Diplomarbeit

„Low Latency Video Streaming“

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines

Dipl.-Ing. (FH) Telekommunikation und Medien

am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von

Dipl. Ing. Grischa Schmiedl

Zweitbegutachtung von

Dipl.-Ing. (FH) Klaus Temper

ausgeführt von

Adam Kogler

tm0110038049

St. Pölten, am 04.04.2008 Unterschrift:

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter beurteilten Arbeit überein.

....

Ort, Datum Unterschrift

Danksagung

Für die große Unterstützung meiner Freunde und Familie während meines Studiums und der Zeit, in der ich diese Arbeit verfasst habe, möchte ich mich an dieser Stelle herzlich bedanken.

Besonders bei meiner Freundin Renate, die während der Erstellung dieser Arbeit viel Geduld aufgebracht hat und immer für mich da war.

Speziell erwähnen möchte ich auch meine Mutter, meinen Vater und meinen Großvater, der die Fertigstellung dieser Arbeit leider nicht mehr miterlebt hat, weil sie mich auf meinem Weg immer ermutigt und unterstützt haben.

Kurzfassung

Streaming wird im Internet immer häufiger benutzt. In dieser Arbeit sollen verschiedene Möglichkeiten zur Erstellung und Distribution von Videostreams gezeigt werden. Der Schwerpunkt liegt dabei auf der Live-Übertragung von Streams mit möglichst geringer Verzögerung (Latency).

Die Vorbereitung für Live-Streaming, die Unterschiede zu On Demand und Progressive Download werden besprochen und Methoden zur Komprimierung von Videosignalen gezeigt.
Es wird ein Überblick der Streaminglösungen, die im Moment auf dem Markt sind, gegeben und auf ihre Eignung für Low-Latency-Streaming eingegangen. Auch auf anwendungsbezogene Lösungen zur Verbesserung der Qualität von Videostreams, bzw. deren Anpassung an Bedürfnisse der Benutzer, wird eingegangen. Die für Streaming im Internet maßgeblichen Protokolle werden gezeigt und ihre Funktionen und Eigenschaften umrissen.
Methoden zur Streamverbreitung für eine große Anzahl gleichzeitiger Benutzer werden erläutert und dabei auch die Funktionsweise, Verbreitung und Zukunftsperspektiven von Multicast besprochen.
In diesem Zusammenhang wird auch die Entwicklung von IPTV in verschiedenen Ländern beleuchtet und Faktoren aufgezeigt die für die Verbreitung und Akzeptanz von IPTV entscheidend sind.

Abstract

Streaming is a growing field in the internet. Different possibilities of production and distribution of video streams are shown in this diploma thesis. The focus is on the live distribution of video streams involving as little latency as possible.

Preparations for live streaming, differences between live and on demand streaming are discussed and an outline of methods for video compression is given.
An overview of current streaming solutions on the market is provided as well as an analysis of their capability for live streaming. Application specific solutions to enhance the quality of the stream or adapt it to the needs of the client are also addressed in this thesis.
Internet protocols which are relevant for streaming are shown and their functions and properties are described.
Methods for stream distribution to a large number of concurrent users are explained and the function, spread and future perspective of multicast is discussed.
In this context the progress of IPTV in various countries is compared and key factors to achieve a high market share and acceptance for IPTV are illustrated.

Inhaltsverzeichnis

Inhaltsverzeichnis 5

1 Einleitung 8

1.1 Definition des Erkenntnisgegenstandes 8

1.2 Forschungsleitende Fragestellung 9

2 Streamen von Videoinhalten 10

2.1 Live vs. On-Demand 10

2.1.1 Aufbau für Live-Streaming 10

2.1.2 Mindestanforderung für Live-Streaming 11

2.1.3 On Demand Streaming (Video on Demand) 12

2.2 Anwendungsgebiete von Videostreaming 13

2.3 Bildqualität und Qualitätskontrolle 14

2.3.1 Bildqualität 14

2.3.2 Komprimierung 14

2.3.3 Qualitätskontrolle 19

2.4 Datenraten 20

2.5 Latency und Buffer-Time 21

3 Content Delivery Networks (CDNs) 23

4 Unicast und Multicast 25

4.1 Definition 25

4.1.1 Unicast 25

4.1.2 Multicast 27

4.2 Multicast-Datenfluss 30

4.2.1 Source Tree (Shortest Path Tree) 30

4.2.2 Shared Trees 31

4.3 Multicast Protokolle 34

4.3.1 Protocol Independend Multicast (PIM) 34

5 Streaming Protokolle 36
5.1 UDP (User Datagram Protocol) 36
5.2 TCP (Transmission Control Protocol) 37
5.3 HTTP (Hypertext Transfer Protocol) 37
5.4 RTP (Real Time Transport Protocol) 39
5.5 RTSP (Real Time Streaming Protocol) 39
5.6 RTCP (Real Time Control Protocol) 40
5.7 RSVP (Ressource Reservation Protocol) 41
5.8 RTMP (Real Time Messaging Protocol) 42
5.9 MMS (Microsoft Media Server Protocol) 43

6 Streaming Server 44
6.1 Flash 44
6.2 Windows Media 45
6.3 Real (Helix-Server) 47
6.4 QuickTime / Darwin 48
6.5 Red 5 49

7 Encoder und Decoder 50
7.1 Flash 51
7.1.1 Encoding 51
7.1.2 Decoding 52
7.1.3 Verwaltung der Buffertime 52
7.2 Windows Media 53
7.2.1 Encoding 53
7.2.2 Decoding 53
7.2.3 Verwaltung der Buffertime 54
7.3 Real 55
7.3.1 Encoding 55
7.3.2 Decoding 55
7.3.3 Verwaltung der Buffertime 56

7.4 Video LAN (VLC) ... 56
7.4.1 Encoding ... 57
7.4.2 Decoding ... 57
7.4.3 Verwaltung der Buffertime ... 57
7.5 Quick Time ... 58
7.5.1 Encoding ... 58
7.5.2 Decoding ... 59
7.5.3 Verwaltung der Buffertime ... 59

8 Low Latency Test Szenarien ... 60
8.1 Versuchsaufbau ... 60
8.2 Flash ... 61
8.3 Windows Media ... 63
8.4 Real ... 66
8.5 Video LAN (VLC) ... 68
8.6 Ergebnisse ... 69

9 Internet Protocol Television (IPTV) ... 71
9.1 Anbieter und Technologie ... 71
9.2 Middleware ... 73
9.3 Interaktives IPTV ... 74
9.4 Verbreitung von IPTV ... 76
9.5 Latency bei anderen Digitalen Fernsehformaten ... 76

10 Schlussfolgerungen ... 77

11 Abbildungsverzeichnis ... 80

12 Quellen ... 83

1 Einleitung

1.1 Definition des Erkenntnisgegenstandes

In dieser Arbeit soll Low-Latency-Streaming erforscht werden. Dieses hebt sich von herkömmlichem Streaming dadurch ab, dass eine möglichst geringe Pufferung des übertragenen Contents angestrebt wird, um bei Live-Übertragungen die Aktualität zu erreichen, die auch beim Fernsehen Standard ist.
Dieses Thema wird immer relevanter, da der Anteil der Internetnutzer, die über einen Breitbandanschluss verfügen und dadurch in der Lage sind Videoinhalte über das Internet zu empfangen immer größer wird.

Auch wenn es sich beim Fernsehen um einen etablierten Standard handelt, der sicherlich noch lange Zeit in der jetzigen Form bestehen wird, bietet das Live-Streaming im Internet wichtige Möglichkeiten zur Ergänzung dieses Angebots.
Die Interaktivität des Konsumenten ist durch die bereits vorhandene Rückleitung leicht umsetzbar und auch für das Fernsehen unrentable Inhalte, die eine zu kleine Zielgruppe ansprechen, können umgesetzt werden.

Einer der wichtigsten Gründe für die Umstellung von analoger auf digitale TV-Übertragung war der Mangel an Frequenzen, die für Fernsehkanäle zur Verfügung standen.
Dies führte dazu, dass die Frequenzen am TV Markt sehr umkämpft waren. Die Umstellung auf digitales Fernsehen brachte eine Reduktion der Bandbreite die erforderlich ist, um einen Fernsehkanal zu übertragen und ermöglichte damit weitere Sendeplätze.

Hier kann auch das Live-Streaming, das sich qualitativ durch die besseren Internetverbindungen stetig verbessert, einen Beitrag zu mehr Vielfalt leisten.

Abgesehen davon ist Streaming auch durch die wesentlich niedrigeren Produktionskosten interessant, weil sich damit auch für Privatpersonen und kleine Unternehmen die Möglichkeit bietet Videoinhalte zu verbreiten.

1.2 Forschungsleitende Fragestellung

Ziel der Arbeit ist die Erforschung der Möglichkeiten um Low-Latency-Streaming zur Verfügung zu stellen. Dabei soll der Begriff Low Latency für eine Verzögerung zwischen Encoding und Wiedergabe von weniger als 3 Sekunden stehen. Erreicht wird dies in erster Linie durch eine massive Reduktion der Pufferzeiten bei Encoding und Decoding. Nachdem bei den meisten Anbietern von Streaminglösungen der übliche (vorgesehene und voreingestellte) Wert der Verzögerung mit 15-20 Sekunden angegeben ist, sollen folgende Thesen untersucht werden:

Es ist möglich einen 600 kbit Stream mit einer Verzögerung von weniger als 3s zur Verfügung zu stellen, der in Österreich von Nutzern mit Breitband-Internet störungsfrei abgerufen werden kann.
Als Breitbandanschluss wird hierbei ein Downstream von mindestens 2048 kbit/s nominell angenommen.

Low-Latency-Streaming (mit Verzögerung < 3s) benötigt eine Bandbreite, die um mehr als 20% höher ist, als wenn mit hoher Verzögerung (ca. 15-30s) gestreamt wird um eine störungsfreie Wiedergabe zu gewährleisten.

2 Streamen von Videoinhalten

2.1 Live vs. On-Demand

Der grundlegende Unterschied zwischen Live- und On Demand Streaming liegt darin, dass die Daten beim Live-Streaming, während der Encoding-Vorgang läuft, auf den Server gestreamt werden, während bei On Demand Streaming die vorkodierten Dateien auf den Server geladen werden und bei Bedarf vom Server zum Empfänger gestreamt werden.
Weiters gibt es auch die Möglichkeit Videos mittels Progressive Download anzubieten. Dies ist allerdings kein Streaming im eigentlichen Sinn. In Kapitel 2.1.3 wird der Punkt Progressive Download etwas ausführlicher besprochen.

Ein großer Unterschied von Live- und On Demand Streaming ist die Hardware-Anforderung auf der Erzeugerseite. Während On Demand Videos, wenn genügend Zeit zur Verfügung steht, mit sehr einfacher Hardware hergestellt werden können, ist bei Live-Streaming je nach gewünschter Qualität des Streams entsprechend gutes Equipment nötig. Dies liegt daran, dass die von der Kamera empfangenen Videodaten in Echtzeit encodet werden müssen. Auch der Bedarf an Upstream-Bandbreite auf der Encoder-Seite ist beim Live-Streaming höher, da die fertig encodeten Videodaten mit der Datenrate des Videos gestreamt werden müssen, während es beim Upload von On Demand Videos kaum störend ist, wenn der Upload auf den Server ein paar Minuten länger dauert.

2.1.1 Aufbau für Live-Streaming

Für den Aufbau eines Live-Streamings wird ein Computer mit Videoeingang, im Idealfall mit eigener Videokarte benötigt. Das Bildsignal kommt entweder von einer Kamera oder über einen Videomischer von mehreren Kameras.

Wie auch in Punkt 2.3.1 besprochen ist dabei gutes Ausgangsmaterial wichtig, obwohl das stark komprimierte Video, das am Ende gestreamt wird meist nicht einmal an die Qualität einer normalen Consumer-Kamera heranreicht.

Entscheidend ist, dass an dem Ort von dem das Ereignis live gestreamt werden soll ein Internetanschluss mit ausreichender Upload-Bandbreite zur Verfügung steht. Dabei ist es immer ratsam nicht die komplette Bandbreite der Internetverbindung zu benutzen. Dies gilt besonders für Low-Latency-Streaming, weil hier bereits kurze Reduktionen der Bandbreite, z.B. durch Netzüberlastung, die Qualität des Streams stark beeinflussen.

Ein Streamingserver wird für einen Live-Stream im Regelfall ebenfalls benötigt. Wird direkt vom Encoder zum Benutzer gestreamt ist die Anzahl der gleichzeitigen Nutzer des Streams stark beschränkt. Das ergibt sich zum Einen aus Softwarebeschränkungen (viele Encoder lassen nur eine begrenzte Anzahl gleichzeitiger Zugriffe zu) und im Besonderen weil bei Unicast Übertragungen für jeden Benutzer ein eigener Stream generiert wird, der Upload Bandbreite beansprucht.

2.1.2 Mindestanforderung für Live-Streaming

Eine Minimalausrüstung für Live-Streaming wäre eine Kamera, ein Computer mit Videokarte (einige Encoder unterstützen auch Signale über DV oder Firewire), der in der Lage ist, Videosignale in Echtzeit zu codieren und eine Breitbandanbindung an das Internet mit einer Upload Rate, die größer ist als die Datenrate des Streams.

Dieses Setup würde allerdings genau für einen Stream zu einem Benutzer ausreichen und das auch nur, wenn der verwendete Encoder diese Streaming Variante unterstützt. Will man ein größeres Publikum ansprechen, wird je nach Anzahl der zu erwartenden Benutzer ein Streaming Server oder ein Content Delivery Network benötigt.

2.1.3 On Demand Streaming (Video on Demand)

Die Anforderungen bei On Demand Streaming sind geringer als bei Live-Streaming und erlauben mitunter auch eine bessere Qualität des Videos. Bei den Vorbereitungen für On Demand Streaming wird Videomaterial mit einem Encoder komprimiert, wobei im Gegensatz zu Live-Streaming auch Codecs eingesetzt werden können, die nicht echtzeitfähig sind, aber eine bessere Videoqualität erzeugen. Einige Codecs bieten z.B. die Möglichkeit mit der „Two-Pass-Encoding“ Technologie zu arbeiten. Dabei wird im ersten Durchgang das Video analysiert, um die Bereiche zu bestimmen, die ohne großen Qualitätsverlust stärker komprimiert werden können und solche, die eine höhere Datenrate benötigen um eine gute Qualität zu behalten. Erst im zweiten Durchgang wird dann die eigentliche Kompression durchgeführt.

Ist die Videodatei fertig encodet, wird sie auf einen Streaming Server geladen und kann von dort abgerufen werden. Dies kann entweder durch direktes Öffnen des Streams im Player über die Eingabe der Serveradresse geschehen, oder der Stream wird in eine Webseite eingebettet und kann dann im Browser wiedergegeben werden, sofern das nötige Plug-In installiert ist.
Dabei wird eine Verbindung zwischen Benutzer und Streaming Server aufgebaut und das Videomaterial als konstanter Datenstrom abgespielt, ohne beim Benutzer gespeichert zu werden.

Progressive Download wird auch immer wieder als Videostreaming bezeichnet, dies ist aber nicht korrekt. Im Unterschied zum Streaming wird bei Progressive Download eine Videodatei als gewöhnlicher Download übertragen. Dazu ist kein Streaming Server nötig, ein normaler Web-Server ist dafür ausreichend. Die Datei wird auf die Festplatte des Clients gespeichert und, sofern der Player dies unterstützt, bereits während des Downloads mit der Wiedergabe begonnen.
Dadurch kann der Benutzer nur auf Bereiche des Videos zugreifen, die sich bereits auf der Festplatte befinden. Bei echtem Streaming kann jederzeit ein beliebiger Punkt im Video gewählt werden und der Streaming Server beginnt die Übertragung der Daten von diesem Punkt aus.

2.2 Anwendungsgebiete von Videostreaming

Videostreaming verbreitet sich immer stärker im Internet. Hier ist vor allem Video On Demand für viele Anbieter interessant. Es lässt sich mit relativ geringem technischem Aufwand umsetzen und ist bis zu einer gewissen Anzahl gleichzeitiger Benutzer auch kostengünstig, weil Server selbst betrieben werden können. Sollten tatsächlich zu einem gewissen Zeitpunkt so viele Nutzer auf einmal zugreifen, dass der Server nicht über ausreichend Bandbreite verfügt, so können Benutzer zu einem späteren Zeitpunkt wieder störungsfrei auf den Stream zugreifen. Ist der Server allerdings regelmäßig überlastet, muss eine Lösung gefunden werden mit mehr Bandbreite zu streamen. Entweder durch eine bessere Internetanbindung für den Server oder über ein Content Delivery Network.
Für Privatpersonen und kleine Firmen, die nicht so viele Zugriffe auf ihre Videos verzeichnen, ist Videostreaming durchaus eine billige Variante der Verbreitung von Inhalten.

Privatpersonen stehen abgesehen davon im Internet diverse Portale zur Verfügung, die Videos auf ihrem Server gratis hosten und sich über Werbung finanzieren. Diese Variante des Videosharing boomt derzeit im Internet und generiert enormen Wert für Portale, die diese Videos hosten. So wurde das Videosharing-Portal YouTube im Oktober 2006 von Google um 1,65 Milliarden Dollar oder 1,31 Milliarden Euro (Kurs vom 09.10.2006) gekauft. [1]

Aber auch Fernseh- und Radiosender nutzen Videostreaming. Sehr viele Fernsehsender bieten Teile ihres Programms nach der Ausstrahlung als On Demand Videostream an. Radiosender übertragen entweder Live-Streams aus dem Studio oder von Events die der Sender veranstaltet.

Firmen werben auf ihren Webseiten für ihre Produkte und Wettanbieter zeigen Sportereignisse als Videostream und nehmen Live-Wetten an.
Web-TV Portale streamen entweder eigenen Content oder schließen Verträge mit herkömmlichen Fernsehstationen ab und übertragen deren Programm live ins Internet.

2.3 Bildqualität und Qualitätskontrolle

2.3.1 Bildqualität

Die Bildqualität eines Videostreams hängt sehr stark vom encodeten Ausgangsmaterial ab. Obwohl der endgültige Stream normalerweise stark komprimiert ist und daher nicht an die Qualität des Ausgangsmaterials heranreicht, wird der encodete Stream eine bessere Qualität haben, wenn die Videoquelle ein gutes Signal liefert.
Ideal für Videostreaming wäre ein hochqualitatives, digitales Signal (**S**erial **D**igital **I**nterface) mit geringem Rauschpegel, weil jedes analoge Signal ohnehin digitalisiert werden muss. Viele Videokarten haben jedoch analoge Videoeingänge, weil dies der Standard bei Videoübertragung ist. Hier wäre ein Component-Signal einem Composite-Signal vorzuziehen, üblich sind aber eher Composite-Verbindungen. [2]

Bei Live-Streamings macht es auch durchaus einen Unterschied, ob für die Aufnahmen hochqualitative Kameras wie z.B. BetacamSP, DigiBeta oder DVCPRO verwendet werden oder eine DV-Kamera.

Wohl der wichtigste Faktor für die Qualität des Streams ist der Codec, der zur Komprimierung des Videos benutzt wird.

2.3.2 Komprimierung

Die Komprimierung von Videodaten ist die Grundvoraussetzung um eine Übertragung im Internet zu ermöglichen. Unkomprimiertes Video im PAL Format hat ohne Audio eine Datenrate von 29,25 MB/s. Selbst DV Material mit einer Kompressionsrate von 1:5 kommt noch auf 3,5 MB/s. Zur Komprimierung von Bildern gibt es verschiedene verlustfreie und verlustbehaftete Verfahren. Der Unterschied dieser Verfahren besteht darin, dass bei verlustfreien Verfahren der gesamte Bildinhalt rekonstruiert werden kann, während die verlustbehafteten Verfahren die Schwächen der menschlichen Wahrnehmung nutzen und in

ren die Schwächen der menschlichen Wahrnehmung nutzen und in erster Linie Bildinhalte verwerfen, die das menschliche Auge nicht wahrnehmen kann.
Die Kompressionsrate, die mit verlustbehafteten Verfahren erreicht werden kann, ist bedeutend höher als bei verlustfreien. Für Datenreduktionen, wie sie für die Übertragung im Internet zumeist benötigt werden, ist es allerdings nicht ausreichend ausschließlich auf Bildinhalte zu verzichten, die keine wahrnehmbare Veränderung des Bildes bewirken.

Der erste Schritt zur Komprimierung von Videosignalen ist die Codierung des Bildes in einzelne Pixel, falls das Ausgangsmaterial nicht ohnehin in digitaler Form vorliegt. Ist dies geschehen, kann die Datenmenge mit Hilfe verschiedener Algorithmen reduziert werden.

- **Variable Längencodierung (VLC)**

Die variable Längencodierung nutzt die Tatsache, dass einige Werte öfter auftreten als andere. Eine Variante dieser Kompressionsmethode ist der Huffman Code. Es wird eine Code-Tabelle erstellt, in der häufig auftretende Werte ein sehr kurzes Codewort zugewiesen bekommen und selten auftretende Werte ein höheres. Dadurch kann die Gesamtlänge des Codes verkürzt werden. [3]

Das Prinzip kann mit jeder beliebigen Zeichenfolge gezeigt werden. Ausgehend von der Annahme die Zeichenfolge AABBCCCCDAAAAAEFFFFF solle mit der variablen Längencodierung vereinfacht werden, würde folgende Wahrscheinlichkeitstabelle erstellt:

Zeichen	Anzahl	Auftrittswahrscheinlichkeit	Code
A	7	0,35	11
F	5	0,25	10
C	4	0,2	01
B	2	0,1	001
D	1	0,05	0001
E	1	0,05	0000

Abbildung 1: Beispiel einer Huffman Code-Tabelle

„Dabei werden zunächst für alle auftretenden Zeichen die Auftrittswahrscheinlichkeiten ermittelt und nach Häufigkeit geordnet. Die Einzelwahrscheinlichkeiten der am seltensten auftretenden Zeichen werden zu einer gemeinsamen Wahrscheinlichkeit addiert und dann die Summe mit den restlichen Zeichen erneut nach Häufigkeit geordnet. Der Prozess wird solange wiederholt, bis nur noch zwei übrig bleiben, denen die Codeworte 1 und 0 zugeordnet werden können. Anschließend werden die Stufen zurückverfolgt, und den Stufen entsprechend zunächst die kurzen und dann die längeren Codeworte vergeben“ [1]

- **Run Length Encoding (RLC)**

Run Length Encoding nutzt die Tatsache, dass im binären Zahlensystem sehr oft eine lange, ununterbrochene Folge von Nullen übertragen werden muss. Anstatt nun jede Null einzeln zu übertragen, werden Folgen zusammengefasst und als X mal 0 übertragen.

- **DPCM (Differential Puls Code Modulation)**

Bei der DPCM wird der Umstand genutzt, dass bei Bildmaterial viele ähnliche Werte auftreten. Dies kann sowohl innerhalb eines Bildes zwischen benachbarten Pixels sein, wenn Flächen dargestellt werden (Intraframe), als auch zwischen aufeinander folgenden Bildern eines

[1] Quelle: Schmidt, Ulrich: Professionelle Videotechnik. - Berlin [u.a.] : Springer, 2000 S.116-117

Videos (Interframe). Abgesehen von Bildern bei denen ein Schnitt im Video liegt oder bei sehr schnellen Bewegungen werden sich zwei Nachbarbilder immer ähnlich sein.
Dies wird ausgenutzt indem nicht das komplette Bild übertragen wird, sondern eine Vorhersage getroffen wird, wie das nächste Bild wahrscheinlich aussehen wird. Dann wird nur die Abweichung des Bildes von dieser Vorhersage übertragen. Die Effektivität der Kompression kann weiter gesteigert werden, indem nicht nur das vorhergehende Bild für die Vorhersage herangezogen wird, sondern auch das folgende. Dafür muss natürlich beim Encoder das Eintreffen der Folgebilder abgewartet werden, bevor das Encoding stattfinden kann. Dies ist zwar auch ein Faktor, der sich auf die Latency auswirkt, allerdings kann dieser durch eine geringe Anzahl der voneinander abhängigen Bilder (**G**roup **o**f **P**ictures) relativ klein gehalten werden. Eine GOP setzt sich aus I-Frames (Intra Frame), P-Frames (Predicted Frames) und B-Frames (Bi-Directional Predicted Frames) zusammen. Durch die bidirektionale Vorhersage können auch gleichmäßige Bewegungen im Bild verarbeitet werden. Wenn sich ein Objekt in einem Bild an einem bestimmten Punkt des Bildes befindet und zwei Bilder danach an einem anderen, kann davon ausgegangen werden, dass es sich im Bild dazwischen auf einem Punkt befindet der auf der Linie zwischen diesen beiden Punkten liegt. [2]

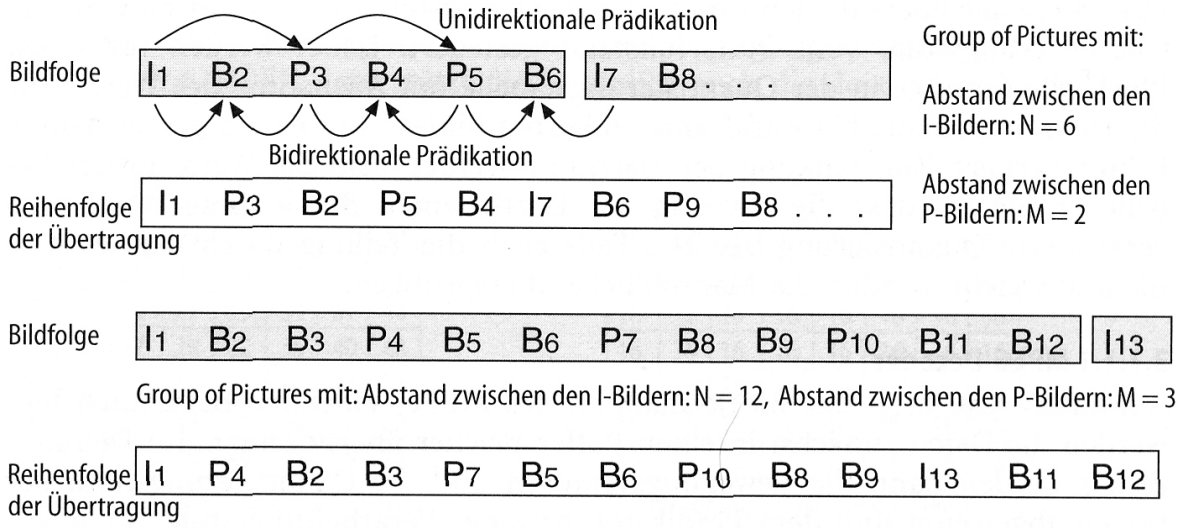

Abbildung 2: Aufbau und Übertragungsreihenfolge einer Group of Pictures (GOP) [2]

[2] Quelle: Schmidt, Ulrich: Professionelle Videotechnik. - Berlin [u.a.] : Springer, 2000 S. 134

- **Diskrete Cosinus Transformation (DCT)**

Bei der Diskreten Cosinus Transformation wird das Frequenzspektrum eines Videosignals zur Komprimierung verwendet. Tiefe Frequenzen ergeben sich durch größere Strukturen und langsamen Anstieg oder Abfall der Helligkeit, hohe Frequenzen durch schnelle Strukturübergänge und Helligkeitsänderungen.

Das Bild wird bei der DCT in Bildblöcke (üblich sind 8x8 Pixel) zerlegt und einzeln transformiert. Das Ergebnis ist eine Transformationskoeffizienten-Matrix.

„Man erhält je 8*8-Block drei Matrizen mit DCT-Koeffizienten. In jeder Matrix steht links oben der DC-Koeffizient (Gleichanteil), nach rechts und nach unten hin stehen die so genannten AC-Koeffizienten der höherfrequenten Grundfunktionen.“[3]
„Die DCT konzentriert die Signalenergie eines Blockes in den 'niedrigen' Koeffizienten, vor allem im DC-Koeffizienten. Die höheren AC-Koeffizienten sind meist 0 oder fast 0, da der überwiegende Anteil der visuellen Information eines Bildes mit kontinuierlich verteilten Werten im Bereich niedriger Frequenzen liegt. Der Grundgedanke dabei ist das in einem natürlichen Bild (Foto, etc.) scharfe Kanten, Linien, abrupte Farbwechsel etc. eher selten sind. Daher ist der Unterschied zwischen 2 benachbarten Farbpunkten meist sehr gering. Deshalb konzentriert sich der größte Teil des Ausgangssignals in den unteren Frequenzen.“[4]
Dies bringt optimale Bedingungen für eine Kompression mit sich, da wichtige Werte auf kleine Bereiche beschränkt werden.

- **Bedeutung für Videostreaming**

Die meisten Kompressionsformate die heute für Streaming Inhalte eingesetzt werden nutzen eine Kombination der oben besprochenen Kompressionsmethoden.

[3] http://www.improve-mtc.de/Veroffentlichungen/Multimedia2/multimedia2.html

[4] http://www.fim.uni-linz.ac.at/lva/Rechtliche_Aspekte/2001SS/Stegano/verfahren_bild.htm

2.3.3 Qualitätskontrolle

Die Qualität eines Videostreams zu überprüfen ist nicht für jeden Endbenutzer möglich. Durch Unterschiede bei der Bandbreite kommt es zwangsläufig zu unterschiedlichen Qualitäten. Will man eine solche Kontrolle erreichen, müsste jedem Benutzer eine Möglichkeit zur Bewertung des Streams gegeben werden, welche wiederum nur eine subjektive Beurteilung der jeweiligen Person bringen würde. Dies würde ein sehr ungenaues Bild wiedergeben, da je nach Anspruch und Erfahrung des Benutzers beim betrachten von Videostreams, ein Stream derselben Qualität von einem Benutzer als schlecht und von einem anderen als sehr gut bewertet werden könnte.
Sinnvoller ist es Mechanismen zu nutzen, die eine Qualitätssicherung bzw. Erhöhung der Qualität von technischer Seite bewirken.

Eine Möglichkeit bietet Surestream bzw. Multibit oder Dynamic Bandwidth Detection (Bezeichnungen der verschiedenen Firmen).
Bei diesen Verfahren wird der Stream in verschiedenen Bitraten encodet, die Bandbreite die zwischen Streaming Server und Benutzer zur Verfügung steht wird ermittelt und dann jener Stream übertragen, der die beste Qualität bietet und trotzdem noch ohne Probleme übertragen werden kann. [4]
Bei Surestream (RealNetworks) und Multibit (Microsoft) sind diese Technologien direkt in die jeweiligen Encoder eingebunden. Bei Dynamic Bandwidth Detection (Adobe) wird Actionscript und JavaScript genutzt, um die verfügbare Bandbreite zu ermitteln und den passenden Stream auszuwählen. [5, 6, 7]

Natürlich wäre auch eine Erhöhung des Puffers bei der Wiedergabe des Streams eine Maßnahme, die der Qualität des Streams speziell bei Schwankungen der verfügbaren Bandbreite zugute kommen würde. Da sich diese Arbeit aber mit Möglichkeiten befassen will die Verzögerung zwischen Encoding und Wiedergabe des Streams möglichst gering zu halten, werden hier vorwiegend andere Mechanismen behandelt.

Bei Multicast in geschlossenen Netzwerken ist eine Qualitätskontrolle wesentlich einfacher durchführbar. Einerseits liefert das RTCP (Real Time Control Protocol) wichtige Daten zur Qualität der Verbindung und andererseits ist es in geschlossenen Netzwerken möglich von dem Best Effort Prinzip, welches im Internet angewendet wird, abzuweichen. Dieses besagt das alle Daten gleichermaßen, so schnell wie möglich, vom Ausgangspunkt zum Endpunkt transportiert werden. In geschlossenen Netzwerken gibt es die Möglichkeit bestimmte Daten mit höherer Priorität zu behandeln als andere oder Bandbreiten für bestimmte Anwendungen zu reservieren. Dies wäre zwar rein technisch auch im Internet möglich, allerdings stellt sich hier die Frage, wer Bandbreiten wofür reservieren darf.

2.4 Datenraten

Die Entscheidung welche Datenrate für das Streaming sinnvoll ist, hängt von verschiedenen Faktoren ab.
Die wichtigste Frage dabei ist, für welche Benutzer der Stream gedacht ist und welche Bandbreite deren Internetanbindung in der Regel hat. Diese Frage muss geklärt sein bevor Faktoren wie die Bild- oder Tonqualität des Streams in Betracht gezogen werden, da es sinnlos ist zuerst die gewünschte Qualität sicherzustellen, wenn die Empfänger den Stream dann gar nicht oder nur mit Unterbrechungen empfangen können.

Wenn der Stream nicht ausschließlich eine Zielgruppe anspricht, die mit hoher Wahrscheinlichkeit über einen Breitband Anschluss verfügt, sollte genau abgewogen werden, ob auch mit einer niedrigeren Datenrate eine ansprechende Qualität erreicht werden kann.

Auch ist zu bedenken, dass im Normalfall nicht die volle nominelle Bandbreite für den Videostream zur Verfügung steht. Daher sollte immer eine Datenrate gewählt werden, die etwas unterhalb der wahrscheinlichen Bandbreite der Zielgruppe liegt.

Verbindung	Video Daten	176 x 132	240 x 180	320 x 240	640 x 480
28.8 kbps Modem	20 kbps	hervorragend	gut	brauchbar	schlecht
56 kbps Modem	34kbps	hervorragend	gut	brauchbar	schlecht
64 kbps single ISDN	45kbps	hervorragend	gut	gut	schlecht
112 kbps dual ISDN	80kbps	hervorragend	hervorragend	hervorragend	brauchbar
256 kbps DSL	225 kbps	hervorragend	hervorragend	hervorragend	hervorragend
384 kbps DSL	350 kbps	hervorragend	hervorragend	hervorragend	hervorragend
512 kbps DSL	450 kbps	hervorragend	hervorragend	hervorragend	hervorragend

Abbildung 3: Durchschnittliche Datenraten über verschiedene Internetanschlüsse (2004) [5]

Die in der Tabelle angegebenen Qualitäten können in diesem Fall nur als Richtwert verstanden werden und sind auch abhängig vom Codec, der zur Komprimierung des Videosignals verwendet wird.

2.5 Latency und Buffer-Time

Die Verzögerung von dem Moment in dem ein Video Stream erzeugt und ins Netz eingespeist wird, bis zu dem Moment in dem der Empfänger das Bild sehen kann, wird als Latency (Latenzzeit, Wartezeit) bezeichnet. Diese setzt sich aus der Dauer des Encodings, des Upstreams zum Server, des Downstreams zum Empfänger, der Buffertime (Pufferzeit) und des Decodings zusammen. „Pufferzeit gibt den Zeitraum in Sekunden an, der vor dem Beginn der Videowiedergabe für die Zwischenspeicherung von Videomaterial erforderlich ist.“ [6]
Diese ist veränderbar und damit der entscheidende Faktor bei der Verkürzung der Latency. Wird von einem Live-Stream ausgegangen, so läuft das Encoding in Echtzeit. Damit ist hier keine Zeitersparnis möglich. Die Qualität der Verbindung ist zwar ein wichtiger Faktor bezüg-

[5] Quelle: http://www.uni-koeln.de/rrzk/multimedia/dokumentation/streaming/streaming.html, 02.04.2008

[6] Quelle: http://livedocs.adobe.com/dreamweaver/8_de/using/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=18_med39.htm, 29.02.2008

lich der Latenzzeit, allerdings kann sie weder vom Sender noch vom Empfänger beeinflusst werden.

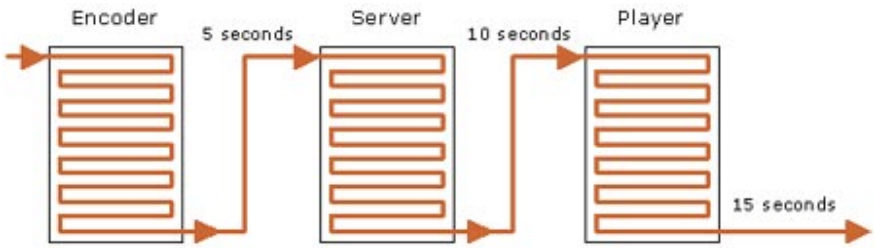

Abbildung 4: Latency durch Pufferung bei Encoder, Server und Player [7]

Eine geringe Pufferzeit hat den Vorteil, dass ein Videostream schnell mit dem Abspielen beginnt, da das Füllen des Pufferspeichers nicht viel Zeit beansprucht. Für sehr zeitkritische Anwendungen wie Videokonferenzen, Sportwetten und andere Arten der Interaktion sind geringe Pufferzeiten essentiell. Streams können auch ohne Pufferzeit wiedergegeben werden, was aber eine konstante Netzwerkverbindung voraussetzt. Bei Schwankungen der verfügbaren Bandbreite verhindern gepufferte Daten, dass der Stream zu stocken beginnt. Die nominelle Bandbreite einer Verbindung sollte beim Streaming, wie schon in 2.4 erwähnt, nicht voll ausgeschöpft werden, da selbst bei geringen Schwankungen, denen Internetverbindungen praktisch immer unterworfen sind, auf den Pufferspeicher zurückgegriffen werden muss und selbst wenn die Verbindung wieder die volle Leistung bringt, kein nachladen in den Pufferspeicher möglich ist.

Bei Streaming mit geringem oder ohne Pufferspeicher ist besonders darauf zu achten, dass deutlich mehr Bandbreite zur Verfügung stehen muss, als die Datenrate des übertragenen Videos, da bei Schwankungen sonst sofort die Wiedergabe ins Stocken gerät.

[7] Quelle: http://www.microsoft.com/windows/windowsmedia/howto/articles/BroadcastDelay.aspx, 17.03.2008

3 Content Delivery Networks (CDNs)

Content Delivery Networks sind im Internet momentan der Weg, um mit Unicast Streaming eine große Anzahl an Benutzern überall auf der Welt zu erreichen. Die Preise, die an Content Delivery Networks gezahlt werden, richten sich nach der übertragenen Datenmenge. Streaming ist eine wichtige, aber bei Weitem nicht die einzige, Anwendung bei der CDNs zum Einsatz kommen. Auch statische Inhalte, wie Bilder, Webseiten oder File-Transfers können bei einer großen Anzahl von Nutzern Bandbreitenprobleme mit sich bringen, die man durch den Einsatz von CDNs vermeiden kann.

Der Aufbau eines CDNs besteht aus vielen, an verschiedenen Orten betriebenen Servern, auf denen Daten, die von einer großen Anzahl von Empfängern benötigt werden gespiegelt sind und die so eine größere Bandbreite zur Verfügung stellen. Außerdem ist auch ein eigenes Routing-Verfahren nötig, um einen Empfänger auf den richtigen Server zu leiten.
Dabei müssen verschiedene Faktoren berücksichtigt werden. Nur jeden Empfänger auf den geographisch am nächsten gelegenen Server zu routen wäre zu wenig. Wenn der nächste Server zum Beispiel bereits durch den Zugriff sehr vieler Empfänger überlastet ist oder ein Router auf dem Weg dorthin ausfällt, könnte auch ein weiter entfernter Server den zuverlässigeren Datenstrom liefern. In den meisten Fällen wird der am nächsten gelegene Server aber auch der günstigste für den jeweiligen Empfänger sein. [8]

Für Content Delivery Networks ist es entscheidend gute Anbindungen an Backbones von großen ISPs zu haben, um für möglichst viele Nutzer eine gute Qualität der Datenübertragung zu erreichen. Dies gilt besonders weil ISPs kein großes Interesse daran haben Daten die nicht für ihre eigenen Kunden bestimmt sind auf dem besten Weg zu übertragen. Dies zeigt sich auch in der Beantwortung einer Frage durch einen Mitarbeiter eines Content Delivery Networks:
„As video content in the internet keeps growing, which are the weakest points in delivering large amounts of data except for the last mile (where are the most likely points for network congestion)?"

“The peering between the providers is the biggest problem. One colleague once called the way the provider treat transit traffic the “hot potato principle”. Every provider wants to get rid of the transit traffic as soon as possible, since it usually does not bring revenue.”

4 Unicast und Multicast

4.1 Definition

4.1.1 Unicast

Direkte Übertragung von Daten zwischen zwei Punkten. Selbst wenn mehrere Empfänger zur gleichen Zeit die gleichen Daten von einem Server anfordern, wird für jeden Empfänger ein eigener Datenstrom erzeugt. [9]

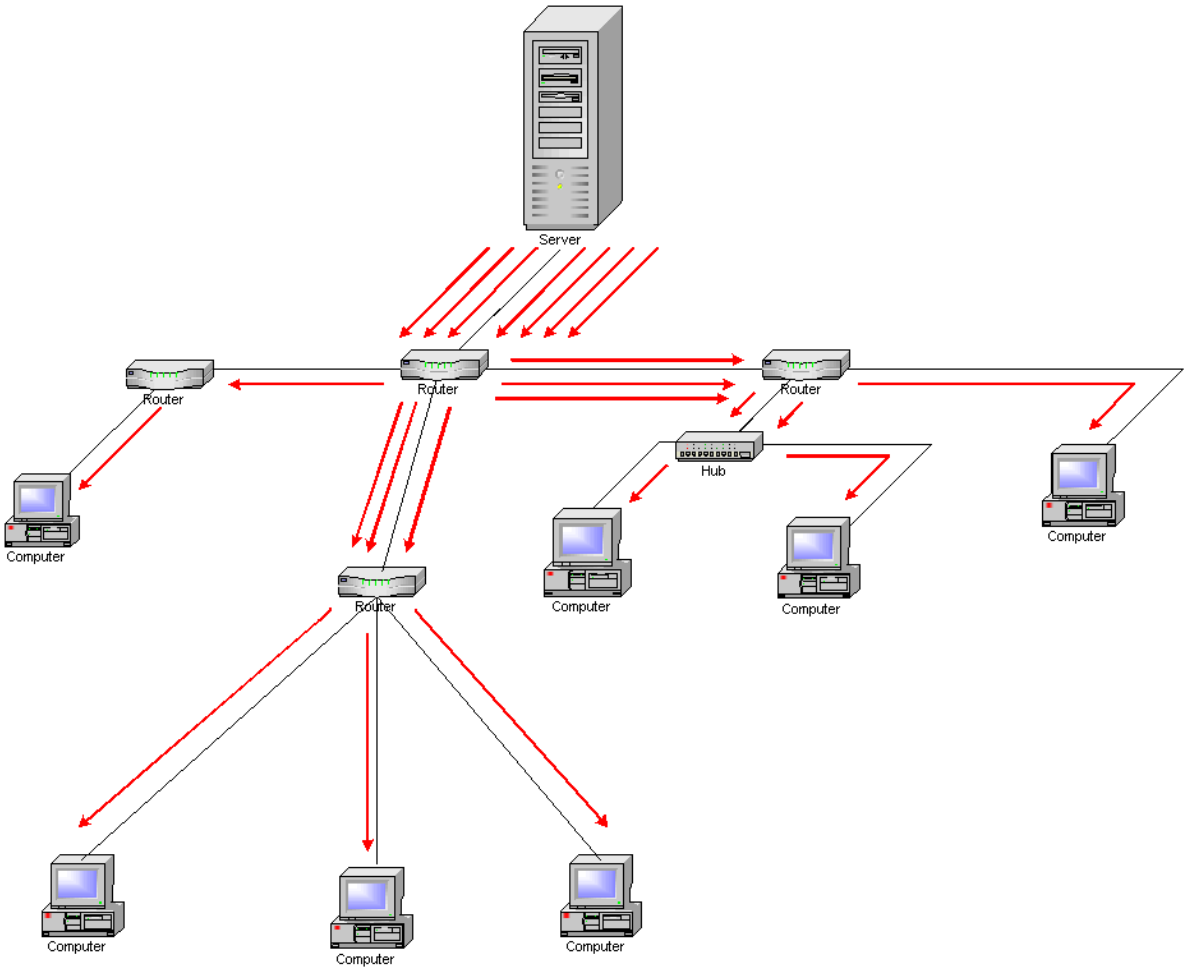

Abbildung 5: Übertragung im Unicast Verfahren [8]

Unicast ist die übliche Kommunikationsform im Internet. In der Regel werden Daten von einem Server angefordert und dieser stellt sie dem Empfänger in einer Point-to-Point Verbindung zur Verfügung.

[8] Quelle: www.surfnet.nl, 08.02.2008

4.1.2 Multicast

Übertragung eines Datenstroms an eine Gruppe von Empfängern. Der Datenstrom wird dabei nur einmal gesendet und von Routern die Multicast-Protokolle unterstützen vervielfältigt, wenn sich der Weg zu den verschiedenen Empfängern verzweigt.

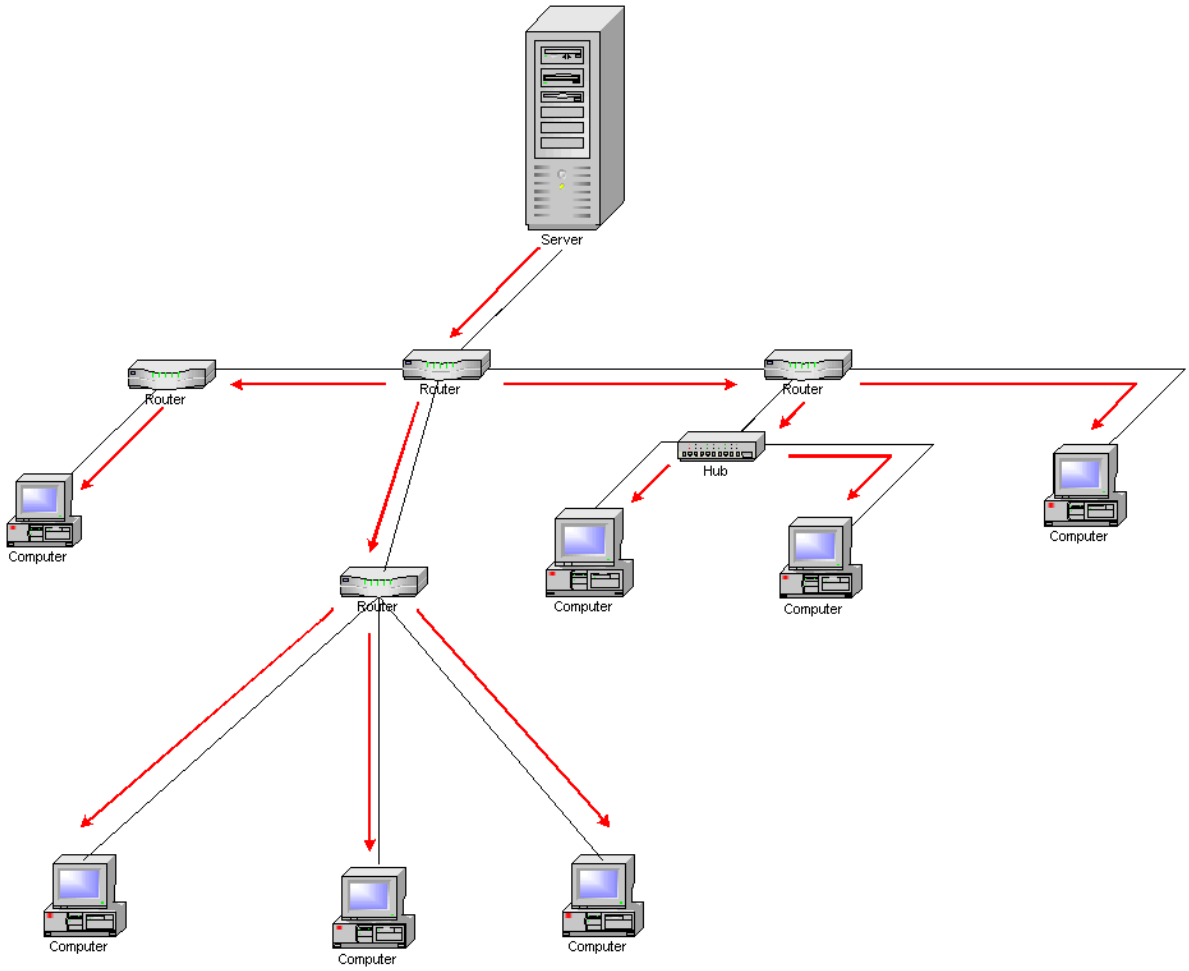

Abbildung 6: Übertragung im Multicast Verfahren [9]

[9] Quelle: www.surfnet.nl, 08.02.2008

Im Moment wird Multicast im Internet nicht von vielen Routern unterstützt. Daher begann die IETF (Internet Engineering Task Force) bereits 1992 ein virtuelles Netzwerk, das MBONE, aufzubauen. Dieses Netzwerk nutzt die bereits vorhandenen Ressourcen des Internets (Kabel, Router, etc.) und erfordert nur an den Endpunkten spezielles Equipment.
Die Umsetzung dieses Netzwerks erfolgte durch Protokolle, die es ermöglichten Multicast-Protokolldaten in Unicast Protokoll Pakete zu verpacken (engl.: tunneling) und so eine Übertragung zwischen nicht Multicast fähigen Routern zu ermöglichen. Wenn ein solches Datenpaket einen Multicast fähigen Router erreicht, wird es als Multicast Paket erkannt und weiter verarbeitet.

Im großen Umfang durchgesetzt hat sich das MBone allerdings nicht. Momentan ist es kaum möglich aktuelle Informationen über das MBone zu finden. Allerdings kann man zum Beispiel auf der Webseite des Greek Research and Technology Network unter http://vod.grnet.gr/mbone_en.php einige Organisationen finden, die zumindest zeitweise über das MBone Videos übertragen. Auch eine Seite mit Terminen für Live-Broadcasts (richtigerweise müsste es Multicast heißen, da nur an eine bestimmte Gruppe gesendet wird) ist abrufbar. Voraussetzung um diese Streams abrufen zu können ist, dass der eigene Internet Sevice Provider Multicast Übertragungen unterstützt.
Durch die hohe Bandbreite, die für Videostreaming benötigt wird, wäre Multicasting geeignet die Datenmengen, die im Internet übertragen werden, deutlich zu reduzieren.

Im Jahr 2007 machte der Anteil von Videostreaming in Deutschland 8,26 % aus. Noch nicht in diesen 8,26% enthalten sind Videostreams, die über HTTP übertragen wurden. [10]
Es ist in den kommenden Jahren noch ein deutlicher Anstieg zu erwarten, da es einen klaren Trend zu mehr Breitband-Internetanschlüssen in privaten Haushalten gibt, die eine Nutzung von Videostreaming Inhalten möglich machen.

	2003	2004	2005	2006	2007
	Breitbandinternetanschluss (in % aller Haushalte)				
Europäische Union (27 Länder)		14	23	30	42
Europäische Union (25 Länder)		14	23	32	43
Europäische Union (15 Länder)			25	34	46
Deutschland	9	18	23	34	50
Österreich	10	16	23	33	46

Abbildung 7: Verbreitung von Breitband Internet in der Europäischen Union [10]

In erster Line wäre die Multicast Technologie für Live-Streaming interessant, da bei On Demand Streams die Möglichkeit den Startzeitpunkt des Streams zu wählen oder zu einer beliebigen Stelle des Videos zu springen durch Multicast verloren gehen würde.

Zur zukünftigen Verbreitung von Multicast im Internet nahm ein Mitarbeiter eines Content Delivery Netzwerks in einem Interview Stellung:

„From my experience in my former company were we sold test equipment and also had solutions for multicast testing my take on this question is that the complexities of setting up a multicast network within one provider is challenging enough in terms of setting it up, maintaining and troubleshooting. I don't see a critical mass of internet providers setting up a large scale multicast network. Also a multicast network can most probably only help for live streaming. In fact, the way our content delivery networks work today is pretty close to a multicast network since we pull the live and ondemand streams through the backbones only once and the edge servers which are already in the provider backbones split the live and ondemand streams out into local unicasts to each enduser. Therefore our approach works with existing networks and helps with both live streaming and ondemand streaming. I also talked to some contacts from a large worldwide ISP and they also thought that this will not happen. One of the reasons

[10] Quelle: Eurostat, 26.02.2008

they saw was, how do you keep track of what goes were and how does fit into the current billing models.”

4.2 Multicast-Datenfluss

4.2.1 Source Tree (Shortest Path Tree)

Bei Source trees, die auch als Shortest Path Tree bezeichnet werden, ist die Quelle des Datenstroms auch die Wurzel. Clients die einen Datenstrom empfangen möchten, der entlang eines Source Trees übertragen wird, müssen eine (S, G) Join-Anforderung an den letzten Router vor dem Empfänger senden. S steht hierbei für Source und G für Group. Um den Multicast Stream zu empfangen muss also die IP-Adresse der Quelle bekannt sein. Für den Fall der in Abbildung 5 gezeigt ist, wäre das also Join (196.7.25.12, 239.194.0.5).
Würde ein weiterer Sender an diese Gruppe senden, müsste dafür ein eigener Source Tree erstellt werden. [11]

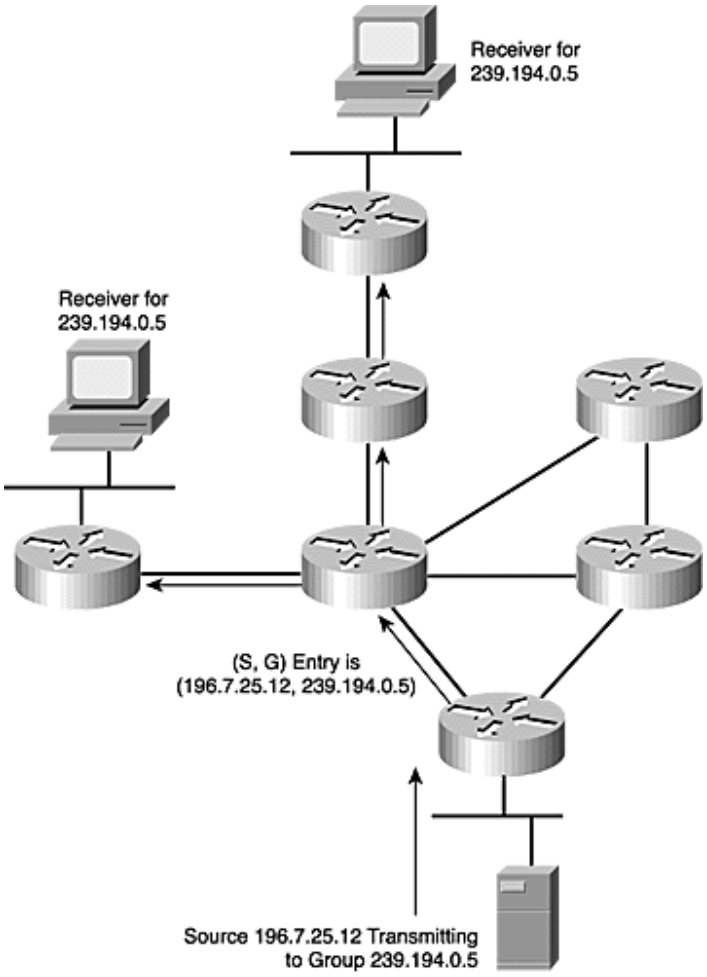

Abbildung 8: Multicast Source Tree [11]

4.2.2 Shared Trees

Bei Shared Trees ist die Wurzel ein beliebiger, vereinbarter Punkt im Netzwerk. Dieser wird als Rendezvous Point (RP) bezeichnet. Alle Multicast Sender, die an eine bestimmte Gruppe senden, müssen ihre Daten an den RP schicken, von wo sie zu den Empfängern im Netzwerk übertragen werden. Für verschiedene Gruppen (Adressbereiche) im Netzwerk können ver-

[11] Quelle: http://www.ciscopress.com/content/images/chap0x_1587051225/elementLinks/th0xfig01.gif, 26.02.2008

schiedene RPs definiert werden, es können aber auch alle Gruppen in einem Netzwerk den selben RP nutzen.
In diesem Fall muss der Empfänger also lediglich die Gruppe wissen, deren Multicast-Daten er empfangen möchte. Die Notation ist in diesem Fall (*, G), da alle Quellen die an diese Gruppe senden den gleichen RP benutzen müssen. [11]

Bei Shared Trees wird der Unicast Routing Table verwendet, um die Upstream- und Downstreamwege voneinander zu trennen. Diese Methode wird als Reverse Path Forwarding (RPF) bezeichnet und dient dazu mehrfaches Senden von Datenpaketen zu verhindern.
Durch das Konzept von Multicast kann es vorkommen, dass ein bereits von einem Router versendetes Datenpaket auf anderem Weg wieder zu demselben Router zurückkommt. Deswegen überprüft RPF, ob ein Datenpaket durch das Interface zu dem Router kommt, das zur Quelle des Datenpakets führt. Ist dies der Fall wird es weitergeleitet, wenn nicht wird es verworfen. [12]

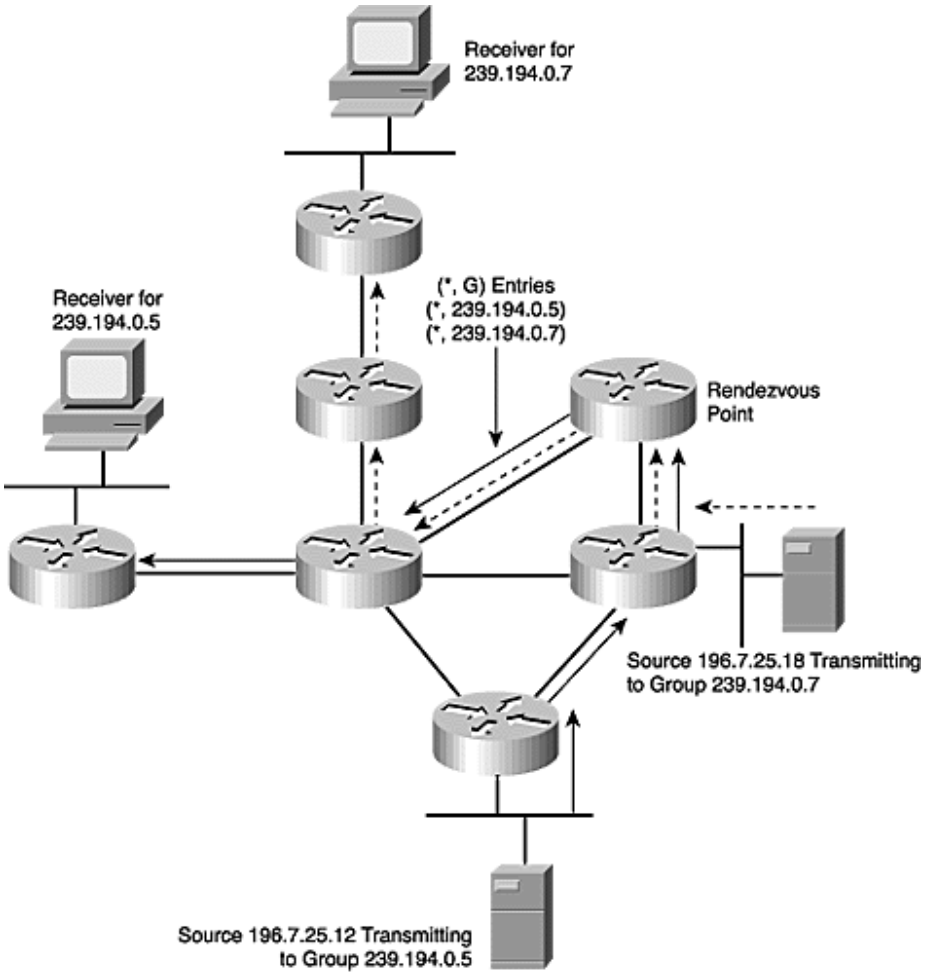

Abbildung 9: Multicast Shared Tree [12]

[12] Quelle: http://www.ciscopress.com/content/images/chap0x_1587051225/elementLinks/th0xfig02.gif, 26.02.2008

4.3 Multicast Protokolle

4.3.1 Protocol Independend Multicast (PIM)

PIM stellt eine Sammlung von verschiedenen Multicast Protokollen dar. Dabei werden Daten entlang eines Multicast Distribution Trees, bestehend aus Multicast-fähigen Routern, zu allen Empfängern weitergeleitet.

PIM Sparse Mode (PIM-SM) ist eine Pull-Methode bei der Datenpakete nur dann weitergeleitet werden, wenn sie von einem Client explizit angefordert wurden. Hierfür wird eine PIM join Nachricht an den Router versandt und von diesem den Multicast-Tree entlang weiter gegeben. Wird der Datenstrom nicht mehr benötigt, wird dies durch ein „Prune“[13] (Ablehnung, wörtlich Abschneiden) an den Router kommuniziert und die Verbindung beendet.
Dieser Mode ist am weitesten verbreitet und eignet sich für Datenübertragungen, bei denen man davon ausgehen muss, dass die meisten Clients bzw. Subnetze in dem Netzwerk nicht an den übertragenen Daten interessiert sind. Im Gegensatz zum Dense Mode wird unnötiger Traffic reduziert, da nur Daten übertragen werden, die auch tatsächlich benötigt werden. [12, 13]

PIM Dense Mode (PIM-DM) ist eine Push-Methode bei der alle Router im Netzwerk die Datenpakete erhalten. Jeder Router gibt die Pakete dann an seine benachbarten Router weiter (wenn das Paket über das Interface kommt, das zu der Quelle führt) und das setzt sich fort, bis die Daten zum Client kommen oder ein Router von all seinen Clients die Mitteilung bekommt, dass die übertragenen Daten nicht benötigt werden. Dies geschieht durch die Rücksendung eines Prune an den Router. Hat ein Router nun ein Prune von allen Clients erhalten, schickt er seinerseits ein Prune an den Router, von dem er die Datenpakete erhält.

[13] Prune: Der Router signalisiert dem im Multicast Distribution Tree über ihm liegendem Router, das kein unter ihm liegender Knoten mehr den Stream benötigt.

Das setzt sich nun fort, bis der Multicast-Tree soweit geschrumpft ist, dass nur noch Router die Datenpakete erhalten, deren Clients an der Übertragung interessiert sind.
Diese Prune Nachrichten haben allerdings eine begrenzte Time to Live (TTL), nach deren Ablauf der Datenstrom wieder an alle Router im Netzwerk weitergeleitet wird, bis diese erneut eine Prune Nachricht zurück schicken.
Daher ist der PIM-DM in seltenen Fällen und dann eher in kleinen Netzwerken geeignet, wenn davon ausgegangen wird, das sehr viele Clients im Netzwerk die Daten erhalten möchten. Mittlerweile wird diese Methode nur noch vereinzelt eingesetzt. [12, 13]

5 Streaming Protokolle

5.1 UDP (User Datagram Protocol)

UDP ist ein verbindungsloses Protokoll der Transportschicht mit sehr wenigen Protokollmechanismen. Es gibt keine Garantie für die korrekte Übertragung von Datenpaketen oder neuerliche Übertragung bei Fehlern. Der Header des UDP enthält lediglich den Quell-Port, den Ziel-Port, die Länge und eine Prüfsumme (optional) die angibt, ob das Paket fehlerhaft übertragen wurde. Diesem wird noch der IP-Header vorangestellt. [14]

Durch die einfache Beschaffenheit des Protokolls erzeugt es wenig Overhead und ist für Videostreaming besser geeignet als TCP oder HTTP. Grund dafür ist nicht nur der geringere Overhead, sondern auch, dass Mechanismen die eine korrekte Übertragung von Datenpaketen garantieren, sich bei Videostreaming meist eher störend auswirken.
Dies gilt insbesondere für Low-Latency-Streaming. Würde ein Datenpaket fehlerhaft übertragen und müsste vom Server erneut angefordert und gesendet werden, könnte es vorkommen (bei Low-Latency-Streaming durch die niedrige Pufferzeit speziell), dass dieses Datenpaket erst eintrifft, wenn die Wiedergabe des Streams diesen Punkt bereits erreicht hätte. Würde die Anwendung auf das eintreffen dieses Pakets warten, stünde der Stream so lange still, und die gepufferten Daten würden mehr. Nach eintreffen des Pakets könnte man nun entweder die Wiedergabe beschleunigen, um den Puffer wieder auf die gewünschte Größe zu reduzieren oder es würde sich jedes mal, wenn auf ein Paket gewartet wird sowohl der Speicherbedarf des Puffers als auch die Latency erhöhen.

Deshalb wird bei Videostreamings wenn möglich UDP verwendet. Allerdings lassen einige Firewalls die Übertragung mit UDP nicht zu, daher wird auch TCP und HTTP genutzt.

5.2 TCP (Transmission Control Protocol)

TCP ist ein verbindungsorientiertes Protokoll der Transportschicht, das eine Verbindung so lange aufrecht erhält, bis sie explizit getrennt wird. Zum Aufbau der Verbindung wird der „Three-Way-Handshake“ eingesetzt. Bei diesem wird von dem Computer der die Verbindung initiiert eine „Sequence Number“ im Header des TCP generiert und übertragen, der Empfänger antwortet mit einer „Acknowledgement Number“, welche wiederum von der anderen Seite bestätigt wird. Damit ist die Verbindung hergestellt.

Bei der eigentlichen Übertragung der Datenpakete wird immer der Empfang bestätigt. Es wird jedoch nicht jedes Datenpaket einzeln bestätigt, sondern immer eine Gruppe von Paketen. Bleibt die Bestätigung aus, so muss die Gruppe noch einmal übertragen werden. [4]

Das TCP ist in erster Linie für Anwendungen geeignet, für die korrekte Übertragung der Daten wichtiger ist als die Übertragungsgeschwindigkeit. Beispiele dafür sind der Versand von E-Mails, File Transfers und die Übermittlung von Webseiten. Natürlich ist auch bei diesen Anwendungen eine hohe Übertragungsgeschwindigkeit ein Vorteil, aber die Bedingung, dass die Daten ohne Fehler ankommen sollen, wiegt schwerer.

5.3 HTTP (Hypertext Transfer Protocol)

HTTP ist ein zustandsloses Protokoll der Anwendungsschicht und ist in den meisten Fällen auf TCP aufgesetzt. Seit 1990 wird die erste Version von HTTP (HTTP/0.9) verwendet. Mit HTTP können verschiedene Arten von Daten übertragen werden. [15] Um eine Kommunikation zwischen Anwendungen zu ermöglichen, bei der festgelegt wird welche Art von Daten übertragen wird, wurden MIME-Typen (Multipurpose Internet Mail Extensions) eingeführt. Ursprünglich wurden die MIME-Typen für E-Mails mit Anhängen genutzt, um klar zu trennen, welche Daten dem Mail direkt und welche dem Anhang zuzuordnen sind. Dies war nötig, da diese Mails die gesamten Daten in einer Datei enthielten.

Auch bei Daten die im Internet übertragen werden hat sich dieses Schema durchgesetzt. Web-Server und Web-Browser haben jeweils die bekannten Videotypen gespeichert. Nachdem immer wieder neue MIME-Typen erschaffen werden, kann es bei weniger bekannten Anwendungen vorkommen, dass der MIME Typ nicht auf dem Server oder im Browser gespeichert ist. Diese Listen können aber jederzeit erweitert werden. Bei Servern muss dies durch den Administrator erfolgen, bei Browsern gibt es meist Plug-Ins, die für die Erweiterung der MIME-Typen sorgen. [16]

Videoart	MIME-Type	Dateierweiterung
Flash	flv-application/octet-stream	*.flv
Shockwave	application/x-shockwave-flash	*.swf, *.cab
Quick Time	video/quicktime	*.qt, *.mov
	video/mp4	*.mp4, *.mpg4
Real	video/vnd.rn-realvideo	*.rv
	application/vnd.rn-realmedia	*.rm
Windows Media	audio/x-ms-wmv	*.wmv
	video/x-ms-asf	*.asf, *.asx
	video/x-ms-wm	*.wm
Microsoft AVI	video/x-msvideo	*.avi
VLC	video/x-mpeg	*.mp2, *.mp3
MPEG	video/mpeg	*.mpg, *.mpeg, *.mpe

Abbildung 10: Auflistung einiger bekannter Video MIME-Typen [14]

[14] Quellen: http://de.selfhtml.org/diverses/mimetypen.htm, RealNetworks, Microsoft, Adobe, 18.03.2008

HTTP ist zwar nicht das am besten geeignete Protokoll für Video Streaming, wird aber trotzdem oft eingesetzt, weil viele Firewalls so konfiguriert sind, dass sie Ports blockieren, die für Streaming mit besser geeigneten Protokollen benötigt werden.

5.4 RTP (Real Time Transport Protocol)

RTP ist ein offener, von der IETF spezifizierter Standard, der speziell für den Transport von Echtzeitdaten wie Audio und Video entwickelt wurde. Am effizientesten ist es wenn RTP auf das UDP Protokoll aufsetzt, es kann allerdings auch mit anderen Protokollen der Netzwerk- oder Transportschicht verwendet werden.
RTP kann sowohl für Unicast als auch Multicast Übertragungen eingesetzt werden, wenn diese vom Netzwerk unterstützt werden. RTP besitzt selbst keinerlei Qualitätssicherungsmechanismen, die garantieren würden, dass ein Datenpaket rechtzeitig oder überhaupt beim Empfänger einlangt. Jedes Datenpaket besitzt eine Sequenznummer. Diese kann genutzt werden um mit anderen Protokollen eine Übertragung in der richtigen Reihenfolge zu gewährleisten oder von der Applikation, um sie richtig anzuordnen.
Es handelt sich hier um ein Open Source Protokoll das ständig weiterentwickelt wird. Die neuesten Entwicklungen können unter http://www.cs.columbia.edu/~hgs/rtp/ abgerufen werden. [17]

5.5 RTSP (Real Time Streaming Protocol)

RTSP wurde von der IETF (Internet Engineering Task Force) spezifiziert und ist ein offener und erweiterbarer Standard.
Es ist ein Steuerungsprotokoll auf der Anwendungsschicht, das benutzt wird um die Übertragung von kontinuierlichen Datenströmen (in erster Linie Audio und Video) zu steuern. Die Quelle können dabei sowohl gespeicherte Dateien als auch Live-Datenströme sein.
Die Funktionalität von RSTP überschneidet sich in bestimmten Bereichen mit HTTP, in einigen Punkten unterscheidet es sich jedoch wesentlich davon. So ist RSTP im Gegensatz zu

HTTP ein symmetrisches Protokoll. Sowohl Clients als auch Server können Anfragen stellen. Ein weiterer Unterschied ist, dass die Daten (Payload) in einem separaten Protokoll übertragen werden. In erster Linie wurde das Protokoll als Ergänzung zu RTP entwickelt, ist aber nicht von einem bestimmten Übertragungsprotokoll abhängig. Das RTSP Protokoll kann auf der Transportschicht sowohl auf UDP, multicast UDP als auch TCP aufsetzen. [18]

5.6 RTCP (Real Time Control Protocol)

Das RTCP wird bei Multimediaübertragungen die mit RTP erfolgen als Kontrollprotokoll eingesetzt, da RTP selbst keine Kontrolleigenschaften besitzt.
RTCP wird auf einem separaten Port gesendet und wird in regelmäßigen Abständen an alle Teilnehmer der Session übermittelt. Durch die Übermittlung an alle kann jeder Empfänger erkennen, wie viele Teilnehmer die Session in dem Moment hat. Bei einer großen Anzahl von Teilnehmern würde in einer Multicast Session der Traffic, der durch Audio bzw. Video verursacht wird, nicht besonders ansteigen, da davon auszugehen ist, dass jeweils nur ein bis zwei Teilnehmer senden und beim Multicast Verfahren, unabhängig von der Anzahl der Empfänger, nur ein Stream für alle Teilnehmer gesendet werden muss.
Der Kontroll-Traffic würde allerdings linear mit der Zahl der Teilnehmer ansteigen, da alle Kontrollpakete an alle Empfänger gesendet werden. Deshalb wird der Abstand, in dem Kontrollpakete ausgesendet werden, immer länger, je mehr Teilnehmer an der Session beteiligt sind.

Es gibt zwei Arten von RTCP Paketen, den Sender Report (SR) und den Receiver Report (RR). Der SR unterscheidet sich vom RR nur durch einen 20-byte großen „Sender Information“ Teil, der als Information für andere Sender in der gleichen Session dient. Alle Pakete, egal ob SR oder RR, enthalten Report Blocks in denen jeweils für einen Sender mitgeteilt wird, wie viele Pakete von diesem Sender seit dem letzten Report nicht oder nur fehlerhaft empfangen wurden, die Nummer des letzten SR der von diesem Sender empfangen wurde und die Zeit, die seither vergangen ist (DSR – Delay since last SR). Die RTCP Pakete können unverschlüs-

selt oder verschlüsselt übertragen werden. Bei verschlüsselten Paketen wird ein zufälliger 32-bit Integer am Beginn des Pakets übertragen. [17]

Die erhaltenen Informationen können genutzt werden, um die Parameter des Streams entsprechend anzupassen. So kann bei einem Anstieg der verlorenen Pakete oder der Jitter („Jitter bedeutet eine Schwankung in der Genauigkeit eines Signals. Eine solche Schwankung kann beliebige Aspekte des Signals betreffen; wesentlich ist lediglich, dass ein Signal vom Standard abweicht.“[15]) die Bitrate oder Framerate reduziert werden, um eine flüssige Darstellung des Streams zu gewährleisten.
„Anwendungen können dem Protokoll Informationen wie bytes sent, packets sent, lost packets, jitter, feedback und round trip delay entnehmen und somit QoS-Eigenschaften überwachen und einhalten.“[16]

5.7 RSVP (Ressource Reservation Protocol)

Das RSVP basiert auf Intserv (Integrated Services), das dazu dienen soll Daten nach ihrer Dringlichkeit zu klassifizieren und deren Übertragung zu steuern. Es ist ein vom Empfänger initiierter Vorgang. Bei einer erfolgreichen Reservierung von Bandbreite wird diese jeweils nur in einer Richtung (vom Sender zum Empfänger) garantiert. RSVP ist ein so genanntes Soft-State Protokoll: jede Pfadreservierung enthält ein „Cleanup Timeout“, welches dafür sorgt, dass die reservierte Bandbreite wieder freigegeben wird, wenn nicht regelmäßig eine „Path Message“ empfangen wird, die weiterhin die Reservierung anfordert.
Es stehen zwei Quality of Service (QoS) Klassen zur Verfügung (in Intserv sind zwar mehr definiert, jedoch sind nur diese beiden für RSVP spezifiziert):

- **Controlled Load Service**
- **Guaranteed Service**

[15] Quelle: http://e-words.at/w/Jitter.html, 02.04.2008

[16] Quelle: http://internet-sicherheit.de/glossar.html, 12.02.2008

Controlled Load Service ist eine erhöhte Prioritätsstufe für einen bestimmten Datenstrom. Dabei wird diesem Datenstrom, unabhängig von der momentanen Gesamtbelastung des Netzwerks, jene Bandbreite zur Verfügung gestellt, die er bei einer geringen Netzauslastung nutzen könnte.

Guaranteed Service ist eine vereinbarte Mindestbandbreite und maximale Verzögerung beim Transport eines Datenstroms. Jeder Router der zwischen dem Sender und dem Empfänger liegt reserviert dann die benötigte Bandbreite explizit für diesen Datenstrom. [19]

Guaranteed Service ist für Low Latency Videostreaming besonders wegen der garantierten, maximalen Latenzzeit interessant. Allerdings bringt es das Problem mit sich, dass wahrscheinlich jeder Internetnutzer und vor allem Firmen, die mit der Arbeit im Internet ihr Geld verdienen, gerne fixe Bandbreiten auf gewissen Übertragungswegen reservieren würden. Deshalb stehen dieser Art von Datenstromverwaltung im Internet große Hürden gegenüber, die in absehbarer Zeit wohl unüberwindbar sind.

5.8 RTMP (Real Time Messaging Protocol)

RTMP ist ein geschütztes Protokoll, das von Macromedia (seit Ende 2005 Adobe) entwickelt wurde und von Flash Media Server unterstützt wird. Red5 ist eine Open Source Server Lösung für Flash Streaming (http://osflash.org/red5), die zumindest einige Funktionen abdeckt, die auch ein regulärer Flash Server liefert.

Adobe gibt an, durch das nicht öffentliche Protokoll, die gestreamten Inhalte besser zu schützen, da Rekorder schwieriger zu programmieren sind, wenn das Streaming Protokoll nicht bekannt ist. [20]
Allerdings bringt die Geheimhaltung des Protokolls auch mit sich, dass keine anderen Server für RTMP Flash Streams benutzt werden können und so Adobe die Möglichkeit gibt höhere Lizenzgebühren zu verlangen.

5.9 MMS (Microsoft Media Server Protocol)

MMS ist ein von Microsoft entwickeltes Unicast Streaming Protokoll, das sowohl über UDP als auch TCP und HTTP streamen kann.
Früher wurde die Datenübertragung über die Protokolle MMSU (Microsoft Media Server Protocol/UDP), MMST (Microsoft Media Server Protocol/TCP) oder HTTP durchgeführt, wobei ein Protocol Rollover integriert war. Dieser hatte die Funktion mit dem Client die Art der Übertragung zu vereinbaren. Zuerst wurde versucht die Daten mit MMSU zu übertragen. Wenn dies fehlschlug wurde MMST für die Übertragung benutzt. Konnte auch damit keine Verbindung erzielt werden wurde über HTTP gesendet.

Mittlerweile wird MMS auch bei Windows Media Streams nur noch für das oben beschriebene Protocol Rollover genutzt. Die Datenübertragung läuft mittlerweile auch hier über RTSP, wobei zwischen Server und Client vereinbart wird, ob über UDP, TCP oder HTTP gestreamt wird.

Bis Windows 2000 Server, WMS 4.1 bzw. Windows Server 2003, WMS 9 Series (wenn auf einen Mediaplayer früher als Version 9 gestreamt wird) auf Senderseite bzw. Windows Media Player 10 auf Empfängerseite (wenn von Windows Media Services 4.1 oder früher gestreamt wird) wird MMS noch als Datenübertragungsprotokoll genutzt.
Trotzdem werden nach wie vor MMS URLs benutzt, um den Protocol Rollover zu initialisieren. [21]

6 Streaming Server

6.1 Flash

Die Benutzung von Flash Media Server bringt als Vorteil die hohe Verbreitung des Flash Players, der zum Abspielen der Inhalte benötigt wird. Laut Angaben von Adobe sind über 98% aller Personal Computer (plattformübergreifend) mit dem Flash Player ausgestattet und selbst wenn dies etwas hoch gegriffen scheint, ist es unbestritten der Player mit dem größten Marktanteil. Nachteil bei Flash ist das proprietäre Übertragungsprotokoll, das Open Source Lösungen für Flash-Streams sehr schwierig macht.

Flash Media Server 3 bietet Streaming, außer im bereits zuvor angebotenem On2 VP6 Format, auch in MPEG4/H.264 an und implementiert damit bei entsprechender Datenrate auch HD-Unterstützung. Auch eine Erweiterung des von Adobe entwickelten RTMP-Protokolls um eine verschlüsselte Version (RTMPE) ist enthalten. [22]

Flash Media Server 3 wird in zwei Versionen angeboten. Der Flash Media Streaming Server um $995 oder €654 (Stand: 06.03.2008) [23] ist die reduzierte Variante des Flash Media Interactive Server um $4.500 (€2.958). [24]
Die Interactive Version bietet im Gegensatz zur Streaming Version unter anderem die Unterstützung von serverseitigem Actionscript, Archivierung von gestreamten Inhalten am Server, serverseitigen Playlist Support und Unterstützt C++ Plug-Ins.

Es wird auch eine gratis Flash Media Development Server Version angeboten. Diese ist vom Leistungsumfang her nahezu identisch mit der Vollversion des Interactive Server, ist allerdings auf 10 gleichzeitige Clients beschränkt.

6.2 Windows Media

Im Microsoft Windows Server 2003 ist bereits der Windows Media Server enthalten (außer in der Web Edition). Die unterstützten Formate für das Streaming über einen Windows Media Server sind ASF, WMA, WMV, MP3 und JPG. Als Streamingprotokoll wird RTSP benutzt. MMS wird für den Protocol-Rollover und für Streaming zu älteren Mediaplayer Versionen verwendet. Zuerst wird versucht eine Verbindung über UDP aufzubauen, wenn dies misslingt wird eine Verbindung über TCP aufgebaut. Schlägt auch diese Verbindung fehl, wird über HTTP gestreamt.

Bei Tests von Microsoft wurde ermittelt, dass Inhalte die über UDP gestreamt werden den Server-Prozessor weniger belasten, als wenn über TCP oder HTTP gestreamt wird. Über UDP konnten 53% mehr Clients vom gleichen Server bedient werden als über TCP oder HTTP. Im Gegensatz dazu war die Nutzung des Arbeitsspeichers am Server beim Streaming über MMSU um 60% höher als bei MMST oder HTTP.
Sowohl die Prozessorauslastung als auch die Speichernutzung steigt dabei nicht im Verhältnis zur Bitrate des Streams. So benötigt ein 220 kbps Stream mit 50 Clients wesentlich weniger Prozessorleistung als ein 22 kbps Stream mit 500 Clients. Auch bei der Speichernutzung braucht ein 22 Kbit Stream mit 730 Byte Speichernutzung pro kbps im Verhältnis wesentlich mehr Speicher als ein 220 Kbit Stream, bei dem pro kbps nur 90 Byte Arbeitsspeicher belegt werden.
Ein weiterer Faktor ist, ob ein Stream live übertragen wird oder On Demand vom Server abgerufen wird. Dabei beanspruchen Live-Streams wesentlich weniger Ressourcen als On Demand Streams. [25]

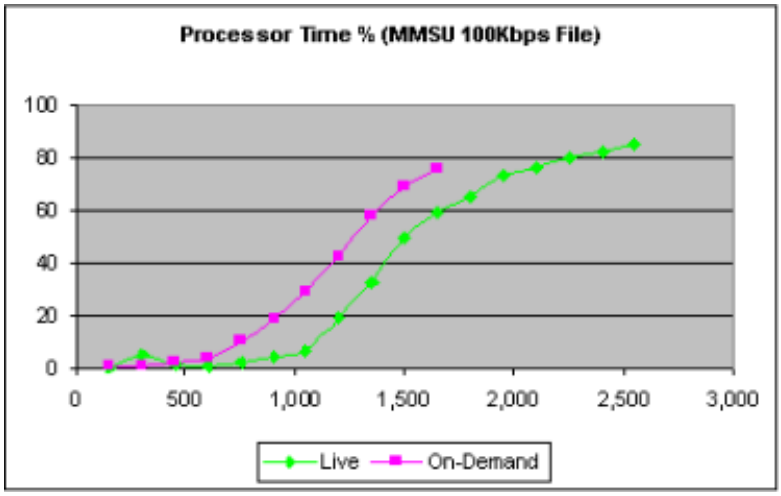

Abbildung 11: Windows Media Server, Prozessorauslastung bei Live- und On Demand Streaming eines 100 kbps Streams [17]

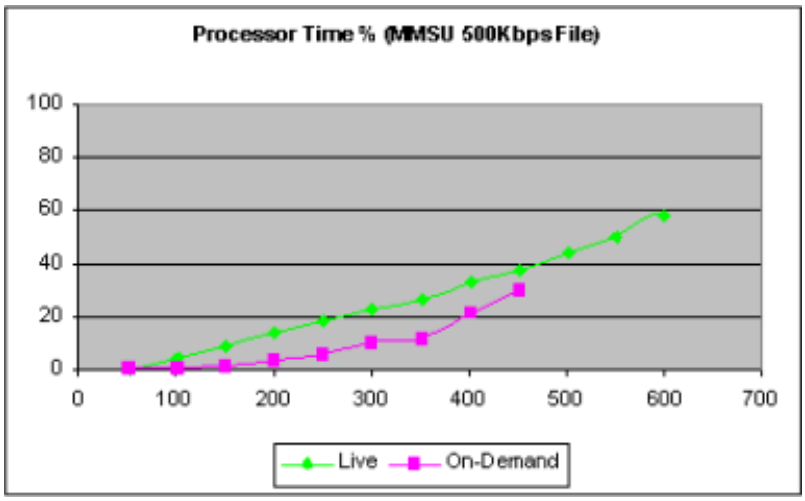

Abbildung 12: Windows Media Server, Prozessorauslastung bei Live- und On Demand Streaming eines 500 kbps Streams [17]

[17] Quelle: Microsoft, 18.03.2008

	On Demand (100 kbps)	Live (100 kbps)	On Demand (500 kbps)	Live (500 kbps)
Arbeitsspeicher/Client	103 kbps	16,7 kbps	138 kbps	20kbps
Arbeitsspeicher/kbps Datenrate	1035 bytes	168 bytes	695 bytes	100 bytes
CPU-Auslastung/ 100 Clients	Nicht linear	Nicht linear	Nicht linear	5%
Empfohlene Anzahl Clients	1050	1400	800	1000
Maximale Anzahl Clients	1750	2500	1200	1450

Abbildung 13: Windows Media Server, Vergleich Prozessorauslastung und Arbeitsspeicherbedarf bei Stream Distribution mit 100 kbps und 500 kbps [18]

6.3 Real (Helix-Server)

Der Helixserver von Real Networks ist ein Multitalent, wenn es um Videoformate geht. Er unterstützt neben Real Audio und Real Video auch Windows Media, Quick Time und 3GPP.
Es gibt drei Versionen des Helix Servers: Helix Server 25, Helix Server 100 und Helix Server Unlimited. Die Namenserweiterungen 25, 100 und Unlimited stehen dabei für die Anzahl der gleichzeitigen Streams, die mit der jeweiligen Version bereitgestellt werden können. Auch einige andere Funktionen wie Multicast für Netzwerkinterne Streams, Multicast zwischen verschiedenen Servern und Sure Stream sind exklusiv in der Unlimited Version vorhanden. [26, 27]
Die Preise für die verschiedenen Versionen und der Angebotenen Update- und Supportverträge sind in der folgenden Tabelle zusammengefasst. Der Unterschied zwischen Silver- und Bronze-Support besteht in erster Linie darin, dass der Silver Support rund um die Uhr erreichbar ist und Probleme mit erhöhter Priorität behandelt werden. [28]

[18] Quelle: Microsoft, 18.03.2008

	Helix Server Unlimited	Helix Server 100	Helix Server 25
Server-Software	$ 13.000 (€ 8.546)	$ 5.000 (€ 3287)	$ 2.000 (€ 1.315)
Essential Support - Silver	$ 5.200 (€ 3.418)	$ 2.600 (€ 1.709)	$ 1.400 (€ 920)
Essential Support - Bronze	N/A	$ 2.000 (€ 1.315)	$ 800 (€ 526)

Abbildung 14: Preisliste für Helix Server Produkte von RealNetworks [19]

6.4 QuickTime / Darwin

QuickTime Streaming Server ist im Mac OS X Server Paket enthalten. Dieses Serverpaket kostet auf 10 gleichzeitige User beschränkt $499 (€326), die unlimitierte Version kommt auf $999 (€653). [29]
Der Programmcode des QuickTime Streaming Servers ist nahezu identisch mit dem Darwin Server. Unterschied sind erweiterte Medienverwaltungs- und Administratorfunktionen. Abgesehen davon gibt es natürlich beim Darwin Server, der allerdings auch unter Linux oder Windows läuft, keinen Support von Apple. [30]

Unterstützt werden neben dem QuickTime Format auch MPEG4/H.264 und 3GP, sowohl im Unicast als auch im Multicast Modus. Für On Demand Streaming und Simulated Live-Streaming (vorgefertigte Clips, die mittels Playlist abgespielt werden, keine Interaktionsmöglichkeit für den Empfänger) müssen so genannte Hint-Tracks für jede Audio- und Videospur erzeugt werden. Diese enthalten Informationen, die dem Server mitteilen, wie die Daten verpackt werden müssen um mittels RTP gestreamt werden zu können.
Hint-Tracks können in QuickTime Pro oder anderen Videoprogrammen, die QuickTime exportieren können, erstellt werden. Wird der Quick Time Streaming Server verwendet erzeugt der Publisher, der für die Erstellung von Playlists verwendet wird, die Hint-Tracks automatisch. [31]

[19] Quelle: www.videomediasolutions.com, 06.03.2008

6.5 Red 5

Der Red5 Server ist eine Open Source Alternative zum Flash Media Server. Seit im Oktober 2005 die erste Version (0.1) von Red5 released wurde, sind schon sehr viele Funktionen implementiert worden. Die aktuelle Version ist 0.7 (Rel.: 23.02.2008) und unterstützt neben anderen Neuerungen auch Skripts (JavaScript, Groovy, Beanshell, JRuby, Jython). Ab der Version 0.8, die als „Public Beta Release“ bezeichnet wird soll intensiv getestet und Fehler ausgemerzt werden. Die Version 1.0 soll dann den vollen Funktionsumfang bieten. [32]

Informationen zu diesem Projekt, sowie der Source Code von Red5 sind unter http://osflash.org/red5 abrufbar.

7 Encoder und Decoder

Die meisten Encoding-Tools haben mittlerweile Möglichkeiten eingeführt um die Latenzzeit durch Senkung der Buffertime zu reduzieren. Dabei unterscheiden sich Methoden für das Live-Streaming und On Demand Content. Dies liegt in erster Linie daran, dass auch die Gründe warum eine geringe Buffertime gewünscht wird, bei live und On Demand nicht komplett übereinstimmen. Live-Streaming soll möglichst aktuell sein und je nach Anwendung können 15-30 Sekunden Latenzzeit schon zu viel sein. Dies betrifft insbesondere Web-TV Anwendungen die bei live Übertragungen in punkto Aktualität nicht hinter herkömmlichem Fernsehen zurück stehen möchten, sowie alle interaktiven Anwendungen, bei denen der Konsument auf die Ereignisse die er sieht reagieren soll. Bei On Demand Streaming geht es ausschließlich darum, dass das Video möglichst schnell nach dem Aufruf abgespielt werden kann, ohne das lange auf das Befüllen des Pufferspeichers gewartet werden muss. Bei Live-Streaming ist dieser Anspruch, zusätzlich zur Aktualität, ebenfalls vorhanden. In diesem Punkt sind teilweise Kompromisse zwischen Aktualität und kurzen Pufferzeiten notwendig, da die Funktionen die zu einer Verkürzung der Pufferzeit führen darauf angewiesen sind, dass bereits genügend Daten encodet wurden. Dies ist der Fall, weil die Reduktion der Pufferzeit erreicht wird, indem die volle Bandbreite für die Übertragung ausgenutzt wird um den Puffer zu füllen, anstatt zuerst den Puffer zu füllen und dann mit der Wiedergabe zu beginnen. Das bedeutet, wenn ein 600 kbps Stream über eine 1 Mbit Leitung übertragen wird, beginnt die Wiedergabe bereits bevor der Puffer gefüllt ist. Dafür wird zu Beginn des Streamings die volle Leitungskapazität ausgenutzt und nicht nur die 600 kbps, die der Stream benötigt. Dadurch kann mit den überschüssigen Daten von 400 kbps der Pufferspeicher gefüllt werden. Ist nun eine Pufferzeit von 5 Sekunden gewünscht und steht zu Beginn des Streamings die volle Bandbreite zur Verfügung, dauert es 7,5 Sekunden bis der Pufferspeicher mit 3 Mbit (600 kb x 5 Sekunden) gefüllt ist. Dies ist bei Live-Streaming grundsätzlich möglich, wenn der Encoder bereits genügend Daten encodet hat, um diesen Datenüberschuss zur Verfügung zu stellen. Dies widerspricht jedoch dem, was bei Low-Latency-Streaming erreicht werden soll, nämlich einer möglichst aktuellen Übertragung der Ereignisse.

7.1 Flash

7.1.1 Encoding

Der Flash Media Encoder 2 ist das von Adobe entwickelte Tool für Live-Streams. Auch die Erzeugung von Flash Video Dateien, die dann als On Demand File zur Verfügung gestellt werden können ist möglich. Der Flash Media Encoder encodet Videos ausschließlich mit dem On2 VP6 Codec. Es ist allerdings damit zu rechnen, dass die Nachfolgeversion auch in MPEG4 H.264 encoden können wird, da der neueste Flash Player (Version 9,0,115,0; Rel.: 03.12.2007) diesen bereits decoden kann.

On Demand Videos können auch mit anderen Videobearbeitungsprogrammen erstellt werden, die MPEG4/H.264 oder On2VP6 unterstützen.

Bei der Erstellung von MPEG4/H.264 Videos gibt es eine Eigenheit, wenn sie als Progressive Download angeboten werden. Ein MPEG4 File besteht aus diskreten Einheiten, so genannten Atoms (oder Boxes). Normalerweise ist ein solches Atom entweder ein Parent und beinhaltet andere Atoms oder es ist ein Informationsträger, aber nicht beides. Das Moov-Atom ist ein Parent und setzt sich aus mehreren Atoms zusammen in denen wichtige Informationen über Dauer und Dimensionen des Videos enthalten sind. Dieses Moov-Atom muss immer am Beginn des MPEG4 Files stehen, damit der Progressive Download funktioniert. Ist dies nicht der Fall, muss das komplette File herunter geladen werden, bevor die Wiedergabe starten kann. Bei den Adobe Produkten Premiere und After Effects Version CS3 wird das Moov-Atom momentan noch (Stand 20.08.2007) an das Ende des Files gesetzt. Es ist aber zu erwarten, dass dieses Problem entweder durch einen Patch oder spätestens in der nächsten Version behoben sein wird. Bis dahin müssen mit diesen Programmen erstellte Dateien, die für Progressive Download verwendet werden sollen „repariert“ werden. Dies kann zum Beispiel mit dem Programm qt-faststart erreicht werden (Sourcecode: http://svn.mplayerhq.hu/ffmpeg/trunk/tools/qt-faststart.c?view=markup).

7.1.2 Decoding

Jeder Benutzer, der ein Flash-Video wiedergeben möchte, muss den kostenlosen Flash-Player installieren, um die Videos decoden zu können. Wie bereits zu Beginn dieses Kapitels erwähnt, ist ab Version 9.0.115.0 auch das Decoding von MPEG4/H.264 Videos möglich. Damit können nun mp4, m4v, m4a, mov und auch 3gp Files abgespielt werden. Es wird allerdings nur MPEG4/H.264 (Base, Main, High und High 10 Profile) unterstützt und nicht MPEG4 Part 2 (Xvid, DivX, usw.). Für 3GP Files ist noch interessant, das ab dieser Version auch 3GPP timed text specification (3GPP TS 26.245) unterstützt wird. Dieser Standard bietet die Möglichkeit 3GP Files Untertitel (falls gewünscht in mehreren Sprachen) hinzuzufügen. Für die richtige Formatierung und Platzierung des Texts im Video muss allerdings der Erzeuger des Videos sorgen. Hierfür können Tools wie z.B. MP4 Box (http://gpac.sourceforge.net/doc_ttxt.php) verwendet werden.
Auf der Audio-Seite hat sich auch etwas geändert. So wird jetzt AAC Main, AAC LC und mit Spectral Band Replication verbesserte Codierung unterstützt. Mehrkanal AAC Files (5.1) werden zwar wiedergegeben, aber auf zwei Känäle gemischt und mit einer Abtastrate von 44,1 KHz. Eine Unterstützung von echtem Mehrkanal-Audio ist für eine der nächsten Flash-Player Versionen geplant. [33]
Ein großer Vorteil von Video im Flash Format ist die hohe Verfügbarkeit des Players auf privaten Computern. Auf mehr als 98% aller an das Internet angeschlossener Computer ist der Flash Player installiert. [34]

7.1.3 Verwaltung der Buffertime

Auf der Seite des Encodings ist im Flash Media Encoder 2 keine Möglichkeit gegeben die Pufferzeit zu verändern.
Bei der Wiedergabe des Videos kann auf Empfängerseite, je nach Software mit der das Video decodet wird, die Pufferzeit gewählt werden. Bei Wiedergabe mit dem in einen Browser eingebundenen Flash Player besteht auch die Möglichkeit die Pufferzeit mit Actionscript einzustellen. Dies bringt den Vorteil, dass der Empfänger die Einstellung nicht selbst vorzunehmen

braucht und daher auch kein Anwenderwissen zur Umsetzung eines Low-Latency-Streams nötig ist. Dafür kann die „NetStream.bufferTime“ Eigenschaft angesprochen werden. Im Gegensatz zu einigen anderen Streamingprodukten kann hier auch der Wert 0 gewählt werden.

7.2 Windows Media

7.2.1 Encoding

Der Windows Media Encoder, der als kostenfreier Download zur Verfügung steht ist die Standard-Anwendung für das Encoding und das Streamen von Windows Media Videos. Die encodeten Videos werden immer im wmv-Format ausgegeben. Bei den verwendeten Codecs kann man zwischen den Windows Media Video Codecs Version 7, 8 und 9 sowie dem ISO MPEG-4 Video V1 Codec wählen. Im Allgemeinen kann gesagt werden, dass der Windows Media Video Codec 9, der in 4 Profilen zur Auswahl steht, hier zu bevorzugen ist. Auch die Abwärtskompatibilität sollte hier keine Probleme machen, da es für den Windows Media Player 6.4 ein Codec Update gibt, mit dem auch Windows Media Video 9 abgespielt werden kann und in allen Nachfolgeversionen ist dieser Codec standardmäßig installiert. Bei der Audiocodierung bietet der Windows Media Encoder auch die Möglichkeit Surround-Sound zu encoden. Der Windows Media Audio Codec bietet im Profil Professional die Varianten 5.1 oder auch 7.1 an, wobei mit bis zu 768 kbps und einer Abtastrate von 96 kHz kodiert wird. Auch unkomprimiertes Audio (PCM) ist mit bis zu 1411 kbps (44 kHz, Stereo) möglich. Diese Varianten eignen sich wegen der hohen Datenrate allerdings kaum für Streaming-Anwendungen.

7.2.2 Decoding

Der Windows Media Player verdankt seine relativ hohe Verbreitung in erster Linie der Tatsache, dass auf jedem Windows Computer der Media Player im Betriebssystem inkludiert ist.

Mit einem Anteil von 83% aller an das Internet angeschlossenen Computer liegt er nach dem Flash Player an zweiter Stelle der streamingfähigen Mediaplayer. [34]
Neben den Windows-Formaten wmv und wma können eine Vielzahl von anderen Formaten wiedergegeben werden. Einige Codecs sind in der Standard-Version des Players integriert und es gibt eine große Anzahl an Codecs, die für den Windows Media Player zusätzlich erhältlich sind. Eine Aufstellung der unterstützten Dateiformate ist unter http://support.microsoft.com/kb/316992 abrufbar.

7.2.3 Verwaltung der Buffertime

Bei der Verwaltung der Pufferzeit können bei Windows Media Encoding an den beteiligten Komponenten jeweils eigene Einstellungen vorgenommen werden. Die Puffergröße in den Einstellungen des Windows Media Encoders steht nach der Installation auf 5 Sekunden, wählt man aber daneben die Checkbox „Standard verwenden“, so wird der Wert auf 3 Sekunden gestellt. Hier kann aber auch ein beliebiger Wert eingegeben werden, der zwischen einer und 100 Sekunden liegen muss.

Auch am Server kann die Pufferung deaktiviert werden. Dazu muss in der Kategorie General die Fast Cache Option ausgeschaltet werden. Dies ist allerdings nur dann möglich, wenn ein eigener Windows Streaming Server betrieben wird. Werden die Streams über ein Content Delivery Network verbreitet, ist diese Möglichkeit die Latency zu reduzieren nicht gegeben. [35]

Bei der Wiedergabe des Streams im Windows Media Player kann auch die Pufferzeit verändert werden. Hier ist dies aber nur durch den Benutzer selbst möglich. In den Optionen des Media Players kann unter dem Punkt Leistung der Puffer manuell eingestellt werden. Der Standardwert ist hier 5 Sekunden.

Eine verbesserte Lösung für Low-Latency-Streaming wird bei Microsoft momentan in dem Forschungsprojekt Conference XP gesucht. Ziel dieses Projekts ist es in Kooperation mit mehreren amerikanischen Universitäten einen institutsübergreifenden Unterricht mittels Videokon-

ferenz umzusetzen. Auch die Aufzeichnung der Unterrichtseinheiten ist geplant, um so Studenten, die den Unterricht versäumt haben oder die Inhalte noch einmal sehen möchten, die Möglichkeit zu geben die gesamte Unterrichtseinheit vom Server als Video On Demand abzurufen.

7.3 Real

7.3.1 Encoding

Für das Encoding von Real Media Videos gibt es zwei Versionen des Real Producers. Der Real Producer Plus um $199,95 oder €129,75 (Stand: 14.3.2008) ist die Vollversion mit allen Einstellungsmöglichkeiten und der Möglichkeit auf mehrere Server zu streamen (Backup-Stream). Der Real Producer Basic ist die kostenlose Version der Encodingsoftware, bei der allerdings nur die voreingestellten Streaming-Varianten zur Verfügung stehen und einzelne Parameter nicht verändert werden können.

7.3.2 Decoding

Zum Abspielen von Real Media Inhalten kann der Real Player in der Basic Version gratis heruntergeladen werden. Auch hier gibt es eine erweiterte Version, den Real Player Plus um €24,99 (Stand: 14.3.2008). Die Hauptunterschiede zwischen den beiden Versionen sind, dass beim Real Player Plus auch eine Brennfunktion für CDs und DVDs integriert ist und Umwandlungen zwischen verschiedenen Formaten möglich ist.

Beide Versionen können außer Real Videos auch die Formate Flash, MP3, AAC, WMA, MPEG, QuickTime, AVI, DivX, WAV und MIDI abspielen.

7.3.3 Verwaltung der Buffertime

Real Networks hat bei der neuen Version 11 von Real Producer und Helix Server neue Möglichkeiten implementiert die Latency zu verringern.

Latency Mode	Expected End-to-End Latency	Preroll for CBR Video	Preroll for VBR Video	Preroll for Audio-Only Clips	Error Resiliency
0–normal	10 or more seconds	4 seconds	4 seconds	23 milliseconds to 2.3 seconds depending on the codec	high
1–reduced	4 or more seconds	2 seconds	2 seconds	500 milliseconds	moderate
2–minimal	2 or more seconds	1 second	0.5 seconds	23 to 480 milliseconds depending on the codec	low

Abbildung 15: Latency-Modes bei Real Streaming [20]

Bisher waren in diesem Bereich nur wenige Einstellungen möglich. Der Real Producer Plus bot zwar die Möglichkeit die Pufferzeit zu verringern, jedoch konnte bei Server und Player keine weitere Reduktion erreicht werden.
Mit den nun auf die Anforderung einer geringen Verzögerung abgestimmten Latency Modes sollte eine Reduktion unter das Niveau der Vorgängerversionen möglich sein.

7.4 Video LAN (VLC)

Die Videolan Client (VLC) Software ist eine Open Source Software, die ursprünglich von Studenten des École Centrale Paris entwickelt wurde. In der Zwischenzeit sind viele weitere Entwickler dazugekommen und haben VLC zu einem internationalen Projekt gemacht.

[20] Quelle: Real Producer 11, User's Guide

7.4.1 Encoding

Der Videolan Client kann sowohl für das Encoding, als auch für das Decoding von Videomaterial verwendet werden. Beim Streamen über RTP/UDP stehen beim Encoding die Codecs MPEG-1, MPEG-2, MPEG-4, und MPEG-4/H.264 zur Verfügung. Es kann grundsätzlich auch in DivX, WMV, H/I 263 und MJPEG gestreamt werden. Diese Formate sind allerdings nicht Standard und daher nicht voll kompatibel.
VLC kann sowohl Unicast als auch Multicast Streams erzeugen. [36]
Allerdings ist das Streamen von Signalen die über eine Video- oder Soundkarte in den Computer übertragen werden momentan nur unter Windows und Linux möglich. [37]

Durch das Open Source Prinzip sind die Release-Intervalle sehr kurz. Sollte es also Anwendungen geben, die im Moment mit VLC nicht umgesetzt werden können, lohnt es sich die Weiterentwicklung der Software im Auge zu behalten. Zusätzlich zum VLC gibt es auch einen Videolan Server (VLS), allerdings ist die Client Version ebenso gut für Streaming zu verwenden.

7.4.2 Decoding

Auch das Decoding funktioniert mit dem VLC. Ein großer Vorteil von VLC ist, dass er sehr viele Videoformate wiedergeben kann und praktisch für alle Betriebssysteme verfügbar ist. Eines der wenigen Videoformate die nicht unterstützt werden ist Real, aber die Integration von RV30 (Real Video 8) und RV40 (Real Video 9 und 10) ist für die Version 0.9 geplant. Die momentan neueste Version (Rel.: 27.02.2008) ist 0.8.6e.

7.4.3 Verwaltung der Buffertime

Zur Verwaltung der Pufferzeit gibt es im Menü „Datei“ -> „Aufnahmegerät öffnen…“ die Möglichkeit bei Signalen, die über eine Videokarte in den Computer gelangen unter „Erwei-

terte Optionen…“ den Input Puffer („Caching“) festzulegen. Weiters kann rechts neben „Stream/Sichern“ der Caching Wert für die Stream Ausgabe festgelegt werden. Die Standardwerte sind allerdings mit 200ms beim Video Input und 300ms beim Stream Output ohnehin recht niedrig angesetzt. Hier kann also im besten Fall noch eine geringe Reduktion der Gesamt-Latency erzielt werden.

7.5 Quick Time

7.5.1 Encoding

Der Quick Time Broadcaster ist das Live-Streaming Tool von Apple, das nur für den Mac erhältlich ist. Er wird als gratis Download auf der Webseite von Apple angeboten. Die wichtigste Neuerung in der Version 1.5.2 (Rel.: 13.12.2007) ist die Unterstützung von MPEG4/H.264 Live-Streaming. Auch stehen weiterhin alle anderen Quick Time Codecs, sowie Codecs von Drittanbietern zur Verfügung, sofern sie echtzeitfähig sind. Außerdem wurde die Unterstützung für mobile Endgeräte mit der Integration von 3GPP verstärkt.
Als Übertragungsprotokoll wird RTP/RTSP benutzt. [38]

Es werden drei Übertragungsmodi angeboten:

- **Manual Unicast (Streaming zu einem einzelnen anderen Rechner)**
- **Multicast**
- **Automatic-Unicast (Streaming über einen Streaming-Server)**

Multicast setzt dabei natürlich eine entsprechende Unterstützung durch die Netzwerkkomponenten voraus und ist in erster Linie für das Streamen in begrenzten Netzwerken geeignet.

7.5.2 Decoding

Für das Decoding von Quick Time Inhalten kann der Quick Time Player gratis für Mac und Windows von der Apple-Webseite heruntergeladen werden. Die Version 7.4.1 (Rel.: 06.02.2008) unterstützt neben MPEG4 noch eine Reihe anderer Codecs wie z.B. DV-PAL und NTSC, Cinepak, Sorenson Video 2 und 3 sowie Motion JPEG. Eine komplette Auflistung der unterstützten Audio- und Videocodecs kann unter http://www.apple.com/quicktime/player/specs.html abgerufen werden. [39]

Auch bei Quick Time gibt es eine erweiterte, kostenpflichtige Version des Players. Der Quick Time Player Pro kostet $30 oder €19 auf der US-Seite. Auf Apple Deutschland wird er um € 30,49 angeboten (Stand: 19.03.2008). Er bietet zusätzlich zu den Funktionen des Quick Time Players die Möglichkeit Videos in verschiedene Dateiformate zu konvertieren und einfache Schnittfunktionen zu benutzen. [40]

7.5.3 Verwaltung der Buffertime

Beim Encoding eines Live-Streams kann im Quick Time Broadcaster zur Einstellung der Pufferzeit auf der Einstellungskarte „Netzwerk“ eine beliebige Pufferzeit numerisch eingegeben werden. Beim Decoding im Quick Time Player kann unter dem Menüpunkt „Bearbeiten“ -> „Einstellungen“ -> „Quick Time Einstellungen“ im „Streaming“-Tab die „Sofortige Wiedergabe“ aktiviert werden und der Regler auf „Sofort“ gestellt werden.

Der Quick Time Streaming Server bietet Funktionen zur Kontrolle der Latency, in den Verfügbaren Quellen konnte jedoch kein Hinweis auf Einstellungsmöglichkeiten gefunden werden. [41]

8 Low Latency Test Szenarien

8.1 Versuchsaufbau

Ziel der Versuche war es mit verschiedenen Anwendungen Videostreams mit möglichst geringer Latency zu erstellen.

Die tatsächliche Latency des Encodings isoliert zu messen ist problematisch. Obwohl viele Encoder den Input und den Output des Videosignals anzeigen können und damit die reine Encodingdauer gemessen werden könnte, ist diese alleine noch nicht aussagekräftig. Dem Anteil des Encodings an der Verzögerung muss auch der Input Puffer, also Material das vor dem Encoding in einem Pufferspeicher abgelegt wird, um eine effiziente Kompression zu ermöglichen, zugerechnet werden.
Deshalb wurde, wo dies Encoder und Player zuließen, die Latency von dem Zeitpunkt als der Encoder den Output anzeigte, bis zu dem Moment als der Player das decodete Bild anzeigte gemessen und dann mit der Latency verglichen, die sich ergab, wenn zusätzlich auf einen Server übertragen wurde.

Für die Versuche wurde als Videoquelle ein Computer mit DeckLink Videokarte genutzt, auf dem in Premiere Pro 2.0 die Aufzeichnung eines Fußballspiels, in DV-PAL Qualität, in die Zeitleiste importiert wurde. Der Videoeffekt „Timecode“ wurde auf das Material angewendet, um während des Abspielens des Videomaterials stets die verstrichene Zeit eingeblendet zu haben. Dies ermöglichte die Zeit zu messen, die bei Encoding, Decoding und Übertragung des Streams benötigt wird.
Über die DeckLink Videokarte wurde ein analoges Bildsignal ausgegeben, welches mit Hilfe einer Videomatrix zum Encoding-Rechner übertragen wurde.

Als Encoding-Rechner diente ein Pentium D mit zwei 3,2 Ghz Prozessoren und 2 GB RAM. Dieser war mit einer Osprey-210 Videokarte ausgestattet, über die das Videosignal analog in den Rechner eingespeist wurde.
Dadurch wurde eine Live-Streaming Situation simuliert, da auch beim Live-Streaming von einem analogen Videosignal auszugehen ist und keine Datei als Videoquelle vorliegt. Zusätzlich wurde mit der Software Netpeeker der Datentraffic gemessen, um die tatsächlich übertragene Datenmenge festzustellen.

Um eine Abbildung der Ergebnisse zu vereinfachen wurde für den Empfang des Streams der gleiche Computer genutzt, auf dem der Stream erstellt wurde. Durch Überprüfung des Streams auf einem zweiten Kontrollrechner wurde dabei sichergestellt, dass dieses Setup die Latency in keiner Weise beeinflusst.

8.2 Flash

Für das Encoding wurde der Flash Media Encoder verwendet. Die Encodingeinstellungen für das Video waren ein Format von 320x240, eine Framerate von 25 Bildern/Sekunde und eine Bitrate von 450 kbps. Dazu wurde das Audiosignal im MP3 Format mit einer Abtastrate von 22050 Hz und einer Datenrate von 48 kbps encodet.

Die bei anderen Encodern zu Vergleichszwecken herangezogene Variante, bei welcher der Stream direkt vom Encoder (ohne Umweg über den Server) abgerufen wird, ist bei Flash nicht möglich, da kein Pull-Streaming (Stream wird an einer public IP über einen freien Port bereitgestellt und muss vom Empfänger oder Server abgerufen werden) unterstützt wird.

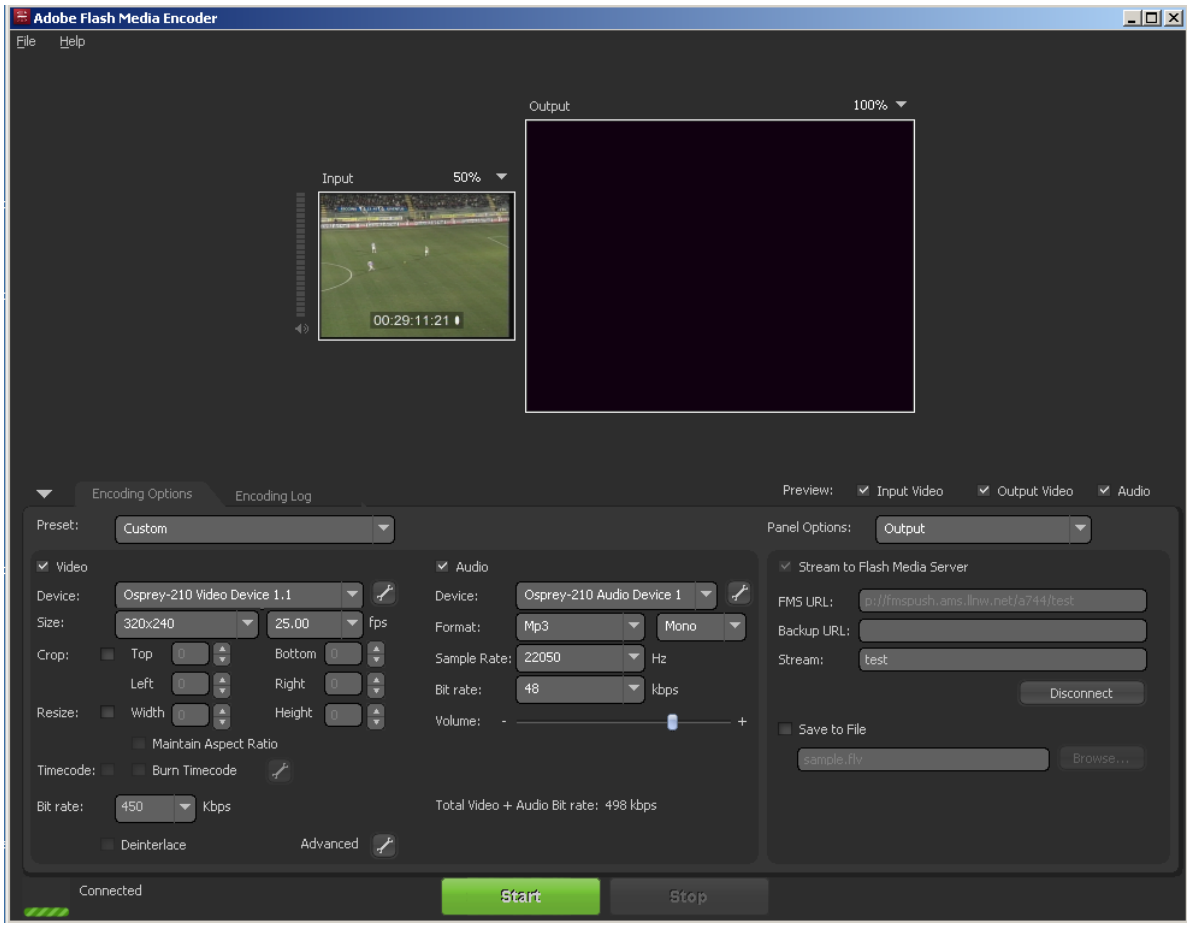

Abbildung 16: Verzögerungsmessung, Einstellungen des Flash Media Encoders

Das Ergebnis dieses Versuchs war eine Verzögerung von Encoding des Streams bis zur Anzeige im Player von 13 Frames. Bei der verwendeten Framerate von 25 Bildern/Sekunde entspricht dies einer Verzögerung von 0,52 Sekunden.

Während des Streamings wurden zwischen 60 und 70 KB/s (Kilobyte pro Sekunde) Datentraffic in jede Richtung gemessen. Dies entspricht zwischen 480 und 560 kbps (Kilobit pro Sekunde). Subtrahiert man nun von der höchsten benötigten Datenrate von 560 kbps die Datenrate des Streams von 498 kbps (450 kbps Video und 48 kbps Audio), so erhält man einen maximalen Overhead von etwa 60 kbps.

Eine ganz exakte Berechnung macht hier keinen Sinn, da die Datenrate des Streams je nach gezeigtem Inhalt und in diesem Zusammenhang mit der Effizienz des Codecs stark schwankt. Dies ist bereits an dem großen Unterschied von kleinster und größter übertragener Datenmen-

ge zu sehen. Daher ist es wahrscheinlich, dass bei anderer Materialauswahl für den Stream auch ein abweichender Wert für den Overhead entstehen würde. So unterschreitet zum Beispiel der niedrigste übertragene Wert von 480 kbps die nominelle Datenrate um 18 kbps. Dies lässt die Annahme zu, dass bei Material welches dem Codec perfekte Kompressionsmöglichkeiten bietet, überhaupt kein Overhead entstehen würde.

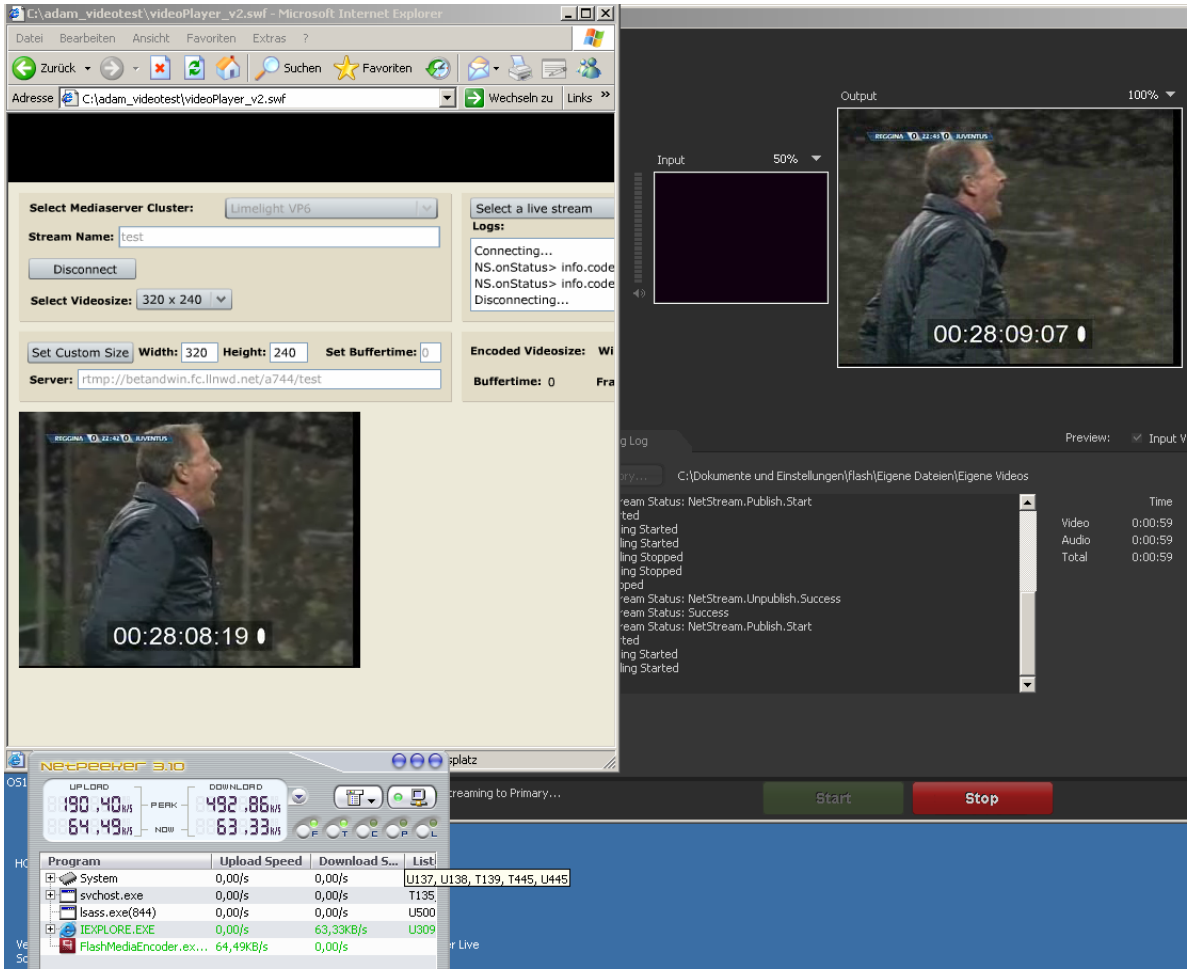

Abbildung 17: Verzögerungsmessung, Encoder und Player bei einer Flash-Übertragung

8.3 Windows Media

Für das Encoding wurde der Windows Media Encoder 9 verwendet. Die benutzten Codecs waren der Windows Media Video 9 und der Windows Media Audio 9.1. Das Video wurde in

einer Größe von 320x240, mit einer Framerate von 25 Bildern pro Sekunde und einer Bitrate von 450 kbps encodet. Das Audio-Encoding erfolgte mit einer Abtastrate von 44kHz und einer Datenrate von 48 kbps. Audio und Video hatten also gemeinsam eine nominelle Datenrate von 498 kbps.

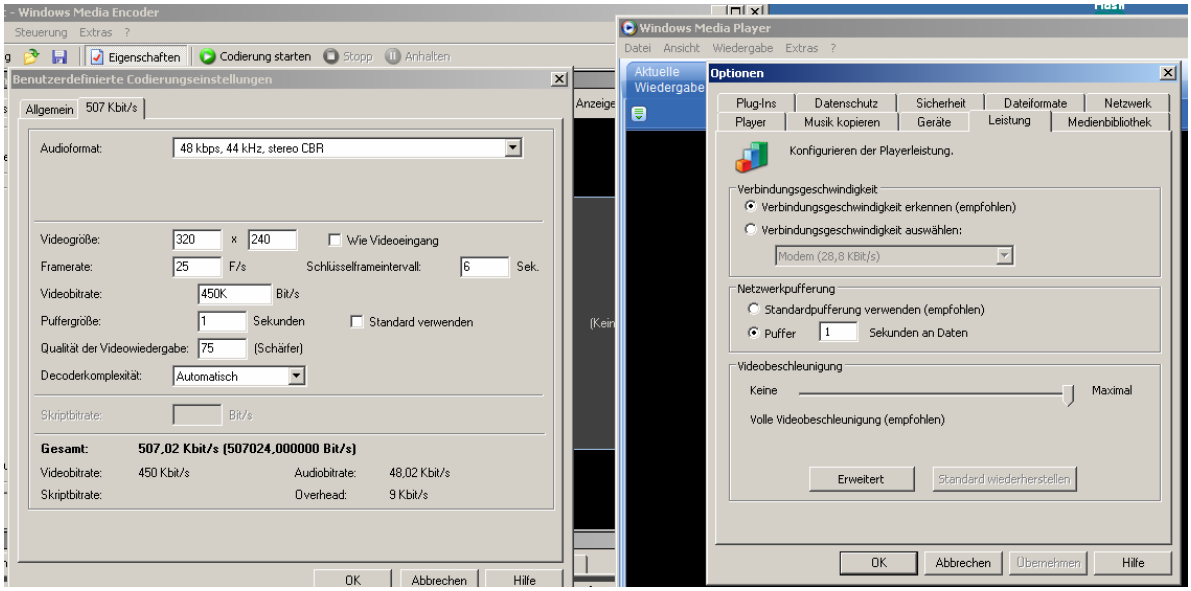

Abbildung 18: Verzögerungsmessung: Einstellungen Windows Media Encoder und Media Player

Zuerst wurde die Latency, die bei der Übertragung zu einem Windows Media Server eines Content Delivery Network entsteht, gemessen. Die Pufferzeit war sowohl beim Encoding mit dem Windows Media Encoder, als auch beim Empfang mit dem Windows Media Player auf den kleinstmöglichen Wert von einer Sekunde eingestellt.

Bei diesem Test wurde eine Latency von fast 24 Sekunden (23 Sekunden, 24 Frames) gemessen. Die durch den Server verursachte Latency konnte in diesem Fall nicht beeinflusst werden, da es sich um den Server eines Content Delivery Network handelte, bei dem keine Einstellungen vorgenommen werden konnten.

Die Datenrate bei der Übertragung schwankte zwischen 56 KB/s und 64 KB/s (448 – 512 kbps). Damit betrug der maximale Overhead 14 kbps.

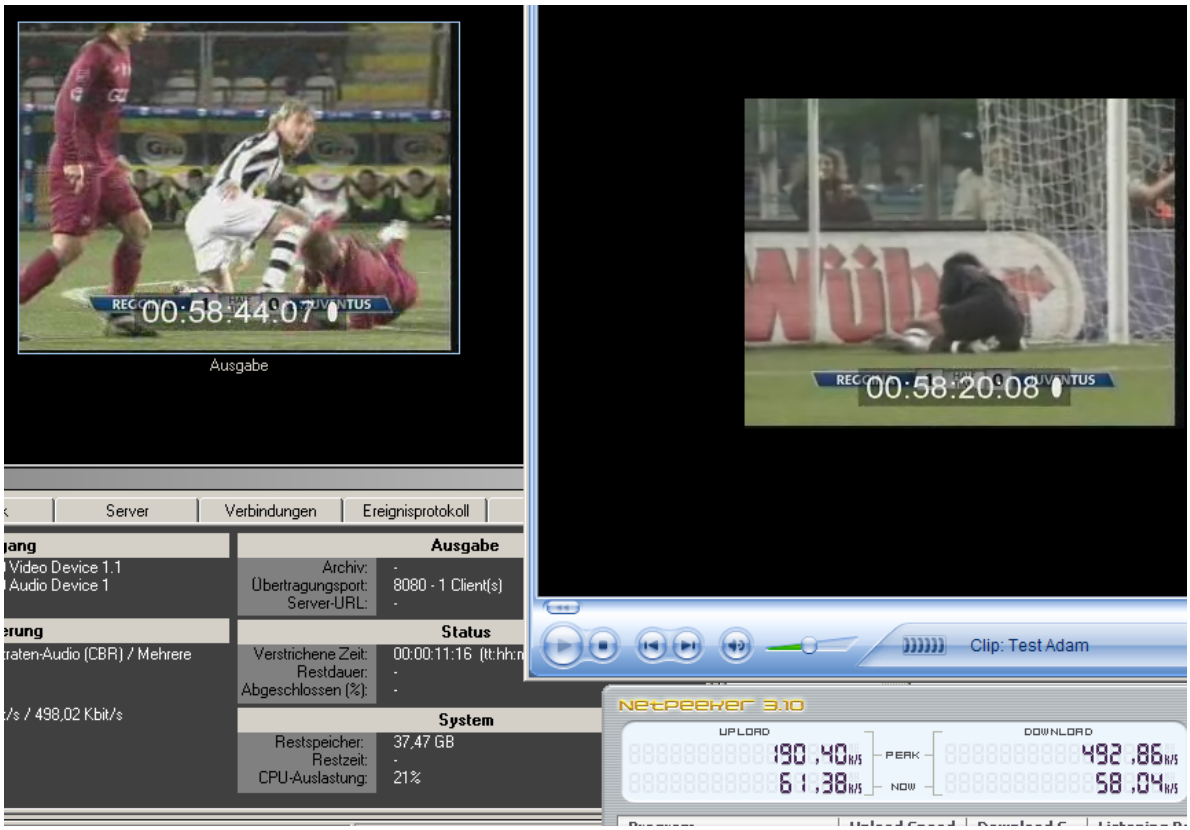

Abbildung 19: Verzögerungsmessung, Encoder und Player bei einer Windows Media Übertragung über ein Content Delivery Network

Zum Vergleich und um die Verzögerung festzustellen, die durch den Weg zum Server, die Pufferung dort und den Weg zum Empfänger verursacht wird, wurde der Stream direkt vom Encoder abgerufen.

Dabei greift der Windows Media Player direkt über die IP-Adresse des Encodingrechners auf einen offenen, im Media Encoder angegebenen Port zu. Dabei wurde eine Verzögerung von 7,56 Sekunden (7 Sekunden, 14 Frames) gemessen. Dies ist die rein bei Encoding und Decoding entstehende Latency. Diese liegt damit doch sehr deutlich über dem eingestellten Pufferwert von einer Sekunde beim Encoding und einer Sekunde beim Decoding.

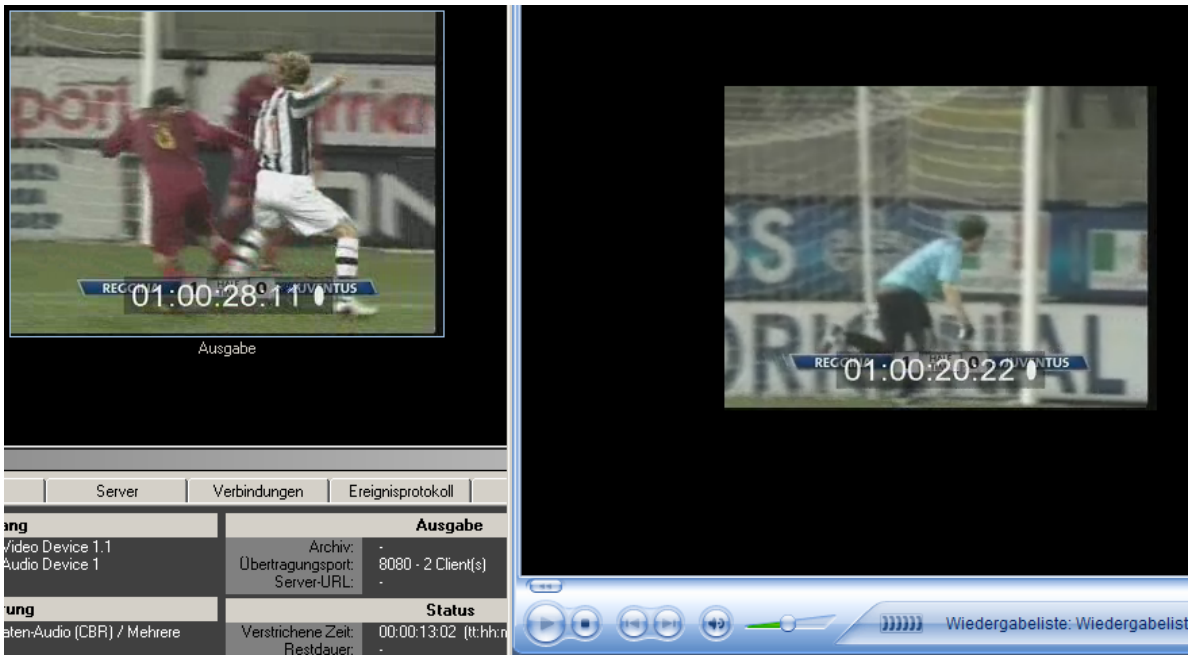

Abbildung 20: Verzögerungsmessung, Encoder und Player bei einer Windows Media Übertragung (ohne Server)

8.4 Real

Für das Encoding wurde der Real Producer Basic Version 10 benutzt. Die Basic Version unterbindet die meisten Benutzereinstellungen und lässt in erster Linie Streaming mit Default-Werten zu. Der Benutzte Real Server war ebenfalls ein Server der Real 10 Generation, daher wäre eine Verwendung der verschiedenen Latency-Modes auch mit der Vollversion des Real Producer 11 nicht möglich gewesen. Für die Kompression des Vidoematerials wurde der Real Video 10 Codec benutzt. Sämtliche Einstellungen waren durch das Profil („512k DSL or Cable“) vorgegeben und konnten in der Basic Version nicht angepasst werden. Der Audio Voice Codec war auf 64 kbps – Real Audio und der Music Codec auf 96 kbps Stereo Music Real Audio eingestellt. Auch die Framerate des Videos war mit 30 Bildern/Sekunde vorgegeben und es wurde mit einer Gesamtdatenrate von 450 kbps (Video und Audio) gestreamt.
Im Menüpunkt „Advanced Video Options“ wäre noch die Maximum Startup Latency einzustellen, aber auch dieser Wert ist in der Basic Version mit 4 Sekunden vordefiniert.

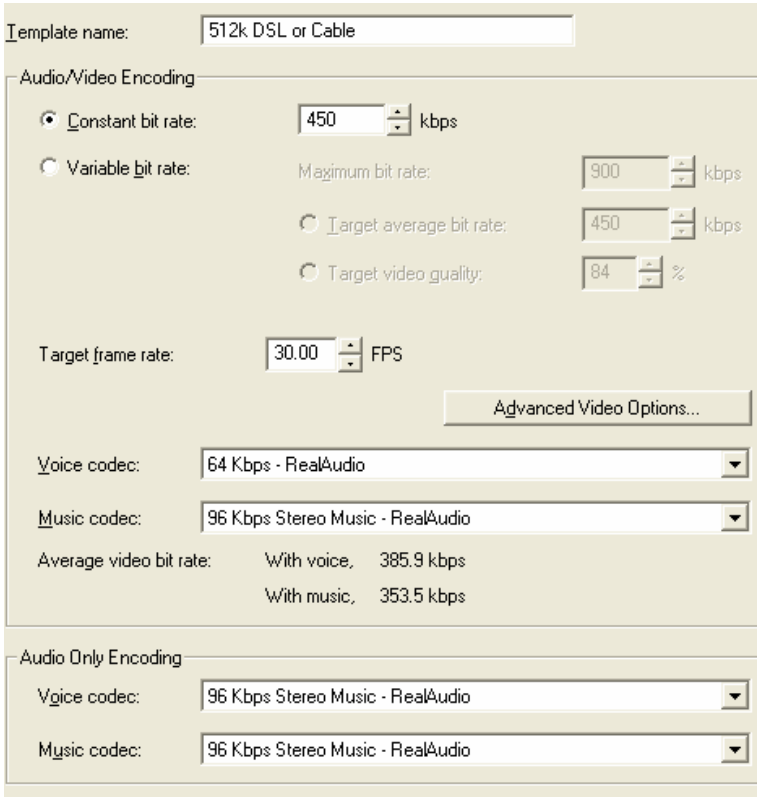

Abbildung 21: Verzögerungsmessung: Einstellungen Real Media Producer

Gemessen wurde die Latency bei der Übertragung zu einem Real Media Server eines Content Delivery Networks.

Für das Decoding wurde der Real Player 11 verwendet und bei diesem in den Optionen, bei der Funktion „Zwischenspeichern des Clips vor der Wiedergabe, falls erforderlich“, ein Wert von einer Sekunde festgelegt (Standardwert: 30s).
Bei diesem Versuch wurde eine Verzögerung von 6 Sekunden und 2 Frames (6,08 Sekunden) gemessen. Der Datentraffic bewegte sich zwischen 49 KB/s und 56 KB/s (392 – 448 kbps) und blieb damit immer im Bereich unter der nominellen Datenrate von 450 kbps.

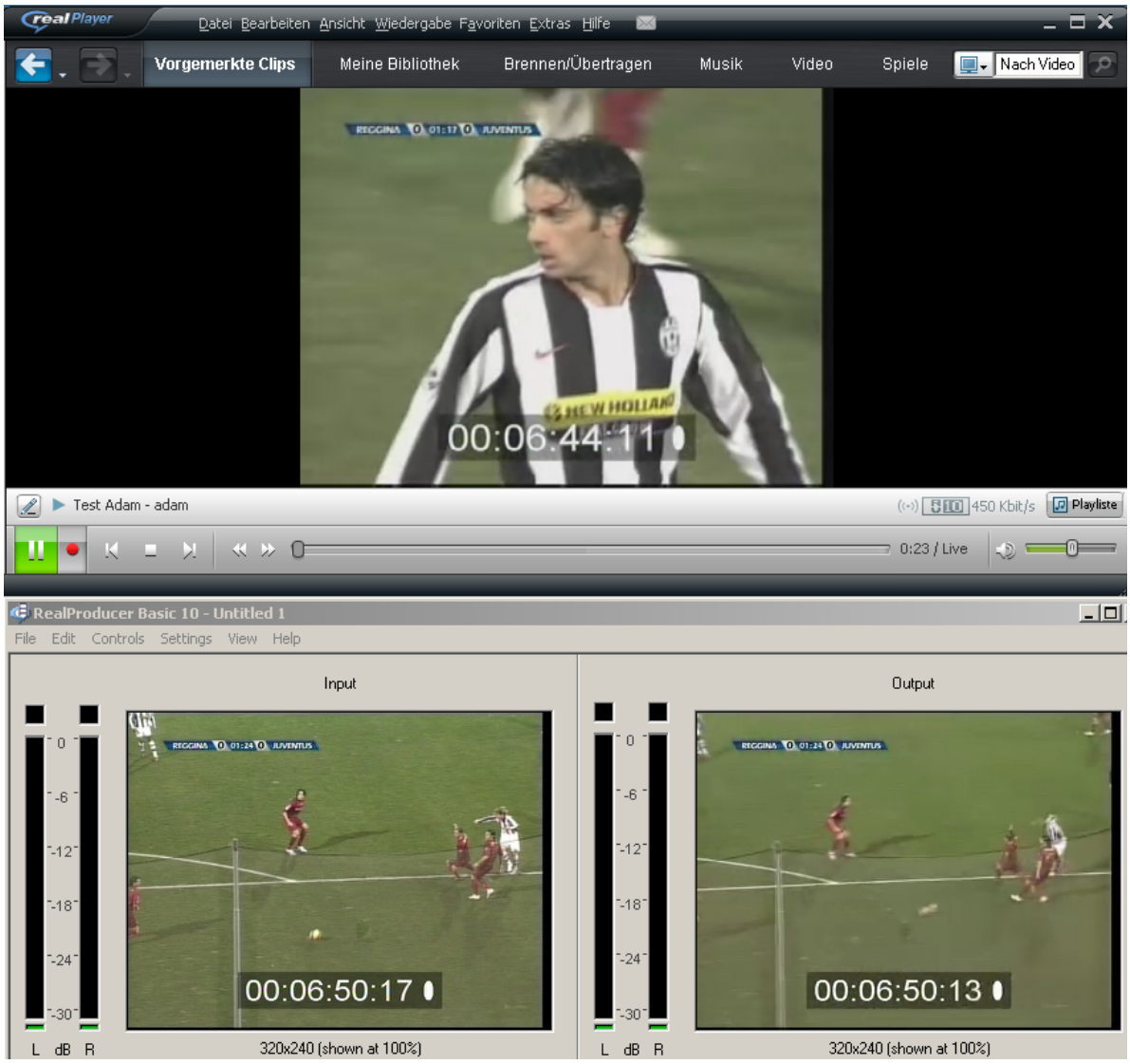

Abbildung 22: Verzögerungsmessung, Encoder und Player bei einer Real Media Übertragung über ein Content Delivery Network

8.5 Video LAN (VLC)

In diesem Versuch wurde lediglich die Latency bei Encoding und Decoding ohne Netzwerkübertragung gemessen. Es wurde jeweils eine Instanz des VLC (Version 0.8.6e) benutzt um den Stream zu generieren und ihn lokal wieder abzurufen.

Es wurde ein Bild mit einer Auflösung von 320x240 bei einer Datenrate von 512 kbps mit MPEG4 encodet. Audio wurde mit MPEGA bei einer Datenrate von 64 kbps Stereo encodet.

Als Cache-Wert beim Signal Input wurden 50ms festgelegt, der Puffer beim Stream Output wurde auf dem Standard-Wert von 300ms belassen.

Abbildung 23: Verzögerungsmessung, Encoder und Player bei einer VLC Übertragung

Bei diesem Versuch wurde eine Verzögerung von 1 Sekunde und 8 Frames (1,32 Sekunden) gemessen. Diese Verzögerung beinhaltet allerdings keine Netzwerkübertragung, ist also die reine beim Encoding und Decoding entstehende Latency.

8.6 Ergebnisse

Bei den Versuchen zeigte sich, dass die Adobe Flash Architektur, bei der ein Vergleich mit reinem Encoding und Decoding nicht möglich war, da ein Streaming Server benötigt wird und der Stream nicht direkt vom Encoder abgerufen werden kann selbst mit dieser zusätzlichen Übertragung weniger Latency erzeugte (0,52 Sekunden), als alle ohne Netzwerkübertragung getesteten Lösungen.
Bei VLC war im Test die reine Zeit von Encoder Output bis der Player das Bild anzeigte mit 1,32 Sekunden noch in einem Bereich, bei dem Low-Latency-Streaming unter günstigen Um-

ständen auch mit Netzwerkübertragung möglich wäre, wenn am Server entsprechende Einstellungen zur Reduktion der Latency möglich sind.

Die Real Architektur zeigte bei einer Übertragung über einen Server eine end-to-end Latency von 6,07 Sekunden. Dies genügt zwar nicht den Ansprüchen, die in dieser Arbeit für einen Low-Latency-Stream zugrunde gelegt werden, allerdings muss gesagt werden das Real Networks in den neuesten Versionen von Producer, Server und Player neue Möglichkeiten implementiert hat, die helfen sollen die Latency zu reduzieren. Da für die Tests nur ein Helix-Server der Version 10 zur Verfügung stand, konnte diese Technologie nicht berücksichtigt werden. Ein Faktor, der sich bei allen Übertragungen die im Rahmen dieser Arbeit durchgeführt wurden nicht beeinflussen ließ, waren die Einstellungen des Servers. Dies lag daran, dass alle Versuche über Content Management Netzwerke liefen und damit ein direkter Zugriff auf die Servereinstellungen nicht möglich war. Dies ist allerdings ein Szenario, das sich durchaus bei vielen Anwendungen von Low-Latency-Streaming ergeben kann, da ab einer gewissen Anzahl von Clients ein Content Delivery Network beinahe unumgänglich ist.

Mit der Streaming Lösung von Microsoft wurde ohne Übertragung über das Netzwerk bereits eine Zeit von 7,56 Sekunden gemessen, bis das Bild vom Player decodet war. Bei der Übertragung über ein Content Delivery Netzwerk stieg die Gesamt-Latency dann auf fast 24 Sekunden an. Obwohl Microsoft momentan an einem Projekt mit amerikanischen Universitäten arbeitet, bei dem Unterricht in Echtzeit praktisch ohne Verzögerung übertragen werden soll, sind die Erkenntnisse aus diesem Projekt offensichtlich noch nicht in den Bereich der angebotenen Streaming Lösungen eingeflossen.

Encoder	Decoder	Datenrate des Streams	Latency über Server	Reine Encoding-Latency
Flash Media Encoder	Flash Player 9.0.115	498 kbps	0,52 s	nicht verfügbar
Real Producer 10 Basic	Real Player 11	450 kbps	6,08 s	nicht verfügbar
Windows Media Encoder 9	Windows Media Player 10	498 kbps	23,96 s	7,56 s
VLC 0.8.6e	VLC 0.8.6e	576 kbps	nicht verfügbar	1,32 s

Abbildung 24: Testergebnisse Low-Latency-Streaming

9 Internet Protocol Television (IPTV)

9.1 Anbieter und Technologie

IPTV bezeichnet die Übertragung von Fernsehsignalen über das Internet Protokoll. Dabei wird im Gegensatz zur digitalen Satellitenübertragung, die mit MPEG 2 erfolgt, zumeist MPEG4/H.264 beim Encoding eingesetzt. Dadurch kann die Datenrate reduziert werden, ohne merkbare Einbußen bei der Bildqualität hinnehmen zu müssen. Dazu ist allerdings zu bemerken, dass in der Praxis wohl meist ein zuvor MPEG2 encodetes Signal, welches über Satellit empfangen wird durch das Encoding in MPEG4/H.264 bereits zum zweiten mal verlustbehaftet encodet wird und daher nicht dieselbe Qualität habe kann, wie ein über Satellit empfangenes Signal. [42]
IPTV wird in der Regel mit 3 - 4 MBit/s übertragen, was etwa der Hälfte eines MPEG 2 Signals in DVD-Qualität entspricht.

In Österreich bietet im Moment nur die Telekom Austria mit aonTV eine IPTV Lösung an und auch das nur in Ballungsräumen. Dabei wird ein MPEG2 Stream gesendet, der qualitativ nicht an einen MPEG4/H.264 Stream mit der gleichen Datenrate heranreicht. In Deutschland hingegen gibt es mittlerweile mit T-Online, Arcor und Alice drei Anbieter mit IPTV Angeboten. Diese setzen alle auf die Übertragung mit dem MPEG4/H.264 Standard.
Voraussetzung dafür sind DSL Anschlüsse mit einer Datenrate von mindestens 6 MBit/s, besser ist allerdings ein 16 MBit/s Anschluss, da IPTV oft in Kombination mit Internet und Telefon angeboten wird, welche ebenfalls Bandbreite benötigen. Wird eine zweite Set-Top-Box benötigt, um einen zweiten Fernseher anzuschließen oder um einen anderen Kanal aufzuzeichnen als jenen, den man gerade im Fernsehen sieht ist eine 16MBit Anbindung das Minimum. [43]
T-Online bietet in einigen Städten bereits VDSL (Very High Speed Digital Subscriber Line) mit einer Übertragungsrate von standardmäßig 25 MBit/s und gegen Aufzahlung 50 MBit/s

an. Mit solchen Anschlüssen ist auch IPTV in HD Qualität möglich, selbst wenn 2 Receiver zum Einsatz kommen. [44]

Im Gegensatz zu so genanntem Web-TV, das eigentlich normales Videostreaming (live oder On Demand) im Internet ist, wird IPTV zwar über die gleichen Leitungen und Router übertragen, dann beim Benutzer aber direkt in einer Set-Top-Box decodiert und auf dem Fernsehgerät wiedergegeben. Bei IPTV Lösungen werden in der Regel herkömmliche Fernsehkanäle übertragen, die auch über Satellit oder Kabel zu empfangen sind. [45]
Die Übertragung von IPTV erfolgt in abgeschlossenen Netzen, die vom Anbieter alleine betrieben werden. Daher ist es möglich Multicast zur Übertragung einzusetzen und Fernseh-Inhalte mit einer höheren Priorität zu behandeln als andere Daten im Netz. Die einzelnen Sender werden dabei jeweils als ein Stream übertragen. Wird das Programm gewechselt, fordert die Set-Top-Box vom Router einen anderen Stream an. Dadurch entstehen beim Umschalten zwischen den Programmen Wartezeiten von mehreren Sekunden. Bei T-Online wird dieses Problem gelöst, indem beim Wechsel von Programmen auf Unicast-Übertragung gewechselt wird und erst wenn der Multicast-Stream für den gewählten Sender bis zur Set-Top-Box geroutet wurde, wieder der Umstieg auf Multicast-Übertragung erfolgt. [46]

Timeshifting, bei dem das Fernsehprogramm an einer beliebigen Stelle angehalten werden kann oder Stellen übersprungen werden können, bietet von den genannten Providern momentan nur T-Online an. Durch das bei der Übertragung eingesetzte Multicast, bei dem ja jeder Empfänger den gleichen Stream bekommt ist es natürlich nicht möglich diesen Dienst serverseitig anzubieten. Erreicht wird die Timeshifting Funktion durch eine in der Set-Top-Box integrierte Festplatte, die jedes gesehene Programm aufzeichnet und dann bei Bedarf zeitversetzt wiedergibt. Das Überspringen von Programmteilen ist demnach nur möglich, wenn die Wiedergabe zuvor angehalten wurde oder erst nach Beginn der Sendung mit der Wiedergabe begonnen wurde und bereits Material auf die Festplatte aufgezeichnet werden konnte. [47]

Arcor hat eine serverseitige Lösung entwickelt und diese Timeshift Restart genannt. Dabei ist es möglich einen Film, der bereits begonnen hat, noch einmal vom Anfang an abzuspielen. Diese Möglichkeit wird allerdings momentan nur für einzelne Sender angeboten. Dabei wird

ein Verfahren eingesetzt, bei dem der Videodatenstrom am Server aufgezeichnet wird und bereits während die Aufzeichnung läuft abrufbar ist. Dadurch können auch Sendungen, die nicht von Beginn an angesehen wurden, neu gestartet werden. Dies ist bei der Timeshift Funktion mit in der Set-Top-Box integrierter Festplatte nicht möglich. Allerdings gibt es mit diesem Verfahren im Moment noch einige rechtliche Probleme. Mit Stand November 2007 hatten erst drei Fernsehstationen die Einwilligung zur Online-Aufzeichnung ihrer Sendungen gegeben und auch gegen den generellen Einsatz des Aufnahmeverfahrens wird wegen vermuteter Verletzung von Patentrechten gerichtlich vorgegangen. Dieser Dienst wird den Kunden außerdem nur gegen Aufzahlung (0,49 € pro Neustart) angeboten. [48, 49] Dies könnte allerdings auch eine Schutzmaßnahme sein, da diese Dienste natürlich wieder über Unicast übertragen werden müssen und so ein Problem entstehen könnte, wenn zu viele Kunden diesen Dienst gleichzeitig nutzen.

9.2 Middleware

Als Middleware wird bei IPTV jener Teil der Technologie bezeichnet, der für die Benutzeroberfläche und für die Kommunikation der Set-Top-Box mit dem Server bezüglich Bereitstellung von Streams, Zugangsberechtigungen und Abrechnung verantwortlich ist. Ein Teil davon ist das Content Management System (CMS) um den Electronic Program Guide (EPG) zu verwalten. Der Customer Relationship Manager (CRM) ist für die Verwaltung und Vergabe von Zugangsrechten verantwortlich. Ein weiterer Teil der Middleware, das Billing-System, ist für die korrekte Abrechnung verantwortlich und muss erfassen, welche Dienste (Paketangebot und Zusatzleistungen wie Video on Demand) der jeweilige Kunde in Anspruch genommen hat. In der Conditional Access (CA) und Digital Rights Managment (DRM) Verwaltung werden je nach den Parametern, die vom CRM gesetzt werden CA Freischaltungen oder DRM Lizenzen für den jeweiligen Nutzer vergeben. Diese können zeitlich begrenzt oder dauerhaft festgelegt werden. [50]

9.3 Interaktives IPTV

Ein großes Thema auf dem Weg zur Einführung von IPTV war immer die Interaktivität, die durch den vorhandenen Rückkanal umgesetzt werden kann. Im Moment bieten die IPTV Provider allerdings nur sehr wenige Services, die ihren Kunden Interaktivität ermöglichen. In erster Linie wird Video on Demand angeboten, wobei sich der Benutzer gegen eine zusätzliche Gebühr Filme aus einer „virtuellen Videothek“ bestellen kann. Weiters werden zum Beispiel bei aonTV einzelne Sendungen ohne zusätzliche Kosten on Demand bereitgestellt.

In Ländern, wo IPTV schon länger verfügbar ist und intensiver genutzt wird ist allerdings zu erkennen, dass wesentlich mehr Einsatzgebiete für interaktives Fernsehen vorhanden sind. In Hong Kong, wo Pacific Century Cyberworks (PCCW) schon seit 2003 IPTV anbietet, gibt es auch wesentlich mehr interaktive Inhalte, wie Shoppingkanäle, Videospiele und Essensbestellung. [51]
Besonders erfolgreich bietet PCCW Trailer zu aktuellen Kinofilmen, in Kombination mit der Möglichkeit Karten für den Film zu kaufen, an. Dieser Service wird so gut angenommen, dass in Hong Kong bereits mehr Kinokarten über IPTV als über das Internet verkauft werden. [52]

Auch in den USA hat sich IPTV bereits etabliert und auch hier werden zusätzliche interaktive Angebote eingesetzt, um sich von den klassischen TV-Anbietern abzuheben. AT&T bietet seinen Kunden die Möglichkeit Informationen über Wetter, Börsenkurse, Sport und Verkehrsinformationen in der so genannten „U-bar“ abzurufen. Was dieses Service von normalem Teletext abhebt ist, dass jene Kunden die auch den AT&T Yahoo! Breitbandanschluss verwenden über ihren Account bei AT&T die U-Bar ihres Fernsehers individuell gestalten können. So kann man beispielsweise das Wetter für einen spezifischen Ort anzeigen lassen, oder nur jene Aktienkurse in die U-bar aufnehmen, die im eigenen Portfolio enthalten sind. Auch die gelben Seiten werden als interaktiver Dienst über das IPTV-Netz angeboten und Videospiele gehören auch bei AT&T zum Angebot. [53]
Ein weiterer Service den AT&T seinen Kunden bietet ist die Möglichkeit den digitalen Videorekorder über das Handy zu programmieren. [54]

Besonders in Asien ist auch Karaoke On Demand eine viel genutzte interaktive Anwendung. Ebenfalls in Asien, speziell in Südkorea, wird auch interaktiven Anwendungen die der Bildung dienen sollen eine große Zukunft vorausgesagt. Es wird geschätzt, dass etwa 2,6% des Bruttoinlandsproduktes in Südkorea für Nachhilfe ausgegeben werden. Auch in China werden bereits interaktive Lerninhalte über IPTV angeboten und werden von den Konsumenten angenommen. [55]

Aber auch in Deutschland beginnt bereits der Trend zu interaktiven Anwendungen. Die Deutsche Telekom veranstaltet mit dem „Deutsche Telekom Interactive Award“, der im März 2008 gestartet wurde und bis Ende Jänner 2009 laufen soll, einen Wettbewerb, der zur Entwicklung von Interaktiven Anwendungen führen soll. Dabei werden Privatpersonen, aber auch kleine und mittelständische Unternehmen und Institutionen aufgefordert Anwendungen zu entwickeln, die besonders in den Bereichen Participation TV[21], Community/Personalisierung[22] und interaktive B2B/B2C Modelle[23] angesiedelt sind.
Als Beispiele werden von der Deutschen Telekom unter anderem Abstimmungen, Quiz, Multiplayer-Games und Messaging genannt. Auch hier eröffnet eine geringe Latency mehr Möglichkeiten, besonders bei der Interaktion mit Live-Sendungen oder auch bei Multiplayer Anwendungen. Die Gesamtdotation des „Deutsche Telekom Interactive Award“ beträgt eine Million Euro. [56]

[21] „Applikationen, die im direkten Zusammenhang mit Live-Sendungen stehen. Dabei hat der Nutzer die Möglichkeit, sich interaktiv in Echtzeit über den Rückkanal der TV-Plattform an Live-Sendungen zu beteiligen.“ [56]

[22] „Applikationen, die den Aufbau und die Pflege persönlicher und sozialer Netzwerke unterstützen, vereinfachen und/oder zu einem noch höheren Grad der Individualisierung des Fernseherlebnisses beitragen.“ [56]

[23] „Applikationen, die sich durch ihr zugrunde liegendes Geschäftsmodell auszeichnen und neue Umsatzpotenziale im interaktiven Fernsehen erschließen.“ [56]

9.4 Verbreitung von IPTV

Die Einführung von IPTV verläuft je nach Land sehr unterschiedlich erfolgreich. Generell kann gesagt werden, dass die Nachfrage nach IPTV in erster Linie in Ländern hoch ist, in denen zuvor der Prozentsatz der Haushalte die nicht über Kabel- oder Satelliten TV verfügen sehr hoch war. Aus diesem Grund war in Europa die Markteinführung von IPTV besonders in Frankreich sehr erfolgreich. In Frankreich beträgt der Anteil von Haushalten mit Kabelanschluss 17%, und auch Satellitenanschlüsse sind selten (18%), da es in Ballungsräumen nicht erlaubt ist Satellitenschüsseln an Häusern anzubringen. [51, 57]

9.5 Latency bei anderen Digitalen Fernsehformaten

Auch bei Antennenfernsehen geht der Trend ganz klar in Richtung digital. In Österreich und Deutschland ist die Umstellung auf DVB-T[24] nahezu abgeschlossen und analoges Antennenfernsehen kaum mehr zu empfangen.
Diese Umstellung bringt neben einem geringeren Bedarf an Bandbreite auch eine größere Latency mit sich, da die Signale auf der Senderseite encodet und auf der Empfängerseite decodet werden müssen. Auch die integrierten Fehlerkorrekturen benötigen Rechenzeit auf der Empfängerseite. [58, 59]
Insgesamt entsteht bei DVB-T dadurch eine Verzögerung von 4 bis 7 Sekunden bis das Bild auf dem Empfangsgerät zu sehen ist. [60]

Digitale Satellitenübertragungen mit DVB-S bzw. dem Nachfolgestandard DVB-S2 sind mit deutlich niedrigerer Verzögerung möglich. Je nach verwendetem Codec und Fehlerkorrekturmechanismen kann hier eine end-to-end Latency von unter einer Sekunde erreicht werden. [61]

[24] Digital Video Broadcast - Terrestrial

10 Schlussfolgerungen

Ziel dieser Arbeit ist die Untersuchung von Möglichkeiten, um Low-Latency-Streams verfügbar zu machen. Dies stellt Anforderungen an alle Komponenten, die für die Erzeugung, Übertragung und Darstellung des Streams verantwortlich sind.

Der Encoder muss in der Lage sein ein Videosignal in Echtzeit zu verarbeiten und dabei mit möglichst wenig Datenpufferung auszukommen. Damit kann der Anteil der Latency, der beim Encoding entsteht niedrig gehalten werden.
Legt man die Grenze für Low-Latency-Streaming bei 3 Sekunden Gesamtverzögerung, darf das Encoding nur einen Bruchteil davon beanspruchen, da damit gerechnet werden muss, dass auch die anderen Komponenten, die am Streaming beteiligt sind noch Latency verursachen.

Bei den Versuchen zeigte sich, dass Flash zum jetzigen Zeitpunkt die am besten geeignete Lösung für einen Low-Latency-Stream darstellt, wenn auch eine relativ teure. Sie bietet allerdings auch den Vorteil, dass die Verbreitung des Players bei nahezu 100% liegt und damit ein sehr großes Publikum angesprochen werden kann.
Mit VLC gibt es allerdings eine Alternative, wenn der Kostenfaktor ein wichtiges Argument ist. Real konnte bei den Tests nicht ganz mit der niedrigen Latency von Flash mithalten, wobei hier angemerkt werden muss, dass mit der kostenpflichtigen Real Producer Version bzw. mit dem Helix Server 11 möglicherweise bessere Werte erzielt werden könnten.
Windows Media konnte im Bereich Low Latency nicht wirklich überzeugen. Selbst wenn angenommen wird, dass mit zusätzlichen Servereinstellungen noch eine Verkürzung der end-to-end Latency möglich wäre, hat alleine das Encoding und Decoding mehr Zeit in Anspruch genommen, als für einen Low-Latency-Stream tolerierbar ist.

Es muss hier auch betont werden, dass fast alle Anbieter von Streaming-Lösungen entweder gerade neue Technologien integrieren oder ankündigen dies zu tun, um die Latency bei Streaming zu reduzieren. Daher wird es in der Zukunft interessant sein zu beobachten, wie sich die Angebote verändern und ob der zusätzliche Wettbewerb weitere Innovationen begünstigt.

Obwohl in vielen Publikationen darauf hingewiesen wird, dass bei geringer Pufferung eines Streams, wenn die Bandbreite nahezu ausgelastet wird, die Gefahr von Störungen bei der Übertragung besonders hoch ist, war kein Hinweis darauf zu finden wie groß der Überschuss an verfügbarer Bandbreite tatsächlich sein sollte, um einen störungsfreien Stream zu gewährleisten.

Die Lösung der Bandbreitenproblematik bei Videostreams erfolgt heute auf der Erzeugerseite ausschließlich über Content Delivery Netzwerke. Die Anzahl der verfügbaren Anbieter steigt in diesem Bereich und es gibt sowohl regionale als auch internationale Anbieter. Trotz der relativ hohen Kosten, die mit dem Einsatz von CDNs verbunden sind, ist es für eine Verbreitung von Streams in größerem Ausmaß unerlässlich deren Dienste in Anspruch zu nehmen. Die Preise für Inhalte, die über CDNs verbreitet werden, sind in den letzten Jahren gefallen und werden vermutlich durch den hohen Konkurrenzdruck weiter nach unten gehen. Der Wettbewerb ist allerdings für Kunden nur schwer auszunutzen, da nur wenige Anbieter öffentlich fixe Paketpreise anbieten. Damit ist der Preis der für die Auslieferung von Daten durch ein CDN zu bezahlen ist in erster Linie Verhandlungssache und in großem Ausmaß davon abhängig wie viel Datenvolumen übertragen werden soll.

Auf der Konsumentenseite zeichnet sich ein sehr starker Trend zu Breitband Internetanschlüssen ab. Die Anzahl der Haushalte, die über einen solchen verfügen wächst stetig an und auch die durchschnittlichen Datenraten werden immer höher.
Es ist daher anzunehmen, dass abgesehen von Streams in Fernsehqualität (im besonderen HD) die Downloadraten auf der Endkundenseite kein Problem für Streaming darstellen.
Relativieren muss man dies für Kabellose Internetverbindungen, da die Datenrate und Übertragungsqualität hier sehr stark von der Empfangsstärke des Signals abhängig sind.

Bei der Thematik der Streamauslieferung an den Endnutzer und der dafür genutzten Technologie hat sich gezeigt, dass Multicast im Internet in den nächsten Jahren kein großes Thema sein dürfte. Hier stellt sich das Problem, dass Investitionen in die Infrastruktur nötig wären und kaum Firmen bereit sind dieses Geld aufzubringen, speziell da nicht gewährleistet ist, dass

der Nutzen alleine dem Investor zufällt. Die Bereiche in denen Multicast im Internet genutzt wird, sind eher weniger geworden und liegen auf Universitäts- und Forschungsebene.

Ein anderes Bild zeigt sich hier in abgeschlossenen Netzwerken, da ISPs die Möglichkeit sehen durch IPTV einen Mehrwert ihrer Angebote zu schaffen, der Kunden dazu animieren soll Verträge für Breitbandanschlüsse mit sehr hohen Übertragungsraten abzuschließen. Zur Umsetzung dieses Konzepts müssen die ISPs Multicast in ihren Netzwerken umsetzen, da sonst bei der angestrebten Kundenzahl selbst leistungsfähige Server und Netzwerke mit der Datenlast vollkommen überfordert wären.
Dieser Trend entwickelt sich weltweit sehr unterschiedlich. Zwar werden die Länder in denen IPTV von gewissen Providern angeboten wird immer mehr, jedoch sind die Kundenzahlen höchst unterschiedlich. Generell wird dem IPTV-Markt ein großes Wachstum in den nächsten Jahren prophezeit, allerdings zeigt sich bereits jetzt, dass die Akzeptanz für IPTV in jenen Ländern besonders hoch ist, die keine weitreichende Abdeckung von Kabel- und Satellitenanschlüssen haben.
Die Provider versuchen ihr Angebot in den Ländern, die besser mit Kabel- und Satellitenanschlüssen versorgt sind, von denen der herkömmlichen Anbieter abzuheben. Dies geschieht einerseits durch im Paket enthaltene Inhalte, die sonst nur im Pay-TV zu sehen sind und andererseits durch interaktive Angebote. Ein wichtiger Faktor ist hier die Timeshift Funktion, mit der es möglich ist Filme anzuhalten und zu einem späteren Zeitpunkt an der gleichen Stelle fortzusetzen. Es werden allerdings auch immer mehr interaktive Inhalte angeboten. In der Zukunft wird es interessant sein, welche Art von interaktiven Inhalten sich durchsetzt. Dabei ist schon nach der heutigen Entwicklung abzuschätzen, dass auch dies so wie die Verbreitung von IPTV in den verschiedenen Ländern oder Kulturkreisen sehr stark variieren könnte. Denn auch jetzt setzen sich in verschiedenen Kulturkreisen verschiedene Anwendungen durch.

11 Abbildungsverzeichnis

Abbildung 1: Beispiel einer Huffman Code-Tabelle..... 16
Quelle: Schmidt, Ulrich: Professionelle Videotechnik. - Berlin [u.a.] : Springer, 2000 S. 116-117

Abbildung 2: Aufbau und Übertragungsreihenfolge einer Group of Pictures (GOP) 17
Quelle: Schmidt, Ulrich: Professionelle Videotechnik. - Berlin [u.a.] : Springer, 2000 S. 134

Abbildung 3: Durchschnittliche Datenraten über verschiedene Internetanschlüsse (2004) ... 21
Quelle: http://www.uni-koeln.de/rrzk/multimedia/dokumentation/streaming/streaming.html, 02.04.2008

Abbildung 4: Latency durch Pufferung bei Encoder, Server und Player 22
Quelle: http://www.microsoft.com/windows/windowsmedia/howto/articles/Broadcast Delay.aspx, 17.03.2008

Abbildung 5: Übertragung im Unicast Verfahren 26
Quelle: www.surfnet.nl, 08.02.2008

Abbildung 6: Übertragung im Multicast Verfahren 27
Quelle: www.surfnet.nl, 08.02.2008

Abbildung 7: Verbreitung von Breitband Internet in der Europäischen Union 29
Quelle: Eurostat, 26.02.2008

Abbildung 8: Multicast Source Tree 31
Quelle: http://www.ciscopress.com/content/images/chap0x_1587051225/elementLinks/ th0xfig01, 26.02.2008

Abbildung 9: Multicast Shared Tree 33
Quelle: http://www.ciscopress.com/content/images/chap0x_1587051225/elementLinks/ th0xfig02.gif, 26.02.2008

Abbildung 10: Auflistung einiger bekannter Video MIME-Typen 38
Quellen: http://de.selfhtml.org/diverses/mimetypen.htm, RealNetworks, Microsoft, Adobe, 18.03.2008

Abbildung 11: Windows Media Server, Prozessorauslastung bei Live-..... 46
Quelle: Microsoft, 18.03.2008

Abbildung 12: Windows Media Server, Prozessorauslastung bei Live-..... 46
Quelle: Microsoft, 18.03.2008

Abbildung 13: Windows Media Server, Vergleich Prozessorauslastung und Arbeitsspeicherbedarf bei Stream Distribution mit 100 kbps und 500 kbps ... 47
Quelle: Microsoft, 18.03.2008

Abbildung 14: Preisliste für Helix Server Produkte von RealNetworks 48
Quelle: www.videomediasolutions.com, 06.03.2008

Abbildung 15: Latency-Modes bei Real Streaming 56
Quelle: Real Producer 11, User's Guide

Abbildung 16: Verzögerungsmessung, Einstellungen des Flash Media Encoders..... 62

Abbildung 17: Verzögerungsmessung, Encoder und Player bei einer Flash-Übertragung 63

Abbildung 18: Verzögerungsmessung: Einstellungen Windows Media Encoder und Media Player 64

Abbildung 19: Verzögerungsmessung, Encoder und Player bei einer Windows Media Übertragung über ein Content Delivery Network 65

Abbildung 20: Verzögerungsmessung, Encoder und Player bei einer Windows Media Übertragung (ohne Server) 66

Abbildung 21: Verzögerungsmessung: Einstellungen Real Media Producer 67

Abbildung 22: Verzögerungsmessung, Encoder und Player bei einer Real Media Übertragung über ein Content Delivery Network 68

Abbildung 23: Verzögerungsmessung, Encoder und Player bei einer VLC Übertragung 69

Abbildung 24: Testergebnisse Low-Latency-Streaming 70

12 Quellen

[1] http://www.techcrunch.com/2006/10/09/google-has-acquired-youtube, 21.03.2008

[2] Austerberry, David. The technology of video and audio streaming. Elsevier, Amsterdam, Netherlands, 2006. ISBN 978-0-240-80580-1.

[3] Schmidt, Ulrich. Professionelle Videotechnik. Springer, Berlin [u.a.], 2000. ISBN 3-540-66854-3

[4] Thobias Künkel. Streaming Media – Technologien, Standards, Anwendungen. Addison-Wesely Verlag, München, Germany, 2001. ISBN 3-8273-1798-3.

[5] www.adobe.com/devnet/flashcom/articles/flvideo_bandwidth.html, 07.02.2008

[6] www.realnetworks.com/products/producer/features.html, 07.02.2008

[7] www.microsoft.com/presspass/press/1999/Apr99/WMT4Betapr.mspx, 07.02.2008

[8] http://www.cs.mu.oz.au/~apathan/CDNs.html, 07.03.2008

[9] http://www.techweb.com/encyclopedia/defineterm.jhtml?term=unicast, 08.02.2008

[10] http://www.ipoque.com/media/internet_studies/internet_study_2007, 26.02.2008

[11] http://www.ciscopress.com/articles/article.asp?p=32100, 26.02.2008

[12] Inc Cisco Systems. Internetworking Technologies Handbook. Macmillan Technical Publishing, 2004. ISBN 978-1587050015.

[13] http://www.dataconnection.com/multicast/pimprotocol.htm, 08.02.2008

[14] http://www.ietf.org/rfc/rfc768.txt, 18.03.2008

[15] http://www.ietf.org/rfc/rfc2616.txt, 18.03.2008

[16] http://de.selfhtml.org/diverses/mimetypen.htm, 18.03.2008

[17] http://tools.ietf.org/html/rfc3550#page-4, 15.02.2008

[18] www.ietf.org/rfc/rfc2326.txt, 12.02.2008

[19] White, Paul P. RSVP and Integrated Services in the Internet: A Tutorial, IEEE Communications Magazine, May 1997, S. 100-106

[20] Kevin Towes. Video content protection measures enabled by Adobe Flash Media Server http://www.adobe.com/devnet/flashmediaserver/articles/protecting_video_fms.pdf, 11.02.2008

[21] http://www.microsoft.com/windows/windowsmedia/licensing/netprokit.aspx, 15.02.2008

[22] http://www.adobe.com/de/products/flashmediaserver/pdfs/fms_wp_ue.pdf, 28.02.2008

[23] http://www.adobe.com/products/flashmediastreaming/, 06.03.2008

[24] http://www.adobe.com/products/flashmediainteractive/, 06.03.2008

[25] http://msdn2.microsoft.com/en-us/library/ms867206.aspx, 18.03.2008

[26] http://docs.real.com/docs/rn/datasheet/Helix_Server_25-100_S301.pdf, 06.03.2008

[27] http://docs.real.com/docs/rn/datasheet/Helix_Server_Unlimited_S316.pdf, 06.03.2008

[28] http://www.videomediasolutions.com/real_support.html, 06.03.2008

[29] http://store.apple.com/AppleStore/WebObjects/BizCustom.woa, 07.03.2008

[30] http://developer.apple.com/opensource/server/streaming/index.html, 07.03.2008

[31] http://images.apple.com/quicktime/pdf/QT_Streaming_TB.pdf, 06.03.2008

[32] http://osflash.org/red5/roadmap, 20.03.2008

[33] http://www.kaourantin.net/2007/08/what-just-happened-to-video-on-web_20.html, 10.03.2008

[34] http://www.adobe.com/products/player_census/flashplayer/, 13.03.2008

[35] http://www.microsoft.com/windows/windowsmedia/howto/articles/BroadcastDelay.aspx, 02.03.2008

[36] http://www.videolan.org/streaming-features.html, 20.03.2008

[37] http://www.videolan.org/doc/play-howto/en/ch03.html, 20.03.2008

[38] http://www.apple.com/quicktime/broadcaster/, 19.03.2008

[39] http://www.apple.com/quicktime/player/specs.html, 19.03.2008

[40] http://www.apple.com/quicktime/pro/, 19.03.2008

[41] http://images.apple.com/quicktime/pdf/QTSS_Modules.pdf, 19.03.2008

[42] http://www.iptv-anbieter.info/testberichte/alice-hometv/alice-iptv-test.html, 26.02.2008

[43] IPTV – das ultimative Entertainment-Erlebnis, FKT 10/2006, S. 622-624

[44] http://entertain.eki.t-home.de/, 27.02.2008

[45] IPTV: TV Programme in voller Qualität, FKT 1-2/2008, S. 66

[46] http://www.zdnet.de/mobile/tkomm/0,39023192,39153383-5,00.htm, 27.02.2008

[47] http://www.dslweb.de/t-home-entertain.php, 10.03.2008

[48] http://www.digitalfernsehen.de/news/news_224028.html, 28.02.2008

[49] http://www.artec.de/de/aktuelles/news/news-singleview.html?tx_ttnews %5Btt_news%5D=77, 28.02.2008

[50] IPTV – Fernsehen über Internet-Protokoll, FKT 1-2/2007, S. 58-64

[51] Dieter, Stefan. Schrameyer, Dr. Dirk. IPTV – Über Internet anders Fernsehen?! – Mehrwert oder nur mehr Kosten?. Landesanstalt für Medien Nordrhein-Westfalen, Düsseldorf, 2008. ISBN 978-3-940929-00-6

[52] http://telcotv-view.blogspot.com/2007/11/ictv-providing-interactive-tv-to-pccw.html, 04.03.2008

[53] http://www.tvover.net/2007/12/18/ATT+Uverse+IPTV+Adds+New+Interactive+ Features+In+Wisconsin.aspx, 20.03.2008

[54] http://www.telecommagazine.com/newsglobe/article.asp?HH_ID=AR_3643, 20.03.2008

[55] http://www.reuters.com/article/sphereNews/idUSSEO3777620080226?sp=true& view=sphere, 20.03.2008

[56] http://www.interactive-tv-award.de/dev/faq, 01.04.2008

[57] IPTV – Fernsehen nicht für jedermann?!, FKT 10/2007, S. 523-530

[58] http://www.dvb-t.at/wann-wie-umstellen/zeitplan.html, 31.03.2008

[59] http://www.zdnet.de/news/tech/0,39023148,39143377,00.htm, 31.03.2008

[60] http://www.movie-college.de/filmschule/medien/dvb-t.htm, 31.03.2008

[61] http://broadcastengineering.com/hdtv/hd-sng-benefit-h264-dvb-s2-fijitsu/, 31.03.2008