Bindeglied zur ATM-Schicht und tauscht mit dieser auf der Benutzerebene 48-Byte-Nutzdaten aus. Diese werden in Abhängigkeit der Dienstklasse bearbeitet und an die jeweilige Konvergenzschicht weitergeleitet.

SBM Subnet Bandwidth Manager: Erkennen von Ressourcen in Layer-2-LANs und Mapping von Layer-2- und Layer-3-QoS-Mechanismen

Sustainable Cell Rate (mittlere Zellrate): Die zulässige Zellrate, die für die Verbindung dauerhaft zur Verfügung gestellt werden kann. Dabei müssen alle definierten Verkehrsparameter eingehalten werden.	*SCR*
Verwürfelung von Daten (Informationen), um diese gegenüber Übertragungsfehlern oder Abhörmöglichkeiten abzusichern.	*Scrambling*
Synchronous Digital Hierarchy: digitale Netzhierarchie im WAN; siehe SONET.	*SDH*
Session Description Protocol: dient der Beschreibung der Nutzlastformate innerhalb von Multimedia-Sitzungen beim Session Initiation Protocol	*SDP*
Die Selektoren bilden den IP-Verkehr auf eine IPsec-Anwendungsstrategie ab. Sie identifizieren bestimmte Anteile des Verkehrs als fein- oder grobkörnig; Selektoren sind: IP-Zieladresse, IP-Ursprungsadresse, Namen, übergeordnete Protokollschichten, Ports von Ziel oder Ursprung, eine Stufe von Datensensitivität (wenn IPsec-Systeme auch die Sicherheit für den Datenfluss unterstützen).	*Selektoren*
Hat das Format einer $\text{FILTER}_{\text{SPEC}}$: diese Nachricht wird der PATH-Message beigefügt. An ihr werden unterschiedliche Sender innerhalb einer RSVP-Sitzung identifiziert.	$\textit{SENDER}_{\text{TEMPLATE}}$
Secure Hash Algorithm: von der NSA entwickelter Hash-Algorithmus mit längerer Ausgabe als MD5	*SHA*
Eine DiffServ-Einheit, die das Shaping für einen bestimmten Datenstrom vornimmt.	*Shaper*
Ein Prozess der Verzögerung der Pakete, damit sie dem vereinbarten Profil entsprechen.	*Shaping*
Secure Hash Standard: enthält u.a. den SHA.	*SHS*
Secure Hypertext Transfer Protocol: in der Spezifikation RFC-2660 beschrieben; ein Kommunikationsprotokoll, welches für den sicheren Einsatz mittels SSL in Verbindung mit HTTP entwickelt wurde.	*S-HTTP*
Session Initiation Protocol: ist ein Signalisierungsprotokoll zum Einrichten, Modifizieren und Beenden von Multimedia-Konferenzen, IP-Telefonie-Verbindungen und ähnlichen Anwendungen.	*SIP*
Simple Key Management over IP: Key-Management-Protokoll von Sun Microsystems	*SKIP*

SLA Service Level Agreement: eine Vereinbarung (Service Contract) zwischen einem Dienstabnehmer, auch Kunde (Customer) genannt, und dem Service Provider, der den Forwarding-Service, den der Kunde bekommt, spezifiziert. Dabei kann als Dienstabnehmer eine andere DS-Domäne oder eine Nicht-DS-Domäne (Source Domain) auftreten.

SLIP Serial Line Protocol: Standard für die Übertragung von TCP/IP-Paketen über serielle Schnittstellen

SLS Service Level Specification: um eine SLA umzusetzen, werden Dienstparameter für entsprechende Datenströme in einer SLS zusammengefasst.

S/MIME Secure/Multipurpose Internet Mail Extensions

SMP Source Merge Points : ein Punkt, an dem Verkehrsströme von unterschiedlichen Quellen zusammenlaufen.

SMTP Simple Mail Transfer Protocol: Internetstandard, welcher festlegt wie E-Mails im Internet verteilt/transportiert werden.

SN Sequence Number: ist für die Erfassung der Zellverluste (Cell Loss) und Zellintegrität (Cell Integrity) bei O.191-Messungen verantwortlich.

SNA Systems Network Architecture: SNA ist die IBM-Netzarchitektur und ihrer Bauart nach ein hierarchisch orientiertes Netz zur Steuerung von Terminals und zur Unterstützung des freizügigen Zugriffs von Terminals auf Anwendungen im Host.

SNAP Sub-Network Access Protocol: IP-Protokoll, welches zwischen der Netzwerkinstanz eines Subnetzes und der Netzwerkinstanz im Endsystem arbeitet.

SONET Synchronous Optical NETwork: der ITU-SDH-Standard unterscheidet zwei SDH-Derivate – das europäische ETSI-SDH (155,52 Mbit/s) und das nordamerikanische ANSI-SONET (51,84 Mbit/s).

Source Domain Quelldomäne: eine Domäne, die Verkehrsquellen enthält. Diese Quellen nutzen einen vereinbarten DS-Dienst.

SPD Security Policy Database: Datenbank aller Sicherheitsanwendungsstrategien bei IPsec

SPE Sprachpaketempfänger: Verzögerung der empfangenen Sprachpakete, um Laufzeitunterschiede auszugleichen sowie digitale Signale in analoge Werte zu dekodieren.

SPI Security Parameters Index: ein 32-Bit-Wert, der die zur Zieladresse des IPsec-Datenpakets gehörige Security Association (SA) identifiziert.

Switched Path Label: Kennzeichnung der Datenpakete im ARIS-Verfahren von IBM	*SPL*
Single-Point-of-Failure: Schwachstelle eines Netzwerks, welches keine oder ungenügende Redundanz besitzt.	*SPOF*
Service Provisioning Policy: eine Policy, die die Konfiguration der Traffic Conditioners in den DS-Grenzknoten beschreibt. Sie enthält auch eine Beschreibung, wie der Datenverkehr beim Eintreten in eine DiffServ-Wolke gemappt wird.	*SPP*
Strict Priority Queueing: Priorisierung der Datenpakete am Ausgangsspeicher eines Routers, um sie nicht nach dem FIFO-Verfahren abzuarbeiten.	*SPQ*
Sprachpaketsender: In diesem wird mit Hilfe des Kodieres die Sprache mit den entsprechenden Verfahren digitalisiert und gegebenenfalls komprimiert sowie in Pausen- und Sprachblöcke aufgeteilt. Anschließend werden Pakete gebildet.	*SPS*
Synchronous Residual Time Stamp: Methode bei ATM zur Übertragung einer Taktfrequenz	*SRTS*
Source Service Access Point	*SSAP*
Service Specific Convergence Sublayer: anwendungsspezifische Konvergenzteilschicht (AAL-Typ 2, 3/4 und 5)	*SSCS*
Secure Shell Protocol: stellt einen sicheren Ersatz für die Berkley-Utilities dar und wird für Telnet-Verbindungen eingesetzt.	*SSH*
Secure Socket Layer: ermöglicht sichere Verbindungen über das Internet für beliebige Nutzer. Es setzt auf einem zuverlässigen Transportprotokoll wie TCP auf.	*SSL*
Synchronous Transport Module, Ebene 1: 155,52-Mbit/s-SDH-Rahmen (Europa)	*STM-1*
Synchronous Transport Module, Ebene 16: 2,45-Gbit/s-SDH-Rahmen (Europa)	*STM-16*
Synchronous Transport Module, Ebene 256: 39,8-Gbit/s-SDH-Rahmen (Europa)	*STM-256*
Synchronous Transport Module, Ebene 4: 622,08-Mbit/s-SDH-Rahmen (Europa)	*STM-4*
Synchronous Transport Module, Ebene 64: 9,95-Gbit/s-SDH-Rahmen (Europa)	*STM-64*

SVC Switched Virtual Circuit: signalisierte ATM-Verbindung

T.120 ITU-Empfehlung: Data protocols for multimedia conferencing

TAN Transaktionsnummer

TC Traffic Conditioning: ein Überwachungsprozess, der die Einhaltung der TCAs überwacht – Traffic Conditioning kann Metering, Marking, Shaping und Policing beinhalten.

TC Transmission Convergence: Generierung eines kontinuierlichen ATM-Zellstroms bei gleichzeitiger Entkopplung der Zellrate durch Idle-Zellen von der zur Verfügung stehenden Bandbreite

TCA Traffic Conditioning Agreement: ein Abkommen zwischen einem Diensterbringer und Dienstnutzer bzgl. Klassifizierungsregeln

TCP Transmission Control Protocol: Transportschicht 4 von OSI für den verbindungsorientierten Datenaustausch mit Fehlererkennung

TCS Traffic Conditioning Specification: Hier sind die Abkommen der TCA in Klassifizierungsregeln und Verkehrsregeln für den Kundenverkehr festgehalten.

TCSB Traffic Control State Block: Ressourceninformationen bezüglich einer Schnittstelle werden hier bei RSVP abgespeichert.

TDP Tag Distribution Protocol: Zum Informieren der anderen TSRs über die Tag Bindings wird beim Tag Switching der Transport über Routing-Protokolle bevorzugt. Für den Fall, dass dies aufgrund des verwendeten Routing-Protokolls nicht möglich ist, hat Cisco Systems das TDP entwickelt.

TE Traffic Engineering: Beeinflussung des Verkehrs eines Netzes durch gezielte Verkehrssteuerung

Teardown $\mathrm{PATH}_{\mathrm{TEAR}}$ und $\mathrm{RESV}_{\mathrm{TEAR}}$. Eine Teardown-Nachricht löscht den PSB bzw. RSB in den Netzknoten entlang des Datenpfades

TFIB Tag Forwarding Information Base: Komponente des Tag-Switching von Cisco, welches allgemein den Funktionen der LS-Forwarding Tabelle entspricht.

TIPHON Telecommunications and Internet Protocol Harmonization Over Networks: ETSI-Projekt zur Schaffung einer offenen Konzeption mit der Konzentration auf Spezifikationen, um eine weltweite Akzeptanz und Anwendung der IP-Telefonie zu erreichen.

TLF Type Length Value: Teil einer LDP-Nachricht, welche die LDP-Funktionalität bei MPLS beinhaltet.

Transport Layer Security: ist in Version 1.0 identisch mit SSL Version 3.0.	*TLS*
Telecommunication Objective Speech Quality Assessment; Messverfahren zur Ermittlung der Sprachqualität in VoIP-Netzen; entwickelt von der Deutschen Telekom T-Nova.	*TOSQA*
Traffic Profile: Wird auch als Verkehrsprofil bezeichnet und beschreibt die Datenstromeigenschaften für einen Microflow bzw. Datenfluss-Aggregat. Ein Verkehrsprofil kann z.B. mit Hilfe des Token Bucket Filter beschrieben werden.	*TP*
Eine Einheit im DS-Knoten, die das Traffic Conditioning durchführt. Traffic Conditioning wird in aller Regel in DS-Boundary-Nodes angewandt. Ein Traffic Conditioner führt das Re-Marking bzw. Dropping oder Shaping des behandelten Datenstroms durch, damit der Ausgangsverkehr des TC dem gewünschten Traffic Profile entspricht.	*Traffic Conditioner*
Transport Layer Service Access Point: durch TSAP ist das Multiplexen mehrerer Kanäle über dieselbe Netzwerkadresse möglich.	*TSAP*
Traffic Stream: eine zusammengesetzte Menge von Microflows, die einen Link passieren. Es wird dafür auch der Begriff Datenstrom verwendet.	*TS*
Time Stamp: misst die Zellverzögerung (Cell Delay), den Zell-Jitter (Cell Jitter) und die Zellverteilung (Cell Distribution) bei O.191-Messungen.	*TS*
Eine Datenstruktur, die den Datenverkehr eines RSVP-Senders beschreibt. Sie beinhaltet Parameter wie mittlere Datenrate, maximale Datenrate, maximale Burstgröße etc. Diese Datenstruktur wird vom RSVP-Prozess an den Routing-Prozess bzw. an den Anwendungsprozess unverarbeitet weitergereicht.	T_{SPEC}
Tag Switching Router: Komponente des Tag-Switching von Cisco, welches allgemein den Funktionen einer Label Switch Routers entspricht.	*TSR*
Time-to-Live: Feld im IP/RSVP-Header zur Begrenzung der Lebensdauer von ziellosen Paketen zur Vermeidung von Netzschleifen	*TTL*
Tributary Unit Group: SDH-Multiplexschema	*TUG*
Unspecified Bit Rate: Dienstklasse im ATM für einfachen LAN-Datenverkehr, welche ausschließlich Best-effort anbietet.	*UBR*
Unregister_Confirm-Nachricht	*UCF*
User Datagram Protocol; Transportschicht 4 von OSI für den verbindungslosen Datenaustausch	*UDP*
Unit Interval: Einheit einer Jitter-Messung	*UI*

UI Unnumbered Information

UNI User-to-Network-Interface: ATM-Schnittstelle vom ATM-Client zur ATM-Vermittlungsstelle/-Switch

UP DS Domain Upstream DiffServ Domain: eine DS-Domäne, aus der Daten in die aktuell betrachtete Domäne einfließen.

URQ Unregister_Request-Nachricht

VC Virtual Connection: virtuelle ATM-Verbidnung, die Pfade und Kanäle beinhaltet.

VCC Virtual Circuit Connection: virtuelle ATM-Kanal-Verbindung

VCI Virtual Channel Identifier: Kanalidentifizierung bei ATM

VebrKrG Verbraucherkreditgesetz
Verkehrsmeter Eine DiffServ-Einheit im DS-Knoten, die das Metering für einen bestimmten Datenstrom vornimmt.

VLAN Virtual Local Area Network: logische Aufteilung eines Netzes, entkoppelt von der physikalischen Struktur

VoCoder Voice und Codierer: Alle Vocoder analysieren senderseitig die Sprache nach Grundfrequenz und Lautbildung und übertragen diese durch Parameter. Auf der Empfangsseite wird mit den Parametern das elektrisches Modell zur Spracherzeugung abgeglichen und so die Sprache wiedergewonnen.

VoIP Voice-over-IP: Sprache über das Internetprotokoll, das sowohl über den Backbone eines Netzes geschehen kann, als auch im lokalen Bereich (à IP-Telefonie).

VPC Virtual Path Connection: virtuelle ATM-Pfad-Verbindung

VPI Virtual Path Identifier: Pfadidentifizierung bei ATM

VPN **Virtual Private Network: allgemeiner Begriff zur Beschreibung der Verbindung von Unternehmensnetzen über öffentliche Netze**

WAN Wide Area Network: Weitverkehrsnetz

WFQ Weighted Fair Queueing: Datenflüssen wird eine bestimmte Priorität zugeteilt, wobei entgegen dem SPQ-Verfahren jede Priorität eine garantierte Mindestbandbreite auf der Schnittstelle besitzt.

WRR Weighted Round Robin: es wird der gesamte für das Queueing vorgesehene Speicherplatz in mehrere Warteschlangen eingeteilt. Dabei wird für jede Queue eine Gewichtung festgelegt. Sind in jeder Queue ausreichend Datenpakete vorhanden, so wird eine Anzahl an Bytes proportional zur Gewichtung aus der Queue verarbeitet.

World Wide Web: grafische Benutzerschnittstelle des Internets; kann als eigener Dienst bezeichnet werden.	*WWW*
Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)	*X.209*
The Directory: Overview of Concepts, Models and Services: enthält Konzepte zu Verzeichnissen und Directory Services.	*X.500*
The Directory: Authentication Framework: Zertifikatstandard der ITU, welcher innerhalb des Directory-Modells nach X.500 angeordnet ist.	*X.509*
eXtensible Markup Language	*XML*

A.2.1 Weiterführende Informationen

Cisco Systems	http://www.cisco.com	***Herstellerauswahl***
Enterasys Networks	http://www.enterasys.com	
Extreme Networks	http://www.extremenetworks.com	
Juniper Networks	http://www.juniper.net	
3Com	http://www.3com.com	
Unisphere	http://www.unisphere.com	
Marconi	http://www.marconi.com	
Nortel Networks	http://www.nortelnetworks.com	
Lucent Technologies	http://www.lucent.com	
Toshiba	http://www.toshiba.com	
Checkpoint	http://www.checkpoint.com	
Bull	http://www.bull.com	
Microsoft	http://www.microsoft.com	
DECOIT e.K.	http://www.decoit.de	***Whitepapers***
Nevarsa GmbH	http://www.nevarsa.com	
Ohio State University	http://www.cis.ohio-state.edu	
NCP	http://www.ncp.de	
IETF	http://www.ietf.org	***Standardisierungs-***
IEEE	http://www.ieee.org	***gremien***
ATM Forum	http://www.atmforum.com	
ITU	http://www.itu.org	
MPLS Forum	http://www.mplsforum.org	
QoS Forum	http://www.qosforum.com	
ANSI	http://www.ansi.org	

Index

I

J

K

L

X

Z