

**Hauptseminararbeit**  
**Sommersemester 2004**

Thema:

**Live-Streaming-Lösungen**

*Recherche und Analyse existierender Live-Streaming-Lösungen  
(Infrastruktur, organisatorischer Aufbau) aus technischer Sicht*

Bearbeiter: Andy Wohlfahrt

Betreuer: Dipl.-Inf. Thorsten Strufe

|  |           |
|--|-----------|
| <b>1 EINLEITUNG .....</b>                                  | <b>3</b>  |
| <b>2 PRINZIPIELLER AUFBAU EINER STREAMING LÖSUNG.....</b>  | <b>4</b>  |
| 2.1 DIE STREAMERZEUGUNG.....                               | 4         |
| 2.1.1 <i>Das Capturen der Mediadaten</i> .....             | 4         |
| 2.1.2 <i>Das Encoden der Mediadaten</i> .....              | 5         |
| 2.1.3 <i>Das Bereitstellen des erzeugten Streams</i> ..... | 6         |
| 2.2 DIE STREAMVERTEILUNG.....                              | 6         |
| 2.2.1 <i>Übertragungsverfahren</i> .....                   | 7         |
| 2.2.2 <i>Techniken der Streamverteilung</i> .....          | 8         |
| 2.3 DIE STREAMVERWENDUNG .....                             | 11        |
| <b>3 DOMINIERENDE STREAMING LÖSUNGEN .....</b>             | <b>12</b> |
| 3.1 WINDOWS MEDIA SERIES.....                              | 12        |
| 3.1.1 <i>Windows Media Encoder</i> .....                   | 12        |
| 3.1.2 <i>Windows Media Server</i> .....                    | 13        |
| 3.1.3 <i>Windows Media Player</i> .....                    | 15        |
| 3.2 REALNETWORKS HELIX .....                               | 16        |
| 3.2.1 <i>Helix Producer</i> .....                          | 16        |
| 3.2.2 <i>Helix Server</i> .....                            | 17        |
| 3.2.3 <i>Real Player</i> .....                             | 19        |
| 3.3 APPLE QUICKTIME .....                                  | 20        |
| 3.3.1 <i>QuickTime Broadcaster</i> .....                   | 20        |
| 3.3.2 <i>QuickTime Streaming Server</i> .....              | 21        |
| 3.3.3 <i>QuickTime Player</i> .....                        | 22        |
| <b>4 ZUSAMMENFASSUNG .....</b>                             | <b>24</b> |
| <b>5 ABBILDUNGSVERZEICHNIS .....</b>                       | <b>25</b> |
| <b>6 LITERATUR- UND QUELLENANGABEN.....</b>                | <b>25</b> |

# 1 Einleitung

Seit der Einführung kostengünstiger Breitbandanschlüsse auch für Privatanwender, ist ein Wandel des Internets weg von der reinen Informationsbereitstellung, in Text und Bild, hin zur Bereitstellung von multimedialen Diensten spürbar. Möglich machen dies Streaming Technologien, welche es erlauben, dass der Endanwender nicht erst die gesamte Datei auf seinen eigenen Computer herunterladen muss, um ein multimediales Angebot zu nutzen, sondern dass der multimediale Inhalt durch einen kontinuierlichen Datenfluss schon beim Download genutzt werden kann.

Doch was genau ist eigentlich Streaming, laut [www.wikipedia.org](http://www.wikipedia.org) versteht man unter Streaming die kontinuierliche Übertragung von Daten, z.B. Audio- und Videosequenzen, über ein Netzwerk, z.B. das Internet. Ziel dabei ist es die Daten möglichst ohne Zeitverlust übertragen und darstellen zu können, weshalb z.B. Audio- und Videosequenzen sich schon während des Ladevorganges wiedergeben lassen.

Beim Live Streaming werden, im Gegensatz zu On Demand Angeboten, die aufgenommenen Inhalte sehr zeitnah zur Aufnahme ins Internet eingespeist. Wenn ein Anwender das Live Streaming Angebot nutzen will, so kann er sich zu jeder Zeit am Server mit seinem Client anmelden und den bereit gestellten Stream nutzen. So können mit einer recht geringen zeitlichen Verzögerung, z.B. Fernsehprogramme, auch auf einem PC mittels Live Stream abgespielt werden.

Im Folgenden wird aufgezeigt, mit welchen Techniken bestehende Live Streaming Lösungen arbeiten und welche Lösungen der Markt anbietet, um Live Content im Netz zur Verfügung zu stellen.

## 2 Prinzipieller Aufbau einer Streaming Lösung

Eine (Live-)Streaming Lösung lässt sich prinzipiell in 3 Teile zerlegen, in die Streamerzeugung, die Streamverteilung und die Streamverwendung beim Endnutzer, wie die folgende Abbildung zeigt.

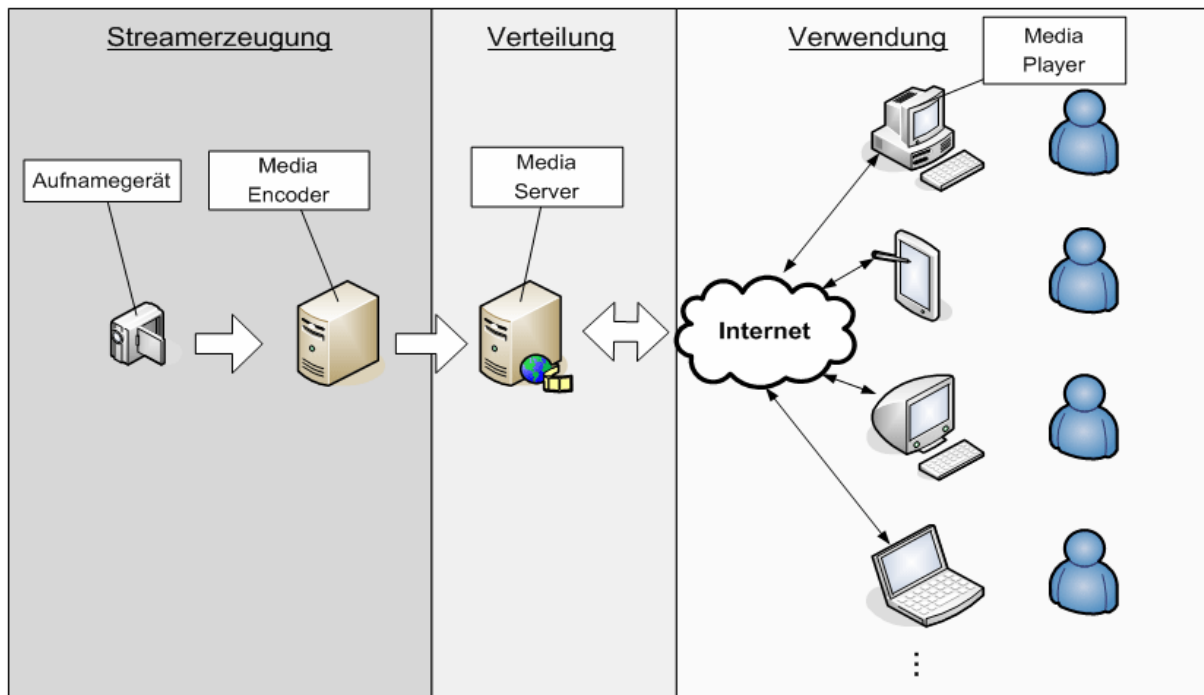


Abb. 1: Prinzipieller Aufbau einer Streaming Lösung

### 2.1 Die Streamerzeugung

Die Streamerzeugung gliedert sich in zwei Hauptbestandteile, einerseits in die Aufnahme der Mediadaten, dem Capturing, und zum anderen der Formatierung und Komprimierung der Inhalte, dem Encoding.

#### 2.1.1 Das Capturen der Mediadaten

Um Mediadaten zu capturen genügt schon einfachste Hardware, wie z.B. eine Webcam oder ein einfaches Mikrofon, welche über eine geeignete Schnittstelle oder spezielle Capturekarten mit dem Media Encoder verbunden ist. Jedoch ist die Qualität der Ausgangsdaten im Wesentlichen mitentscheidend für die Qualität des im Netz zur Verfügung gestellten Streams. So können Fehler, wie z.B. ein Rauschen im Video, später nur sehr aufwändig oder gar nicht wieder behoben werden. Daher setzt die Produktion eines Streams schon am Anfang eine gründliche Planung voraus. So mag im privaten Bereich eine einfache Webcam noch ausreichend sein, aber um ein

kommerzielles Angebot im Netz zur Verfügung zu stellen, ist man auf professionelle Kamera-/Tontechnik angewiesen.

### 2.1.2 Das Encoden der Mediadaten

Um die nun erzeugten Mediadaten im Netz als Stream zur Verfügung zu stellen, ist ein PC oder Macintosh PC nötig, welcher als Media Encoder fungiert. Dieser, meist mit spezieller Crabbererhardware<sup>1</sup> ausgestattet, erhält nun die aufgenommenen Rohdaten, um sie zu formatieren und zu komprimieren. Aus ihnen wird danach ein permanenter komprimierter Stream zu produziert.

Die Bandbreiten der Nutzer des Streamingangebotes variieren stark, von einer Modemverbindung bis hin zu LANs, daher ist es notwendig schon beim kodieren der Mediadaten auf diese Gegebenheit einzugehen. Die nachfolgende Tabelle veranschaulicht die verschiedenen Formate für die jeweiligen Zielgruppen.

| Zielgruppe   | Auflösung | Framerate | Bitrate Video | Samplerate | Bitrate Audio |
|--------------|-----------|-----------|---------------|------------|---------------|
| LAN/Intranet | 576x432   | 25 fps    | 1,5 MBit      | 44,1 kHz   | 128 kBit      |
| DSL          | 352x240   | 18 fps    | 450 kBit      | 44,1 kHz   | 96 kBit       |
| ISDN/Modem   | 196x132   | 12 fps    | 35 kBit       | 22 kHz     | 48 kBit       |

**Tabelle 1: Formate**

Um optimale Ergebnisse beim Kodiervorgang zu erreichen, stehen mehrere Modi und Varianten zur Verfügung. Als Kodierungsmodi stehen das ConstantBitRate (CBR) Enncoding und das VariableBitRate (VBR) Encoding zur Verfügung. Beim ConstantBitRate Encoding bleibt die Bitrate während des Streams, bis auf kleine Variationen, konstant. Der Nachteil von CBR-Encoding besteht darin, dass die Ausgangsdaten nicht homogen sind, dies bedeutet, dass einige Inhalte schwerer zu komprimieren sind als andere. Das führt zu Variationen in der Qualität des Ausgabestreams. Beim VBR-Encoding kann auf diese Gegebenheit eingegangen werden, in dem auf die Komplexität der Daten eingegangen wird. Den weniger komplexen Daten werden weniger Bits zugewiesen, welche dann bei komplexeren Teilen verwendet werden können.

Um beim CBR-Encoding optimale Ergebnisse zu erreichen besteht die Möglichkeit, unter Verwendung des two-pass Verfahrens, die Media Daten optimiert zu kodieren.

<sup>1</sup> Crabberhardware: spezielle Schnittstellenkarten, zum digitalisieren analoger Video-/Audiodaten

Im Gegensatz zu dem single-pass Verfahren, welches die Media Daten in einem Durchlauf bearbeitet, beruht das two-pass Verfahren auf zwei Durchläufen. Wobei der erste Durchlauf zu einer Analyse der Daten genutzt wird und im zweiten, auf Grund der Analyse, optimiert zu kodieren. Da aber beim two-pass Verfahren die zu kodierenden Daten vorliegen müssen, ist es nur für On Demand Streaming geeignet.

### **2.1.3 Das Bereitstellen des erzeugten Streams**

Der Media Encoder stellt als Ergebnis des Encodingvorgangs einen einzelnen formatierten und komprimierten Stream dem Media Server zur Verfügung.

Da die Bandbreiten der Endanwender aber stark variieren, ist es für die meisten Anbieter von Streaming Angeboten obligatorisch verschieden formatierte Streams, je nach Bandbreite, anzubieten. Um Redundanzen zu vermeiden und nicht für jeden Stream einen speziellen Media Encoder bereit zu stellen, wurden Verfahren entwickelt und in die Encoding Software implementiert, welche es erlauben, dass ein Media Encoder, als Resultat des Encoding Prozesses, mehrere unterschiedlich formatierte Streams dem Media Server bereit stellt. Diese Verfahren werden, je nach Hersteller, als Surestream oder Multibit Rate bezeichnet.

Das von Microsoft entwickelte Multibit Rate Verfahren und das Surestream Verfahren von RealNetworks erlauben das schreiben von mehreren verschiedenen Bitraten in ein File. Der Media Server handelt dann mit dem Client des Endanwenders die Qualitätsstufe des Streams aus. So kann man in einem Encoding Vorgang optimierte Qualitäten für Modem- bis hin zu LAN-Verbindungen anbieten.

## **2.2 Die Streamverteilung**

Die Aufgabe der Verteilung, des vom Media Encoder erzeugten Streams, übernimmt der Media Server. Dieser erhält vom Media Encoder den fertig komprimierten Stream, welchen er dann an die Clients verteilt.

Die Anforderungen an die Hardware des Streaming Servers sind für eine kleine Anzahl von Clients eher moderat, jedoch steigt sie proportional zu der Anzahl und der Bitrate der angeforderten Streams. So gibt z.B. Microsoft als Mindestvoraussetzungen für die Hardware eines Windows Media Servers einen PC mit 233 MHz an, mit welchem bis zu 1000 Clients mit einer 28.8 Modemverbindung simultan versorgt werden können. Jedoch kommt es zu steigenden Anforderungen

an die Hardware, sobald die Anzahl der Verbindungen steigt bzw. höhere Bitraten angeboten werden.

## 2.2.1 Übertragungsverfahren

Für die Übertragung von Streams über das Internet existieren grundsätzlich 2 Übertragungsmöglichkeiten, das Unicast- und das Multicastverfahren, welche im folgenden Abschnitt näher erläutert werden.

### Unicast

Beim Unicast initiiert jeder Client eine eins zu eins Verbindung, was zur Folge hat, dass die Anzahl der zu übertragenden Streams linear mit der Anzahl der verbundenen Clients steigt. Dies führt zu sehr hohen Anforderungen an die Bandbreite sowie an die Hardware des Streamingserver. Als Vorteil des Unicast-Verfahrens ist jedoch zu sehen, dass es keine Anforderungen an das Netzwerk stellt und der Server auch individuell auf die einzelnen Clients reagieren kann, was Voraussetzung für das On Demand Streaming ist.

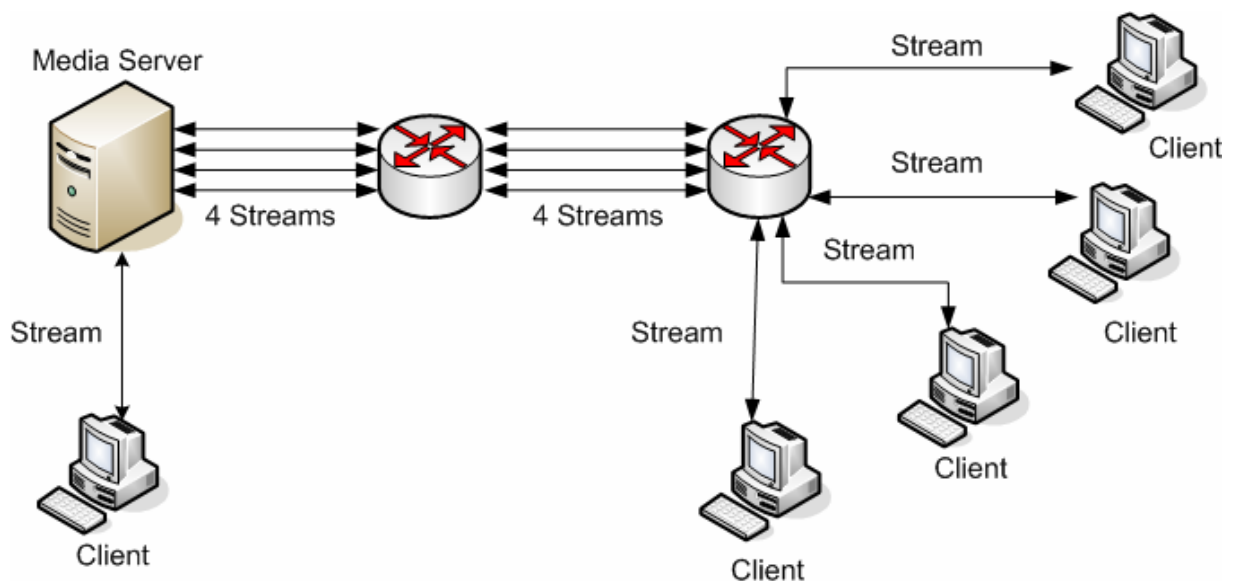


Abb. 2: Unicast

### Multicast

Beim Multicast-Verfahren wird, im Gegensatz zu dem Unicast-Verfahren, nicht eine eins zu eins Verbindung vom Client zum Server aufgebaut, sondern der Server sendet die Daten an eine Empfängergruppe. Dies jedoch setzt eine "Intelligenz" der Router voraus, daher unterstützen auch noch nicht alle Netze Multicastfunktionalität.

Der große Vorteil von Multicast ist jedoch, dass die Anzahl der zu versorgenden Clients sich nicht direkt auf die Bandbreite auswirkt.

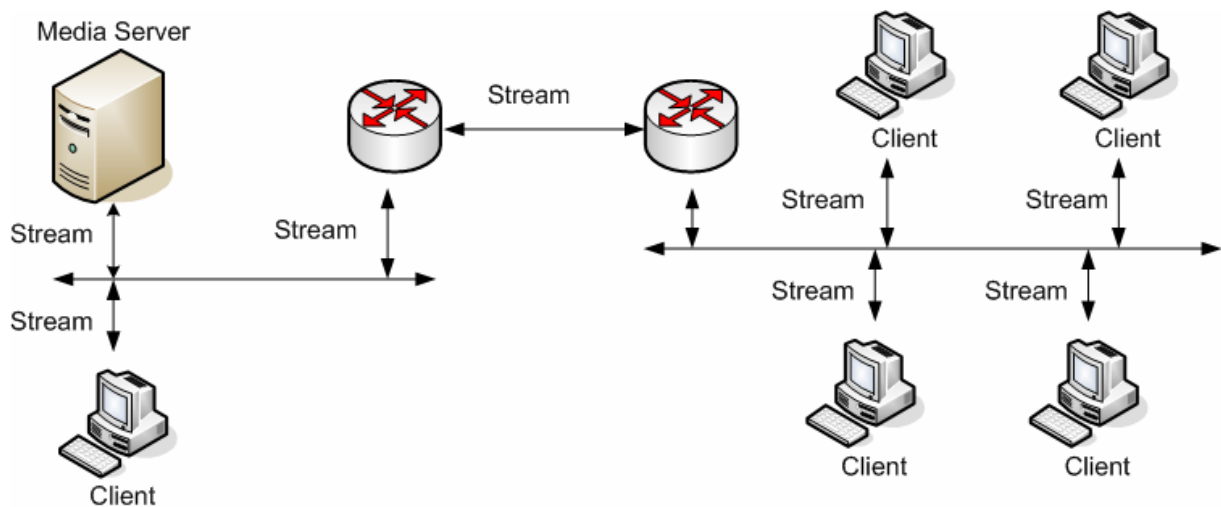


Abb. 3: Multicast

## 2.2.2 Techniken der Streamverteilung

### Centralized Content Serving

Der erste Ansatz Multimedia Streams im Netz bereit zu stellen, beruhte auf dem Centralized Content Serving. Dieses Prinzip beruht auf einem zentralen Server, bzw. Servercluster, welcher die Verteilung der Streams regelt. Dies bedeutet, dass jeder Client, egal wie weit er vom Server entfernt ist, eine Verbindung zu diesem Server aufbaut und das Angebot abrufen.

Durch die großen Distanzen, welche beim Centralized Content Serving auftreten können, kann es jedoch zu Beeinträchtigungen im QoS<sup>2</sup> kommen. Dies macht sich vorrangig in hohen Latenzzeiten und Paketloss bemerkbar, und wirkt sich negativ auf das Angebot aus.

Ein weiterer gravierender Nachteil des Centralized Content Serving ist, daß der zentrale Server über eine sehr hohe Bandbreite verfügen muss, damit der Dienst für möglichst viele Nutzer angeboten werden kann, speziell bei der Verwendung von Unicast als Übertragungsverfahren. Auch liegt es auf der Hand, dass ein einzelner zentraler Server eine schlechte Lösung im Bezug auf Ausfallsicherheit und Überlastung ist.

<sup>2</sup> QoS: Quality of Service



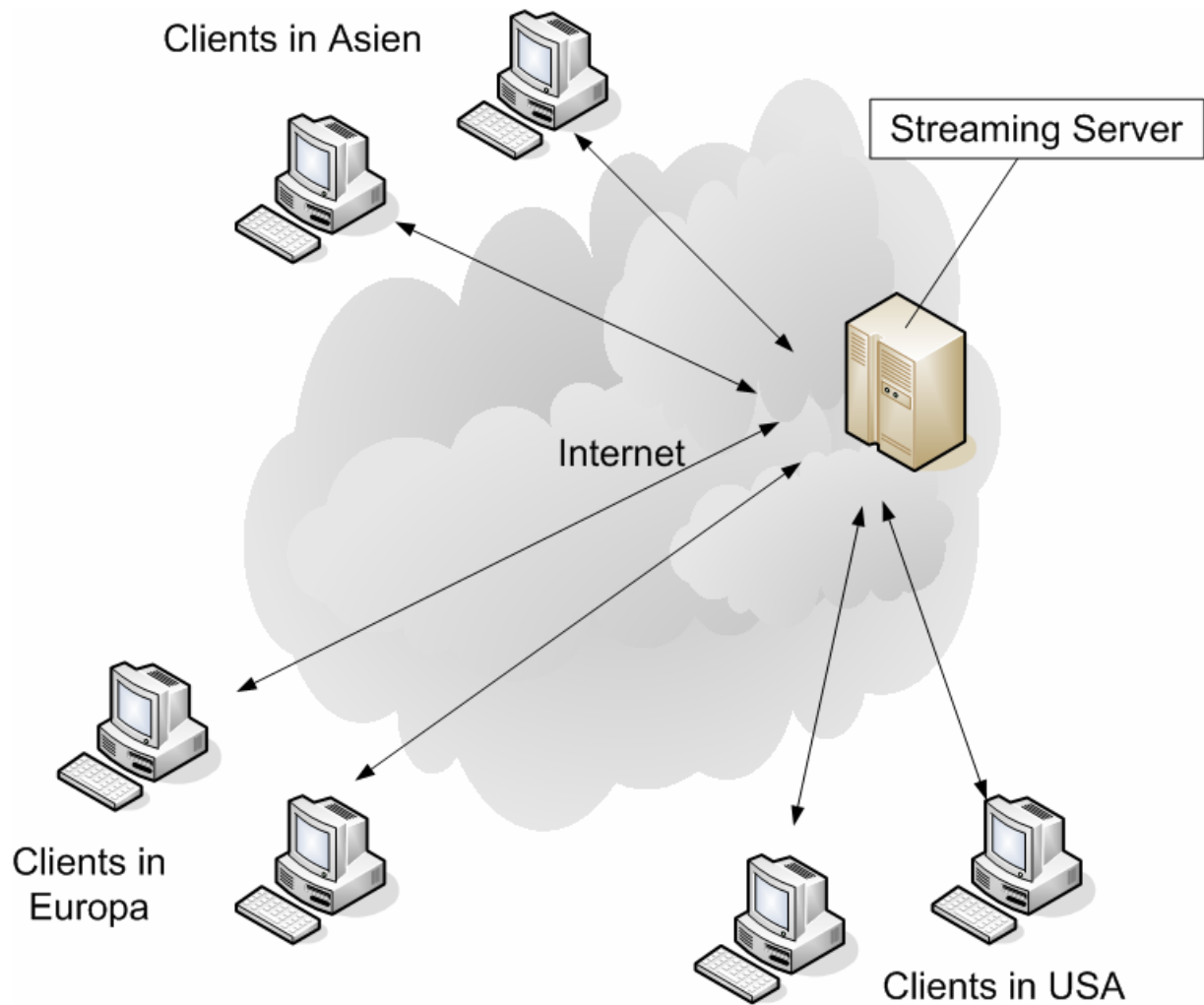


Abb. 4: Centralized Content Serving

## Load Balancing

Zwei entscheidende Qualitätsmerkmale für ein Streaming Angebot sind die Verfügbarkeit und die Kapazität. Um eine Überlastung oder einen Ausfall eines einzelnen Streaming Servers zu kompensieren, ist Redundanz ein wichtiges Instrument. So werden die Streams nicht von einem einzelnen Streaming Server dem Netz zur Verfügung gestellt, sondern über einen Load Balancing Server verteilt. Dieser stellt virtuell den Stream für das Netz bereit, verteilt aber nur die Anfrage auf verschiedene Streamingserver. Dabei kann es sich um lokale Servercluster oder, wie im Abschnitt Distributed Content Serving beschriebene, Edge Server handeln. So umgeht der Einsatz von Load Balancing z.B. die Bandbreitenbeschränkung bei Unicast

Einerseits erreicht nun durch Load Balancing eine Aufteilung der Serverlast auf mehrere Server, andererseits ergibt sich durch das Spiegeln der Daten und der Aufteilung auf mehrere Server auch eine erhöhte Ausfallsicherheit. Sollte es dazu

kommen, dass ein Streaming Server ausfällt, so übernimmt ein anderer Streaming Server den ausgefallenen Stream.

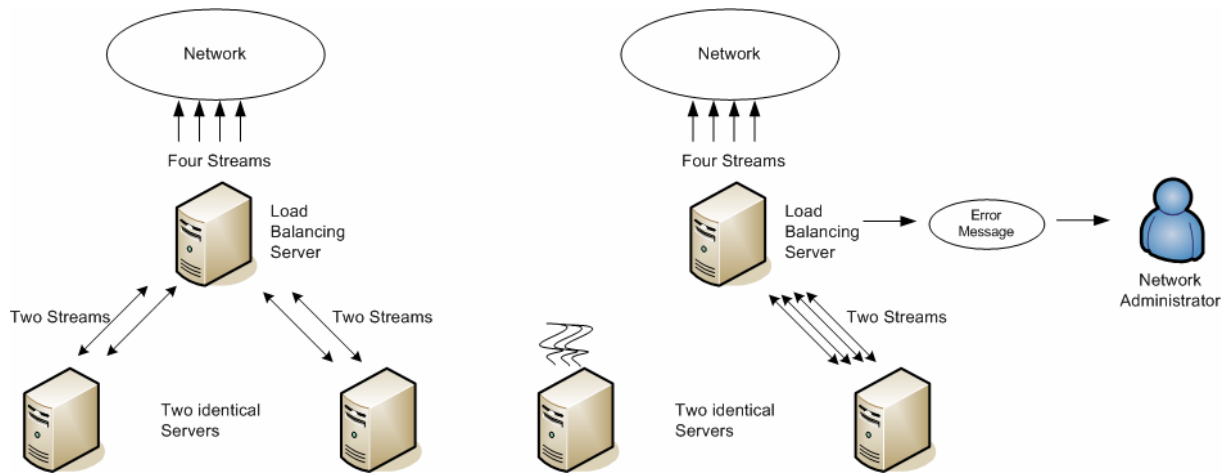


Abb. 5: Load Balancing

Üblicherweise geschieht das Verteilen der Anfragen, welche den Load Balancing Server erreichen, über DNS Rotation, auch round robin DNS genannt. Diese Technik erlaubt es, verschiedene IP Adressen auf einen einzelnen Hostnamen im DNS Eintrag abzubilden. Wenn nun der DNS Server eine Anfrage erhält, so verteilt er diese rotationsartig an die verschiedenen IP Adressen der Streaming Server. Dieses Verfahren wird auch direkt, laut RFC 2326, vom RTSP unterstützt.

### Distributed Content Serving (Edge Server)

Um den Nachteilen zu begegnen, die mit der Verwendung eines zentralen Servers in Verbindung stehen, wird zunehmend das Distributed Content Serving eingesetzt. Die Grundidee dieses Verfahrens ist es, den Content nah beim Endnutzer zur Verfügung zu stellen. Dies geschieht über so genannte Edge Server, welche sich sozusagen am "Rand" des Netzes befinden und somit in der Nähe der Clients. Der Client verbindet sich über ein dynamisches Client to Server Mapping automatisch zu dem Edge Server, der im am nächsten liegt. Die Distanz zu den Servern wird dabei über die Hop- Anzahl ermittelt.

Auf Grund der Nähe zum Client können die Probleme, welche beim Centraliced Content Serving im Bezug auf QoS auftreten, umgangen werden. Auch kommt es durch die Lastverteilung auf viele verteilte Server auch zu dem positiven Effekt, dass die Störungsanfälligkeit minimiert wird. So verbindet sich der Client, sollte es zu einer Störung des Edge Servers kommen, zu dem nächst näheren Edge Server. Der

zentrale Streaming Server bedient beim Distributed Content Serving auch nur die Edge Server mit Streams, dies führt wiederum zu einer erheblich niedrigeren Bandbreitenvoraussetzung.

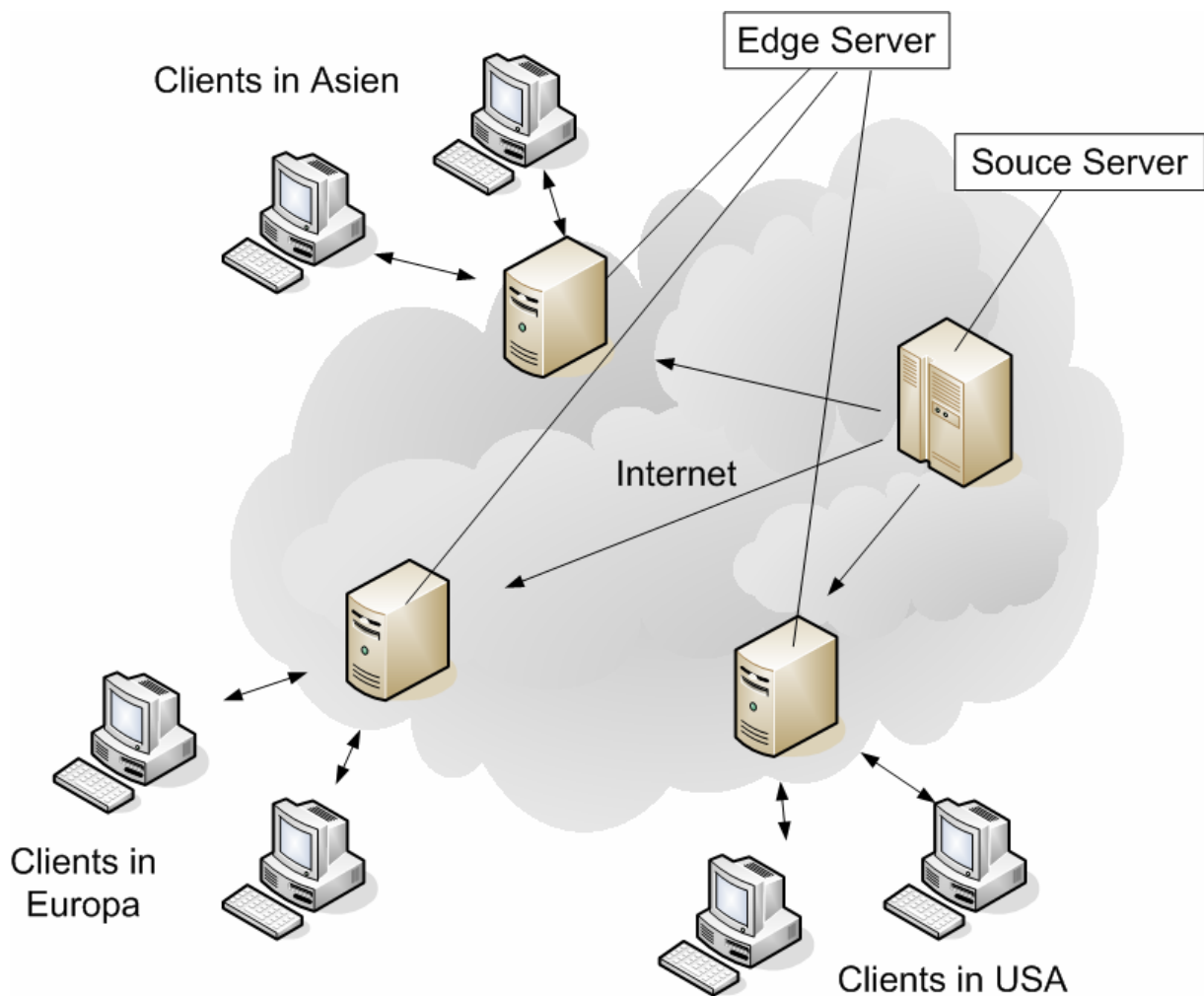


Abb. 6: Edge Server Prinzip

### 2.3 Die Streamverwendung

Am häufigsten geschieht das Abrufen eines Streams durch einen PC, welcher über eine entsprechende Clientsoftware verfügt. Die meisten Anbieter von Clientsoftware erlauben auch ein Einbinden, als Plug-In, in den Internetbrowser. Das erlaubt ein Abspielen des Streams direkt im Webauftritt. Häufig wird diese Möglichkeit bei Internetradio oder zu Werbezwecken genutzt.

Eine weitere Möglichkeit gestreamte Medieninhalte abzurufen, bieten so genannte Set-Topboxen, dies können beispielsweise Spielekonsolen, wie die X-Box von Microsoft, sein. Diese Geräte verfügen über einen Internetzugang und über die entsprechende Hardware, welche das Abrufen eines Live oder On Demand Streams erlauben.

## 3 Dominierende Streaming Lösungen

### 3.1 Windows Media Series

#### 3.1.1 Windows Media Encoder

Der Windows Media Encoder ist ein umfangreiches Werkzeug zur Erzeugung von Streams für die Windows Media Platform. Mit ihm ist es möglich, sowohl Live Content als auch Media Daten zu encoden, die als On Demand angeboten werden sollen.

#### Leistungsmerkmale<sup>3</sup>

- flexible Encoding Modi:
  - o constant/variable bitrate encoding
  - o one-/two-pass encoding
- Improved Multibit Rate Encoding:
  - o MBR Audio, z.B. für Mehrsprachunterstützung
  - o skalierbare Videoauflösungen und verschiedene Framegrößen in einem Stream
- Live DRM:
  - o Real-Time Verschlüsselung zur sicheren Bereitstellung von Content
- Windows Media Video 9 Screen:
  - o Möglichkeit zum Capturen des Desktops bzw. zum Aufzeichnen von Arbeitsvorgängen auf diesem
- Improved Source Switching:
  - o verbessertes Schalten zwischen verschiedenen Medienquellen, wie z.B. zwischen Livequellen und bestehenden Filmdaten
- Device Controll:
  - o Möglichkeit zum direkten Ansteuern von Digital Video (DV) und Video Tape Recorder (VTR) Geräten für one-/two-pass encoding

---

<sup>3</sup> Quelle: Windows Media 9 Deployment Guide

- Multi Channel Audio Sourcing:
  - o Möglichkeit zum capturen von Multichannel Audio, z.B. für Surround Sound
- Erweiterbarkeit und Automation:
  - o Plugin Fähigkeit
  - o Automation des Encodingvorganges durch Skriptsteuerung

### **Hardware/Software Voraussetzungen**

Die Hardwarevoraussetzungen für den Windowsmediaencoder variieren sehr stark, so kann man lt. Microsoft, schon mit einem PC-System mit 266 MHz und 64 MB RAM mit dem Media Encoder arbeiten. Jedoch ist das Encoden in Echtzeit ein rechenintensiver Vorgang, daher wird empfohlen, dass man als Minimum einen PC mit 866 MHz einsetzt, welcher über 256 MB RAM verfügt. Die Anforderungen an die Hardware steigen auch immens, je mehr Features des Windows Media Encoders genutzt werden.

Des Weiteren werden die Betriebssysteme Windows 2000 oder Windows XP vom Windows Media Encoder unterstützt.

#### **3.1.2 Windows Media Server**

Der Windows Media Server ist die zentrale Komponente, welche für die Verteilung der Streams auf die Clients zuständig ist. Er ist Bestandteil in den Serverbetriebssystemen Microsoft Windows Server 2003, in der Standard-, Enterprise- und Datacenter Edition. Somit entstehen keine weiteren Kosten, falls eines dieser Betriebssysteme vorhanden ist.

Mit dem Windows Media Server besteht die Möglichkeit, sowohl Live Content als auch bestehende Media Daten in Verbindung mit dem Windows Media Encoder und dem Windows Media Player, als Stream zur Verfügung zu stellen.

Der Windows Media Server verfügt über die im Folgenden beschriebenen Leistungsmerkmale.

## **Fast Streaming**

Um ein verbessertes Streaming Angebot anzubieten, verfügt der Windows Media Server über Fast Streaming. Fast Streaming ist eine Gruppe von vier Features, dem Fast Start, Fast Cache, Fast Recovery und Fast Reconnect.

- Fast Start ist eine Technik, welche ein sofortiges Wiedergeben des Streams, ohne Wartezeiten durch ein Zwischenspeichern bzw. Puffern, ermöglicht.
- Fast Cache bedeutet, dass der Inhalt zu dem Windows Media Player Cache so schnell übertragen wird, wie das Netzwerk erlaubt. Dies soll die Möglichkeit, dass das Abspielen des Streams durch Netzwerkprobleme unterbrochen wird, reduzieren.
- Fast Recovery arbeitet mit Forward Error Correction (FEC), somit werden redundante Pakete zu den Clients gesendet, welche über eine kabellose Verbindung zum Netz verfügen.
- Fast Reconnect verbindet den Client automatisch wieder mit dem Server, falls es zu einem Abreißen der Verbindung gekommen ist. Dies ermöglicht das unterbrechungsfreie Betrachten des Streams, obwohl es zwischenzeitlich zu Verbindungsstörungen gekommen ist.

## **Server-Side Playlist**

Der Windows Media Server verfügt außerdem über Server-Side Playlist Funktionalität, die es ihm ermöglicht Live Streams und auch On Demand Streams von einer, vom Server ausgeführten, Playlist wieder zu geben. Dieses Verfahren kann sowohl bei Multicast als auch Unicast genutzt werden.

Die zu verwendeten Playlisten basieren auf dem SMIL<sup>4</sup> 2.0 Standard.

## **Cache/Proxy Unterstützung**

Der Windows Media Server ermöglicht eine umfassende Proxy-, sowie Cache-Unterstützung, dies reduziert Bandbreitenprobleme, Latenzzeiten und Serverlast, wie im Abschnitt Load Balancing (bzw. Distributed Content Serving) schon konkret beschrieben wurde.

---

<sup>4</sup> SMIL: Synchronized Multimedia Integration Language

## **Administration**

Um die Administration des Windows Media Servers zu erleichtern, stehen dem Administrator verschiedene Tools, zum Monitoring und der allgemeinen Administration, zur Verfügung. Welche es unter Anderem erlauben, den Windows Media Server über ein Webinterface zu administrieren.

## **Hardware/Software Voraussetzungen**

Wie auch schon beim Windows Media Encoder, sind die Minimalvoraussetzungen, zum Betreiben des Windowsmedia Servers, sehr moderat. Es besteht schon die Möglichkeit mit einem PC, welcher mit einem 233 MHz Prozessor und 256 MB RAM ausgestattet ist, ein Streaming Angebot zu betreiben. Empfohlen wird jedoch schon ein PC mit einem 550 MHz Prozessor und 1 GB RAM.

Als zu verwendendes Betriebssystem gibt Microsoft Windows Server 2003, in den Versionen Standard-, Enterprise- oder Datacenter Edition, an.

### **3.1.3 Windows Media Player**

Der Windows Media Player ist bei den Betriebssystemen von Microsoft enthalten, es existieren aber auch Versionen für Mac OS, die zum freien Download zur Verfügung stehen. Mit ihm ist es nicht nur möglich Streams abzuspielen, sondern er ermöglicht auch die Wiedergabe von lokalen Media Dateien, angefangen von .wav bis hin zu Videos im AVI Format und DVDs.

Des Weiteren existiert zum Windows Media Player eine API, welche es Drittanbietern ermöglicht, eine große Anzahl von Plug-Ins zur Verfügung zu stellen, die den Funktionsumfang des Media Players erheblich erweitern.

Es ist auch eine Einbindung direkt in den Browser möglich, welche eine Einbindung des Streamingangebotes, z.B. bei Webradio, direkt in den Internetauftritt erlaubt.

## **Hardware/Software Voraussetzungen<sup>5</sup>**

Minimalvoraussetzung:

- OS: Microsoft Windows 98, 2000, Me, XP
- Prozessor: 233 MHz Pentium II, oder kompatibel
- RAM: 64 MB

---

<sup>5</sup> Quelle: [www.microsoft.com/windows/windowsmedia](http://www.microsoft.com/windows/windowsmedia)

- Sound: 16-bit Soundkarte

empfohlen:

- OS: Windows XP
- Prozessor: 500 MHz oder schneller
- RAM: 138 MB
- Sound: 5.1 Multichannel Audio Karte

## **3.2 RealNetworks Helix**

### **3.2.1 Helix Producer**

Zum Encoden von Content für die Helix Streaming Server Lösung, steht der Helix Producer zur Verfügung. Er ist kostenlos und ist auch, unter RealNetworks Community und RealNetworks Public Source License, als Quellcode verfügbar.

### **Leistungsmerkmale**

Die freie Helix Producer Software verfügt über folgende Eigenschaften:

- Deinterlace Filter:
  - o Entfernt Fehler, welche beim Encoden von NTSC oder PAL Videos entstehen können
- Inverse Telecine Filter:
  - o Entfernt extra Frames, welche beim konvertieren von analogen Film zu digitalen Video entstehen können
- Black Level Correction Filter:
  - o Nimmt Einstellungen beim Kontrast des Inputs vor, so dass Schwarz dunkler und Weiß heller wirkt. Dieser Filter hilft auch um die Sättigung eines Videos zu verbessern
- Noise Reduction Filter:
  - o Entfernt das Videorauschen
- Improved Variable Bitrate (VBR) Encoding:
  - o Erlaubt eine gleich bleibende Videoqualität, optimiert auf die Bitrate des Endanwenders.



- Two-Pass Encoding:
  - o 2 Durchläufe beim Kodieren, 1. zur Analyse des Materials, 2. zum optimierten Kodieren
- Audio Gain Control:
  - o Filter, welcher Eingangsmaterial angleicht und resampled
- Firewire (IEEE 1394) live capture support
- Forward Error Correction
- Loss Protection:
  - o Senden redundanter Informationen um verlorene Pakete zu rekonstruieren

## Hardware/Software Voraussetzungen

Windows:

- OS: Windows 98 SE/ME/NT 4.0/2000/XP
- Prozessor: Pentium III 400 MHz, oder höher
- RAM: 96 MB, oder mehr
- Sound: Windows kompatible Soundkarte

Linux:

- OS: Linux Kernel 2.2 oder höher
- Prozessor: Pentium III 400 MHz oder höher
- RAM: 96 MB, oder mehr
- Sound: OSS<sup>6</sup> kompatible Soundkarte

### 3.2.2 Helix Server

Der Helix Streaming Server ist ein universell einsetzbarer Streaming Server, welcher sowohl on demand als auch Live Streams zur Verfügung stellen kann.

Er kann eingesetzt werden um Streams der folgenden Formate<sup>7</sup> bereit zu stellen:

- RealNetworks: RealAudio (.rm), RealVideo (.rm, .rmvb), RealPix (.rp) und RealText (.pt)
- Macromedia: Flash (.swf)

---

<sup>6</sup> OSS: Open Sound System

<sup>7</sup> Quelle: Helix Universal Server Administration Guide

- Microsoft: Windows Media (.asf, .wma, .wmv)
- Apple: QuickTime (.mov)
- Standards-Based: MPEG-1, MPEG-4, MP3
- Image-Formats: GIF, JPEG, PNG
- Andere: AU, AIFF, WAV

Um die verschiedenen Formate der unterschiedlichen Hersteller zu unterstützen, arbeitet der Helix Server mit den verschiedenen Encoding Softwarelösungen zusammen. So können die Eingangsstreams vom QuickTime Encoder, Windows Media Encoder, Helix Producer oder anderen Encodern kommen, sofern diese eines der unterstützten Formate bereitstellen. Dieses Verfahren bezeichnet RealNetworks als Universal Media Encoding and Delivery.

## **Protokolle**

Der Helix Streaming Server unterstützt RTP<sup>8</sup>/RTSP<sup>9</sup> mit seinen Möglichkeiten im Bezug auf Proxy- und Cachingtechniken. Die Unterstützung von RTP/RTSP erlaubt es, mit allen Versionen des RealPlayers ab Version G2 zu kommunizieren. Es wird auch das ältere proprietäre Protokoll PNA<sup>10</sup> unterstützt, um kompatibel zu älteren Versionen des RealPlayers zu sein. Da auch Windows Media unterstützt wird, kann auch das microsoft-eigene MMS<sup>11</sup> Protokoll verwendet werden. Des Weiteren unterstützt der Helix Server die Weiterleitung von Mediadaten über TCP, UDP Unicast und UDP Multicast.

## **Splitting<sup>12</sup>**

Splitting erlaubt das versenden eines Broadcast Streams von einem Helix Server zu anderen, welche als Empfänger agieren. Diese versenden den empfangenen Stream dann an die verschiedenen Media Player. Dadurch wird ein verbessertes Aufteilen der Serverlast, sowie die Unterstützung von Firmennetzen, welche durch Firewalls geschützt werden, ermöglicht.

---

<sup>8</sup> RTP: Real-Time Transport Protokoll (RFC 1889)

<sup>9</sup> RTSP: Real-Time Streaming Protokoll (RFC 2326)

<sup>10</sup> PNA: Progressive Networks Audio

<sup>11</sup> MMS: Microsoft Media Server Protokoll

<sup>12</sup> Quelle: Helix Universal Server Administration Guide

## **Archiving<sup>13</sup>**

Da Live Events nicht als Dateien vorliegen, stellt der Helix Server eine Archivierungsfunktion zur Verfügung, welche ein Mitschneiden des Streams erlaubt, um ihn später als on demand zur Verfügung zu stellen. Dieses Feature unterstützt aber nur die RealMedia und MP3 Formate.

## **Redundante Encoder**

Speziell bei Live Broadcasts ist ein störungsfreies und zuverlässiges encoden notwendig. Um dies zu gewährleisten, unterstützt der Helix Streaming Server redundante Encoder, das heißt, er arbeitet mit multiplen Eingangsquellen. Falls ein Encoder ausfallen sollte, schaltet er automatisch zum Nächsten.

## **Administration**

Die Administration des Helix Streaming Servers kann lokal über XML-Dateien oder via Webfrontend geschehen. Über JAVA-Appletts ist eine Monitoringfunktion über die angeschlossenen Clients und die Nutzung der Dateien/Streams gegeben.

### **3.2.3 Real Player**

Der Real Player erlaubt eine Wiedergabe von Streams in den meistverwendeten Formaten, wie AVI, MP3, RealAudio, RealVideo, WAV Audio und Windows Media.

Neben der frei downloadbaren RealPlayer Version, bietet RealNetworks auch die, ca. 20\$ teure, Premium Version des Players an, welche unter anderem das Brennen von CDs, Analoges Aufnehmen und MP3 Encoding unterstützt.

Wie beim Windows Media Player steht auch beim RealPlayer eine Schnittstelle für Plug-Ins zur Verfügung, um die Funktionalität des Players zu erweitern. So gibt es z.B. Erweiterungen, welche einen Export von RealMedia Daten oder Präsentationen erlauben.

---

<sup>13</sup> Quelle: Helix Universal Server Administration Guide

## **Hardware/Software Voraussetzungen**

Windows:

- OS: Windows 95, 98, Me, NT 4, 2000, XP
- Prozessor: Pentium II 233 MHz, oder schneller
- RAM: 64 MB
- Sound: 16-bit Soundkarte

Mac OS:

- OS: Mac OS 8.6, oder höher
- Prozessor: PowerPC mit 233 MHz, oder schneller
- RAM: 128 MB
- Sound: 16-bit Soundkarte

### **3.3 Apple QuickTime**

#### **3.3.1 QuickTime Broadcaster**

Das Pendant von Apple, speziell für Live Streams, zum Windows Media Encoder und Helix Producer ist der QuickTime Broadcaster. Der QuickTime Broadcaster unterstützt die Übertragung von MPEG-4 kompatiblen Audio- und Videodaten an jeden ISO konformen Player.

#### **Leistungsmerkmale<sup>14</sup>**

- MPEG-4 Unterstützung
- 3GPP [Auf MPEG-4 basierender Standard speziell für breitbandige mobile Netzwerke] Unterstützung
- Video-on-Demand:
  - o Aufzeichnung der Live-Sendung auf Speichermedium, mit direkter Einbindung einer Steuerspur
- Echtzeit-Feedback:
  - o Versendung von statistischen Daten an den Admin, wie Datenrate, Bildrate, Prozessorauslastung etc.

---

<sup>14</sup> Quelle: [www.apple.com/de/quicktime/products/broadcaster](http://www.apple.com/de/quicktime/products/broadcaster)

- Aufzeichnung über Firewire
- AppleScript Unterstützung

## **Hardware / Software Voraussetzungen**

Zum Encoden mit dem Quicktime Broadcaster benötigt man als Minimalvoraussetzung einen Macintosh Computer mit PowerPC Prozessor, 128 MB RAM. Als Betriebssystem muss Apple Mac OS X verwendet werden, und der PC muss über QuickTime 6 verfügen.

Die empfohlene Konfiguration ist ein Macintosh PC mit PowerPC G4, welcher über 256 MB RAM verfügt.

### **3.3.2 QuickTime Streaming Server**

Der QuickTime Streaming Server ist Bestandteil des Mac OS X Server Betriebssystems ("Panther") und ist vorinstalliert auf Apple Server Hardware. Es entstehen nach dem Erwerb des Betriebssystems also keine weiteren Lizenzkosten. Wie auch bei den Server Lösungen von RealNetworks und Microsoft, ist es mit dem QuickTime Streaming Server möglich, sowohl Live als auch On Demand Angebote bereit zu stellen. Die unterstützten Formate sind neben dem appleeigenen .mov auch MPEG-4 und MP3. Die native Unterstützung von MPEG-4 und 3GPP erlaubt eine Wiedergabe der gestreamten Inhalte auf den standardmäßigen Media-Playern der Mac-, Windows- und Linux Plattformen.

Außer dem QuickTime Streaming Server bietet Apple noch den Darwin Streaming Server als Open Source Version an. Dieser steht, im Gegensatz zum QuickTime Server, auch für verschiedene Betriebssysteme zur Verfügung.

## **Protokolle**

Der QuickTime Streaming Server (QTSS) sowie der Darwin Streaming Server (DST) verwenden zur Bereitstellung der Media Daten die standardisierten Protokolle RTP und RTSP. Somit stehen die verschiedenen, schon in den vorherigen Kapiteln beschriebenen, Möglichkeiten zur Skalierung und Optimierung zur Verfügung.

Wie die Serverapplikationen der anderen Hersteller, unterstützen sowohl der QTSS und der DST den Transport der Streams mittels Unicast und Multicast.

## **Relay<sup>15</sup>**

Der QTSS und DST können auch als Relay eingesetzt werden, welches die eingehenden Streams auf einen oder mehrere Streaming Server, z.B. in anderen Subnetzen, verteilt. Das ermöglicht eine bessere Ressourcennutzung, ähnlich dem Distributed Content Serving.

## **Instand-On<sup>16</sup>**

Instand-On ist eine Technologie, die es ermöglicht, die Zeit, die beim Endnutzer zum Füllen des Buffers verwendet wird, erheblich zu reduzieren. Mit Instand-On ist es möglich im Film, falls es sich um ein On Demand Angebot handelt, ohne größere Wartezeiten vor und zurück zu springen.

Um dies zu ermöglichen nutzt Instand-On das overbuffering, welches durch den QTSS angeboten wird. Overbuffering bedeutet, dass das Zwischenspeichern des Streams schneller als das Abspielen des Streams geschieht, was aber stark von der Bandbreite des Nutzers und der Bitrate des Streams abhängig ist.

## **Hardware/Software Voraussetzungen<sup>17</sup>**

QTSS:

- OS: Mac OS X Server
- Prozessor: PowerPC G3, oder höher
- RAM: 128 MB, oder mehr

DSS:

- OS: Mac OS X, RedHat Linux 7.2, Solaris 8, Windows NT/2000/XP
- Prozessor: 500 MHz, oder höher
- RAM: 128 MB, oder mehr

### **3.3.3 QuickTime Player**

Den QuickTime Player bietet Apple in zwei Varianten an, den QuickTime Player und QuickTime Player Pro. Beide bieten die Funktionalität die Streams, in den vom QTSS bereitgestellten Formaten, abzuspielen. Die Pro Version, welche als Einzelplatzlizens

---

<sup>15</sup> Quelle: QuickTime Streaming Server 5.0 Administration

<sup>16</sup> Quelle: QuickTime Streaming Server 5.0 Administration

<sup>17</sup> Quelle: QuickTime Streaming Server/Darwin Streamin Server Administrator's Guide

40,60€ kostet, ermöglicht, im Gegensatz zur freien Version, eine Wiedergabe des Streams als Vollbild, das lokale Speichern von Streams, das Bearbeiten von Filmen und das Erstellen von MPEG-4 und GPP3 Inhalten.

Beide Versionen unterstützen die Wiedergabe von MPEG-4, AAC<sup>18</sup>, MPEG-2 und MP3.

Um Streams im Browser abspielen zu können, ist eine Einbindung als Browser-Plug-In auch beim QuickTime Player möglich.

### **Hardware/Software Voraussetzungen<sup>19</sup>**

Mac OS:

- OS: Mac OS X oder Mac OS 8.6/9
- Prozessor: PowerPC G3
- RAM: 128 MB

Windows:

- OS: Windows 98, Me, 2000, XP
- Prozessor: Pentium PC oder kompatibel
- RAM: 128 MB

---

<sup>18</sup> Advanced Digital Audio

<sup>19</sup> Quelle: [www.apple.com/quicktime](http://www.apple.com/quicktime)

## 4 Zusammenfassung

Um kleinere Live Streaming Lösungen, wie z.B. im privaten Bereich, zur Verfügung zu stellen, genügen meist schon handelsübliche Multimedia PCs, rudimentäre Video-/Soundhardware, einfache Aufnahmetechnik, wie z.B. eine Webcam, und ein Internetanschluß. So lassen sich schon Testkonfigurationen mit geringem Aufwand zusammenstellen. Die Frage nach der zu wählenden Streaming Server Lösung wird im privaten Bereich höchstwahrscheinlich durch das Budget entschieden und sicherlich mit der Entscheidung für eine der freien Lösungen enden, bzw. durch die vorhandene Hardware bestimmt.

Als eines der Hauptprobleme bei privat betriebenen Streaming Lösungen ist die zur Verfügung stehende Bandbreite anzusehen, da der Upstream, z.B. bei DSL, sehr begrenzt ist und somit die Verbreitung des Streams, speziell über Unicast, nur an ein äußerst begrenztes Publikum zulässt.

Um ein breites Publikum mit einem professionellen Live Stream zu versorgen, ist jedoch eine intensive Planung nötig. So stellt sich nicht nur die Frage nach der zu verwendeten Hard- und Software, bzw. der zu bevorzugenden Streaming Lösung, sondern auch Problemstellungen, wie die Verfügbarkeit, Kapazität und Kosten treten zunehmend in den Vordergrund.

Schon die Aufnahme qualitativ hochwertiger Media Daten stellt hohe Herausforderungen an die Aufnahmetechnik, da bei einer Aufnahme für einen Live Stream keine Nachbearbeitung der Daten möglich ist. Somit würden sich Mängel, die bei der Aufnahme entstehen würden, direkt auf die Qualität des Streaming Angebotes auswirken.

Für den Erfolg des Angebotes ist die Verfügbarkeit im hohen Maße mitentscheidend, somit sind Überlegungen im Bezug auf die Zielgruppe von entscheidender Bedeutung. So stellen sich Fragen, für wie viele Nutzer das Angebot ausgelegt sein soll und wo sich diese Nutzer befinden. Bei einer großen Anzahl von Nutzern, die sich auch auf verschiedenen Kontinenten befinden können, ist meist die Nutzung von Distributed Content Serving, und damit von Content Delivery Networks, wie Akamai, unabdingbar, um die Verfügbarkeit und auch die Kapazitäten zur Verfügung stellen zu können.



## 5 Abbildungsverzeichnis

Abb. 1: Prinzipieller Aufbau einer Streaming Lösung

Abb. 2: Unicast

Abb. 3: Multicast

Abb. 4: Centralized Content Serving

Abb. 5: Load Balancing

Abb. 6: Edge Server Prinzip

## 6 Literatur- und Quellenangaben

- [1] Steve Mack: "Streaming Media Bible", Hungry Minds, 2002
- [2] Eyal Menin: "Streaming Media Handbook", Prentice Hall PTR, 2002
- [3] Detlef Randerath, Christian Neumann:  
"Streaming Media: Produktion und Broadcasting von Audio- und Video-Content im  
Web", Galileo Press, 2001
- [4] Windows Media 9 Deployment Guide
- [5] Helix Universal Server Administration Guide
- [6] QuickTime Streaming Server 5.0 Administration
- [7] QuickTime Streaming Server/Darvin Streaming Server Administrator's Guide
- [8] [www.microsoft.com/windows/windowsmedia](http://www.microsoft.com/windows/windowsmedia)
- [9] [de.real.com](http://de.real.com)
- [10] [www.apple.com/quicktime](http://www.apple.com/quicktime)