Untersuchung der Protokolle xDSL und priorisiertes IP zur Übertragung multimedialer Datenströme

Diplomarbeit

Jeannot Petters

Studiengang: Informatik, HTWK-Leipzig

Betreuer: Prof. Dr. Hänßgen

Ort, Datum: Leipzig, 14. April 2003

INHALTSVERZEICHNIS

1. EINLEITUNG

In den letzten Jahren wuchs nicht nur die Anzahl der Netz-Zugänge sondern auch die Bandbreite der einzelnen Zugänge. Entsprechend steigt auch die Nachfrage an Multimedia-Diensten und Möglichkeiten von Echtzeitübertragungen, z.B. Video-Konferenzen oder einfach Video-Telefonie über das Internet. In den letzten zwei Jahren stieg auch die Anzahl der DSL-Anschlüsse, so stehen den Endanwendern nicht nur ISDN, sondern auch DSL-Anschlüsse in großer Zahl zur Verfügung, die sich zur Übertragung von multimedialen Datenströmen eignen. Auch die sehr leistungsfähigen Prozessoren der 5. und 6. Generation tragen maßgeblich zur Verbreitung multimedialer Anwendungen und Dienste bei. Die modernen Prozessoren unterstützen multimediale Anwendungen nicht nur durch ihre ständig steigenden Taktraten, sondern auch durch spezielle Befehle, die z.B. Codierung und Decodierung von digitalen Video bzw. Audiodaten um ein vielfaches beschleunigen. Voraussetzung für die Nutzung von diesen Befehlen ist, dass beim Erstellen der Anwendung ein entsprechender Compiler verwendet wurde. Dies ist z.B. der von Intel angebotene C++ Compiler für Windows und Linux. Wenn man multimediale Dienste anbieten oder nutzen möchte reicht also nicht nur eine entsprechend ausgebaute Infrastruktur, sondern es bedarf auch eines gut abgestimmten Systems aus Hard- und Software. Dies wird unter anderem in den nächsten Kapiteln näher erläutert.

Diese Arbeit zeigt, welche Möglichkeiten diese Netzwerktechnologien bieten und welche Vor- bzw. Nachteile es gibt. In den nachfolgenden Kapiteln werden Grundlagen bezüglich Multimedia und Netzwerktechnik behandelt. Im zweiten Kapitel werden die Grundlagen der verschiedenen Netzwerktechnologien DSL, TCP/IP und QoS dargelegt. Das dritte Kapitel beschreibt die Grundlagen für das Verständnis der heute eingesetzten Multimediatechnologien. In Kapitel 4 werden verschiedene Szenarien beschrieben, in denen Quality of Service in IP-Netzen genutzt werden kann. Kapitel 5 erläutert unterschiedliche Verfahren und Werkzeuge, die zur Messung der Qualitätsmerkmale von Netzwerkverbindungen verwendet werden kann. Weiterhin werden in dem Kapitel Messungen beschrieben, die z.B. bei einer DSL Verbindungen gemessen wurden und auch verschiedene Effekte erläutert, die bei diesen Messungen aufgetreten sind.

2. NETZWERKGRUNDLAGEN

Das zweite Kapitel legt die Grundlagen für das Verständnis der Netzwerktechnik. Im ersten Abschnitt folgt eine Beschreibung des OSI-Modells, danach werden die verwendeten Protokolle im Detail erklärt. Abschließend erfolgt eine Erläuterung der Möglichkeit einer Qualitätssicherung für IP basierte Netze.

2.1 Das OSI-Referenzmodell

Es bedurfte in den späten 70er Jahren eines Standards, der die Regeln für den Datenverkehr in Computernetzen festlegte, da zunehmend heterogene Systeme in den Netzten zum Einsatz kamen. Dieser Standard wurde von der International Standardisation Organisation(ISO) definiert. Das OSI-Referenzmodell dient als Beschreibung der Datenkommunikation zwischen Computern in Netzwerken. Das OSI-Referenzmodell wird auch als kurz OSI-Modell oder als 7-Schichten-Modell bezeichnet. Es ist lediglich als abstraktes Modell zu verstehen, das es nicht die direkte Implementierung vorgibt. Das OSI-Modell behandelt die ganze Breite von der physikalischen Datenübertragung bis zu Diensten, die durch Anwendungsprogramme benötigt werden. Dazu wird es in 7 Schichten eingeteilt. Jede dieser Schichten übernimmt eine bestimmte Aufgabe, wobei jede Schicht der über ihr liegenden ihre Funktion zur Verfügung stellt und die Funktion der darunter liegende Schicht nutzt. Abbildung 2.1 verdeutlicht dieses normierte Referenzmodell für Kommunikationsprotokolle in offenen Systemen.

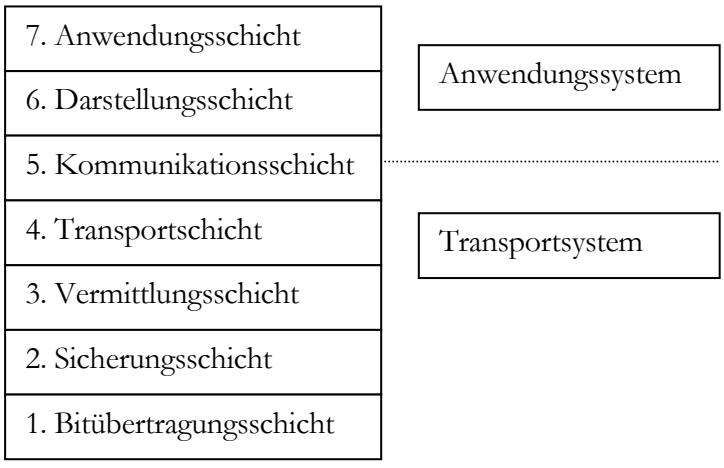

Abbildung 2.1: Schematische Darstellung der Schichten des OSI-Modell

Schicht 1 - Bitübertragungsschicht

Diese Schicht ist für die physikalische Übertragung der Daten verantwortlich. Hier werden die Eigenschaften der Hardware definiert, wie z.B. elektrische Spezifikationen, Signalpegel, Spannungswerte. Ebenfalls wird festgelegt, abhängig vom Übertragungsmedium, ob serielle oder parallele Datenübertragung stattfindet.

Schicht 2 - Sicherungsschicht

Diese Schicht ist für die Sicherung der zu übertragenden Daten zuständig. Die Daten werden dafür in Frames[1] verpackt und entsprechende Prüfsummen(in Form von Korrekturbits) hinzugefügt. In dieser Schicht wird auch für die richtige Adressierung zwischen den beteiligten Stationen gesorgt. Der Erfolg der Übertragung wird durch Quittungen überprüft.

Schicht 3 - Vermittlungsschicht

Die Funktion dieser Schicht ist hauptsächlich für die Kommunikation zwischen 2 Hosts in einem Netzwerk zuständig. Des Weiteren ist sie für das Routing[2] der Datenpakete zuständig.

Schicht 4 - Transportschicht

Die Aufgabe dieser Schicht besteht darin, den Datentransport zwischen 2 Anwendungsprozessen zu regeln. Hier wird auch die Aufteilung der Verbindung für mehrere Benutzer geregelt.

Schicht 5 - Kommunikationsschicht

Dies ist die unterste Schicht des Anwendungssystems und damit verantwortlich für den logischen Verbindungsauf- und abbau. Sie sichert bei Verbindungsabbruch den Neuaufbau der Verbindung und ordnet den Datenpakten die entsprechenden Anwendungen zu.

[1] Datenblöcke, Datenpakete

[2] Wegeauswahlverfahren für die Daten

Schicht 6 - Darstellungsschicht

Diese Schicht sorgt für die einheitliche Umsetzung der Daten auf verschiedenen Systemen. Die Daten werden hier in bestimmte Standardformate(z.B. ACSII) konvertiert.

Schicht 7 - Anwendungsschicht

Die letzte Schicht des OSI-Modells stellt eine Schnittstelle für Netzapplikationen dar, z.B. Telnet oder ftp.

In konkreten Implementierungen ist es nicht unüblich, dass Schichten wegfallen oder Schichten zusammengelegt werden. Das liegt unter anderem daran, dass benachbarte Schichten oft ähnliche Funktionen erfüllen und somit eine Verschmelzung der Schichten nahe liegt.

2.2 Protokolle

Dieses Kapitel erklärt den Aufbau und die Funktion der verwendeten Protokolle.

2.1.1 TCP/IP

Die Architektur des TCP/IP-Referenzmodells wird, im Gegensatz zum OSI-Referenzmodell, das 7 Schichten umfasst, in 4 Schichten unterteilt. Das TCP/IP-Modell unterscheidet nicht zwischen Bitübertragungs- und Sicherungsschicht(siehe [1]). Die unterste Schicht des TCP/IP-Modells wird als Netzwerkschicht bezeichnet. Die zweite Schicht ist die Internetschicht, die für die Steuerung der Datagramme verantwortlich ist. Die darüber liegende Schicht ist die Transportschicht, die die Transportprotokolle TCP und UDP enthält. Die oberste Schicht wird Anwendungsschicht genannt und ist für die Funktion der Netzanwendungen(FTP, Telnet, NFS,…) zuständig. Abbildung 2.2 zeigt die schematische Darstellung des TCP/IP-Modells.

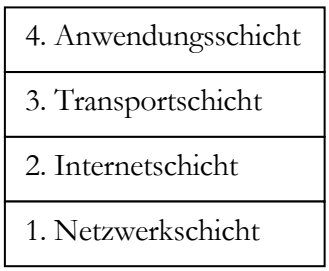

Abbildung 2.2: Schema der TCP/IP-Protokollhierachie

Schicht 1 - Netzwerkschicht

Die unterste Schicht im TCP/IP-Modell übernimmt die Funktionalität der Bitübertragungs- und Sicherungsschicht des OSI-Modells. In dieser Schicht arbeiten beispielsweise die Ethernet-Treiber.

Schicht 2 - Internetschicht

In dieser Schicht ist das Internet Protokoll(IP) angesiedelt. Dieses sorgt für die Bildung der Datagramme und deren korrekte Übermittlung vom Sender zum Empfänger. Es finden sich neben dem Internet Protokoll auch noch das Address Resolution Protokoll(ARP) und das Reverse Address Resolution Protokoll(RARP), das für die Adressauflösung verantwortlich ist. Das Internet Control Message Protocol(ICMP) befindet sich ebenfalls in dieser Schicht und ist für den Transport von Fehler- und Diagnoseinformationen zuständig.

Schicht 3 - Transportschicht

In der Transportschicht befinden sich zwei wichtige Transportprotokolle, das Transmission Control Protokoll(TCP) und das User Datagram Protokoll(UDP). TCP ist ein verbindungsorientertes Protokoll, das Funktionen zur Fehlererkennung und Fehlerkorrektur enthält. UDP ist ein verbindungsloses Protokoll, das keine Fehlerkorrektur durchführt, dafür aber mit weit weniger Overhead[1] auskommt.

Schicht 4 - Anwendungsschicht

Diese Schicht stellt eine Reihe von Anwendungsprotokollen zur Verfügung, z.B. Telnet, FTP, NFS. Diese Dienste können direkt vom Benutzer verwendet werden, dies ist nur in dieser Schicht möglich.

Im nachfolgenden Abschnitt werden die zwei wichtigsten Protokolle, TCP und IP, noch genauer erklärt. Diese Protokolle bilden die Grundlage eines jeden TCP/IP basierten Netzes und sind deshalb von größerer Bedeutung.

[1] Verwaltungsaufwand

Internet Protokoll(IP)

Das Internet Protokoll ist ein verbindungsloses Protokoll, d.h. es besteht keine direkte Verbindung zwischen Sender und Empfänger. Die Informationen, die für die korrekte Übermittlung der Daten notwendig sind, befinden sich im Header[1] des Datagrammes[2]. Zu den Aufgaben von IP gehört:

- Zerlegen und Zusammensetzen von IP-Paketen
- Definition des Adressierungsschemas
- Übermittlung der Daten von der Transportschicht zur Netzwerkschicht
- Routing der IP-Pakete durch das Netz

Das Routing der Pakete erfolgt mittels „Best Effort" Methode, d.h. es wird mit allen Bemühungen versucht das Paket zum Ziel zu befördern. Es gibt jedoch keine Quittung für den korrekten Empfang des Paketes, dies geschieht erst in der Transportschicht. IP besitzt keine Methoden zur Fehlererkennung bzw. Fehlerbeseitigen, lediglich Fehler im Header können erkannt werden. Abbildung 2.3 verdeutlicht den Aufbau eines IP-Paketes.

<table>
<tr><td>Version</td><td>Length</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td>Time to Live</td><td colspan="2">Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source IP Address</td></tr>
<tr><td colspan="5">Destination IP Address</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Abbildung 2.3: Aufbau des IP-Paketes

[1] Kopf

[2] Wird auch als IP-Paket bezeichnet

Ein IP-Paket besteht, wie in Abbildung 2.3 zu sehen, aus dem Header und den Daten. Es gibt einen Teil des Headers mit fester Größe von 20 Byte, der von einem optionalen Teil mit variabler Länge gefolgt wird. Die maximale Größe eines Datagrammes beträgt 64KByte, in der Regel ist ein Datagramm 1500 Byte groß.

Die Felder im Header werden nachfolgend genauer erklärt, in Klammern wird die Länge des jeweiligen Feldes angegeben:

Version(4 Bit): Dieses Feld enthält die Versionsnummer des verwendeten IP-Protokolls. Das hat den Vorteil, das sich mit verschieden Versionen über längere Zeit arbeiten lässt. Das Feld enthält 4 für IPv4 oder 6 für IPv6. (siehe auch [1])

Length(4 Bit): Hier wird die Länge des Datagrammkopfes vermerkt. Die Länge wird als Anzahl der 32-Bit-Worten vermerkt. Der kleinste mögliche Wert beträgt 5, dies sind 5*32bit(=20 Byte). Das wäre der Fall wenn keine Optionen angegeben wurden, durch Anhängen von Optionen kann dieser Wert bis auf 15(15*32bit=60 Byte) erhöht werden.

Type of Service(8 Bit): In diesem Feld haben die Bits verschiedene Bedeutung. Bit 0-2 heißen Precedence und gibt die Priorität von 0(niedrig) bis 7(hoch) an. Die nächsten drei Bits sind als Flags mit unterschiedlicher Bedeutung zu betrachten, Bit 3(Delay[1]), Bit 4(Throughput[2]) und Bit 5(Reliability[3]). Diese 3 Flags ermöglichen es dem Host, zu entscheiden auf welche Aktionen er seine Priorität legt. Die letzten beiden Bits sind reserviert. Dieses Feld wird in IPv4 nicht benutzt.

Total Length(16 Bit): Dieses Feld enthält die Gesamtlänge des IP-Paketes in Byte, d.h. einschließlich Daten und Header. Die maximale Größe ist wegen des 16 Bit Feldes auf 65535 Byte beschränkt.

Identification: Dieses Feld ist eine fortlaufende Nummer, durch die der Empfänger feststellen kann, welches Fragment zu welchen Datagramm gehört. Fragmente eines Datagramms erhalten die gleiche Nummer.

[1] Verzögerung

[2] Durchsatz

[3] Zuverlässigkeit

Flags(3 Bit): Nur zwei Bits werden für die Flags DF(Don't Fragment) und MF(More Fragment) benötigt, das andere bleibt ungenutzt. DF bedeutet, dass ein Datagramm nicht fragmentiert werden darf, wenn es den Wert 1 hat. Steht das Flag MF auf 1, wird angezeigt, das noch weitere Fragmente folgen. Hat das Flag den Wert 0, so folgen keine Fragmente mehr oder das Datagramm wurde nicht fragmentiert.

Fragment Offset(13 Bit): Dieser Wert gibt an, an welche Stelle relativ zum Anfang des Datagramms ein Fragment gehört. Mit dieser Angabe kann der Empfänger das Datagramm durch die richtige Zusammensetzung der Fragmente wiederherstellen. Aus der Größe dieses Feldes ergibt sich die maximale Anzahl von 8192 Fragmenten pro Datagramm.

Time to Live(8 Bit): Dieses Feld wird als Zähler(in Sekunden) verwendet, der die Lebensdauer eines IP-Paketes enthält. Bei jedem Routerdurchlauf muss der Zähler mindestens um 1 verringert werden. Wenn der Zähler bei 0 angelangt ist, wird das Paket verworfen. Dies soll verhindern, dass ein Paket endlos im Netz umherwandert. Maximal kann ein Paket 255 Sekunden im Netz verbleiben.

Protocol(8 Bit): Dieses Feld enthält die Nummer des Transportprotokolls, an das das Paket weitergeleitet wird (→siehe RFC 1700).

Header Checksum(16 Bit): Entsprechend dem Namen, enthält das Feld die Prüfsumme der Felder im Header, außer diesem. Bei jedem Routerdurchlauf muss dieser Wert neu berechnet werden. Der Wert ist die Summe modulo 16 aller 16 Bit Blöcke im Header und wird im Einerkomplement gespeichert.

Source IP Address, Destination IP Address (je 32 Bit): Die Adresse wird byteweise durch 4 Dezimalzahlen, die durch Punkte getrennt sind, notiert. Zum Beispiel ist 141.57.33.11 eine IP-Adresse. Die Adressen lassen in Netzwerk- und Hostadresse einteilen und je nach Länge der Netzwerk- und Hostadresse auch in verschiedene Klassen.

Klasse	1. Byte	2. Byte	3. Byte	4. Byte
A	1-126	0-255	0-255	0-255
B	128-191	0-255	0-255	0-255
C	192-223	0-255	0-255	0-255
D	224-239	–	–	–
E	240-247	–	–	–

Abbildung 2.4: Zeigt die verschiedenen IP-Adressklassen

In Abbildung 2.4 wird die Netzwerkadresse durch Grün und die Hostadresse durch Gelb gekennzeichnet. Die Netzwerkadresse charakterisiert das Netzwerk in dem sich der Host befindet. Die Rechner innerhalb eines Netzes haben folglich die gleiche Netzwerkadresse. Die Hostadresse beschreibt einen bestimmten Rechner in einem Netzwerk.

Klasse A:

Das erste Byte identifiziert die Netzwerkadresse und die übrigen 3 die Hostadresse. Das erste Bit vom ersten Byte ist 0 und somit ergeben sich 126 mögliche Klasse A Netze und 2^{24} mögliche Hostadressen.

Klasse B:

Im ersten Byte ist das erste Bit 1 und das zweite 0, die Wertespanne vom ersten Byte beträgt entsprechend 128 bis 191. Die ersten zwei Bytes bestimmen die Netzwerk-adresse und letzten zwei die Hostadresse. Möglich sind also bis zu 2^{14} verschiedene Netzwerke und 2^{16} verschiedene Hosts.

Klasse C:

Das erste Byte kann Werte von 192 bis 223 annehmen, da die ersten zwei Bits 1 und das dritte 0 ist. Die ersten 3 Byte bestimmen die Netzwerkadresse, das letzte Byte die Hostadresse. Daraus ergeben sich 2^{21} mögliche Netzwerke und bis zu 254 verschiedene Hosts in einem Netzwerk.

Klasse D:

Die Adressen dieser Klasse werden auch Multicastadressen genannt. Das erste Byte dieser Adressen kann Werte von 224 bis 239 annehmen, da die ersten drei Bits 1 und das vierte Bit 0 sind. Diese Adressen werden gebraucht, um z.B. ein Datagramm an mehrere Hosts gleichzeitig zu versenden.

Klasse E:

Diese Klasse ist für zukünftige Verwendungen reserviert.

Options(24 Bit): Dieses Feld wird in der Praxis selten benutzt und spielt nur eine untergeordnete Rolle. (Siehe RFC 791)

Padding(8 Bit): Damit die Länge ein Vielfaches von 32 bit beträgt wird dieses Feld zum Auffüllen benutzt.

Transmission Control Protocol(TCP)

TCP ist in der Transportschicht des TCP/IP-Referenzmodells definiert und stellt ein Verbindungsorientiertes Protokoll dar. Um eine zuverlässige Verbindung zu erstellen wird eine Technik namens PAR[1] verwendet. Das heißt, es wird nur eine positive Quittung gesendet, wenn die Daten korrekt angekommen sind. Bleiben die Quittungen aus, werden, nach Ablauf eines Timers, die Daten erneut gesendet. Jedes TCP-Paket[2] enthält eine Prüfsumme, an der festgestellt werden kann, ob ein Paket fehlerfrei übertragen wurde.

Da TCP ein Verbindungsorientiertes Protokoll ist, muss diese vorher initiiert werden. Dies wird durch den **three-way-handshake** erreicht, der wie folgt funktioniert. Ein Host A der zu Host B eine Verbindung aufbauen will, sendet ein TCP-Segment mit gesetztem SYN-Flag und eine Sequenznummer, die Host A beginnend zum kommunizieren benutzt, zu Host B. Die Sequenznummer wird verwendet um die Daten beim Empfänger in der richtigen Reihenfolge wieder zusammenzusetzen. Wird die Verbindung von Host B angenommen, sendet Host B ein Paket mit gesetztem SYN- und ACK-Flag und die um 1 erhöhte Sequenznummer, die im ACK-Number Feld steht. Im Sequenznummernfeld steht jetzt die Sequenznummer, die Host B beginnend zum kommunizieren verwendet. Zum Schluss wird dieses Segment von Host A dadurch bestätigt, dass Host A ein Paket an Host B sendet, in dem das ACK-Flag gesetzt ist und im ACK-Number Feld steht die um 1 erhöhte Sequenznummer von Host B. Mit diesem Verfahren wird jede TCP-Verbindung etabliert.

TCP zerlegt den Datenstrom in Segmente von maximal 64 kByte, das heißt TCP kann die Paketlänge dynamisch an die Gegebenheiten im Netzwerk anpassen. Diese Segmente werden an die Internetschicht weitergeleitet und als IP-Paket verschickt.

[1] Positive Acknowledgement with Re-Transmission

[2] Wird auch als TCP-Segment bezeichnet

<table>
<tr><td colspan="3">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="5">Sequence Number</td></tr>
<tr><td colspan="5">Acknowledgment Number</td></tr>
<tr><td>Offset</td><td>Reserved</td><td>Flags</td><td colspan="2">Window</td></tr>
<tr><td colspan="3">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Abbildung 2.5: TCP-Segment

Abbildung 2.5 zeigt den Aufbau eines TCP-Segmentes und die Bezeichnung der einzelnen Felder. Ein TCP-Segment besteht aus einem Header und den zu übertragenden Daten.

Im nachfolgenden werden die einzelnen Felder genauer erklärt, in Klammern wird die Länge der jeweiligen Felder angegeben.

Source-,Destination Port(je 16 bit) : Diese Felder dienen zur Identifizierung einer TCP-Verbindung. Mit diesen Portnummern kann TCP die Daten an die bestimmten Anwendungen korrekt weiterleiten. Die IP-Adresse und die Portnummer kennzeichnen einen Kommunikationsendpunkt[1].

Sequence-,Acknowledgment Number (je 32 bit): Die Sequence Number wird benutzt, um die Pakete in der richtigen Reihenfolgen wieder zusammenzusetzen. Dabei wird die Sequence Number in Senderichtung und die Ack. Number zum Bestätigen der empfangenen Pakete verwendet. Die Anfangswerte sind zufällig gewählte Zahlen, die bei jedem Verbindungsaufbau neu initialisiert, gegenseitig gesendet und quittiert werden. Während der Datenübertragung wird die Sequence Number vom Absender um die Anzahl der gesendeten Bytes erhöht. Die Ack. Number gibt an, bis zu welchem Byte der Empfänger die Daten korrekt erhalten hat.

Offset(4 Bit): Um den Beginn der Daten zu bestimmen, wird in diesem Feld die Länge des Headers in 32 bit Blöcken gespeichert.

[1] Wird auch als Socket bezeichnet

Reserved(6 bit): Dieses Feld ist für zukünftige Verwendungen reserviert.

Flags(6 bit): Die folgenden 6 Flags sind gültig wenn sie auf 1 gesetzt sind.

1. URG: Das Feld Urgent Pointer wird benutzt.
2. ACK: Das Feld Acknowledgement Number ist gültig.
3. PSH: Die Daten sollen ohne Pufferung direkt an die entsprechend Anwendung weitergegeben werden.
4. RST: Die Verbindung soll zurückgesetzt werden. Dieses Flag wird meist im Fehlerfall benutzt.
5. SYN: Eine Verbindung soll aufgebaut werden. Siehe auch three-way-handshake.
6. FIN: Die Verbindung soll beendet werden.

Window(16 bit): Diese Feld beinhaltet an Anzahl an Bytes, die der Empfänger zur Verfügung hat. Mit dieser Zahl kann der Empfänger den Datenstrom steuern. Zum Beispiel könnte eine 0 in diesem Feld stehen und der Sender erkennt, dass der Empfänger gerade keine Daten empfangen kann und stoppt den Datenstrom.

Checksum(16 bit): Zur Berechnung der Checksumme wird dem eigentlichen TCP-Header ein so genannter Pseudo-Header vorangestellt. Dieser besteht aus 3 Feldern je 32 bit. Die ersten beiden Felder enthalten die Quell- und Ziel-IP-Adresse. Das letzte Feld besteht aus 8 Bits, die auf null gesetzt sind, danach folgen 8 Bits für die Protokollnummer und schließlich die Länge des TCP-Segments. Der Pseudo-Header soll helfen von IP falsch adressierte Pakete zu erkennen.

Urgent Pointer(16 bit): Dieser Wert ergibt zusammen mit der Sequence Number ein Zeiger auf Daten besonderer Dringlichkeit. Diese Daten sollen sofort gelesen werden.

Options(24 bit): Zurzeit sind 3 Optionen bekannt, **End of Option List**, **No-Operation** und **Maximum Segment Size**(beim Verbindungsaufbau kann die maximale Segmentgröße

festgelegt werden). Die anderen beiden Optionen spielen in der Praxis nur eine untergeordnete Rolle.

Padding(8 bit): Dieses Feld wird benutzt, um die 32 bit Grenze einzuhalten.

2.1.2 xDSL(Digital Subsriber Line)

Das x vor DSL ist als Platzhalter für einen Buchstaben zu sehen, der die verschiedenen DSL Varianten bezeichnet. DSL ist ein Oberbegriff für breitbandige digitale Leitungstechnik, die auf Kupferleitungen basiert. DSL wurde schon in den sechziger Jahren von der Firma Bellcore in den U.S.A. entwickelt. In Europa wird heutzutage eine weiterentwickelte Version dieser Technologie verwendet. DSL macht sich die Tatsache zunutze, dass die vorhandenen Kupferleitungen in ihrer Bandbreite noch nicht voll ausgeschöpft wurden. Die entsprechende Verwertung ungenutzter Bandbreite erhöht die Übertragungskapazitäten um ein vielfaches als z.B. ISDN(64kBit/s) und ist somit auch für die Anwendung im Multimediabereich und andere Hochgeschwindigkeitsanwendungen tauglich geworden.

Es gibt verschiedene DSL Technologien, die sich in der Anzahl der verwendeten Kupferadern, Übertragungsfrequenzen und Modulationsverfahren unterscheiden. Die Trennung von Up- und Downstream Geschwindigkeit ist ebenfalls eines der wichtigen Merkmale von DSL. Die verschiedenen DSL Varianten verwenden auch unterschiedliche Up- bzw. Downstream Geschwindigkeiten. Die bekannteste DSL Art ist ADSL(Asymmetric Digital Subscriber Line) und verwendet, wie der Name schon sagt, einen asymmetrischen Datenstrom. Hier ist der Datenstrom vom Provider zum Kunden(Downstream) größer als umgekehrt(Upstream). Im Gegensatz zu ADSL nutzt SDSL(Single pair Digital Subscriber Line) einen symmetrischen Datenstrom, d.h. die Datenraten vom Kunden zum Provider und in Gegenrichtung ist gleich groß. Weiterhin existiert noch HDSL(High Data Rate DSL), sie ist die älteste DSL Technologie und stammt von von der klassischen Hochgeschwindigkeitstechnik T1/E1 ab. Diese klassische T1/E1 Technik setzte jedoch den Einsatz von Repeatern[1] voraus. Da dies sehr teuer und aufwendig ist, entwickelte die Firma Bellcore HDSL, das den Einsatz von Repeatern überflüssig macht und die vorhandenen Kupferleitungen optimal ausnutzt.

[1] Signalverstärker zwischen Netzwerksegmenten

Die Abbildung 2.6 zeigt eine Übersicht der verschieden DSL Varianten und ihre unterschiedlichen Eigenschaften.

DSL Art	**Downstream**	**Upstream**	**Frequenzband**	**max. Entfernung bis zur nächsten Vermittlungsst.**
ADSL	1,5 MBit/s bis 9,0 MBit/s	bis 1 MBit/s	138 kHz bis 1 MHz	6 km
HDSL	1,5 MBit/s bis 2,0 MBit/s	wie Downstream	0,1 kHz bis 485 kHz	4 km
SDSL	bis 2,3 MBit/s	wie Downstream	0 Hz bis 387 kHz	3 km
VDSL (asym.)	bis 52 MBit/s	bis 6,4 MBit/s	300 kHz bis 20 MHz	300m
VDSL (sym.)	bis 34 MBit/s	bis 34 MBit/s	300 kHz bis 20 MHz	1,3 km

Abbildung 2.6: Übersicht DSL Arten

Im nächsten Abschnit werden die einzelnen DSL Technologien noch genauer erläutert.

ADSL:

Dies ist die bekannteste und meist verwendete DSL Technologie. Dies liegt nicht zuletzt daran, das ADSL sich durch geringere Kosten, gegenüber den anderen DSL Varianten, im Massenmarkt durchgesetzt hat und vorzugsweise von Privatkunden genutzt wird. ADSL ist z.B. hervorragend für Videodienste geeignet. Video-on-Demand wäre ein Beispiel für ein Videodienst, während in Providerrichtung nur die Befehle für START, STOP, VORSPULEN oder ZURÜCKSPULEN übertragen werden, könnten in Kundenrichtung die Videodaten übertragen werden, die ja wesentlich mehr Bandbreite benötigen als die Befehle. Video-on-demand ist als eine virtuelle Videothek zu sehen, die der Kunde von seinem Rechner zu Hause steuern kann. Bereits mit 2,0 MBit/s können Videos in einer Qualität übertragen werden, die dem eines handelsüblichen Videorecorders entspricht. In Deutschland wird ADSL von der Firma Telekom unter dem Marketingnamen „T-DSL“ angeboten. „T-DSL“ ist keine eigenständige DSL Technologie.

ADSL verwendet eine so genannte Kanaltrennung, d.h. das verfügbare Frequenzband wird in verschieden Bereiche(Kanäle) aufgeteilt. Die Frequenzen bis 4kHz sind dem analogen Telefonsystem zugeordnet, bis 130 kHz wird ISDN zugeschrieben und der Bereich darüber steht für Up- und Downstream von ADSL zur Verfügung. Die Technik, die für die Aufteilung des Frequenzspektrums in entsprechende Kanäle verantwortlich ist, wird als FDM(Frequency Division Multiplexing) bezeichnet. FDM bildet die nur die Grundlage für den Datentransfer, der mittels zwei verschiedenen Übertragungsmethoden erfolgen kann. Die erste Methode nennt sich CAP(Carrierless Amplitude/ Phase Modulation) und die zweite heißt DMT(Distcrete Multi-Tone Modulation). Bei CAP handelt es sich eher um eine ältere, nur noch selten eingesetzte Methode(siehe auch [5]).

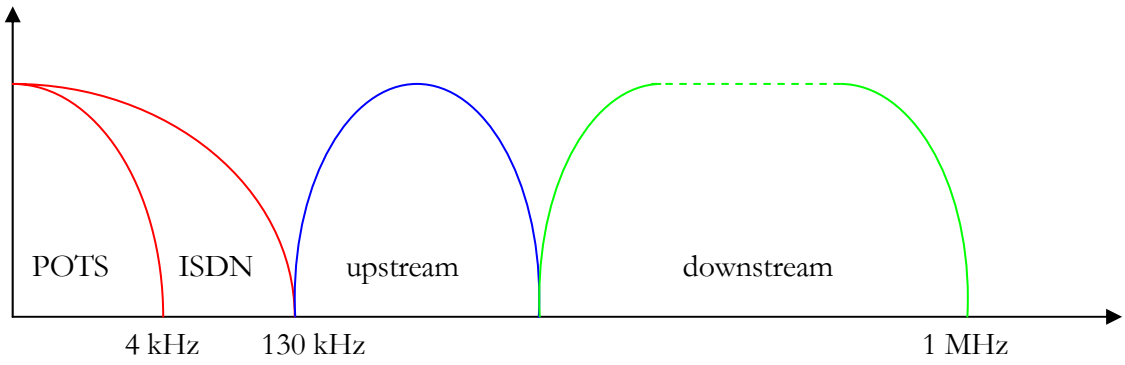

Abbildung 2.7: ADSL mit CAP

Die Abbildung 2.7 verdeutlicht die Aufteilung der Frequenzen in einzelne Kanäle im CAP-Verfahren. Bei CAP wird eine trägerlose Phasen/Amplituden Modulation verwendet, wobei das Trägersignal, durch geschickte Wahl der Frequenz, nicht mit übertragen wird. Die Bezeichnung POTS(Plain Old Telephone System) in Abbildung 2.7 steht für das analoge Telefonsystem. Der Hauptunterschied zwischen CAP und DMT besteht darin, das CAP getrennte Frequenzbereiche für Up- und Downstream benutzt, wie in Abbildung 2.8 zu sehen ist, wird dies bei DMT nicht getan. DMT wurde von der Normierungsbehörde ANSI zum ADSL-Standard ausgewählt und wird heute von den meisten Providern benutzt. Es wurde unter anderem ausgewählt weil es auch bei nicht bekannter Leitungsqualität bessere Ergebnisse liefert als CAP. Wie in Abbildung 2.8 zu sehen, wird das für ADSL vorgesehene Frequenzband in mehrere 4kHz kleine Teilfrequenzbänder(Kanäle) unterteilt.

Für den Downstream werden bis zu 256 Kanäle genutzt, während für den Upstream nur 32 Kanäle benötigt werden.

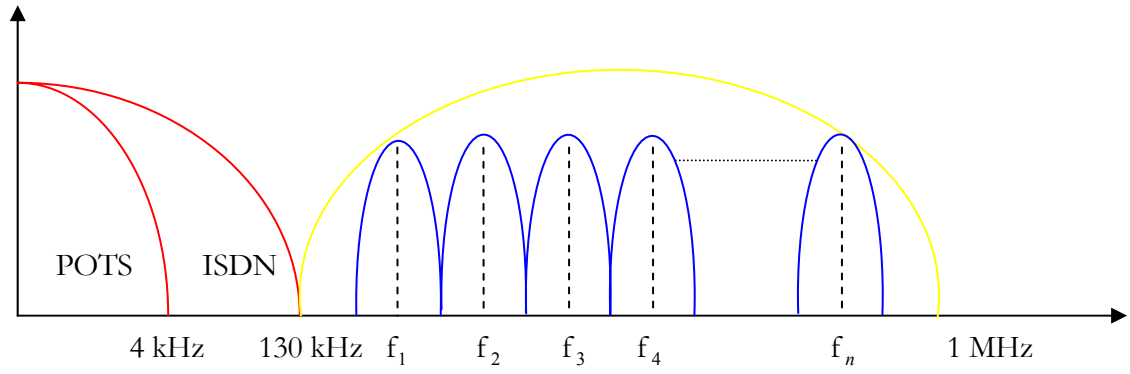

Abbildung 2.8: ADSL mit DMT

Mit DMT ist ADSL wesentlich flexibler, da für jeden Teilkanal unterschiedliche Bitraten(Bits die pro Teilkanal gesendet werden) festgelegt werden können. Es können so störanfällige Frequenzen umgangen und die Gesamtdatenrate somit immer optimal an die Gegebenheiten angepasst werden.

DSL wird in Verbindung mit dem PPPoE(Point-to-Point Protocol over Ethernet) genutzt. Die PPPoE-Pakete werden durch ein DSL-Modem in ATM-Zellen verpackt und zur Gegenstelle gesendet. Die Fehlerkorrektur, die bei ADSL Anwendung findet wird als Forward Error Correction(Vorwärtsfehlerkorrektur) bezeichnet. Hierbei werden vor der Übertragung Kontrollbits an die Daten angehängt, damit kann der Empfänger eventuell auftretende Fehler bis zu einer bestimmte Bitzahl erkennen und einer kleineren Bitzahl korrigieren. Da es auch zu Störimpulsen während der Übertragung kommen kann, die mehrere aufeinander folgende Bytes zerstören können, wird ein so genanntes Interleaving verwendet. Das bedeutet, die PPPoE-Pakete werden so auf ATM-Zellen aufgeteilt, dass aufeinander folgende ATM-Zellen immer nur einige Bytes des 1. Paketes, gefolgt von einigen Bytes des 2. Paketes, …, enthalten. Dies erhöht die Wahrscheinlichkeit, das auch solche Fehler, die durch Störimpulse verursacht werden, mit dem Forward Error Correction Verfahren erkannt und beseitigt werden können. Mit der Interleaving Depth kann die Verschachtelungstiefe eingestellt werden(siehe auch [7]).

HDSL:

Diese DSL-Variante arbeitet ebenfalls mit Kupferleitungen, mindestens auf zwei oder drei Adernpaare verteilt, um Störungen so gering wie möglich zu halten. HDSL erreicht eine Datenrate von 1,5-2,0 MBit/s und arbeitet mit einem symmetrischen Datenfluss. Diese DSL Variante ist vorwiegend für den geschäftlichen Einsatz geeignet, da sie teurer als ADSL ist, vor allem weil mindestens 2 Adernpaare für den Betrieb benötig werden. ADSL braucht grundsätzlich nur ein Adernpaar und ist deshalb auch preiswerter in der Verkabelung.

SDSL:

SDSL(Single pair DSL) benötigt, wie der Name schon verrät, auch nur ein Adernpaar und ist eine modernere HDSL Variante. Wie HDSL verwendet es auch einen symmetrischen Datenfluss. Die höchstmögliche Entfernung zur nächsten Vermittlungsstelle beträgt 3km bei einer maximalen Übertragungsrate von 2,3 MBit/s.

VDSL (Very high data rate DSL):

VDSL ist eine Weiterentwicklung von ADSL bzw. SDSL und soll in Zukunft den Bedarf an noch höheren Datenraten abdecken. Dies wird nur durch eine gut ausgebaute Glasfaser-Infrastruktur möglich sein. Nur noch der Weg von der letzten Vermittlungsstelle zum Kunden soll mit Kupferkabel verbunden werden. Die asymmetrische Variante überträgt im Downstream bis zu 52 MBit/s und im Upstream immer noch bis 6,4 MBit/s. Die symmetrische Variante schafft bis zu 34 MBit/s. Wenn diese Übertragungsraten erreicht werden wollen, darf die Strecke bis zur Vermittlungsstelle nur etwa 300m betragen. Sollte dies in naher Zukunft auch ein „Massenprodukt“ für Privatanwender werden, bedarf es natürlich eines engmaschigen Netzs an Vermittlungsstellen.

2.3 QoS in IP-Netzen

Die Abkürzung QoS steht für Quality of Service und lässt sich mit Dienstgüte übersetzen. In diesem Abschnitt werden die verschiedenen Architekturen vorgestellt die ein QoS in IP-Netzen ermöglicht.

2.3.1 Priorisierung

In IP-Netzen werden alle Pakete gleich behandelt, egal ob sie zeitkritische Daten, wie z.B. Videodaten oder Sprache übermitteln. Für Echtzeitanwendungen ist daher IP eher ungeeignet oder es müssen starke Abstriche bei der Qualität gemacht werden. Für Echtzeitanwendungen wie Videokonferenzen oder interaktive Multimediaanwendungen müssen minimale Qualitätskriterien erfüllt sein, um den Dienst sinnvoll nutzen zu können. Um eine bestimmte Qualität zu erreichen, ist eine Priorisierung unabdingbar. Dabei ist natürlich die vom Anwender subjektiv empfundene Qualität entscheidend. In klassischen IP-Netzen ist keine Zusicherung der Qualität der Übertragung vorgesehen. Die Güte der Verbindung in IP-Netzen hängt vom aktuellen Datenverkehr ab und somit sind für Echtzeitanwendungen keine Möglichkeiten gegeben, die Güte vorherzusehen.

Es gibt verschiedene Eigenschaften mit denen die Qualität einer Verbindung messen kann. Zum einen ist die **Verzögerung**(Latency oder Delay) ein Parameter, der die Zeit, die ein Paket vom Sender zum Empfänger benötigt beschreibt. Ein weiterer Parameter ist **Paketverlus**t(Paketloss), d.h. Prozentzahl der Pakete, die während der Verbindung verloren gehen. Dies trifft auf beschädigte und verworfene Pakete zu. **Jitter** heißt ein Parameter, der die Schwankungen der Paketlaufzeit innerhalb eines Datenstroms kennzeichnet. Diese Parameter werden auch als QoS-Parameter bezeichnet. Um eine effektive Priorisierung vornehmen zu können, müssen die verschiedenen Datenströme in Verkehrsklassen/Dienstklassen eingeteilt werden. Prinzipiell kann man drei verschiedene Dienstklassen unterscheiden, **Echtzeitverkehr**, **Streamingverkehr** und **normaler Datenverkehr.**

Im Fall von Echtzeitverkehr handelt es sich vordergründig um Audio -und Videokonferenzen, die nur geringste Verzögerungen und minimalen Jitter vertragen. Im Fehlerfall sind hier die Auswirkungen am dramatischsten, d.h. bei Paketverlust ist die Zeit meist nicht ausreichend um Pakete erneut anzufordern, da der Empfangspuffer(~10ms)

schon leer ist. Das macht sich vor allem in Bild -und Tonverlust bemerkbar, bei Bildübertragungen hängt dies auch vom jeweiligen Kodierungsverfahren ab.

Übertragungen die zur Kategorie Streamingverkehr gehören, sind zum Beispiel Realaudio und Realvideo oder auch Windows Media Player. Hier sind die Anforderungen bezüglich Jitter, Delay und Paketloss nicht so hoch wie bei Echtzeitverkehr. Entsprechend haben sie auch einen größeren Empfangspuffer(~10s), dennoch benötigen auch diese Dienste eine gewisse Mindestbitrate. Weitere Anwendungen dieser Klasse können auch Datenbanken und Datensicherung sein.

Der normale Datenverkehr besteht aus den älteren Diensten, für die das Internet eigentlich entwickelt wurde, d.h. FTP, HTTP, MAIL….Bei diesen Anwendungen machen sich hohe Verzögerungen, große Jitter und Paketloss lediglich in längeren Wartezeiten bemerkbar.

Eines der wichtigsten Modelle für QoS in IP-Netzen ist, wie oben schon erwähnt, die Priorisierung und beruht auf der Unterscheidung einzelner Verkehrsklassen. Der erste Standard ist die **Differentiated Services Architecture(DiffServ)** und wurde von der IETF (Internet Engineering Task Force) entwickelt und findet vor allem in größeren Netzen Anwendung. DiffServ arbeitet in Schicht 3 des OSI-Referenzmodells. Ein zweiter Standard **IEEE 802.1q** arbeitet in Schicht 2 des OSI-Referenzmodells und ist für LANs vorgesehen. Beide Standards basieren auf einer Unterteilung des Datenverkehrs in Klassen, dazu müssen den Datenpaketen entsprechende Markierungen hinzugefügt werden. Im Fall von DiffServ ist die Markierung im Type-of-Service Feld(siehe 2.1.1) des IP-Headers. Der IEEE 802.1q Standard erweitert zur Markierung den Ethernet-Header, da er sich in Schicht 2 befindet. Die Markierungen dienen zur Einordnung der Pakete in verschiedene Dienstklassen. Sind die Pakete markiert, ist es Aufgabe des **Schedulers**, die Pakete entsprechend ihrer Einordung zu behandeln.

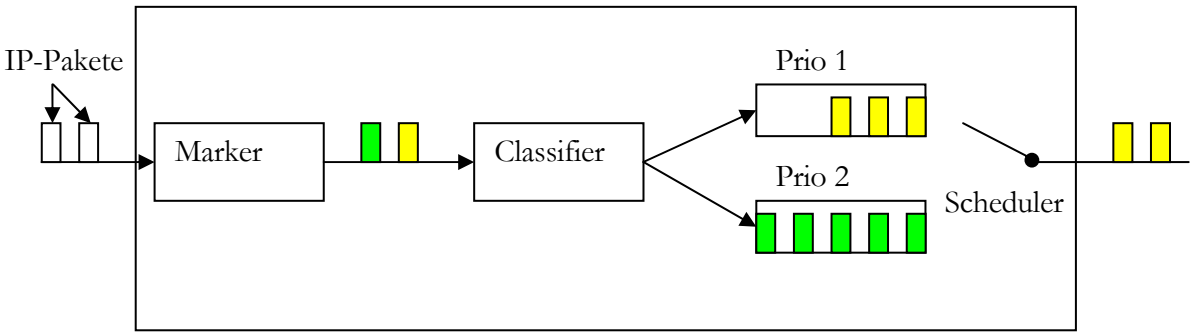

Abbildung 2.9: Schema eines Priorisierungsprozesses

Die Abbildung 2.9 zeigt, wie die Markierung und Einordnung prinzipiell funktionieren. Der Classifier verschiebt die Pakete in die Puffer für die sie markiert wurden. Es existiert also für jede Dienstklasse ein entsprechender Puffer. Ein Puffer wird nur vom Scheduler bedient, wenn kein Puffer mit höherer Priorität Daten zum Senden enthält.

2.3.2 DiffServ

DiffServ nutzt im Fall von IPv4, wie schon weiter oben erwähnt, das ToS-Feld im IP-Header zur Klassifizierung der IP-Pakete. Dieses 8 Bit große Feld wird im DiffServ Standard als DS-Feld bezeichnet. Die ersten 6 Bit dieses Feldes werden als DSCP(Differentiated Service Code Point) definiert und beschreiben die Zugehörigkeit der IP-Pakete zu vordefinierten Klassen, die anderen 2 Bits sind reserviert. Daraus ergeben sich 32 mögliche DSCP Werte, die für eine entsprechende Einteilung in Klassen vorgesehen sind. Für das Verhalten des Schedulers, auch als PHB(Per Hop Behavior) bezeichnet, wurden bei DiffServ zwei Standards festgelegt, **Expedited Forwarding PHB** und **Assured Forwarding PHB Group**. Ein Per-Hop Behavior kann sich entweder auf ein DSCP Wert beziehen oder auf mehrere, dann wird es als PHB Group bezeichnet. Aus diesen beiden Forwarding Regeln ergeben sich die beiden Dienstgüten[1], die bei DiffServ vorgesehen sind. Der **Premium Service** basiert auf dem Expedited Forwarding PHB und erzeugt einen Dienst der eine virtuelle Leitung darstellt. Die Anforderungen an diesen Dienst leiten sich aus der Tatsache ab, dass dieser Service eine Leitung simulieren soll.

[1] Qualität eines Transportdienstes in IP-Netzen

Entsprechende Anforderungen sind:

- garantierte Bandbreite
- geringes Delay und wenig Jitter
- kein Paketloss durch Datenstaus

Die Nutzungsmöglichkeiten für diesen Dienst sind Echtzeitübertragungen wie z.B. Sprachdienste und Multimediadienste. Der DiffServ Standard schreibt in keinem Fall die Implementierung eines Dienstes vor, sondern definiert lediglich die Anforderungen an diesen. So wird von dem Premium Service gefordert, dass der Eingangsverkehr an allen DS-Geräten[2], der weitergeleitet werden soll, kleiner ist, als die Bandbreite die für den Ausgangsverkehr bereit steht. Anderenfalls könnten lange Verzögerungen und Paketverluste auftreten. Für die Einhaltung und Überwachung der Bedingungen sorgt der **Traffic Conditioner**. Dieser besteht aus **Meter**, **Marker**, **Dropper** und **Shaper**. Der Meter ist für die Messung des Verkehrs in der Klasse zuständig, wird die maximale Bandbreite überschritten, dann wird dies den anderen Komponenten mitgeteilt. Das Zusammenspiel aller Komponenten wird in Abbildung 2.10 verdeutlicht. Der Marker kann dem DSCP-Feld eines IP-Paketes einen neuen Wert geben und somit in eine andere Klasse einordnen. Der Dropper hat die Möglichkeit einzelne IP-Pakete zu verwerfen. Der Shaper kann durch Verzögerung einzelner IP-Pakete ebenfalls aktiv den Verkehr beeinflussen.

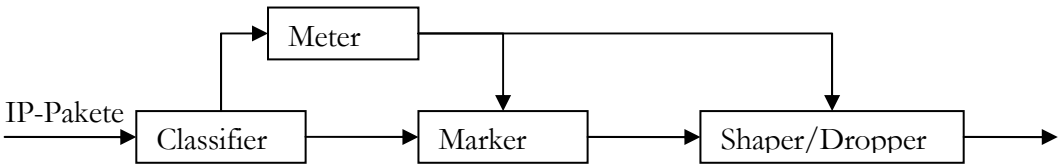

Abbildung 2.10: Darstellung der Arbeitsweise des Traffic Conditioner

[2] DiffServ fähiges Gerät

Die zweite definierte Dienstgüte, Assured Service, basiert auf der Assured Forwarding PHB Group. Bei diesem Dienst wird weniger auf Delay und Jitter geachtet, da dieser weniger anspruchsvoll ist, als der Premium Service. Die Anforderungen an diesen Dienst sind im Wesentlichen eine geringe Paketverlustrate und die Einhaltung eines vordefinierten Durchsatzes. In diesem Service kann noch eine Unterteilung in verschiedene Unterklassen sinnvoll sein, low, medium und high. Mit dieser Unterteilung kann das Dropping Verhalten entsprechend der Auslastung gesteuert werden.

2.3.3 IntServ

Die Bezeichnung IntServ steht für Integrated Services und stellt einen weiteren Ansatz dar, um eine Dienstgüte durch Reservierung zu erreichen. Im IntServ Standard werden zwei Dienstklassen unterschieden, **Controlled Load Service** (CLS) und **Guaranted Service** (GS). Dieser Standard basiert auf Ressourcen Reservierung, die von einer Anwendung aus angefordert wird. Die Applikation sendet eine Anforderung entsprechend eines vorausberechneten Bedarfs an Bandbreite an das Netzwerk. Es kann für jeden Flow[1] eine solche Vereinbarung getroffen werden. Die in der Vereinbarung angegebenen Resourcen werden entlang des Netzwerkpfades reserviert. Hier ist auch der größte Nachteil von IntServ zu sehen, jeder Router muss Zustandsinformationen von jeden einzelnen Flow speichern. Bei einer Dienstgüte-Anforderung wird vorher geprüft, ob diese entlang des Pfades eingehalten werden kann, sonst wird keine Bestätigung vom Netzwerk erteilt. Da eine Reservierung über die Kapazitätsgrenzen hinaus, die garantierten Zusagen an schon bestehenden Flows nicht eingehalten werden können.

Der Controlled Load Service soll eine Güte bereitstellen, die in einem Netzwerk mit geringer Auslastung vorkommt. Dies soll ohne Abhängigkeit von der realen Netzwerkauslastung gewährleistet werden. Es gibt zwei wichtige Bedingungen, auf die es dabei ankommt. Zum einen muss der Paketverlust relativ gering sein und zum anderen darf es keinen hohen Prozentsatz von IP-Pakten geben, die die minimale Übertragungsverzögerung wesentlich überschreitet.

Der Guaranteed Service zeichnet sich durch folgende Merkmale aus, die IP-Pakete werden innerhalb einer vorher vereinbarten Zeit übermittelt und es wird garantiert, dass es zu

[1] Datenstrom von Sender zu Empfänger

keinen Paketverlusten auf Grund von Queue Overflows[1] kommt. Diese Garantien werden vom Netzwerk nur eingehalten, wenn die Anwendung die vorher vereinbarte Bandbreite nicht überschreitet. Jede Anwendung muss vor dem Aufbau einer solchen Verbindung die Verkehrsparameter(Traffic Specification) und die Reservierungsparameter(Reservation Specification) an das Netz übermitteln, um zu prüfen, ob die geforderten Parameter eingehalten werden können. Die Verkehrsparameter, kurz TSpec, dienen zur Beschreibung des von der Applikation verursachten Verkehrs. Die Reservierungsparameter, kurz RSpec, beschreiben die Bandbreitenanforderungen und Verzögerungsparameter.

2.3.4 RSVP

Die Abkürzung RSVP steht für Resource Reservation Protocoll und stellt ein Signalisierungsprotokoll dar, dass in der Schicht 4 des OSI-Modells arbeitet. Das Protokoll soll eine von einer Applikation angeforderte Ressourcenreservierung entlang eines Netzwerkpfades sicherstellen. Ein wichtiges Merkmal von RSVP ist die klare Trennung zwischen Bereitstellung und Anforderungen von Ressourcen. Die Anforderung von Ressourcen wird von RSVP bearbeitet und die Bereitstellung erfolgt durch das **Traffic Control(siehe Abb. 2.11)**. Die Anforderung und die Kommunikation zwischen den RSVP Geräten werden mit Hilfe von RSVP-Messages realisiert. Die Anforderung wird zuerst von der **Policy Control** auf ihre Berechtigung überprüft und die Verfügbarkeit der angeforderten Ressourcen wird von dem Traffic Control ermittelt. Die Traffic Control besteht aus 3 Komponenten: Scheduler, Classifier und Admission Control. Von der Admission Control wird überprüft, ob einem Flow eine Dienstgüte zugesichert werden kann, ohne die bestehenden Flows zu beeinträchtigen. Ist einem Flow eine entsprechende Dienstgüte zugesichert worden, wird von Scheduler und Classifier für die korrekte Bearbeitung der RSVP-Messages entlang des Pfades gesorgt. Nach einer Anfrage wird der Pfad zwischen Sender und Empfänger festgelegt und danach werden auf diesem Pfad die Anforderungen gestellt. Die Abbildung 2.11 verdeutlicht den Informationsfluss zwischen Endgerät und Router, weiterhin ist auf der Illustration auch das Zusammenspiel zwischen den einzelenen Komponenten in einem RSVP-Gerät zu sehen.

[1] Überlaufen eines Zwischenpuffers/Warteschlangen

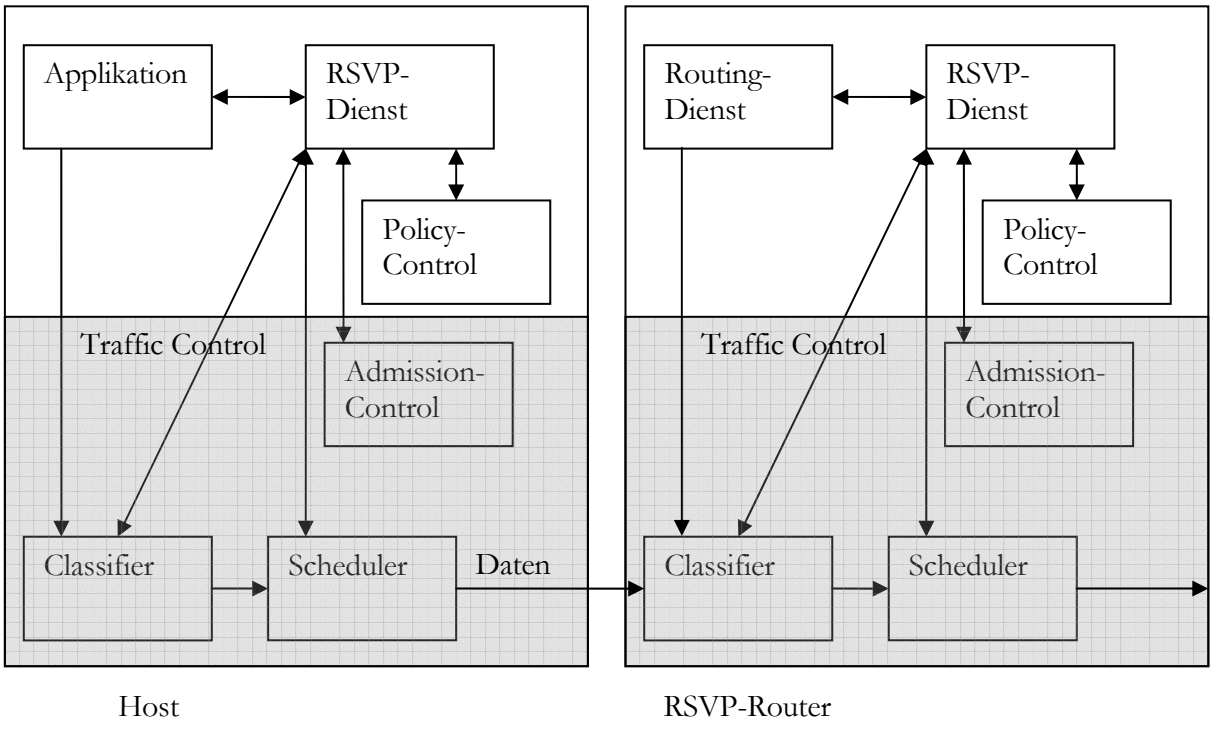

Abbildung 2.11: Zusammenspiel zwischen Host und RSVP-Router

Für die grundsätzliche Kommunikation zwischen den RSVP-Diensten existieren 2 Messagetypen, die **PATH**- und die **RESV**-Message.

Die PATH-Message wird von dem RSVP-Dienst des Senders an alle Empfänger gesendet. Um zu erkennen ob auf dem Pfad zwischen Sender und Empfänger alle Router RSVP-kompatibel sind, existiert ein TTL-Feld in dem Header jeder RSVP-Message. Dieses Feld wird nur von RSVP-fähigen Routern dekrementiert, normale IPv4 Router dekrementieren nur das TTL-Feld im IP-Header, wenn sich beide TTL-Felder in ihrem Wert unterscheiden, existiert mindestens ein Router, der nicht RSVP-kompatibel ist. Die PATH-Message dient verschiedenen Zwecken:

- Unterscheidung der einzelnen Flows
- Kennzeichnung des Netzwerkpfades
- Beschreibung der Eigenschaften der Flows

Für die Identifikation eines Flows existieren zwei Objekte in der PATH-Message, SESSION und SENDER_TEMPLATE. Diese zwei Objekte dienen der Zuordnung von Daten und Messages zu den jeweiligen Datenströmen. Das Objekt SESSION enthält die Zieladresse und den Zielport des Empfängers. Weiterhin wird im SESSION-Objekt auch die IP-Protokollnummer gespeichert. In SENDER_TEMPLATE speichert RSVP die IP-Adresse sowie die Portnummer des Senders.

Zur Kennzeichnung eines Pfades steht das Objekt RSVP_HOP zur Verfügung. Das Objekt wird von jedem RSVP-kompatiblen Gerät, das durchlaufen wird, verändert. Im Detail wird in diesem Objekt die IP-Adresse des letzten RSVP-Gerätes gespeichert, bei einer Weiterleitung trägt der RSVP-Dienst seine eigene IP-Adresse ein und speichert den alten Wert.

Für die Beschreibung der Eigenschaften der Datenströme steht das Objekt SENDER_TSPEC zur Verfügung. Einige der wichtigsten Parameter sind TOKEN_BUCKET_TSPEC und AVAILABLE_PATH_BANDWITH. Der Parameter AVAILABLE_PATH_BANDWITH gibt an, wie viel verfügbare Bandbreite entlang des Netzwerkpfades existiert. Die Werte des Parameters werden in Bytes/s angegeben. Der Parameter TOKEN_BUCKET_TSPEC beschreibt die Eigenschaften eines Flows und wird benutzt, um das voraussichtliche Datenaufkommen anzuzeigen. Der Parameter enthält die Werte Peak Data Rate, Token Bucket Rate(TBR), Token Bucket Size(TBS), Minimum Policed Size und Maximum Packet Size.

Peak Data Rate: Dieser Wert bezeichnet die maximale Datenrate mit der die Anwendung ihre Daten senden darf.

TBR/TBS: Diese beiden Werte sind Teil des Token-Bucket-Algorithmus. Der Token-Bucket bezeichnet hier ein „Eimer“ in dem sich eine bestimmte Anzahl von Platzhaltern(Token) anstatt der eigentlichen Daten befindet. Der Wert TBS bezeichnet die Größe des Buckets. Für jede zu sendende Einheit(z.B. Byte) muss ein Token aus dem Eimer genommen werden. Der Wert TBR bezeichnet die konstante Füllrate von Token in den Bucket. Ist der Bucket leer muss so lange mit dem Senden gewartet werden, bis wieder genug Token im Bucket sind, um weitere Daten senden zu können. Ist die TBR größer als die Senderate, werden die Token, die nicht mehr in dem Bucket Platz finden verworfen.

Das heißt, die maximale Datenrate(R_m) wird auf $R_m <= TBS + tTBR$ begrenzt, wobei t der Zeitintervall ist, der zum Senden benutzt wird. Dieses Verfahren soll vor allem den stoßartigen Datenverkehr(Burst) eingrenzen. Wenn ein Datenstrom relativ viele Bursts enthält muss dann entsprechend die Größe des Buckets(TBS) angepasst werden.

Minimum Policed Size: Dieser Wert beschreibt die minimale Größe der Pakete, die von der Anwendung gesendet werden.

Maximum Packet Size: Dieser Wert bezeichnet die maximale Paketgröße.

Die RESV-Message wird vom Empfänger über den Pfad, der von der PATH-Message vorgegeben wurde, zum Sender befördert. Diese Message enthält unter anderem die Objekte RESV_CONFIRM und SCOPE. Um bei einer erfolgreichen Reservierung eine Bestätigung zu erhalten, wird bei einer Bestätigungsanfrage die IP-Adresse des Empfängers in dem RESV_CONFIRM Objekt gespeichert. Das SCPOPE Objekt dient bei einer Wildcard Reservierung zur Verhinderung von Schleifen. Das heißt, wenn eine einzelne Ressourcen Reservierung für eine unbestimmte Anzahl von Sendern genutzt wird, kann es zu Schleifenbildung kommen, da diese Reservierung an alle neu auftretenden Sender automatisch weitergeleitet wird. Deshalb wird im Scope Objekt eine Liste IP-Adressen von bereits erreichten Sendern gespeichert.

Des Weiteren enthält eine RESV-Message die Objekte SESSION und FILTER_SPEC. Diese Objekte dienen zur Identifikation eines Flows. Diese Identifikation ist abhängig von der Art der Reservierung. Grundsätzlich existieren zwei Arten der Reservierung, die getrennte und die gemeinsame Reservierung. Bei einer getrennten Reservierung verfügt der Sender allein über die Reservierung. Anders bei der gemeinsamen Reservierung, hier teilen sich mehrere Sender eine Reservierung, ohne gegenseitige Beeinflussung. Die Senderauswahl berücksichtigt zwei Varianten, entweder können alle vorhandenen Sender(Wildcard) ausgewählt werden oder die Auswahl wird vom Empfänger getroffen(Explizit). Für die explizite Auswahl können für einzelne Sender Filter eingestellt werden. Der **Wildcard-Filter** ist für eine unbestimmte Anzahl von Sendern bestimmt, die gemeinsam eine Reservierung nutzen. Der Empfänger erhält alle Pakete, die entlang des Reservierungspfades gesendet werden. Der **Shared-Explizit-Filter** generiert eine einzige Reservierung, die von auserwählten Sendern gemeinsam genutzt werden. Der Unterschied

zum Wildcard-Filter besteht darin, das hier einzelne Sender ausgewählt werden können und auch zwischen ihnen gewechselt werden kann. Die gemeinsam genutzten Reservierungen dienen der Schonung von Ressourcen und erhöht die Effizienz der Übertragung. Der **Fixed-Filter** wählt explizit den Sender aus, von dem er Daten empfangen will. Dem ausgewählten Sender steht die angeforderte Reservierung allein zur Verfügung.

Um die verschiedene Flows, die ein Router zu verwalten hat, zu steuern benutzt RSVP Soft-States. Das heißt, die verschiedenen Reservierungen müssen durch ständige Refresh-Messages ihren State[1] aufrecht erhalten. Die Zeitperiode in der dies geschieht, wird in dem Objekt TIME_VALUES. Dieses Objekt wird sowohl für PATH- als auch für RESV-Messages verwendet. Sind diese Timeout-Werte abgelaufen, werden entsprechende Reservierungs- und/oder Pfadinformationen gelöscht und die Ressourcen freigegeben. Diese Methode verhindert die Blockierung von wertvollen Ressourcen bei auftretenden Netzwerkstörungen oder Routenänderungen. Der Zeitwert wird so eingestellt, dass RSVP einen gewissen Verlust an PATH-/RESV-Messages toleriert.

Neben den zwei wichtigsten Message-Typen PATH und RESV gibt es noch weitere Messages.

Die **PathTear-Message** löscht die Zustände in den jeweiligen RSVP-Geräten, die durch PATH- und RESV-Messages angelegt wurden. Entweder wird diese Message vom Sender ausgelöst oder wie oben erwähnt beim Ablauf eines Timeouts.

Die **ResvTear-Message** löscht die Reservierungszustände, die durch entsprechende RESV-Messages angelegt wurden. Falls diese Reservierung noch von anderen Flows benutzt wird, bleibt diese erhalten. Wie bei der PathTear-Message kann diese Message auch explizit, hier durch den Empfänger, oder durch Ablauf eines Timeouts ausgelöst werden.

Eine **PathError-Message** wird ausgelöst, wenn bei der Bearbeitung einer PATH-Message ein Fehler aufgetreten ist. Diese Message besitzt ein Objekt ERROR_SPEC, in dem der Fehler genauer spezifiziert ist, unter anderem die IP-Adresse des RSVP-Routers in dem der Fehler aufgetreten ist, und wird in Richtung Sender geschickt.

[1] Status

Die **ResvError-Message** wird bei Fehlern ausgelöst, die während der Reservierung stattfinden und werden in Richtung Empfänger weitergeleitet. Wie bei der PathError-Message finden sich im Objekt ERROR_SPEC genauere Angaben zum aufgetretenen Fehler, unter anderen auch die IP-Adresse des RSVP-Routers beim dem der Fehler aufgetreten ist.

Die ResvConf-Message ist eine Bestätigungsnachricht, die nach einer erfolgreichen Reservierung ausgelöst wird. Dies geschieht wenn in einer RESV-Message das Objekt RESV_CONFIRM enthalten ist. Die Message wird zum Empfänger weitergeleitet, die IP-Adresse des Empfängers steht im Objekt RESV_CONFIRM.

3. MULTIMEDIAGRUNDLAGEN

Dieses Kapitel stellt die verschiedenen Multimediatechnologien vor und erläutert ihre Funktionsweise näher. Der erste Abschnitt in diesem Kapitel legt die Grundlagen zum Verständnis der Kompressionsalgorithmen bei digitalen Videodaten. Der Abschnitt 3.2 behandelt die digitalen Videoformate MJPEG, MPEG und DivX, im folgenden Abschnitt 3.3 werden die Verfahren der digitalen Audiokompression erläutert. Das Kapitel trägt außerdem noch zum Verständnis der Effekte, die beim Erstellen und Übertragen von Multimediadaten auftreten können, bei.

3.1 Grundlagen Videokompression

Grundsätzlich lassen sich Kompressionsverfahren in zwei verschiedene Arten unterteilen, verlustfreie und verlustbehaftete Kompression. Die Methode der verlustfreien Kompression beseitigt Redundanzen in der zu übertragenden Datenmenge und ist somit auch ein umkehrbarer Prozess. Mit dieser Methode kann im Videobereich aber nur eine sehr geringe Kompression erreicht werden. Anwendungsbereiche für diese Kompressionsart sind vor allem Bereiche in denen die Originaldaten wieder zu 100 Prozent wiederhergestellt werden müssen, z.B. Dokumente oder Programmdateien. Bekannte verlustfreie Kompressions-Algorithmen sind z.B. RLE[1](Run Length Encoding), Huffmann oder LZW[2](Lemple-Ziv-Welch).

Bei verlustbehafteten Verfahren werden Informationen aus dem Videosignal entfernt, so dass der Betrachter keine oder nur geringe Qualitätsverluste bemerkt, dafür aber ein hoher Kompressionsgrad erreicht wird. Dieses Verfahren ist nicht umkehrbar und deshalb lassen sich die Originaldaten nicht wiederherstellen. Je höher die Kompression, desto größer ist auch die Informationsverlust und somit wird die Qualität geringer. Der Kompressionsgrad, der erreicht werden kann, hängt aber auch oft vom Ausgangsmaterial ab. Will man entsprechende Algorithmen auf ihre Qualität prüfen, ist dies nur durch Verwendung

[1] Lauflängenkodierung

[2] Nach den Erfindern benannt

gleicher Testmaterialien möglich. Der Kompressionsgrad beschreibt das Verhältnis zwischen Originaldatenmenge und komprimierter Datenmenge.

3.1.1 RLE(Run Length Encoding)

Dieses Verfahren gehört zu den einfacheren Algorithmen um Daten verlustfrei zu komprimieren. Der Algorithmus ersetzt bei sich wiederholenden Zeichenfolgen, diese durch das Zeichen und die Anzahl der Wiederholungen und einer Markierung. So wird z.B. die Zeichenfolge aaaamsmttttttxxxxxxxx durch $a4msm$t5$x8 kodiert, die Markierung wird durch $ dargestellt. Dieses Verfahren ist ab 4 Wiederholungen effizient. Die Markierungen dienen bei der Dekompression dazu, um zu Erkennen ob ein einfaches Zeichen oder eine komprimierte Zeichenfolge folgt.

3.1.2 Huffmann-Kodierung

Die Huffmann Kodierung arbeitet mit Häufigkeitsanalysen und binären Bäumen. So werden Buchstaben oder Werte, die sehr häufig vorkommen zu kurzen Codewörtern zusammengefasst und selten vorkommende zu längeren Codewörtern. Mit Hilfe der Huffmann Kodierung lassen sich für beliebige Zeichenfolgen immer Codewörter minimaler Länge errechnen. Das Verfahren soll an einem Beispiel deutlich gemacht werden. Die zu kodierende Zeichenfolge lautet „Ulm und um Ulm herum“. In einem ersten Schritt wird die Häufigkeit der auftretenden Zeichen notiert.

Zeichen	n	u	l	m	d	h	R	e
Anzahl	1	5	2	4	1	1	1	1

Aus dieser Tabelle wird nun ein binärer Baum erstellt, in den Knoten wird die Anzahl der Häufigkeiten notiert. Zunächst werden alle Zeichen als Knoten aufgefasst. Danach werden immer 2 Knoten mit den niedrigsten Häufigkeiten zu einem Vaterknoten zusammengefasst. In dem Vaterknoten wird die Summe der Häufigkeiten beider zusammengefassten Knoten gespeichert. Dies wird solange wiederholt bis alle Knoten in den Baum aufgenommen sind. Wenn der Baum komplett ist, wird von der Wurzel an der Pfad zu den Knoten jeweils mit 1(links) oder 0(rechts) gekennzeichnet. So entsteht das Codewort eines Zeichens, in dem man von der Wurzel bis zu dem Knoten die Binärziffern notiert. Der so entstanden Binärbaum wird in Abbildung 3.1 dargestellt.

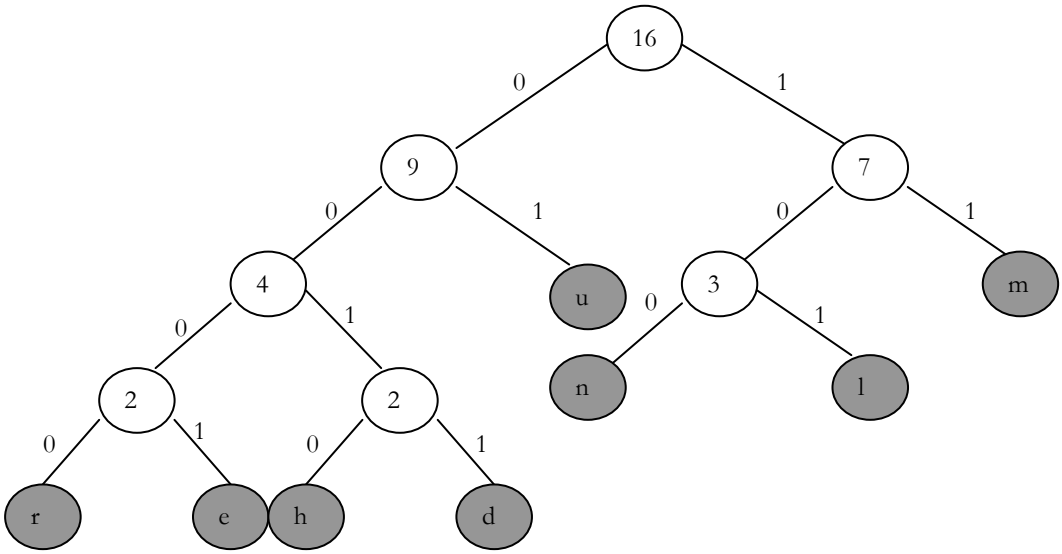

Abbildung 3.1: Binärbaum der Beispielkodierung

Die entstehenden Kodewörter lauten r=0000, e=0001, h=0010, d=0011, u=01, m=11, n=100, l=101. Nun erkennt man, dass Zeichen die häufiger auftreten auch mit kürzeren Codewörtern versehen sind. Diese Eigenschaft ist auch im Morsealphabet benutzt worden. Zur Dekodierung wird der bei der Kodierung gebildete Binärbaum immer benötigt, d.h. er muss zusammen mit den komprimierten Daten gespeichert werden. Die Dekodierung erfolgt, in dem man den Text aus Codewörtern als „Navigationshilfe“ durch den Binärbaum benutzt. Wenn eine 0 folgt nach links und bei einer 1 nach rechts, beim Erreichen eines Blattes wird das entsprechende Zeichen notiert, dann beginnt man wieder von vorn. Zum Beispiel wird der Codetext 0110111, zu 01->u, 101->l und 11->m =ulm dekodiert. Wie man anhand der Kodewörter sehen kann, gibt es immer nur eine Möglichkeit zur Dekodierung. Werden mit Hilfe der Huffman Kodierung nur spezielle Daten(z.B. Texte) komprimiert, eignet sich der Einsatz einer festen Häufigkeitstabelle, somit erspart man sich die Zeit für die Häufigkeitsanalyse am Anfang eines jeden Komprimierungsprozess und muss den Binärbaum nicht mit speichern. Bei Daten deren Zusammensetzung nicht bekannt ist, muss eine individuelle Häufigkeitsverteilung bestimmt werden, um den jeweils günstigsten Huffmann Code zu bestimmen, der dann natürlich mit den komprimierten Daten abgespeichert werden muss.

3.1.3 Diskrete Kosinus Transformation(DCT)

Dieses Verfahren gehört zur Gruppe der Transformationskodierung und wird im JPEG und MPEG Format angewendet. Ziel der Transformation ist, die örtliche Verteilung der Pixel in einem Bild in Frequenz und Amplitudenangaben zu transformieren. Dabei werden regelmäßige Flächen als niedere Frequenzen und detailreichere Bildteile als hohe Frequenzen dargestellt. Um die Datenmenge zu reduzieren, werden die hohen Frequenzanteile geringer bemessen als die niedrigen, da auch das menschliche Auge weniger empfänglich für diese ist. Durch diese Transformation wird noch keine Datenreduzierung erreicht, sondern erst durch das Quantifizieren der sich aus der Transformation errechneten Koeffizienten.

Zur Transformation wird das Bild in 8x8 große Pixelblöcke unterteilt und mit der folgenden Formel transformiert.

$$f(k,n)=\frac{C(k)}{2}\frac{C(n)}{2}\sum_{x=0}^{7}\sum_{y=0}^{7}F(x,y)\cos\left(\frac{\pi(2x+1)k}{16}\right)\cos\left(\frac{\pi(2y+1)n}{16}\right) \quad k,n \in \{0,..,7\}$$

$$C(k),C(n) = \begin{cases} 1/\sqrt{2} \text{ für } k,n = 0 \\ 1 \text{ für } k,n > 0 \end{cases}$$

Durch F(x,y) wird ein 8x8 Pixelblock aus dem Ausgangsbild dargestellt. Die Transformation bewirkt, dass sich die wichtigen Informationen, große Bilddetails, die für das menschliche Auge besser wahrnehmbar sind, in die linke obere Ecke(x=0, y=0) des neuen 8x8 Koeffizientenblocks gespeichert werden und weniger wichtige Informationen, feine Bilddetails, die für das Auge nicht gut wahrnehmbar sind, in der rechten unteren Ecke(x=7, y=7). Der Koeffizient f(0,0) wird dabei als DC-Koeffizient bezeichnet, die restlichen als AC-Koeffizienten. Der DC-Koeffizient bestimmt den Grundton für den gesamten Block. Die Transformation ist ein Verfahren, bei dem keine Informationen verloren gehen, bis auf Rechenungenauigkeiten.

Da die DCT ein umkehrbares Verfahren ist, lässt sich aus der Koeffizientenmatrix auch wieder das Ausgangsbild erstellen. Die Formel für die inverse DCT lautet:

$$F(x,y)=\sum_{k=0}^{7}\sum_{n=0}^{7}\frac{C(k)}{2}\frac{C(n)}{2}f(k,n)\cos\left(\frac{\pi(2x+1)k}{16}\right)\cos\left(\frac{\pi(2y+1)n}{16}\right) \quad x,y \in \{0,..,7\}$$

$$C(k),C(n) = \begin{cases} 1/\sqrt{2} \text{ für } k, n = 0 \\ 1 \text{ für } k, n > 0 \end{cases}$$

Analog zur DCT wird durch F(x,y) ein Pixelblock des Ausgangsbildes dargestellt und f(x,y) beschreibt die Koeffizientenmatrix.

Um nun die erwünschte Kompression zu erreichen müssen die durch die Transformation entstandenen Koeffizienten noch quantisiert werden. Bei der Quantisierung werden die entstehenden Werte in diskrete Stufen eingeteilt. Je nach dem wie viele Stufen für die Quantisierung verwendet werden, kann der Kompressionsfaktor eingestellt werden. Je mehr Quantisierungsstufen, desto höher die Qualität und desto niedriger der Kompressionsfaktor. Je gröber die Einteilung der Quantisierungsstufen desto höher ist auch der Kompressionsfaktor und desto niedriger die Qualität. Für jeden DCT-Koeffizient kann die Quantisierungsstufe einzeln festgelegt werden, gebräuchlicher ist aber, dass der DC-Koeffizient mit 8 und die AC-Koeffizienten mit 6 quantisiert werden. Diese Einstellungsmöglichkeit hat den Vorteil, dass bestimmte Frequenzen mehr Bedeutung bekommen als andere. Somit können hohe Ortsfrequenzen, die der Mensch nur schlecht oder überhaupt nicht wahrnimmt, gezielt weggelassen werden, indem diese Bereiche entsprechend grob quantisiert werden. In diesem Schritt tritt dann natürlich ein Datenverlust auf, der nicht wiederhergestellt werden kann.

Bei dem Quantisierungsverfahren wird die Koeffizientenmatrix im Zick-Zack-Verfahren (siehe Abbildung 3.2) von oben links nach unten recht abgearbeitet, das dient zur Vorsortierung für die RLE-Komprimierung, da in der unteren Hälfte der Matrix viele gleiche Werte durch die grobe Quantisierung entstehen.

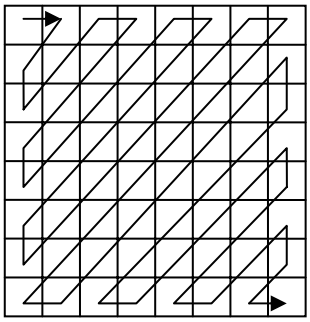

Abbildung 3.2: Zick-Zack-Abtastung der Koeffizientenmatrix

Im praktischen Einsatz, z.B. bei JPEG werden so genannte Quantisierungstabellen verwendet. Jedem AC-Koeffizient wird in dieser Tabelle in eigner Wert zugeordnet mit dem er quantisiert wird, d.h. der Koeffizient wird durch den entsprechenden Wert dividiert und auf eine ganze Zahl gerundet. In den verwendeten Tabellen werden besonders die hochfrequenten Teile sehr grob quantisiert. Diese Tabellen sind auf das Sehvermögen des menschlichen Auges hin erstellt und angepasst worden, getrennt für jede Farbkomponente.

3.2 Digitale Video Formate

In diesem Abschnitt geht es um verschiedene Video Formate, welche Vor- und Nachteile diese besitzen. Weiterhin werden auch verschiedene Einsatzgebiete erläutert, in denen die Codecs[1] eingesetzt werden. Die Formate unterscheiden sich z.B. durch die eingesetzten Kompressionsalgorithmen, Auflösung oder auch in der Bitrate. Diese für die Qualität entscheidenden Merkmale sollen hier näher herausgearbeitet und untereinander verglichen werden.

3.2.1 MJPEG

Die Abkürzung MJPEG steht für Motion Joint Picture Experts Group. Dieses Format basiert auf einem Videostrom aus einzelnen im JPEG-Verfahren komprimierten Einzelbildern. Dies ist auch ein entscheidender Vorteil gegenüber anderen Verfahren, hier kann der Benutzer auf alle Einzelbilder des Videostroms zugreifen. Durch diesen Vorteil ist dieser Codec auch größtenteils im Videoschnittbereich zu finden. Ein weiterer Vorteil von

[1] Abkürzung für **Co**mpressor-**Dec**ompressor

MJPEG ist, dass zwischen den Einzelbildern auch keine Abhängigkeiten bestehen und somit bei einer Netzübertragung nur geringe Verzögerungen entstehen.

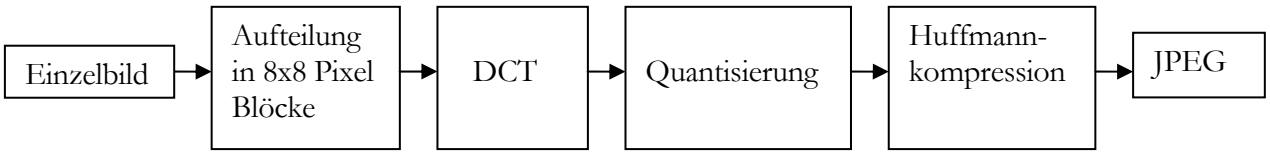

Abbildung 3.3: Ablauf des JPEG Verfahrens zur Kompression von Einzelbildern

Das Verfahren beginnt mit der Aufteilung eines Einzelbildes in 8x8 Pixel große Blöcke. Diese werden dann durch die diskrete Cosinus-Transformation(siehe 3.1.3) in 8x8 große Matrizen transformiert, in der die DCT-Koeffizienten gespeichert werden. Die entstandenen Koeffizienten werden anschließend quantisiert(siehe 3.1.3) und mit dem Huffmann-Algorithmus nochmals komprimiert. In Abbildung 3.3 wird dieser Prozess noch einmal anschaulich skizziert.

3.2.2 MPEG

Im Jahr 1988 wurde von der ISO(International Standardization Organisation) das Standardisierungsgremium MPEG(Motion Picture Expert Group) gegründet. Diese Expertengruppe hat die Aufgabe Standards im Bereich digitale Audio- und Videoverarbeitung zu definieren und zu verabschieden.

Als erstes Ergebnis wurde 1991 der MPEG-1 Standard veröffentlicht. Dieser Standard, auch als ISO-11172 bezeichnet, ist für die Single-Speed CD-ROM Laufwerke ausgerichtet worden. Das bedeutet, das die Übertragungsrate maximal 1.5 MBit/s betrug, bei einer Bildwiederholungsfrequenz von 25-30 Hz. MPEG-1 stellt außerdem auch Möglichkeiten bereit, um Bild und Ton synchron zu synchronisieren. Die Auflösung für MPEG-1 wurde auf maximal 352x288 Pixel beschränkt. Die Anwendung von MPEG-1 fand sich zum Beispiel im CD-I, Video-CD oder auch im Laserdisc Format wieder.

Ein weiterer Standard MPEG-2 folgte 1995, der Hauptunterschied zu MPEG-1 liegt in der höheren Datenrate, diese kann zwischen 1.5 MBit/s und 15MBit/s liegen. Dieser Standard ist wesentlich flexibler werden als der Vorgänger, so ist z.B. eine ganze Reihe von möglichen Auflösungen definiert, mit denen MPEG-2 arbeiten kann. MPEG-2 definiert die Auflösungen Low(352x240), Main(720x480), High1440(1440x1152) und High(1920x1080). Die beiden Auflösungen, die als High1440 und High benannt wurden, sind für das Fernsehformat HDTV(High Definition Teevision) entwickelt wurden. Für normale Fernsehbilder ist die Main Auflösung definiert und Low wurde lediglich aus Kompatibilitätsgründen mit in den Standard aufgenommen. Die bedeutendsten Einsatzgebiete von MPEG-2 sind digitale Fernsehübertragungen über Satellit oder Kabel und DVD. Als Audiosignal wurde erstmals ein Mehrkanalton eingeführt, der die zwei Stereokanäle, den Frontkanal und zwei Raumklang-Kanäle umfasst.

Der MPEG-3 Standard sollte für das HDTV entwickelt werden, da aber MPEG-2 HDTV in vollem Umfang unterstützt wurde die Arbeit an MPEG-3 eingestellt.

Seit 1993 wird an dem MPEG-4 Standard gearbeitet, dieser ist besonders für niedrige Datenraten geeignet sein. Die Eignung für niedrige Datenraten(bis 1MBit/s) soll den Standard vor allem im mobilen Bereich etablieren. Ein wichtiger Punkt stellt auch die Objektorientierung im Videobereich dar, dabei soll es möglich sein, ein Bild in verschiedene Objekte aufzuteilen und entsprechend weiter zu verarbeiten. Diese so genannte Inhaltsbezogene Interaktivität soll die Wiederverwertung von Mutltimediadaten erhöhen. Um diese Objekte zu bearbeiten, definiert MPEG-4 die MSDL(MPEG-4 Syntactic Description Language). Diese Sprache stellt dem Anwender alle Werkzeuge zu Verfügung, die für die Bearbeitung der Multimediaobjekte benötigt werden, ohne dass der Anwender dabei die digitale Darstellung des Objektes kennen muss. Einzelne Objekte einer Szene können z.B. skaliert und stärker betont werden als andere, ebenfalls können Objekte auch in unterschiedlicher Auflösung gespeichert werden. Aufgrund der hohen Komplexität dieses Standards wurde erst 1998 die erste Version dieses Standards verabschiedet. Bis heute existiert noch keine vollständige Implementation des MPEG-4 Standards, lediglich Teile dieses Standards wurden in verschiedenen Codecs implementiert, z.B. in DivX oder in Microsoft Windows Media Video.

MPEG-7 stellt den neuesten Standard im Multimediabereich dar. Der Schwerpunkt für die Entwicklung dieses Standards wurde auf die Indizierung und Beschreibung von Multimediaobjekten gelegt. So soll es möglich sein, mittels Suchmaschine in audiovisuellen Inhalten nach Stichwörtern oder Beschreibungen zu suchen. MPEG-7 definiert dazu das Format mit dem das Material in multimedialen Datenbanken abgelegt wird.

Nachdem die verschiedenen Standards der MPEG Gruppe vorgestellt wurden, folgt im nächsten Abschnitt eine Erläuterung der den MPEG Standards zu Grunde liegenden Technik und eine Beschreibung der Abläufe bei Kodierung und Dekodierung.

Das MPEG-Kodierungsverfahren läuft im Wesentlichen in 5 Teilen ab. Der erste Schritt ist **Reduzierung der Auflösung**(abhängig vom Ausgangsmaterial), danach wird die **Bewegungskompensation** angewendet, nach Transformation durch **DCT** und **Quantisierung** werden die Daten durch **Huffmann Kodierung** nochmals reduziert.

Die Reduzierung der Auflösung wird vor allem bei der MPEG-1 Kodierung angewendet, da MPEG-2 höhere Auflösungen bietet und eine Reduzierung somit überflüssig ist. In diesem Schritt wird auch eine so genannte Farbraumtransformation durchgeführt, d.h. die im Quellmaterial vorliegenden Daten sind oft im RGB[1]-Format gespeichert. Diese werden in das YCbCr[2](Luminanz(Y)Chrominanz(CbCr))-Format umgewandelt. Die Transformation geschieht, weil das menschliche Auge Helligkeitsinformation besser aufnehmen kann als Farbunterschiede. Es wird auf vier Lumianzdaten nur eine Chrominanzinformation gespeichert, deshalb reduzieren sich die Chrominanzdaten um 50 Prozent.

Die Ähnlichkeit aufeinander folgender Bilder in einem Videostrom wird von der blockorientierten Bewegungskompensation(Motion Compensation) ausgenutzt. Für dieses Verfahren werden unterschiedliche Bildarten benötigt. Das **I-Bild**(Intraframe) ist für den wahlfreien Zugriff auf das Video zuständig, da keine Abhängigkeiten zwischen anderen Bildern bestehen, quasi ein Standbild darstellt. Da I-Bilder keinen Abhängigkeiten

[1] Rot Grün Blau

[2] Y(Helligkeitsanteil)/CbCr(Farbanteile)

unterstehen, dienen sie als Referenz für die anderen Bildtypen. Ein typischer Wert für ein MPEG-Datenstrom ist 15 I-Bilder(oder I-Frames) pro Sekunde. Das **P-Bild**(Predicted) wird aus vorherigen I- oder P-Bildern errechnet. Hier besteht also immer die Abhängigkeit zum Vorgängerbild, deshalb muss bei Kodierung und Dekodierung immer das Referenzbild bekannt sein. Um auf ein einzelnes P-Bild zuzugreifen, muss das letzte I-Bild und alle dazwischen befindlichen P-Bilder dekodiert werden. Das **B-Bild**(Bidirectional) berechnet sich aus Vorgänger- und Nachfolgerbild und besitzt somit die höchste Abhängigkeit. B-Bilder können selbst als Referenz auf andere Bilder nicht verwendet werden. Die B-Bilder sind praktisch das Ergebnis einer Interpolation zwischen Vorgänger- und Nachfolgerbild. Die Anzahl der I-Bilder in einem MPEG-Video ist ein maßgeblicher Wert für die Qualität des Videos. Abbildung 3.4 zeigt eine typische Bildfolge in einem MPEG-Datenstrom und zwei Beispiele für mögliche Abhängigkeiten zwischen Bildern.

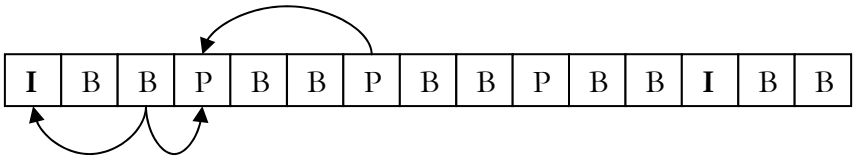

Abbildung 3.4: Typische Bildfolge in einem MPEG-Video

Es gibt zwei unterschiedliche Arten von Bewegungskompensation, die interpolierende und die prädikative Bewegungskompensation.

Bei der interpolierenden Technik wird der Videostrom aus der Interpolation eines Vorgängerbildes und eines Nachfolgerbildes und der Anwendung einer Korrekturfunktion wiederhergestellt. Der Kodierungsprozess hat darauf zu achten, dass die Bilder in der richtigen Reihenfolge gespeichert werden, da zeitlich nachkommende Bilder referenziert werden. Die Art der Bewegungskompensation wird bei der Erstellung der B-Bilder angewendet. Dieser Prozess ist stark asymmetrisch, d.h. die Kodierung ist um ein vielfaches komplexer und zeitintensiver als die Dekodierung.

Bei der prädikativen Bewegungskompensation wird das Bild in 16x16 Pixel große Blöcke aufgeteilt, so genannte Makroblöcke, aufgeteilt. Es wird nun für jeden Makroblock ein Verschiebungsvektor errechnet, d.h. es wird versucht ein Makroblock mit einem minimalen

Prädikationsfehler, relativ zum Ausgangsmakroblock, zu finden. Diese Fehler entstehen natürlich, weil sich Bilddetails nicht ohne Veränderung von Bild zu Bild fortpflanzen. Als Bemessung für den Prädikationsfehler wird z.B. der mittlere quadratische Abstand eingesetzt. In diesem Fall wird von den beiden zu vergleichenden Blöcken die Summe der Differenzen aller Luminanz- und Chrominanzwerte gebildet, wobei der Idealfall der Prädikationsfehler gleich null ist. Diese Technik wird zur Kodierung der P-Bilder eingesetzt.

Zur Kodierung der Einzelbilder wird die in 3.1.3 beschriebene Diskrete Kosinus Transformation und die darauf folgende Quantisierung angewendet. Zum Abschluss wird die Datenmenge mittels Huffmann Kodierung noch weiter reduziert.

3.2.3 DivX

Der Name DivX stammt von einem gescheiterten Ausleihverfahren für DVDs. Das System sollte den Namen „Digital Video Express“ tragen, doch es wurde nie eingeführt. Der Kunde sollte für jeden Abspielvorgang einer ausgeliehenen DVD extra bezahlen. Der fällige Betrag sollte dann per Modem automatisch abgebucht werden.

Heute bezeichnet DivX einen hochkomprimierenden Videocodec, der in der Lage ist einen kompletten DVD-Film auf eine Größe schrumpfen zu lassen, dass er auf eine oder zwei CDs passt, bei akzeptabler Bildqualität. DivX wird auch als gehackter Microsoft MPEG-4 Codec bezeichnet. Der Grund dafür ist, dass DivX eine modifizierte Microsoft Implementation des MPEG-4 Standards ist. So wurden bestimmte Limitationen, die Microsoft in den Codec aufgenommen hat, beseitigt. Microsoft hat die Datenrate auf maximal 256 KBit/s beschränkt, da dies für die Kodierung eines Videofilms in mindestens VHS-Qualität zu gering war, wurde diese Einschränkung aufgehoben. Die Entwickler von DivX erweiterten die Datenrate auf maximal 6000 KBit/s, für eine Qualität, die etwa VHS entspricht, reichen aber schon 650 KBit/s. Ebenfalls wurde vor der Framecodierung ein Weichzeichner vorgeschaltet, da die MPEG Codierung bei hohen Kontrastfrequenzen und einer geringen Datenrate unansehnliche Artefakte[1] verursacht. Um diese hohen Kontrastfrequenzen zu verhindern wird ein Weichzeichner verwendet, der aber als Nachteil eine gewisse Unschärfe produziert.

[1] Blockbildung

3.3 Digitale Audiokompression

Zur Abrundung des Kapitels Multimediagrundlagen gehört natürlich auch ein Abschnitt über Audiokompression dazu. Auch in diesem Bereich hat die MPEG Gruppe entsprechende Standards definiert und verabschiedet. Von dieser Organisation wurden bis jetzt drei verschiedene Audio Standards definiert, MPEG-1 Layer 1 bis MPEG-1 Layer 3. Die Unterschiede der drei Standards werden in Abbildung 3.5 verdeutlicht.

Standard	Datenrate	Komprimierungsverhältnis
MPEG-Layer 1	384 kBit/s	1:4
MPEG-Layer 2	256 bis 192 kBit/s	1:6 bis 1:8
MPEG-Layer 3	128 bis 112 kBit/s	1:10 bis 1:12

Abbildung 3.5: Unterschiede zwischen den MPEG Audiostandards

Die Abbildung 3.5 zeigt einen Vergleich von Eigenschaften der drei MPEG Verfahren unter der Annahme gleich bleibender Audioqualität. Wie zu sehen ist erreicht der Layer 3 das höchste Kompressionsverhältnis und ist zugleich das komplexeste der drei Verfahren. Im nachfolgenden Abschnitt soll dieses Verfahren ein wenig genauer erläutert werden. Es wird auch als Abkürzung für MPEG-1 Layer 3 die Bezeichnung MP3 benutzt.

Als Eingabedaten für die MP3-Komprimierung dienen digitalisierte Audiosignale mit einer Abtastfrequenz von 32, 44.1 oder 48 kHz.

Das menschliche Gehör kann Frequenzen von 20 Hz bis 22 kHz wahrnehmen, dabei reagiert es jedoch nicht auf jede Frequenz gleich empfindlich. Das bedeutet, dass eine Frequenz eine bestimmte Lautstärke besitzen muss, um von Menschen gehört zu werden. Diese Eigenschaft wird auch als Hörschwelle bezeichnet. Wie in Abbildung 3.6 zu sehen, reagiert das menschliche Gehör auf Frequenzen 2-4 kHz am empfindlichsten und Töne ab 14 kHz müssen schon mindestens 35 dB laut sein um gehört zu werden. Diese Hörschwelle wurde mit Hilfe von Testpersonen ermittelt, da es keine andere Möglichkeit gibt, diese zu ermitteln.

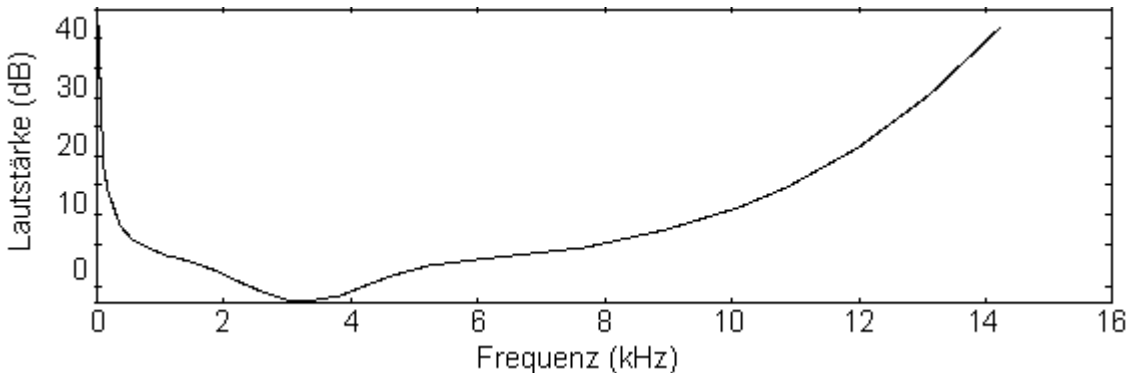

Abbildung 3.6: Hörschwelle des menschlichen Gehörs

Wenn ein Ton von 4 kHz bei 60 dB gehört wird, müssen Töne die sich im nahen Frequenzbereich dieses Tones befinden, nun wesentlich lauter sein um gehört zu werden. Die als simultane Maskierung bezeichnete Eigenschaft verändert die Hörschwelle so, dass bei zunehmender Frequenz des Maskierungstons immer größere Frequenzbereiche ausmaskiert werden, d.h. bei gleicher Lautstärke nicht mehr hörbar sind. Abbildung 3.7 zeigt ein Beispiel wie sich 4 verschiedene Töne auf die Hörschwelle auswirken. Dabei sind die Töne, die unter der jeweiligen Kurve des Maskierungstons liegen nicht mehr hörbar. Der hörbare Frequenzbereich kann in 27 kritische Frequenzbänder eingeteilt werden.

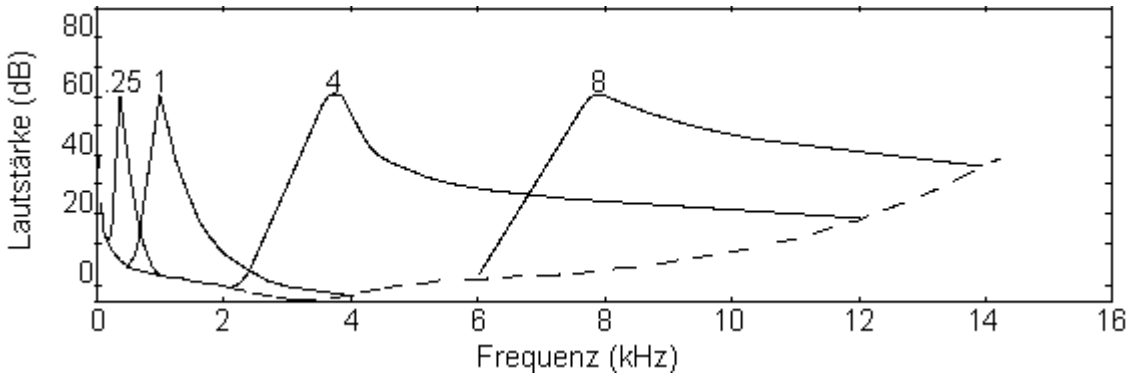

Abbildung 3.7: Maskierung des Frequenzbandes durch 4 verschiedene Töne

Es existiert noch eine weitere Art der Maskierung, die zeitliche Maskierung. Wenn ein Ton vom Gehör wahrgenommen wird, dauert es ungefähr 5-20 ms bis das Ohr leisere Töne ähnlicher Frequenz wieder hören kann. Diese „Verzögerung“ ist abhängig vom Verhältnis der Frequenzen und deren Lautstärke.

All diese Eigenschaften des menschlichen Gehörs werden bei der MPEG-Audio Kodierung berücksichtig und ausgenutzt um die anfallenden Datenmengen zu reduzieren. Die Datenmengenreduktion arbeitet verlustbehaftet.

Die Kodierung der Audiodaten erfolgt in mehreren Teilschritten. Zuerst werden die Daten vom Zeitbereich in den Frequenzbereich transformiert, ähnlich wie bei der JPEG Kodierung. Die Transformation übernimmt die so genannte **Filterdatenbank**, wobei die Daten in 32 Frequenzbänder unterteilt werden und pro Subband ein Sample[1] ausgegeben wird. Diese Einteilung in Subbänder dient als Abschätzung für die kritischen Frequenzbänder. Die Transformation wird, wie auch bei der Videokompression, durch eine diskrete Kosinustransformation durchgeführt.

Im **psychoakustischen Modell** wird dann ermittelt, wie stark die Maskierung in den einzelnen Subbändern nun tatsächlich ist. Um diese Berechnungen zu vereinfachen, werden in dem Modell die Maskierungskurven in Kurven gleicher Größe transformiert. Das psychoakustische Modell nimmt weiterhin eine Unterteilung in tonale und nicht-tonale Objekte vor, um beispielsweise das Geräusch eines Schlagzeuginstrumentes(nicht-tonal) von einem anderen Ton zu trennen, da die beiden Arten unterschiedlich maskiert werden. Für jedes Subband wird eine minimale Maskierung aus der Maskierung der kritischen Frequenzbänder errechnet. Im Detail muss verglichen werden, wie breit ein Subband in einem kritischen Frequenzband liegt. Ist das Subband sehr breit, relativ zum überlappenden kritischen Frequenzband, wird die minimale Maskierung überlappenden kritischen Frequenzbandes übernommen, anderenfalls wird eine durchschnittliche Maskierung aus allen Frequenzen dem Subband zugeordnet. Für die Quantisierung werden noch die Quotienten aus maximaler Signalintensität aus dem entsprechenden Band und minimaler Maskierungsintensität ermittelt.

Im Anschluss daran werden die Audiodaten dem Quantisierungsmodul übergeben. Liegt die Intensität eines Signals in einem Subband unter der Maskierung des Bandes wird es nicht mit kodiert, da es nicht hörbar ist. Es existieren für ein Subband 3x12 Samples, die zu einem Sampleblock zusammengefasst werden. Die Abbildung 3.8 verdeutlicht diese Einteilung der Samples.

[1] Aus der Diskretisierung entstandener Wert

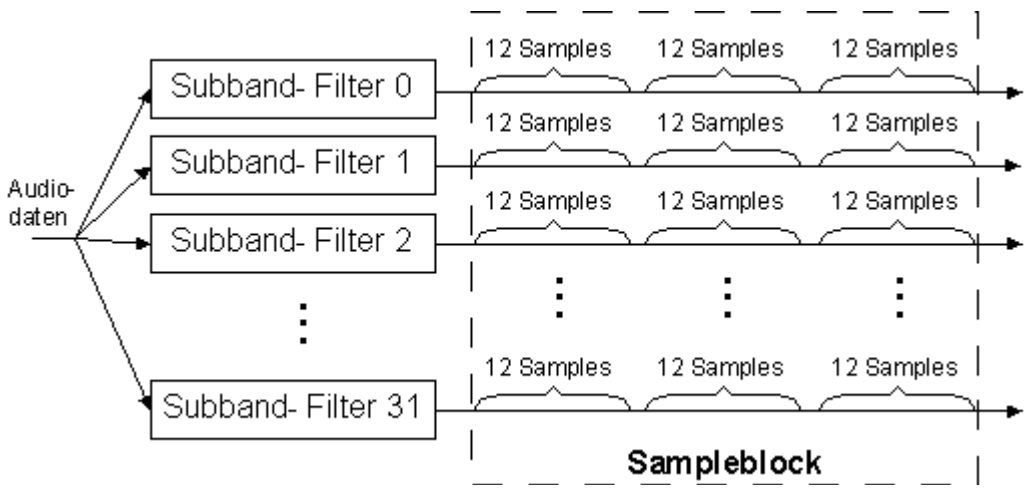

Abbildung 3.8: Einteilung der Samples in Blöcke

Pro Sampleblock wird die Anzahl Bits gespeichert, die zur Quantisierung der Samples benutzt wird. Um den gesamten Bereich des Quantisierers auszunutzen, werden pro Sampleblock noch bis zu 3 Skalierungsfaktoren gespeichert, also maximal ein Faktor für 12 Samples. Der Quantisierungsprozess prüft ob das durch die Quantisierung entstandene Rauschen im nicht-hörbaren Bereich liegt. Wenn dies nicht der Fall ist, wird eine feinere Quantisierung in den betreffenden Subbändern ausgewählt. Die quantisierten Samples werden danach durch eine Huffmann-Kodierung noch komprimiert.

Für die Stereo-Kodierung gibt es verschiedene Möglichkeiten. Im Prinzip ist der Stereo-Effekt eine Pegel/Phasen-Differenz zwischen rechten und linken Kanal. Die bestehende Redundanz zwischen rechten und linken Kanal wird von beiden Verfahren unterschiedlich ausgenutzt. Die erste Kodierungsmöglichkeit ist die Intensitätskodierung. Dabei werden tiefe Frequenzen für jeden Kanal einzeln kodiert und für hohe Frequenzen wird nur ein Summensignal für linken und rechten Kanal kodiert, das aber mit einem Skalierungsfaktor für jeden Kanal ausgestattet wird. Das andere Verfahren wird als MS(Middle/Side)-Kodierung bezeichnet. Im MS Verfahren wird ein Mittensignal, bestehend aus der Summe von linken und rechten Kanal, und ein Seitensignal, bestehend aus der Differenz von linken und rechten Kanal. Das Seitensignal trägt sehr wenig Information(weniger als ein Monosignal) und kann deswegen höher komprimiert werden als das Mittensignal.

4. QOS SZENARIEN

In diesem Kapitel sollen die Vor- und Nachteile der verschiedenen QoS Szenarien dargestellt werden. Es werden Probleme, die bei einem Einsatz von QoS in einem vorhandenen Netzwerk auftreten können und entsprechende Lösungen gezeigt.

4.1 RSVP/IntServ über Ethernet

Damit über Ethernet eine garantierte Übertragungsqualität realisiert werden kann, müssen verschiedene Voraussetzungen erfüllt werden. Im neuen Standard IEEE 802.1Q ist im Ethernet Frame ein Feld für die Priorität definiert. Damit ist es möglich Ethernet-Frames in verschiedene Verkehrsklassen aufzuteilen und entsprechend ihrer Priorität weiterzuleiten. Dies ist eine Grundvoraussetzung um eine Dienstgüte zu garantieren. Ein Problem kann entstehen wenn pro Ethernetsegment mehrere Sender existieren, d.h. bei diesem Shared-Ethernet kann es passieren das mehrere Stationen gleichzeitig versuchen zu senden und sich dabei gegenseitig blockieren. In diesem Fall ist es schwer bis unmöglich Garantien für maximale Verzögerung zu geben. In einem Switched-Ethernet ist dies kein Problem, da hier pro Segment nur ein Sender existiert. Ein Switch der dem oben erwähnten neueren Ethernetstandard genügt, leitet die Ethernetpakete entsprechend ihrer Priorität weiter, so das Garantien für maximale Verzögerung und Paketloss getroffen werden können. Damit Anwendungen und höhere Protokolle QoS des Netzes nutzen können sind noch einige Komponenten notwendig, die in den nächsten Abschnitten genauer erläutert werden.

4.1.1 Bandwidth Manager

Diese Komponente übernimmt die Aufgabe der in Abschnitt 2.3.4 beschriebenen Admission Control. Der Bandwidth Manager besteht aus zwei Teilkomponenten, dem **Requester Module**(RM) und dem **Bandwidth Allocator**(BA)(siehe auch [20], [21]).

Der Bandwidth Allocator ist für die Verwaltung und Zuteilung der Ressourcen in einem Subnetz zuständig. Diese Teilkomponente bearbeitet z.B. Anfragen von Hosts nach Reservierungen oder Änderungen bestehender Reservierungen. Die Kommunikation zwischen Host und BA erfolgt durch das Requester Module, das in jedem Host bzw. jeder

Endstation zu finden ist. Der Bandwidth Allocator kann auf unterschiedlicher Art im Netz implementiert werden, als zentrale oder dezentrale.

Die zentrale Implementation kommt beispielsweise dann in Frage wenn die vorhandenen Netzwerkkomponenten wie Switch oder Bridge keine Bandwidth Allocator Funktion besitzen. Ein Gerät übernimmt dann die Funktion des BA für das gesamte Subnetz. Es besteht dadurch aber auch die Gefahr, dass dies ein Engpass im Netzwerk werden kann. Die RM in den beteiligten Hosts die eine Reservierung anfragen, kommunizieren mit dem BA in ihrem Subnetz. In größeren Subnetzen wird ein Gerät die gesamten Anfragen des Subnetzes möglicherweise nicht bearbeiten können, deshalb ist es nötig weiteren Geräten als BA eine Anzahl von Segmenten in dem Subnetz zuzuweisen. Die folgende Abbildung 4.1 verdeutlicht das Zusammenspiel der beteiligten Komponenten bei der zentralisierten Implementation des Bandwidth Allocator.

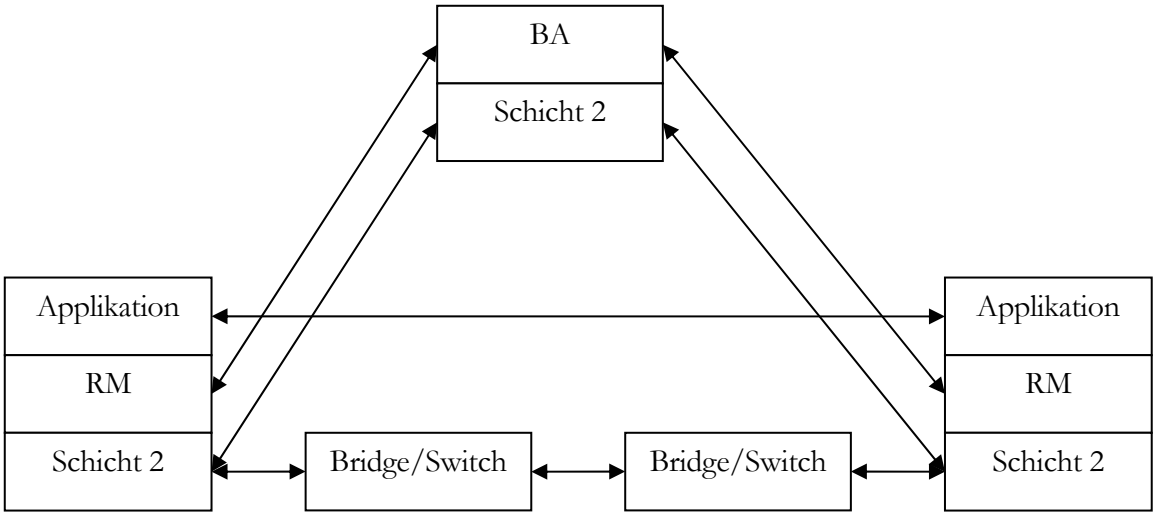

Abbildung 4.1: zentralisierte Implementation des Bandwidth Allocator

Bei einer verteilten Implementation des BA müssen alle Geräte im Subnetz die Funktionalität des BAs besitzen. Die Hosts müssen zusätzlich noch ein Requester Module, wie in der zentralisierten Variante, besitzen. In der verteilten Variante nehmen alle Geräte im Subnetz aktiv an der Admission Control teil. Im Gegensatz zur zentralen Implementation müssen hier die einzelnen BAs in den Geräten nur lokale Topologie

Informationen kennen, wie z.B. welche sind die aktiven Ports und welche Unicast-Adressen sind von welchen Ports aus erreichbar.

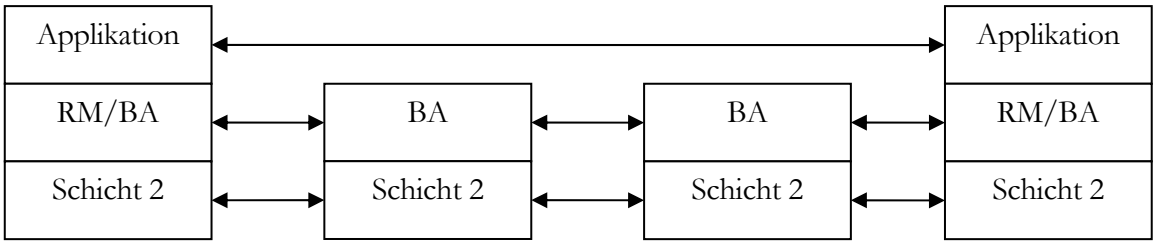

Abbildung 4.2: Verteilte Implementation des Bandwidth Allocator

Der zweite Bestandteil des Bandwidth Managers ist das Requester Module, das sich in jeder Endstation des Subnetzes befindet. Eine wesentliche Funktion besteht darin, den Applikationen und den höher gelegenen Protokollen wie z.B. RSVP ein Interface zum Bandwidth Manager zur Verfügung zu stellen. Bei einer Reservierung muss die Applikation dem RM entsprechende Parameter übergeben, wie der gewünschte Service(Guaranteed oder Controlled Load), Verkehrsbeschreibung TSpec und die Reservierungsbeschreibung RSpec(siehe auch 2.3.3). Das Requester Modul erfüllt im Sender und im Empfänger unterschiedliche Aufgaben.

Im Sender erhält das RM Anfragen von höher liegenden Schichten oder Applikationen, ob einem Flow eine bestimmte Dienstgüte gewährleistet werden kann, im Fall von RSVP kommt diese Anfrage von dem auf dem Sender laufenden RSVP-Prozess. Die Kommunikation zwischen Requester Module und Bandwidth Allocator erfolgt mit Hilfe des Signalisierungsprotokolls SBM(Subnet Bandwidth Manager). Dieses Protokoll wird im nächsten Abschnitt 4.1.2 noch etwas näher erläutert. Wird im Subnetz ein verteilter Bandwidth Allocator eingesetzt müssen bei einer Reservierungsanfrage der lokale und der nächstliegende BA in Richtung des Empfängers befragt werden.

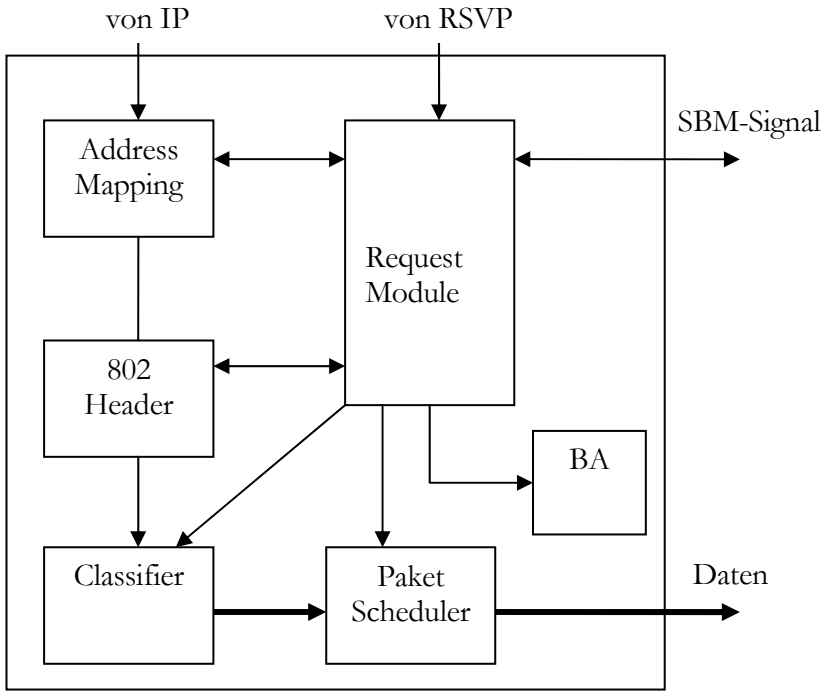

Abbildung 4.3: Zusammenspiel der Komponenten in einem Sender

In Abbildung 4.3 ist die Skizze eines Senders dargestellt, bei der das Zusammenspiel zwischen Requester Modul und den anderen Komponenten zu sehen ist. Die Address-Mapping Komponente sorgt für die Bereitstellung der Adresse des nächsten BAs. Der in Abbildung 4.3 zu sehende lokale Bandwidth Allocator ist natürlich nur im Fall einer verteilten Implementierung vorhanden. Ist eine erfolgreiche Reservierung durchgeführt wurden, erhält das RM eine User Priority vom BA zurück. Mit dieser User Priority werden die zu dieser Reservierung zugehörigen Pakete versendet. In der 802-Header Tabelle wird gespeichert welche User Priority welchem Flow zugeordnet wird. Die Informationen aus dieser Tabelle werden vom Classifier benutzt um die Pakete entsprechend zu ihrer Reservierung zu kennzeichnen.

Der Aufbau des Requester Module im Empfänger ist nicht so komplex wie im Sender, wie in Abbildung 4.4 zu sehen. Das Modul befragt den lokalen BA, ob entsprechende Ressourcen für den Fluss vorhanden sind, wie etwa der Empfangspuffer. Weiterhin besteht die Aufgabe des RM, dem RSVP Protokoll zu melden, ob eine Reservierung möglich ist und die Übermittlung der Antwort zum anfragenden Host mit Hilfe des SBM Protokolls.

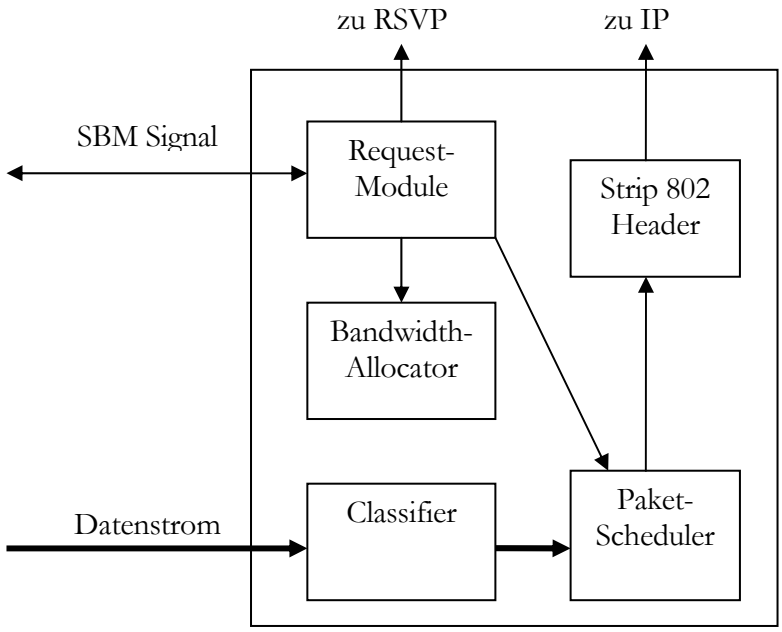

Abbildung 4.4: Zusammenspiel der Komponenten im Sender

Die Abbildung 4.4 zeigt den Aufbau des Senders und das Zusammenspiel zwischen dem Requester Module und den beteiligten Komponenten. Die Komponente **Strip 802 Header** entfernt die für die Übertragung hinzugefügten Prioritäts- und Identifikations-informationen aus den ankommenden Paketen. Wie in Abbildung 4.4 zu sehen, kann das RM auch vorhandene Paketscheduler beeinflussen und konfigurieren, um bestimmte Verkehrsklassen zu identifizieren und entsprechend weiterzuleiten oder zu reservierende Puffer bereitzustellen.

4.1.2 SBM Protokoll

Die Abkürzung SBM steht für Subnet Bandwidth Manager und bezeichnet ein Signalisierungsprotokoll für RSVP-basierte Admission Control in Netzwerken die dem IEEE 802 Standard genügen. Dieses Protokoll wird für die Kommunikation zwischen den RSVP-Hosts und den Layer 2 Geräten, wie Switches oder Bridges eingesetzt, um die Reservierung von Ressourcen der RSVP Flüssen in IEEE 802 Netzwerken gewährleisten. Ein Bandwidth Allocator wird als SBM-fähig bezeichnet. Bei mehreren SBM-fähigen Geräten wird ein Gerät ausgewählt, das als **DSBM**(Designated Subnet Bandwidth Manager) fungiert. In einem Segment muss mindestens ein DSBM vorhanden sein, der dann die Funktion des Bandwidth Allocator für das Segment übernimmt. Ein Segment in

dem ein DSBM vorhanden ist, wird auch als **Managed Segment** bezeichnet. Der DSBM beantwortet Anfragen der RSVP-Clients ob entsprechende Ressourcen in dem Segment vorhanden sind. Die Informationen dafür werden aus den RESV-Messages ausgelesen. Um die RSVP-Messages vom Sender zum Empfänger über Managed Segmente zu leiten, müssen die Messages um einige Objekte erweitert werden. Wird die Admission Control bestanden, sendet der DSBM die erweiterten RESV-Messages zum nächsten RSVP-Prozess. Bevor der RSVP-Prozess die Messages bearbeitet, werden die hinzugefügten Objekte wieder entfernt, da diese nur für den Transport über Managed Segmente benötigt werden. Da nur RESV-Messages von der DSBM weitergeleitet werden, die die Admission Control bestanden haben, kann Bandbreite und Zeit gespart werden, da der RSVP-Prozess nicht nach erhalt einer RESV-Message noch eine Anfrage zur DSBM senden muss, ob eine entsprechende Reservierung gewährleistet werden kann. Folgende Objekte werden den RSVP-Messages beigefügt: RSVP_HOP_L2, LAN_NHOP und TCLASS. Diese Objekte haben folgende Funktionen(siehe auch [19]):

RSVP_HOP_L2:

Dieses Objekt speichert die MAC-Adresse[1] des letzten RSVP Knotens. Das Objekt wird benötig, da nicht alle SBM-Geräte Zugang zu den Layer 3 Routing Informationen haben oder mit Hilfe von ARP die MAC-Adresse aus der IP-Adresse ermitteln können.

LAN_NHOP:

In diesem Objekt werden die IP-Adresse und die MAC-Adresse des nächsten Layer 3 RSVP Knotens gespeichert.

TCLASS:

In TCLASS wird die User Priority für den Transport über ein Managed Segment gespeichert.

[1] Layer 2 Adresse(Ethernet)

4.2 RSVP/IntServ über DiffServ

DiffServ benutzt Gegensatz zur Flussorientierung von RSVP, bei der in allen beteiligten Netzwerkkomponenten der Status der einzelnen Flüsse gespeichert werden muss, eine Klassenorientierung, die an Hand der DSCP(siehe Abschnitt 2.3.2) unterschieden werden. Der Vorteil von DiffServ liegt darin, das es für größere Netze sehr viel besser geeignet ist als RSVP, da es bei einer großen Anzahl von Flüssen, bei Flussorientierter Verarbeitung, zu Engpässen in den RSVP-Routern kommen kann. Kombiniert man diese beiden Technologien, lassen sich die Nachteile von RSVP durch die Vorteile von DiffServ ausgleichen. Vergleichbar ist diese Kombination mit der IntServ/RSVP über Ethernet, hier wird die User Priority im Ethernet Header gespeichert und in einem DiffServ Netz wird dafür das DS-Feld im IP-Header benutzt, in dem der Differentiated Service Code Point(DSCP) dort eingetragen wird. Der DSCP wird in das DS-Feld entweder vom Host selber oder von einem IntServ/RSVP-Router eingetragen, bevor die Pakete durch das DiffServ-Netz geleitet werden. Die Kombination aus diesen beiden Technologien ergeben 3 verschiedene Möglichkeiten der Ressourcenreservierung.

1. **Statische Reservierung:** Im DiffServ Netzwerk existieren keine RSVP-fähige Geräte. Bei einer Übertragung über das DiffServ Netzwerk wird zwischen den IntServ/RSVP-Geräten eine feste Dienstgüte vereinbart mit der der Flow über das DiffServ Netz geleitet wird.

2. **Dynamische Resevierung mit RSVP:** In diesem Szenario sind im DiffServ-Netz RSVP-Geräte vorhanden. Es kann also innerhalb des DiffServ Netzes auf Kapazitäts-Veränderungen flexibel reagiert werden, d.h. mittels RSVP Signalisierung kann bei z.B. frei gegebenen Ressourcen dies den außerhalb des DiffServ Netzes liegenden RSVP-Geräten mitgeteilt und entsprechend reagiert werden. Diese dynamische Anpassung ist zum Unterschied zur statischen Reservierung nicht möglich.

3. **Dynamische Reservierung mittels Aggregated RSVP:** Aggregated bedeutet in diesem Zusammenhang eine Bündelung von einzelnen RSVP Flüssen. Es wird auch von Tunneled RSVP gesprochen. Aggregated RSVP stellt eine Erweiterung zum normalen RSVP dar. Die Erweiterung von RSVP besteht in der Weiterleitung von

RSVP-Messages durch ein DiffServ-Netz, die als normale Datenpakete durch das DiffServ-Netz geleitet werden. Das hat den Vorteil, dass es keine RSVP-Router innerhalb des DiffServ-Netz bedarf, die die RSVP-Messages verarbeiten.

5. VERFAHREN ZUR QUALITÄTSMESSUNG

In diesem Kapitel werden Verfahren vorgestellt, um die Qualität von Netzwerk-Verbindungen zu bestimmen. Die Qualität von Netzwerkverbindungen, die vor allem für Multimediaübertragungen von Bedeutung ist, wird durch die Eigenschaften Delay(Verzögerung), Jitter(Delayvarianz), Throughput(Durchsatz) und Paketloss (Paketverlust) bestimmt. An einigen Beispielmessungen soll gezeigt werden, welche Qualitätsmerkmale z.B. Verbindungen über 56k Modem, DSL oder Ethernet vorweisen und ob diese Verbindungen sich für Multimediaübertragungen eignen.

5.1 Delay

Der Delay lässt sich in zwei Arten unterscheiden, zum einen der One-Way-Delay, der die Verzögerung vom Sender zum Empfänger beschreibt, und zum anderen der Round-Trip-Delay, der die Verzögerung vom Sender zum Empfänger und in Gegenrichtung angibt.

Die Messung eines One-Way-Delay erfolgt durch die Differenzbildung von Empfangs- und Sendezeitpunkt. Dazu werden die Datenpakete vom Sender mit einem Zeitstempel versehen und beim Eintreffen der Pakete im Empfänger wird der Zeitpunkt festgehalten und entsprechend die Differenz der beiden Zeiten gebildet. Ein wesentliches Problem dieser Methode besteht in der Synchronisation der Uhren im Sender und im Empfänger. Um ein verlässliches Ergebnis dieser Messmethode zu erhalten müssen beide Uhren genauer synchronisiert sein, als die kleinste zu messende Zeitverzögerung. Um dieses Problem zu lösen, existiert ein Protokoll namens **NTP**(Network Time Protocol vgl. [15]).

NTP wurde von David L. Mills 1988 entwickelt, als er herausfand, dass über 60 Prozent der von ihm überprüften Server mehr als eine Minute von der wahren Zeit abwichen. Bei 10 Prozent der Server betrug die Abweichung sogar 13 Minuten und mehr. Das Network Time Protocol ist ein hierarchisches Protokoll(siehe RFC 1305). Der NTP-Prozess in einem Server arbeitet dabei selbst als Uhr und ist nicht von der Systemuhr im Rechner abhängig. Über dieses Protokoll können Server eine gemeinsame Zeit ermitteln. Im Zusammenhang mit diesem Protokoll wird der Begriff Stratum(Schicht, Ebene) verwendet.

Jeder NTP-Server wird in ein solches Stratum eingegliedert. Diese Stratum Angabe beschreibt die Entfernung eines NTP-Servers von einer externen Zeitquelle(z.B. Atomuhr). Ein Stratum-1-Server besitzt also als Zeitgeber eine externe Quelle, auf der zweiten Ebene(Stratum-2) werden Stratum-1-Server als Referenz benutzt, auf der dritten Ebene werden Stratum-2-Server als Referenz benutzt, und so weiter. Die höchste Ebene liegt bei 16, in der Praxis reicht jedoch eine Abstufung von 4 aus(siehe auch [14]).

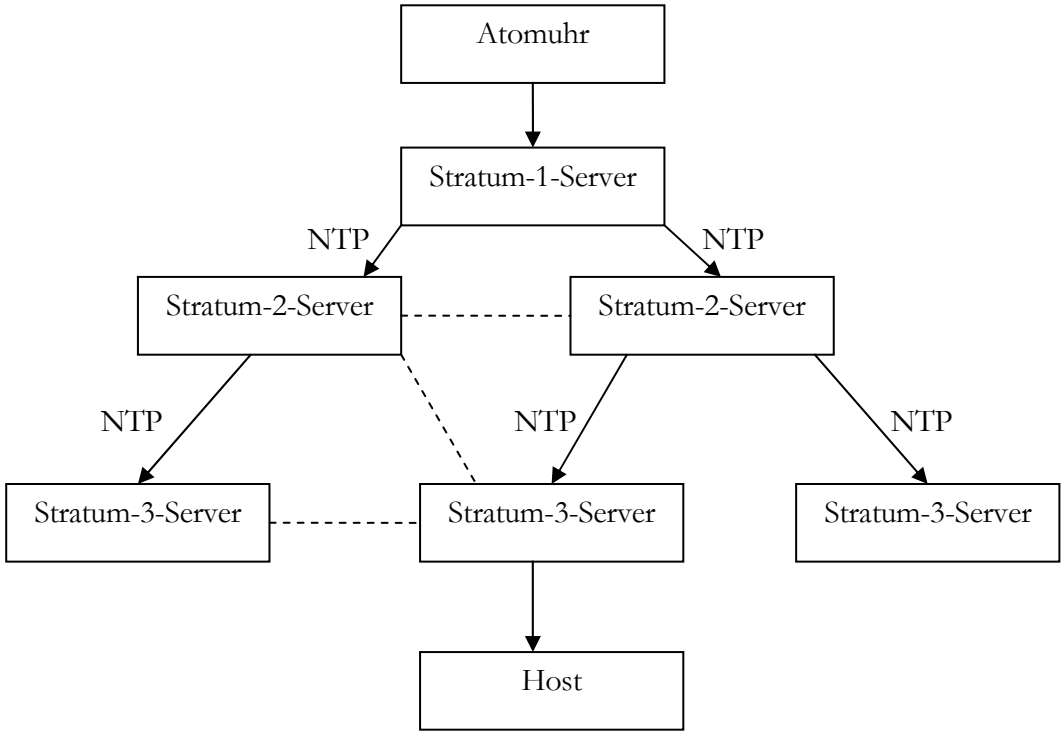

Abbildung 5.1: Hierarchie von Stratum-n-Servern, die mittels NTP kommunizieren

Wie in Abbildung 5.1 zu sehen, sind die NTP-Server in einer Hierarchie organisiert. Die gestrichelten Linien sollen verdeutlichen, dass sich die NTP-Server untereinander sowohl innerhalb der gleichen, als auch von höher gelegenen Schichten abstimmen können, falls es zu Unterbrechungen der Verbindung kommt. Mit aufsteigender Stratumnummer steigt auch die Ungenauigkeit an, aufgrund der serverinternen Verarbeitung. Diese Abweichungen hängen vor allem von der Paketlaufzeit und der jeweiligen Konfiguration des NTP-Servers ab. Die gesetzliche Zeit in Deutschland wird von der Atomuhr der Physikalisch-Technischen Bundesanstalt festgelegt und der entsprechende Stratum-1-Server ist unter ptbtime1.ptb.de erreichbar.

Im praktischen Einsatz wird zur Messung des Delay einer Verbindung das Tool **ping** verwendet. Ping basiert auf dem ICMP(Internet Control Message Protocol). ICMP ist kein Transportprotokoll, es wird als Statusprotokoll bezeichnet. Dieses Protokoll bietet die Möglichkeit Informationen über Verfügbarkeit und Erreichbarkeit von Netzwerkverbindungen abzurufen. Der Vorteil von ping ist, dass dieses Tool auf allen netzwerkfähigen Betriebsystemen verfügbar ist. Als Beispielmessung wurde eine Netzwerkverbindungen zwischen einem Host, der in Dresden lokalisiert war, und einem Host in der HTWK mittels ping durchgeführt.

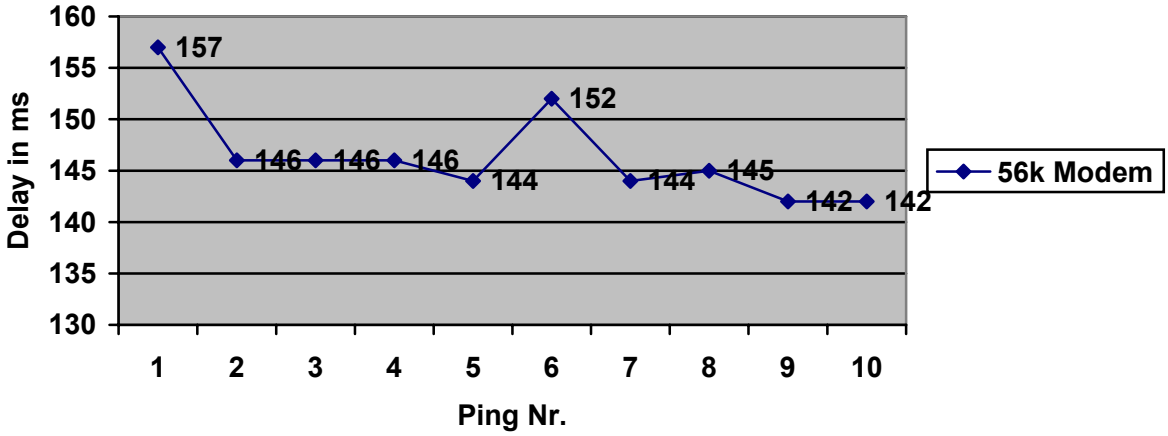

Abbildung 5.2: Delaymessung mit ping(56k Modem)

Diese Messung erfolgte mittels ping und den entsprechenden Parametern:

```
ping -c 10 217.184.84.182
```

Die Parameter, die ping übergeben werden können, sind auf den unterschiedlichen Betriebssystemen nicht einheitlich. Unter Linux wird -c und unter Windows wird -n gefolgt von der Anzahl zu sendender ping-Anfragen benutzt. Eine typische Ausgabe von ping sieht nach der Eingabe der obigen Befehlszeile wie folgt aus:

```
64 bytes from 217.184.84.182: icmp_seq=1 ttl=115 time=157 ms
64 bytes from 217.184.84.182: icmp_seq=2 ttl=115 time=146 ms
64 bytes from 217.184.84.182: icmp_seq=3 ttl=115 time=146 ms
64 bytes from 217.184.84.182: icmp_seq=4 ttl=115 time=146 ms
64 bytes from 217.184.84.182: icmp_seq=5 ttl=115 time=144 ms
64 bytes from 217.184.84.182: icmp_seq=6 ttl=115 time=152 ms
64 bytes from 217.184.84.182: icmp_seq=7 ttl=115 time=144 ms
64 bytes from 217.184.84.182: icmp_seq=8 ttl=115 time=145 ms
64 bytes from 217.184.84.182: icmp_seq=9 ttl=115 time=142 ms
```

```
64 bytes from 217.184.84.182: icmp_seq=10 ttl=115 time=142 ms
```

Diese gemessenen Verzögerungswerte sind in Abbildung 5.2 grafisch dargestellt. Die Ausgabe „time=144 ms“ bedeutet, dass die Paketlaufzeit der ping-Anfrage vom Sender bis zum Empfänger und zurück(Round-Trip-Delay) 144 ms beträgt. Die Ausgabe „icmp_seq=x“ (x ist eine Zahl >0) dient zur Darstellung der Paketnummern bzw. Nummerierung der ping-Anfragen. Die Ausgabe „ttl=115“ entspricht der Bedeutung des TTL Feldes im IP-Header(siehe 2.1.1).

Nach der Empfehlung der ITU(International Telecommunication Union) sollte die maximale Verzögerung in eine Richtung für „Telephonie-artige“ Dienste 150 ms betragen. Dies bedeutet für die gemessene Verbindung mit einem 56k-Modem, dass eine entsprechende Anwendung zur Übertragung von Sprache in Echtzeit in Frage kommt. Dies lässt sich aus der Darstellung 5.2 erkennen, es gibt zwei Messwerte die größer als 150 ms sind, da es sich dabei aber um die Verzögerung in beide Richtungen handelt, stellt dies für eine solche Übertragung kein Problem dar.

Entsprechende Messungen mittels ping wurden auch mit einem T-DSL-Anschluss(ADSL) durchgeführt. Die Messungen erfolgten analog zu denen mit einem 56k-Modem.

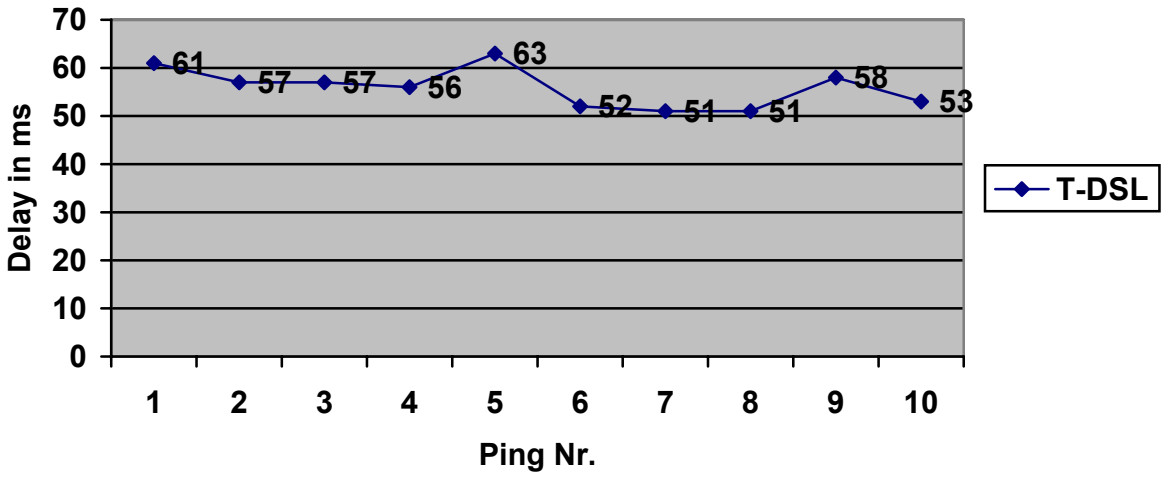

Abbildung 5.3: Delaymessung mit ping(DSL)

Die Abbildung 5.3 zeigt ein Diagramm mit den Delaymessungen eines T-DSL Anschlusses. Auffällig bei den Messwerten der DSL Verbindung sind die relativ hohen Verzögerungs-

Werte im Gegensatz zum Verhältnis Übertragungsgeschwindigkeit und Delay bei einem 56k-Modem. Als Vergleich, in einem unbelasteten Fast-Ethernet liegen die Messwerte zwischen 0,3 und 0,5 ms. Dies ist durch die bei DSL verwendete Fehlerkorrektur zu erklären. Das verwendete Verfahren wird als FEC(Forward Error Correction) bezeichnet(siehe 2.1.2).

Durch die Verschachtelung der Pakete, d.h. wie in 2.1.2 schon erwähnt, die PPPoE-Pakete werden auf die ATM-Zellen so aufgeteilt, das aufeinander folgende ATM-Zellen immer nur einige Bytes des 1. PPPoE-Paketes, die nächste Zelle einige Bytes des 2. PPPoE-Paketes, u.s.w., enthält. Die Verschachtelungstiefe bestimmt, wie viel Pakete ineinander geschachtelt werden. Dies zieht aber auch einen gravierenden Nachteil mit sich, die Daten eines PPPoE-Paketes werden zeitlich weit auseinander gezogen, folglich ergibt sich die Erhöhung der Latenzzeit bzw. die Reaktionszeit des Kommunikationspartners. Je höher die Verschachtelungstiefe, desto höher ist auch die Latenzzeit.

5.2 Durchsatz, Jitter und Paketverlust

Im vorherigen Abschnitt 5.1 wurde die wichtigste QoS Charakteristik Delay genauer erklärt, in diesem Abschitt werden die anderen QoS Charakteristiken behandelt. Da es zwei Werkzeuge gibt die diese drei Eigenschaften zusammen messen können werden sie auch zusammen in diesem Abschnitt behandelt.

Das erste Programm heißt **iperf**([17]) und es verwendet sowohl TCP, als auch UDP als Transportprotokoll. Das Programm arbeitet im Client-/Server-Modus, d.h. iperf wird mit dem Parameter „-s“ als Server gestartet und mit „-c“ [IP-Adresse des Servers] als Client. Der Parameter „-w“ gefolgt von einer Zahl und dem Buchstaben „k“ dient zur Angabe der TCP-Window-Size. Eine Messung mit einem 56k-Modem brachte folgende Ausgabe:

Server-Teil:

```
[jpetters@master iperf-1.6.5]$ ./iperf -s -w 128k
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte
------------------------------------------------------------
[  6] local 141.57.11.119 port 5001 connected with 217.184.84.182 port 1107
[ ID] Interval       Transfer     Bandwidth
[  6]  0.0-18.5 sec   200 KBytes  88.6 Kbits/sec
```

Client-Teil: `[jpetters@master iperf-1.6.5]$ ./iperf -c master`

Die mit 128 kByte angegebene TCP-Window-Size dient dazu, um auszuschließen, dass eine zu geringe TCP-Window-Size die Messergebnisse negativ beeinflussen könnte.

Die in Abschnitt 5.1 erwähnte DSL-Verbindung wurde auch mit diesem Werkzeug ausgemessen. Die Firma Telekom gibt für einen gewöhnlichen T-DSL-Anschluss die Werte für Downstream mit 768 kBit und für den Upstream bis zu 128 kBit an. Diese Werte sind Bruttoangaben, d.h. in diesen Werten ist ein gewisser Anteil an Overhead enthalten, der keine Nutzdaten, sondern nur Steuerinformationen enthält. Die Abbildung 5.4 verdeutlicht die Messdaten, die mit iperf für Upstream innerhalb eines Intervalls von 10 Sekunden gemessen wurde. Die Intervallmessung wird bei iperf mit dem Parameter „-i“ gefolgt von der Anzahl in Sekunden angegeben.

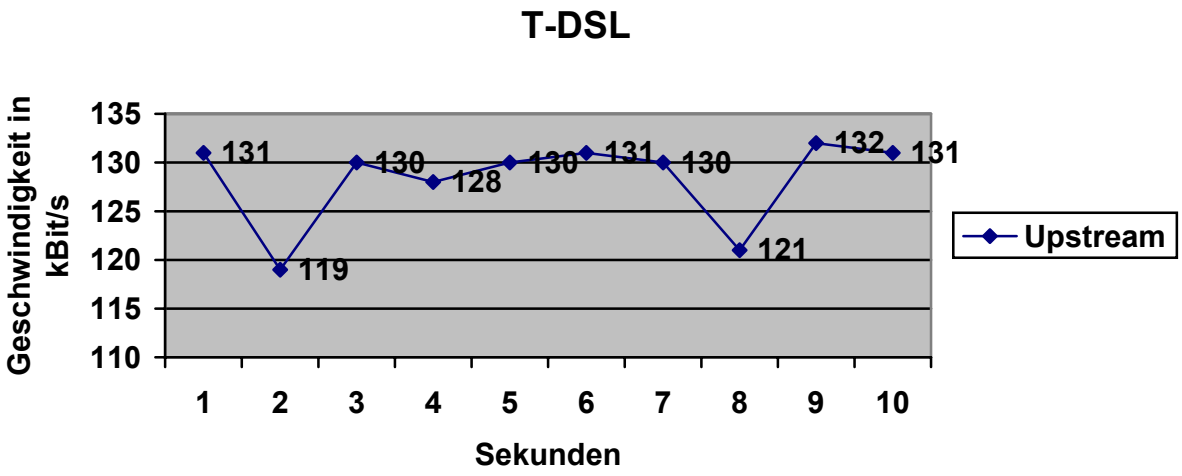

Abbildung 5.4: Diagramm mit den Upstream Messdaten

Wie in Abbildung 5.4 zu sehen liegen einige Messwerte sogar über den von der Telekom angegeben 128 kBit. Dies kann zum einen an der Messgenauigkeit von iperf liegen oder hat die Telekom an einigen Stellen noch kleinere Reserven in der Bandbreite für den Upstream. Die genaue Begründung für diesen Effekt könnte z.B. im Rahmen einer weiteren Arbeit untersucht werden, da dies den Rahmen der vorliegenden Arbeit überschreiten würde. Die relative Standardabweichung der in Abbildung 5.4 dargestellten Messwerte beträgt 3,5%.

Die nächste Abbildung 5.5 zeigt für T-DSL Verbindung die mit iperf gemessenen Downstream-Werte.

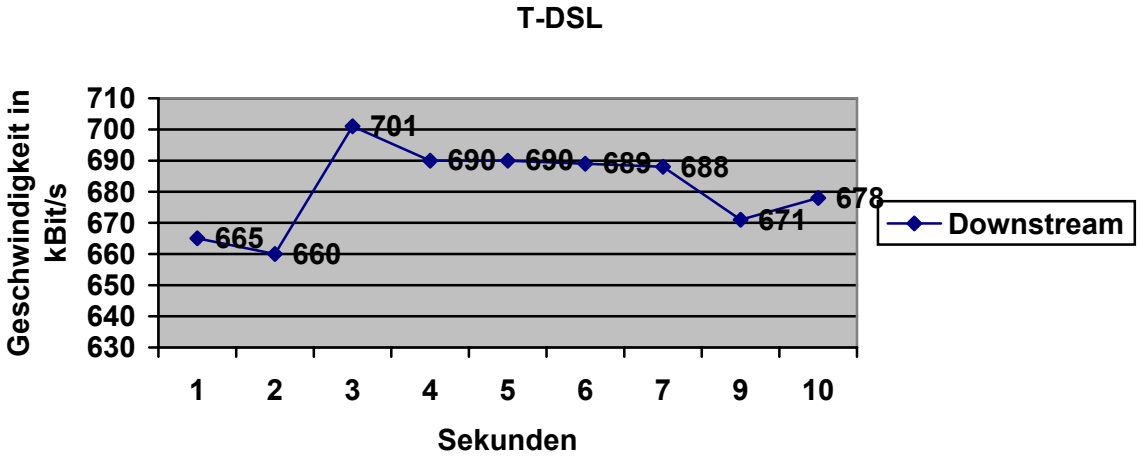

Abbildung 5.5: Diagramm mit den Messdaten des Downstream

Wie in Abbildung 5.5 zu sehen wurde der von der Telekom angegebene Wert von 768 kBit/s nicht erreicht. Der maximale Wert liegt bei 701 kBit/s und der niedrigste Wert bei 660 kBit/s. Der Grund für die Tatsache, dass die 768 kBit/s nicht erreicht wurden liegt vor allen daran, dass eine gewisse Bandbreite für die Steuerinformationen bzw. für den Protokoll-overhead benötigt wird. Die hier gemessenen Datenraten liegen zwischen 85% und 91% von dem von der Telekom angegebenen Wert. Die relative Standardabweichung für die in Abbildung 5.5 abgebildeten Werte beträgt 11,3%.

Ein weiteres Werkzeug dient zur Messung des Durchsatzes von TCP/IP Verbindungen verwendet werden. Der Name des Programms ist NetIO([16]), der Unterschied zu iperf besteht in der Tatsache, dass dieses Programm den Durchsatz in Abhängigkeit von der Paketgröße misst. Das Werkzeug misst Paketgrößen von 1,2,4,8,16 und 32 kByte. Ähnlich wie iperf arbeitet dieses Programm auch im Client-/Servermodus. Dem Programm wird der Parameter „-s“ übergeben wenn es als Server agieren soll und wenn es als Client arbeiten soll wird dem Programm lediglich die IP-Adresse des Servers übergeben. Weiterhin muss dem Client der Parameter „-t“ übergeben werden, wenn TCP/IP Verbindungen gemessen werden sollen. Mit diesem Programm wurden 4 Messreihen mit der T-DSL Verbindung

durchgeführt. In Abbildung 5.6 sind die Durchschnittswerte der Messreihen für den Upstream grafisch dargestellt. In Abbildung 5.7 sind die Durchschnittswerte der Messreihen für den Downstream grafisch dargestellt.

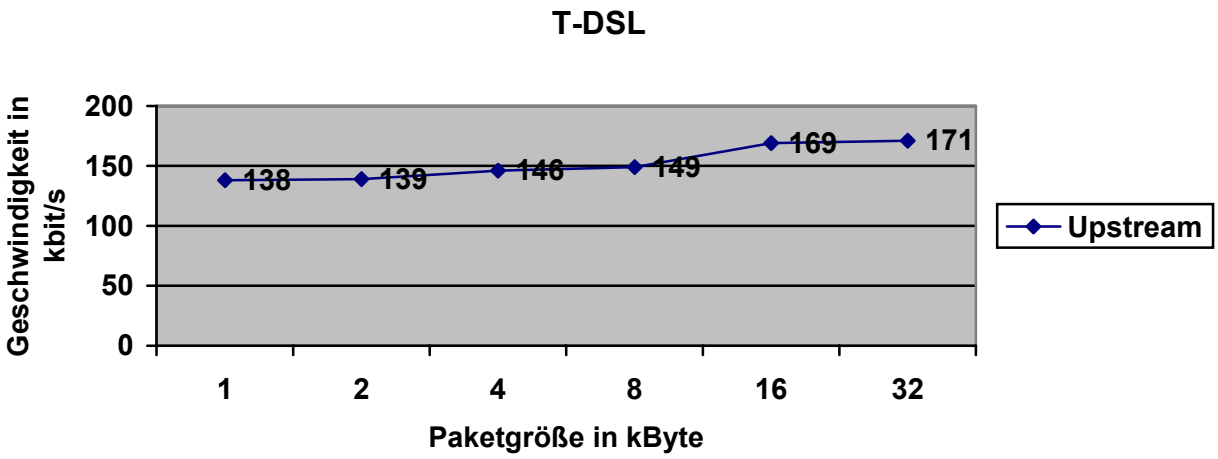

Abbildung 5.6: Diagramm mit den Messdaten des Upstream von NetIO

Wie in Abbildung 5.6 zu sehen, liegen hier auch die von NetIO gemessenen Werte über dem für Upstream von der Telekom angebenen Wert von 128 kBit/s. Weiterhin ist zu sehen, dass sich bei zunehmender Paketgröße auch der Upstream-Durchsatz erhöht. Dies lässt sich darauf zurückführen, dass bei größeren Paketen die Fragmentierung durch die unter TCP liegenden Protokolle geringer ist als bei den kleineren. Bei größeren Paketen können die für Nutzdaten vorgesehenen Puffer besser ausgenutzt werden.

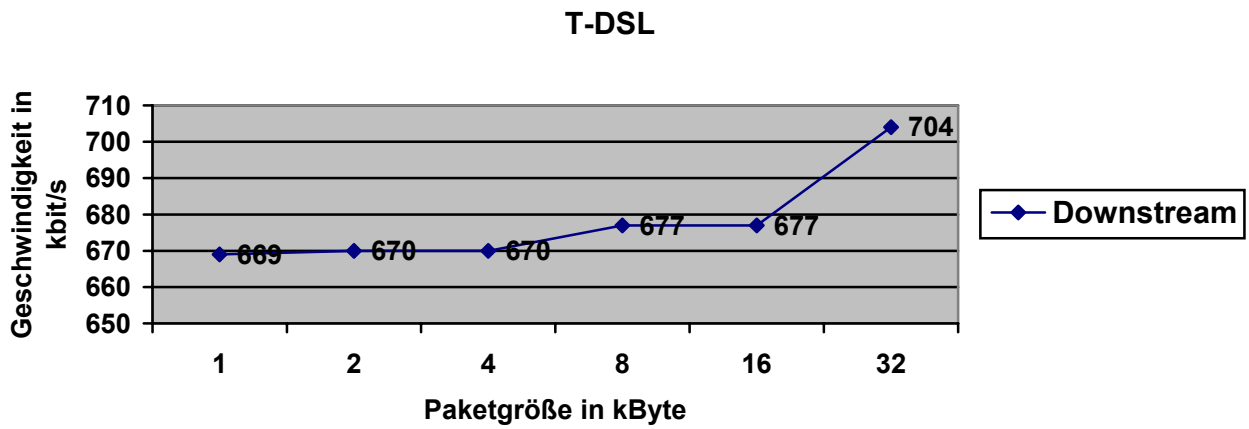

Abbildung 5.7: Diagramm mit den Messdaten des Downstream von NetIO

In Abbildung 5.7 sind die Datenraten für den Downstream bei unterschiedlichen Paketgrößen zu sehen. Auch hier ist deutlich zu beobachten, wie bei den Messungen mit iperf, dass die Datenraten unter dem Wert 768kBit/s liegen. Dies begründet sich aus dem entsprechenden Protokolloverhead und den zu übertragenden Steuerinformationen. Wie bei den Messwerten für den Upstream ist auch für den Downstream zu bemerken, dass bei steigender Paketgröße auch der Downstream-Durchsatz steigt, weil die unter TCP liegenden Protokolle die für Nutzdaten vorgesehenen Puffer besser ausnutzen können und die Fragmentierung geringer ist als bei kleinen Paketgrößen.

Zur Messung der zwei anderen Qualitätsmerkmale Jitter und Paketloss kann ebenfalls iperf benutzt werden. Das Merkmal Jitter wird auch Delay-Variation bezeichnet und ist vor allem für Multimediaübertragungen von Bedeutung. Der Großteil der Multimediaanwendungen benötigt einen konstanten Datenstrom, die entsprechende Kenngröße für die Unregelmäßigkeit in diesem Datenstrom ist der Jitter-Wert. Auf Anwendungsebene können solche Unregelmäßigkeiten mit Zwischenpuffern ausgeglichen werden, aber dies bringt auch einen großen Nachteil mit sich, da durch einen größeren Zwischenpuffer auch eine größere Gesamtverzögerung eintritt. Daher sollten diese Zwischenspeicher so klein wie möglich sein, um die Gesamtverzögerung niedrig zu halten, aber auch so groß gewählt werden das alle auftretenden Delay-Variationen abgefangen werden können.

Um mit iperf die Jitter-Werte zu messen, wird iperf wieder ein Zeitintervall mit „-i“ gefolgt von einer Anzahl Sekunden und der Parameter -u für eine UDP Übertragung übergeben. Auf der Client-Seite wird iperf wieder „-c“ gefolgt von der IP-Adresse des Servers, „-u“ für UDP und „-b“ gefolgt von einer Zahl die die Senderate beschreibt übergeben. Das folgende Beispiel dient zur Erklärung der Ausgaben und zeigt wie die Befehlszeilen aussehen können:

Server-Teil:

```
node2> iperf -s -u -i 1
```

Client-Teil:

```
node1> iperf -c 122.0.0.1 -u -b 10m
```

Eine Ausgabe sieht z.B. auf dem Server so aus und stammt aus der Dokumentation von iperf:

```
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 60.0 KByte (default)
------------------------------------------------------------
[  4] local <IP Addr node2> port 5001 connected with <IP Addr node1> port
9726
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total
Datagrams
[  4]  0.0- 1.0 sec   1.3 MBytes  10.0 Mbits/sec  0.209 ms    1/  894
(0.11%)
[  4]  1.0- 2.0 sec   1.3 MBytes  10.0 Mbits/sec  0.221 ms    0/  892 (0%)
[  4]  2.0- 3.0 sec   1.3 MBytes  10.0 Mbits/sec  0.277 ms    0/  892 (0%)
[  4]  3.0- 4.0 sec   1.3 MBytes  10.0 Mbits/sec  0.359 ms    0/  893 (0%)
[  4]  4.0- 5.0 sec   1.3 MBytes  10.0 Mbits/sec  0.251 ms    0/  892 (0%)
[  4]  5.0- 6.0 sec   1.3 MBytes  10.0 Mbits/sec  0.215 ms    0/  892 (0%)
[  4]  6.0- 7.0 sec   1.3 MBytes  10.0 Mbits/sec  0.325 ms    0/  892 (0%)
[  4]  7.0- 8.0 sec   1.3 MBytes  10.0 Mbits/sec  0.254 ms    0/  892 (0%)
[  4]  8.0- 9.0 sec   1.3 MBytes  10.0 Mbits/sec  0.282 ms    0/  892 (0%)
[  4]  0.0-10.0 sec  12.5 MBytes  10.0 Mbits/sec  0.243 ms    1/ 8922
(0.011%)
```

Die Angabe „Interval“ bezeichnet den entsprechenden Zeitraum für eine Messung, Transfer bezeichnet das übertragene Datenvolumen im entsprechenden Zeitintervall, „Bandwidth“ gibt den gemessenen Durchsatz im jeweiligen Intervall an, „Jitter“ gibt die gemessene Delay-Variation an und „Lost/Total“ stellt die Anzahl jeweils verlorener und

gesendeter Pakete gegenüber. In Abbildung 5.8 ist ein Diagramm mit den gemessenen Jitter-Werten der T-DSL Verbindung zu sehen.

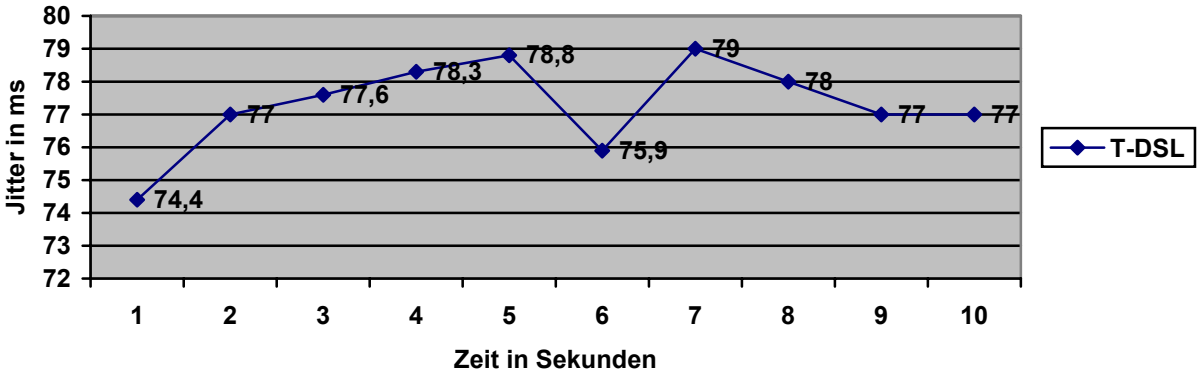

Abbildung 5.8: Jitter Messwerte einer T-DSL Verbindung

Die relative Standardabweichung der in Abbildung 5.8 dargestellten Werte beträgt 1,3%. Diese Jitter-Messungen können vor allem dazu dienen, um in Multimediaanwendungen den Jitter-Buffer optimal auf die Verbindung einzustellen.

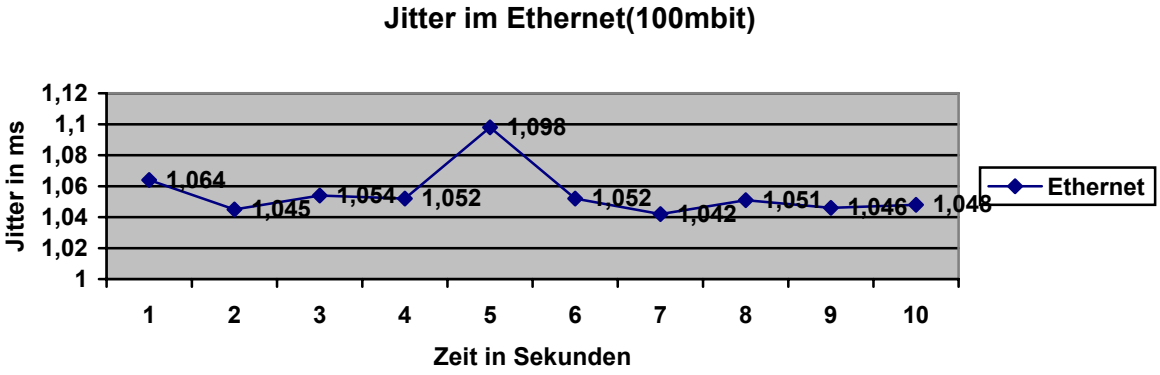

Abbildung 5.9: Jitterwerte gemessen in einem Fastethernet

Die in Abbildung 5.9 dargestellten sollen als Vergleich zu den Werten einer T-DSL Verbindung dienen. Diese Werte wurden zwischen 2 Hosts in einem unbelasteten Fastethernet gemessen und die relative Standardabweichung liegt bei 1,5%. Dies soll die Dimension verdeutlichen, die beide Technologien in dieser Kategorie unterscheidet. Die im

Ethernet gemessenen Werte sind ca. um den Faktor 70 kleiner als die T-DSL Jitter-Werte. Bei Jitter-Messungen bedeutet dies, je kleiner desto besser für Multimediaanwendungen, da auch die entsprechende Gesamtverzögerung gering bleibt und Jitterschwankungen nicht durch einen Puffer ausgeglichen werden muss.

6. ZUSAMMENFASSUNG

In diesem Kapitel wird eine Zusammenfassung der Schwerpunkte dieser Diplomarbeit dargestellt. Diese Arbeit hatte zur Aufgabe zwei unterschiedliche Netzwerktechnologien zu untersuchen. Zum einen die DSL-Technologie, die schon etabliert und weit verbreitet ist und zum anderen eine zukünftige Technologie, QoS fähige IP-Netze.

Das erste Kapitel enthält eine kurze Einleitung zum Thema der Arbeit.

Das zweite Kapitel schafft die Grundlagen für das Verständnis der Protokolle TCP und IP, sowie die für DSL verwendete Technik. Des Weiteren werden auch die verschiedenen Möglichkeiten für QoS in IP-Netzen vorgestellt, wie etwa RSVP, IntServ oder DiffServ.

In Kapitel 3 werden verschiedene Multimediatechnologien im Audio- und Videobereich erklärt. Es enthält auch Grundlagen im Bereich Videokompression und digitaler Videokodierung. Dies dient vor allem dazu, um die Anforderungen an entsprechende Netzwerkübertragungen zu verdeutlichen.

Im vierten Kapitel werden verschieden Szenarien dargestellt, die die Vor- und Nachteile von unterschiedlichen Möglichkeiten QoS basierter IP-Netze darlegen. Im Einzelnen werden auch Verfahren erläutert, um bestehende Netzewerke, wie Ethernet, QoS fähig zu machen.

Im 5. Kapitel liegt der Schwerpunkt auf der Messung der Qualitätsmerkmale von Netzwerkverbindungen, wie Delay, Jitter oder Durchsatz. Ebenfalls werden die für die Messungen verwendeten Werkzeuge erklärt und die Messergebnisse einzelner Untersuchungen an Netzwerkverbindungen beschrieben.

LITERATURVERZEICHNIS

[1] Tanenbaum, A.S.: Computernetzwerke, Prentice Hall, München, 1997, 3. Auflage

[2] Gumm, Heinz-Peter und Sommer, Manfred: Einführung in die Informatik, München, 1998, 3. Auflage

[3] Reimann: Vorlesung und Seminare „Rechnernetze“ und „Betriebssysteme“

[4] Hudecek, Michael: Subnetztrennung mittels Firewall und Masquerading, Diplomarbeit, 2000

[5] Enders, Johannes: DSL - Die schnelle Leitung, c't 16/1999

[6] Manhart, Klaus: So funktioniert DSL, PC-Welt Spezial 1/2002

[7] Odutayo, Maleka und Teulner, Markus: Die Breitbandaufahrt auf den Datenhighway, http://www.cs.fhm.edu/~ifw99235/dako/semthem.html

[8] Krämer, Mathias: Architektur und Technik kommunaler Telekommunikationsnetze, Diplomarbeit, http://www-user.rhrk.uni-kl.de/~mkraemer/mkraemer/diplom/

[9] Queisser, Carsten: Verfügbarkeit und Dienstgüte in IP-Netzwerken - Quality of Service nutzen, http://www.networkworld.de

[10] Fiebig, Matthias: Untersuchung der Online-Videoübertragung unter Windows NT über Dual-Video-Systeme, Diplomarbeit, Universität Leipzig

[11] Breyer, Tobias: MPEG (Video) Proseminar, http://www.uni-karlsruhe.de/~udue/

[12] Scheuermann, Torsten: MPEG (Audio) Proseminar, http://www.uni-karlsruhe.de/~udue/

[13] Friedrich, Martin: Vergleich von Entwicklungen für Quality of Service für IP-Netze, Diplomarbeit, TU München

[14] Ahlers, Ernst: Zeinahme, c't 19/2002

[15] Network Time Protocol, http://www.ntp.org

[16] NetIO 1.16: http://freshmeat.net/redir/netio/18673/url_zip/netio116.zip

[17] Iperf: http://dast.nlanr.net/Projects/Iperf/

[18] Bernsau, Dirk und Lohner, Boris: Messung von QoS, Studienarbeit, Universität München 2002

[19] RFC 2814: SBM - A Protocol for RSVP-based Admission Control over IEEE 802-style networks, Network Working Group, May 2002

[20] RFC 2815: Integrated Service Mappings on IEEE 802 Networks, Network Working Group, May 2000

[21] RFC 2816: A Framework for Integrated Services over Shared and Switched IEEE 802 LAN Technologies, Network Working Group, May 2000

[22] RFC 2998: A Framework for Integrated Services Operation over Diffserv Networks, Network Working Group, November 2000

[23] Holtkamp, Heiko: Einführung in TCP/IP, Universität Bielefeld, Juni 1997

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jeannot Petters
Leipzig, 14. April 2003