Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Markus Vahlenkamp

Content Delivery Networks - Chancen und Konzepte für Internet Service Provider

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Markus Vahlenkamp

Content Delivery Networks - Chancen und Konzepte für Internet Service Provider

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Schmidt
Zweitgutachter: Prof. Dr. rer. nat. habil. Westhoff

Abgegeben am 29. März 2011

Hochschule für Angewandte Wissenschaften Hamburg
Fachbereich Technik und Informatik
Zusammenfassungsblatt zur Bachelorarbeit

Markus Vahlenkamp

Thema der Bachelorarbeit
Content Delivery Networks - Chancen und Konzepte für Internet Service Provider

Stichworte
Content Delivery Network, CDN, Internet Service Provider, ISP, Video on Demand, VoD, Media Flow Controller, Verkehrsoptimierung, Adpative Bitratenanpassung

Kurzzusammenfassung
Um die steigenden Volumen der täglich übertragenen digitalen Inhalte in Netzwerken auszuliefern, werden Content Delivery Network eingesetzt. Diese Arbeit soll klären, welche Möglichkeiten sich aus der Verwendung von CDNs für Internet Service Provider ergeben. Im Rahmen dieser Arbeit wird die Verwendung von CDNs zur Verkehrsoptimierung und zur Schaffung von Mehrwerten gegenüber dem Kunden des ISPs konzeptuell dargestellt und anhand eines CDN-Server Produkts untersucht.

Markus Vahlenkamp

Title of the paper
Content Delivery Networks - Chances and Concepts for Internet Service Provider

Keywords
Content Delivery Network, CDN, Internet Service Provider, ISP, Video on Demand, VoD, Media Flow Controller, Traffic Optimization, Adaptive Bitrate Adjustment

Abstract
Content delivery network are used in order to deliver the increasing volume of daily transmitted digital content in networks. This work will clarify the opportunities arising from the use of CDNs for Internet Service Provider. In this study, the use of CDNs for traffic optimization and the creation of added value is represented conceptually and examined on the basis of a CDN server product.

In Kooperation mit:

Xantaro Deutschland GmbH
ABC-Straße 45
20354 Hamburg
www.xantaro.net

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

Tabellenverzeichnis

Kapitel 1

Einführung

1.1 Ziel der Arbeit

Content Delivery Networks (CDNs) werden seit 1998 eingesetzt, um digitale Inhalte an Inhaltekonsumenten auszuliefern [Douglis und Kaashoek (2001)]. Populäre Internetdienste wie YouTube oder BBC, die große Inhaltsmengen zur Verfügung stellen, schaffen es erst mithilfe von CDNs, der wachsenden Anzahl an Nutzern die angefragten Inhalte zur Verfügung zu stellen. Unternehmen wie Akamai Technologies[1] und Limelight Networks[2] haben sich auf die Auslieferung von großen Inhaltsvolumina spezialisiert. Beispielsweise liefert Akamai laut eigenen Angaben zwischen 10 und 20 Prozent des weltweiten Internetverkehrs über das eigene CDN aus. CDN-Provider bieten ihren Kunden, den Inhalteanbietern, die Möglichkeit, hohe Datenvolumen in einer angemessenen Qualität zu den Inhaltekonsumenten zu übertragen.
Bei Inhalten mit gesteigerten Anforderungen an die Zuverlässigkeit und Kontinuität der Datenübertragung kann es sich beispielsweise um Videoinhalte handeln. Diese erfordern, dass der Datenstrom ohne größere Beeinträchtigungen durch Engpässe oder ähnliche Effekte im Netz übertragen wird. Sobald diese Anforderungen nicht erfüllt werden können, kommt es zu Pausen oder Aussetzern bei der Wiedergabe. Um dies zu vermeiden, werden CDNs eingesetzt, die die effiziente Verteilung von Inhalten unterstützen.
Das Ziel der vorliegenden Arbeit ist, die Chancen, die sich einem Internet Service Provider (ISP) durch den Einsatz eines eigenen CDNs eröffnen, darzustellen. Ein weiteres Ziel besteht in der Darstellung von Konzepten, die beschreiben, wie die Architektur eines vom ISP betriebenen CDNs aussehen könnte.

[1] www.akamai.de
[2] www.limelightnetworks.com

1.2 Aufbau der Arbeit

Im 2. Kapitel werden die Grundlagen zu Content Delivery Networks erläutert. Hierzu werden die vier Teilsysteme Request-Routing, Content Delivery, Distribution und Accounting eines CDNs mit ihren Eigenschaften und Funktionsweisen detailliert dargestellt. Kapitel 3 befasst sich zunächst mit den möglichen Übertragungsarten von Video on Demand (VoD)-Inhalten, bevor unterschiedliche Konzepte zur Realisierung eines VoD-Angebots verifiziert werden. Kapitel 4 thematisiert Probleme, denen ISPs heutzutage gegenüberstehen, bevor anschließend Konzepte zur Steigerung der Umsatzerlöse bzw. Senkung der Kosten unter Verwendung vom ISP betriebener CDNs aufgezeigt werden. In Kapitel 5 wird der Aufbau eines VoD-Dienstes und eines kollaborativen Proxyserververbunds unter Verwendung einer CDN-Serverlösung der Firma Juniper Networks[3] dargestellt. Im Speziellen wird bei der Einrichtung des VoD-Dienstes detailliert auf die Umsetzung der Funktion zur adaptiven Bitratenanpassung des auszusendenden Videomaterials eingegangen. Als weiterer Punkt wird die Konfiguration und Arbeitsweise eines Interception Proxyservers zur Zwischenspeicherung von Videoinhalten der YouTube Plattform[4] und somit zur Verkehrsoptimierung analysiert. Abschließend wird eine inhaltliche Zusammenfassung der Arbeit zusammen mit dem Ausblick auf noch offene und in dieser Arbeit nicht behandelte Forschungsbereiche und Entwicklungen im Umfeld von CDNs gegeben.

[3] www.juniper.net

[4] www.youtube.com

Kapitel 2

Content Delivery Networks

Ein Content Delivery Network (CDN) besteht aus einer Menge von Netzwerkelementen, die dem effizienten Ausliefern von digitalen Inhalten dienen. Wie in der Abbildung 2.1 zu sehen, stellen die Inhalteanbieter ihre digitalen Inhalte in einem CDN zur Verfügung, aus dem die Inhaltekonsumenten die Inhalte anschließend abrufen.
Die Inhalte werden auf Stellvertreterserver repliziert, die sich in verteilten Lokationen befinden. Zu diesen Lokationen gehören unter anderem die Points of Presence (PoPs) verschiedener ISPs [Pathan und Buyya (2007)]. Durch diese Art der Verteilung sind die Inhalte gut erreichbar und nah am Inhaltekonsumenten gespeichert. Hierdurch wird eine effiziente Datenübertragung erreicht, was zu einer Schonung der Backbonebandbreiten und der Transit-Verbindungen führt. Für den Inhaltekonsumenten wirkt sich die Nutzung eines CDNs durch verringerte Latenzen und eine erhöhte Servicequalität positiv aus.

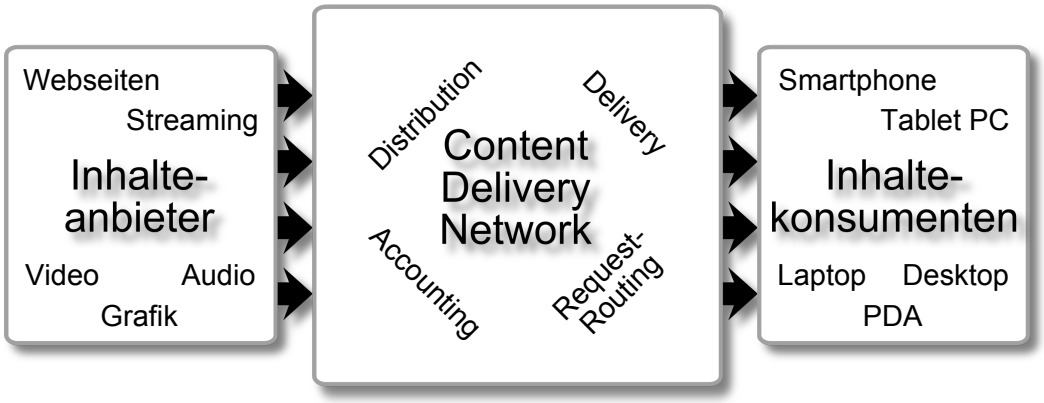


Abbildung 2.1: Abstrakter CDN-Überblick nach Buyya u. a. (2006)

Ein CDN besteht im Allgemeinen aus den folgenden Komponenten [Day u. a. (2003); Hofmann und Beaumont (2005); Yin u. a. (2010)].

Request-Routing System Das Request-Routing System (siehe Kapitel 2.1) leitet die Inhaltsanfragen an geeignete Stellvertreterserver weiter. Hierzu interagiert es mit dem Distribution System, um Informationen über die aktuell im Cache präsenten Inhalte zu erhalten.

Content-Delivery System Das Content-Delivery System (siehe Kapitel 2.2) besteht aus Stellvertreterservern, die die Inhalte an die Inhaltekonsumenten ausliefern.

Distribution System Das Distribution System (siehe Kapitel 2.3) implementiert Mechanismen, mit denen der Inhalt vom Herkunftsserver zu den Stellvertreterservern gelangt. Des Weiteren sorgt es für die Konsistenz der gecachten Inhalte.

Accounting System Das Accounting System (siehe Kapitel 2.4) sammelt statistische Daten der drei oben genannten Systeme. Diese Informationen werden für Verkehrsstatistiken und nutzungsbasierte Abrechnungen verwendet.

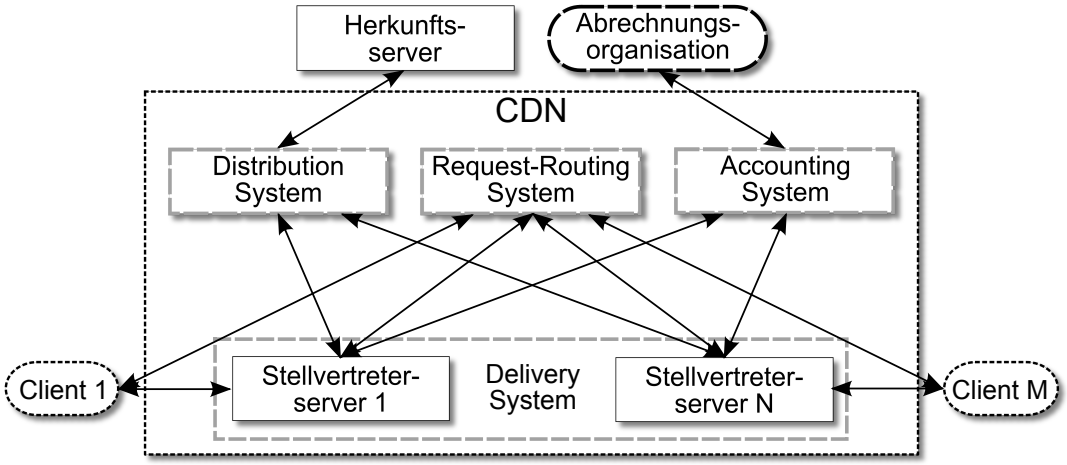


Abbildung 2.2: Komponenten eines CDNs

2.1 Request-Routing System

Das Request-Routing System ist für die Weiterleitung der Clients an einen geeigneten Stellvertreterserver zuständig.

Es setzt sich aus zwei Teilen zusammen [Sivasubramanian u. a. (2004); Pathan und Buyya (2007)]: Dem Algorithmus (siehe Kapitel 2.1.2), der anhand der zur Verfügung stehenden Metriken (siehe Kapitel 2.1.1) entscheidet, welcher Stellvertreterserver der am besten geeignete für die Verarbeitung einer Inhaltsanfrage ist und dem Mechanismus (siehe Kapitel 2.1.3), über den der Inhaltekonsument an den für ihn am besten geeigneten Stellvertreterserver verwiesen wird.

2.1.1 Metriken

Um beim Request-Routing entscheiden zu können, welcher Stellvertreterserver am besten für die Verarbeitung der Inhaltsanfragen eines Clients geeignet ist, müssen zunächst Informationen erhoben und zu Metriken aufbereitet werden. Im Folgenden wird ein Überblick über verschiedene Metriken sowie die Art ihrer Erhebung gegeben.

Proximität (Nähe) Die Proximität in Netzwerken kann durch drei Aspekte definiert werden. Zum einen kann die topologische Proximität zweier Knoten in einem Netzwerk durch die Anzahl der zu traversierenden Router bei einer Ende-zu-Ende-Verbindung bestimmt werden [Obraczka und Silva (2000)]. Da jeder Router eine Warteschlangen- und eine Verarbeitungsverzögerung bei der Paketvermittlung erzeugt, wird davon ausgegangen, dass Wege mit weniger Hops eine höhere Qualität der Datenübertragung liefern können. Die Latenz der Datenübertragung, als auch das Auftreten von Jittern ist geringer.

Ein weiteres Maß für die Proximität liefert das AS_PATH Attribut des Border Gateway Protocols [Rekhter u. a. (2006)]. Es gibt an, welche Autonomous Systems (ASs) auf der Ende-zu-Ende-Verbindung vom Client zu den jeweiligen Stellvertreterservern liegen (AS-Hops). Der AS_PATH gibt die zu überquerenden Providergrenzen an. Er stellt jedoch nicht immer eine gute Abschätzung der Proximität dar, da der AS_PATH manipuliert werden kann und die Anzahl der Router, die sich innerhalb eines AS verbergen, variiert [Hyun u. a. (2003); Paxson (1997); Obraczka und Silva (2000)].

Bei der Latenz-Proximität werden Paketlaufzeiten gemessen und die daraus resultierenden Werte als Maß der Nähe genommen. Die Werte müssen in kurzen Intervallen aktualisiert werden, da die Latenz unter anderem von der schwankenden Verkehrslast im Netz abhängt. Die gemessenen Daten können durch Mittelung zu einer ausreichend aussagekräftigen Metrik zusammengefasst werden.

Server Load Die Auslastung des Stellvertreterservers eignet sich als Metrik, da sich die Auslieferungsverzögerung von Inhalten neben der Übertragungszeit auch aus der Anfrageverarbeitungszeit besteht. Die Auslastung gibt Aufschluss darüber, ob der Stellvertreterserver die Inhaltsanfrage in ausreichender Qualität bedienen kann. Die Stellvertreterserver müssen Inhaltsanfragen entgegennehmen, verarbeiten und die Inhalte im Speicher verfügbar machen, um sie von dort ausliefern zu können. Die hierbei entstehenden Einflussfaktoren können durch verschiedene Metriken beschrieben werden. Beispiele hierfür sind:

- Arbeitsspeicherauslastung
- aktuelle Festplattenauslastung
- aktuelle CPU-Auslastung
- aktuelle Anzahl der Client-Verbindungen

Diese Einzelmetriken können zu einer Gesamtmetrik aggregiert werden.

Paket Loss Der Pfad vom Client zum Stellvertreterserver sollte ein möglichst geringe Paketverlustrate bieten. Gehen häufig Pakete verloren, werden diese bei Transmission Control Protocol (TCP)-Verbindungen erneut übertragen, bei User Datagram Protocol (UDP)-Verbindungen gehen sie ohne weitere Behandlung durch die Transportschicht verloren. Solange in höheren Schichten keine Vorkehrungen getroffen wurden, um einen gewissen Prozentsatz verlorener Pakete zu tolerieren, wirkt sich dies negativ auf die Qualität des Dienstes aus.

Average Bandwidth Die durchschnittlich verfügbare Bandbreite der Ende-zu-Ende-Verbindung ist vor allem für Streaming-Inhalte von großer Bedeutung. Je höher die durchschnittlich zur Verfügung stehende Bandbreite, umso bessere Qualität können die gestreamten Inhalte aufweisen. Wird ein gewisser Mindestwert unterschritten, kann dies beispielsweise bei Videoinhalten zu Verzögerungen oder Aussetzern der Wiedergabe führen.

Die vorgestellten Metriken können auch kombiniert werden, wie es in Krishnamurthy u. a. (2001) mit dem Proximity-Load-Threshold Algorithmus vorgestellt wird. Wie bereits dem Namen des Algorithmus zu entnehmen ist, wird zunächst die Proximitätsmetrik genutzt, um einen nahe gelegenen Stellvertreterserver zu ermitteln. Anschließend wird die Server Load Metrik ausgewertet, um sicherzustellen, dass der Stellvertreterserver nicht überlastet ist.
Das NetAirt Redirection System [Szymaniak u. a. (2003)] für Apache Webserver benutzt die AS-Pfadlänge zwischen dem Inhaltekonsument und den verschiedenen Stellvertreterservern als Metrik für die Wahl des am besten geeigneten Stellvertreterservers. Die Metrik muss periodisch aktualisiert werden, da sich die Netzwerkstruktur über die Zeit gesehen, ändert.

Beim NetAirt System wird zugrunde gelegt, dass sich das Routing im Internet symmetrisch verhält, so dass Pakete von der Quelle zum Ziel denselben Weg nehmen wie Pakete in die entgegengesetzte Richtung.

Die im vorherigen Abschnitt angeführten Metriken müssen vor ihrer Verwendung erhoben und aggregiert zur Verfügung gestellt werden.
Zur Erhebung von Metrikdaten können Network Probing, Traffic Monitoring sowie das Surrogate Feedback verwendet werden (siehe Abbildung 2.3). Diese werden im Folgenden näher erläutert.

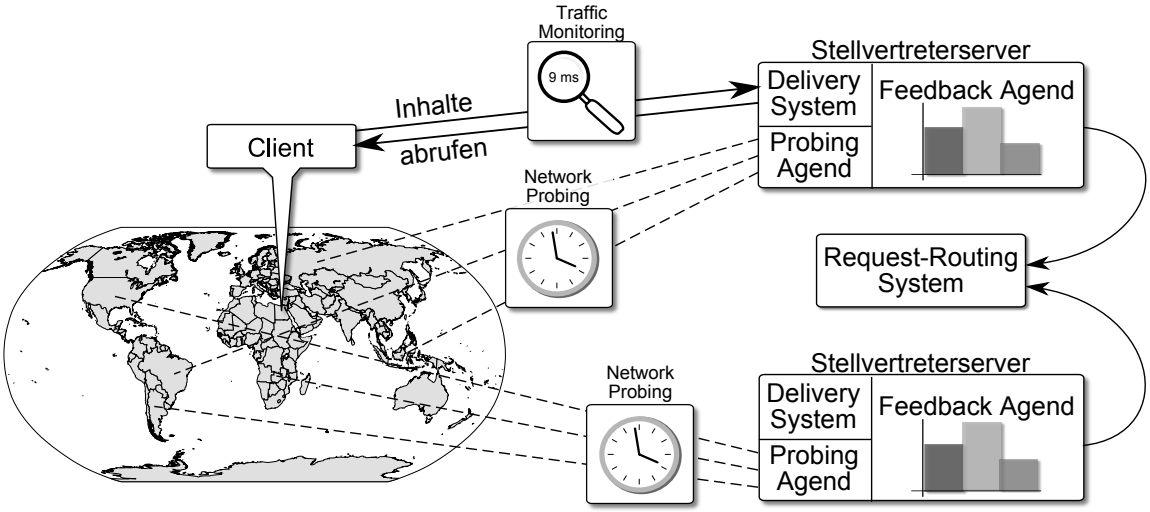


Abbildung 2.3: Erhebung von Request-Routing Metriken

Network Probing Beim Network Probing werden aktiv Messungen durchgeführt. Stellvertreterserver können Internet Control Message Protocol (ICMP) Echo-Nachrichten [Postel (1981)] an potentielle Clients senden. Hierdurch wird eine Datenbasis geschaffen, die Aussagen über die Netzwerk-Proximität zulässt. Die Messungen müssen periodisch wiederholt werden, um die Informationen über den globalen Zustand der Netze aktuell zu halten. Aktive Network Probing Techniken bergen allerdings auch Probleme. Zum einen verursachen sie zusätzliche Verkehrslast im Netz, zum anderen ist das ICMP-Protokoll häufig durch Firewalls blockiert oder wiederholte ICMP-Pakete werden von Intrusion Detection Systeme (IDS) als Denial of Service (DoS)-Angriff gewertet [Merola (2006)].

Traffic Monitoring Beim Traffic Monitoring wird die Kommunikation zwischen Client und Stellvertreterserver auf passive Weise ausgewertet. Hierzu werden Informationen aus dem Verhalten der TCP-Verbindung gezogen. Systeme wie z. B. IDMaps [Francis u. a. (2001)] erstellen dann aus diesen Daten Proximitätsinformationen.

Surrogate Feedback Beim Surrogate Feedback werden Statistiken und Daten über den aktuellen Systemzustand des Stellvertreterservers gesammelt. Zum einen können die Daten durch das Messen und Auswerten periodisch abgesetzter Inhaltsanfragen erhoben werden, zum anderen ist es möglich, dass der Stellvertreterserver selber dem Request-Routing System Rückmeldungen über seinen Zustand gibt. Hierzu wird ein Agent auf dem Stellvertreterserver eingerichtet, der periodisch oder ereignisgetrieben Statistiken und Gesundheitsinformationen an das Request-Routing System weiterreicht.

2.1.2 Request-Routing Algorithmen

Request-Routing Algorithmen beschreiben die Auswahlkriterien zur Bestimmung von geeigneten Stellvertreterservern. Mit ihnen wird entschieden, an welchen Stellvertreterserver die Clients ihre Inhaltsanfragen richten.
Request-Routing Algorithmen werden in nicht-adaptive und adaptive Algorithmen unterteilt [Pathan und Buyya (2007); Sivasubramanian u. a. (2004)].

Nicht-adaptive Algorithmen Die nicht-adaptiven Algorithmen passen sich nicht den aktuellen Gegebenheiten (z. B. Latenz, Server Auslastung) im Gesamtsystem an, weshalb sie einfacher zu implementieren sind. Nicht-adaptive Algorithmen nutzen beispielsweise Heuristiken zur Auswahl von Stellvertreterservern. Eine Bedingung für die Skalierung der nicht-adaptiven Algorithmen leitet sich aus den Eigenschaften der Anfragen ab. Diese sollten möglichst gleicher Art und Größe sein, da es sonst zu einer ungleichmäßigen Verteilung der durch die Anfragen entstehende Last auf den Stellvertreterservern („hot-spots“ [Arlitt und Jin (2000)]) kommen kann. Hierdurch würde die Gesamtleistungsfähigkeit des Systems negativ beeinflusst.
Eine einfache Form des nicht-adaptiven Request-Routings stellt das Round-Robin Verfahren dar. Hierbei werden alle eingehenden Inhaltsanfragen gleichmäßig auf die verfügbaren Stellvertreterserver verteilt. Es wird vorausgesetzt, dass alle Stellvertreterserver dieselbe Leistungsfähigkeit besitzen und die Caches identische Inhalte aufweisen. Das Round-Robin Verfahren birgt in weit verteilten Systemen das Problem, dass die gewählten Stellvertreterserver weiter entfernt sein können als der Herkunftsserver.

Adaptive Algorithmen Adaptive Algorithmen berücksichtigen den aktuellen Zustand des CDNs. Hierzu werden Informationen genutzt, die sich aus Metriken (siehe Kapitel 2.1.1) wie z. B. der Last auf den Stellvertreterserver sowie der Auslastung von Netzwerkverbindungen zusammensetzen. Die adaptiven Verfahren zeigen durch ihre Eigenschaften ein gutes Verhalten in Ausnahmesituationen. Ein Beispiel für eine solche Ausnahmesituation stellen beispielsweise Flash Crowds [Arlitt und Jin (2000)]

dar, bei denen eine Webseite einen sprunghaften, zeitlich begrenzten Anstieg an Inhaltsanfragen verzeichnet. Adaptive Algorithmen sind in diesen Fällen in der Lage, Clients von überlasteten Stellvertreterservern oder Netzwerkbereichen weg zu anderen Stellvertreterservern zu lenken.

2.1.3 Request-Routing Mechanismen

Request-Routing Mechanismen werden verwendet, um die Inhaltsanfragen der Clients auf Stellvertreterserver des CDNs zu verteilen. Das Lenken der Clients an die zuvor durch den Request-Routing Algorithmus gewählten Stellvertreterserver geschieht durch unterschiedliche Mechanismen [Barbir u. a. (2003); Pathan und Buyya (2007)].
Entweder auf Clientebene, so dass jede Anfrage eines Clients an den gleichen Stellvertreterserver geleitet wird oder auf Objektebene, so dass für jedes durch den Client angefragte Objekt ein anderer Stellvertreterserver genutzt werden kann.
Konkrete Request-Routing Mechanismen werden im Weiteren detailliert aufgeführt.

DNS-based Request-Routing Beim DNS-based Request-Routing werden spezielle DNS-Server in den Prozess der Namensauflösung eingebunden [Barbir u. a. (2003)]. Diese DNS-Server sind in der Lage, anhand von Metriken den für eine Inhaltsanfrage am besten geeigneten Stellvertreterserver auszuwählen und dessen IP-Adresse zurückzuliefern. Um diese Art des Request-Routings durchzuführen werden A-, NS- und CNAME-Records [Mockapetris (1987)] verwendet. Barbir u. a. (2003) beschreiben die folgenden Techniken, um die Inhaltekonsumenten zu einem geeigneten Stellvertreterserver zu leiten [Barbir u. a. (2003); Pathan und Buyya (2007)].

Single Reply Bei diesem Ansatz liefert der authoritative (für die entsprechende Domain zuständige) Request-Routing DNS-Server [Mockapetris (1987)] die IP-Adresse des gewählten Stellvertreterservers an den anfragenden lokalen DNS-Server.

Multiple Replies Bei diesem Ansatz werden A-Records mehrerer geeigneter Stellvertreterserver an den anfragenden Client-DNS-Server übermittelt. Der Caching DNS-Server auf der Clientseite kann nun bei weiteren Clientanfragen die IP-Adressen im Round-Robin Verfahren zurück liefern. Hierdurch verteilt auch der lokale Caching DNS-Server die Last seiner lokalen Clients auf mehrere Stellvertreterserver.

Multi-Level Resolution Bei diesem Ansatz werden mehrere Request-Routing Client-DNS-Server in die Namensauflösung einbezogen. Dies wird genutzt, um Entscheidungen auf mehreren Ebenen treffen zu können. Der erste DNS-Server trifft eine grobe Entscheidung, an welchen weiteren DNS-Server er verweist.

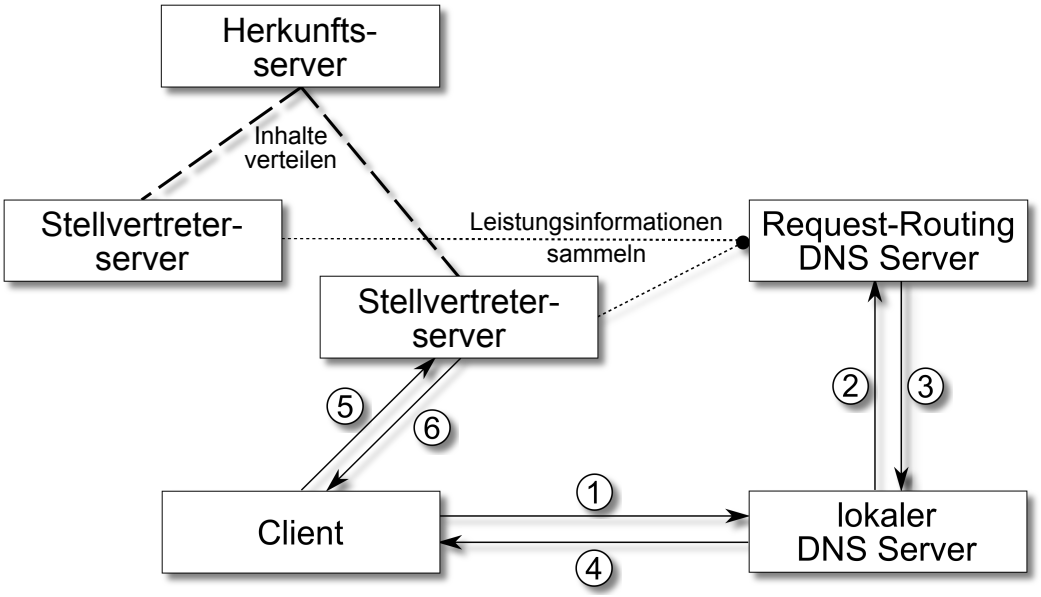


Abbildung 2.4: DNS-based Request-Routing

Anschließend trifft der zweite DNS-Server eine präzisere Entscheidung, welche IP-Adressen von Stellvertreterservern er dem Caching DNS-Server des Clients übermittelt. Grundlage für diesen Mechanismus bieten NS- und CNAME-Records.
Werden NS-Records verwendet, kann ein DNS-Server für seine Subdomains authoritative Name-Server angeben. Diese Server sind dann zuständig für die Verwaltung der Subdomains. Diese Subdomain DNS-Server können nun speziellere Request-Routing Entscheidungen treffen.
Werden CNAME-Records verwendet, kann die Namensauflösung an eine komplett andere Domain weitergeleitet werden. Dies bietet im Gegensatz zur Verwendung von NS-Records den Vorteil, dass die Anzahl der Indirektionen und somit die Anzahl der Entscheidungen nicht durch die Struktur des Uniform Resource Locators (URLs) vorgegeben sind.

DNS-based Request-Routing bietet die Möglichkeit, die Clients transparent an bestimmte Stellvertreterserver zu verweisen. Es sind keine Anpassungen der Clientsoftware notwendig. Die Umlenkung an die Stellvertreterserver ist transparent und wird von den Clients nicht bemerkt.
Nachteilig ist, dass der lokale DNS-Server und nicht der Client selber am Request-Routing System anfragt. Hierdurch wird bei der Suche nach dem am nächsten gelegenen Stellvertreterserver die Netzwerklokation des lokalen DNS-Servers als Referenz

herangezogen. Dies kann z. B. in großen ISP-Netzen zu falschen Entscheidungen bezüglich des am besten geeigneten Stellvertreterservers führen [Penno u. a. (2010)]. Darüber hinaus ist die Verwendung von rekursiven DNS-Anfragen nicht möglich, da hierbei die Lokation des anfragenden Clients durch die rekursiven Anfragen der DNS-Server verschleiert wird. Es ist zu jedem Zeitpunkt nur der aktuell letzte DNS-Server in der Kette für den nächsten DNS-Server sichtbar.

HTTP Redirection HTTP bietet durch die 3xx Status Codes [Fielding u. a. (1999)] im HTTP-Header die Möglichkeit, Clients zu signalisieren, dass sie ihre Anfrage an einen anderen Server erneut stellen sollen. Für die Nutzung als Request-Routing Mechanismus in CDNs hat sich der Status Code 302 Found durchgesetzt. Dieser informiert den Client, dass der Inhalt temporär unter einer anderen URL erreichbar ist. Hierdurch können Webserver implementiert werden, die die Inhaltsanfrage entgegennehmen, einen geeigneten Stellvertreterserver ermitteln und einen Redirect mittels HTTP Status Code 302 senden.

Diese Methode des Request-Routings ist einfach zu implementieren und bietet die Möglichkeit der Anfrageumleitung auf Objektebene. Der Nachteil besteht jedoch in der zusätzlich entstehenden Latenz, die durch das Senden des Redirect Headers entsteht. Der Redirect wird zudem für den Client nicht transparent durchgeführt [Barbir u. a. (2003); Pathan und Buyya (2007)].

URL Rewriting Beim URL Rewriting werden die in einem HTML-Dokument enthaltenen Verweise zu eingebetteten Objekten so manipuliert, dass sie von einem geeigneten Stellvertreterserver abgerufen werden können. Hierzu gibt es mehrere Varianten.

A priori URL Rewriting beschreibt den Mechanismus, bei dem der Inhalteanbieter bereits während der Erzeugung der Inhalte den Stellvertreterserver, von dem der eingebettete Inhalt abgerufen werden soll, festlegt. Dieser Mechanismus alleine unterstützt kein dynamisches Request-Routing. Dies kann in Kombination mit DNS-based Request-Routing erreicht werden.

On-Demand URL Rewriting beschreibt einen Mechanismus, bei dem der Inhalt modifiziert wird, sobald er vom Inhaltekonsumenten angefragt wird. Im Gegensatz zum DNS-based Request-Routing ist die Client IP-Adresse bekannt, wodurch besser geeignete Stellvertreterserver bezüglich der Lokalität des Inhaltekonsumenten gewählt werden können.

URL Rewriting Mechanismen bieten wie auch HTTP Redirections den Vorteil, dass sie auf Objektebene arbeiten und die Granularität, mit der geeignete Stellvertreterserver bestimmt werden, feiner ist. Einen Nachteil stellt der erhöhte Rechenaufwand dar, der durch das Parsen und Umschreiben der Verweise generiert wird [Barbir u. a. (2003); Hofmann und Beaumont (2005); Pathan und Buyya (2007)].

Global Server Load Balancing (GLSB) Beim Server Load Balancing (SLB) werden mehrere Webserver an einen Content Switch angeschlossen. Die Kombination aus Content Switch und Webservern bezeichnet man auch als SLB-Cluster (siehe Abbildung 2.5). Jeder SLB-Cluster besitzt eine virtuelle IP-Adresse, die dem Content Switch zugewiesen ist. Der Content Switch sammelt Statistiken und Statusinformationen der einzelnen Webserver. Aufgrund dieser Informationen wird die Last der an die virtuelle IP-Adresse gesendeten Clientanfragen auf die Webserver verteilt.
Beim Global Server Load Balancing (GSLB) existieren mehrere solcher SLB-Cluster. Sie sind global verteilt und tauschen gegenseitig Statistiken und Statusinformationen über ihren Zustand aus [Hofmann und Beaumont (2005); Pathan und Buyya (2007)].

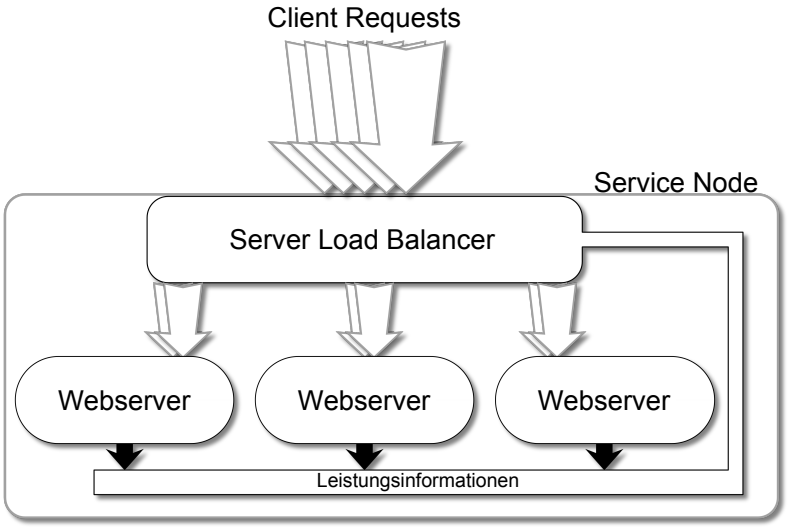


Abbildung 2.5: SLB-Cluster / GSLB Service Node

Diese Informationen werden in Hofmann und Beaumont (2005) als global awareness beschrieben. Smart authoritative DNS ist die Umsetzung von DNS-based Request-Routing im Kontext des GSLB mit dessen Hilfe die Informationen über den globalen Zustand des GSLB-Systems genutzt werden, um die Last der Inhaltsanfragen global auf die beteiligten Service Nodes zu verteilen.

Anycasting Der Anycast Mechanismus kann in zwei Ansätze unterteilt werden.
Beim IP-Anycast Ansatz [Partridge u. a. (1993)] wird jedem Stellvertreterserver dieselbe IP-Anycast-Adresse zugewiesen. Jeder Router im Netz besitzt eine Route zu dieser Anycast-Adresse. Da Router nur die Route mit der geringsten Routing-Metrik zu einer Adresse / einem Adressbereich in die aktive Routing Tabelle aufnehmen [Baker (1995)], wird die Kommunikation mit dem Stellvertreterserver aufgenommen, der die

beste Routing-Metrik ausgehend vom lokalen Router des Clients besitzt. Dies bedeutet, dass unterschiedliche Router Pakete für dieselbe IP-Adresse an unterschiedliche Stellvertreterserver senden.
Beim Application-level Anycast Ansatz [Fei u. a. (1998)] wird zunächst eine Anfrage an einen Server, den Anycast Resolver gestellt. Dieser Anycast Resolver ist für die Abbildung von Anycast Domänen Namen auf IP-Adressen zuständig. Trifft eine Anfrage für einen Anycast Domänen Namen ein, entscheidet der Anycast Resolver anhand der Daten in seiner Metrikdatenbank, welche IP-Adresse zurückgeliefert wird. Ein Vorteil des Application-level Anycast Ansatzes ist seine Flexibilität, nachteilig ist, dass der Client des Inhaltekonsumenten über eine Softwareschnittstelle für die Anfrage beim Anycast Resolver verfügen muss. Diese Art des Request-Routings ist für den Client nicht transparent [Barbir u. a. (2003); Hofmann und Beaumont (2005); Pathan und Buyya (2007)].

2.2 Content Delivery System

Das Content Delivery System ist für die Auslieferung der Inhalte an den Inhaltekonsumenten zuständig. Zur Planung des Content Delivery Systems gehört sowohl die Platzierung der Stellvertreterserver im Netzwerk (siehe Kapitel 2.2.1) als auch die Auswahl der Protokolle (siehe Kapitel 2.2.2), die bei der Auslieferung der angefragten Inhalte unterstützt werden.

2.2.1 Stellvertreterserver Platzierung

Die Stellvertreterserver eines CDNs werden an unterschiedlichen Positionen in einem oder mehreren Netzen platziert. Für diese Verteilung der Stellvertreterserver gibt es unterschiedliche Ansätze [Leighton (2009); Yin u. a. (2010)]. Sie unterscheiden sich unter anderem in ihren Eigenschaften bezüglich Betriebskosten, Managementaufwand und Proximität zum Inhaltekonsumenten. Im Folgenden werden die Ansätze Highly Distributed CDN, Big Datacenter CDN, Cloud CDN sowie Peer2Peer-Assisted CDN dargestellt.

Highly Distributed CDN Bei der Highly Distributed Architektur werden die CDN-Server direkt in den PoPs der ISPs platziert. Die Idee ist, die Inhalte möglichst nah am Inhaltekonsumenten vorzuhalten. Durch die Platzierung im PoP kann die Latenz gering gehalten werden und der Datendurchsatz des Hausanschlusses kann effektiv genutzt werden.
Zu den Problemen dieses Architekturansatzes gehört die Administration der Systeme, die durch die weite Ausdehnung zeitaufwendig und kostspielig ist. Dieser Ansatz wird von der Firma Akamai Technologies, dem aktuellen Marktführer unter den CDN-Anbietern, umgesetzt.

Big Datacenter CDN Beim Big Datacenter Ansatz werden die digitalen Inhalte in wenigen großen Rechenzentren vorgehalten. Die Standorte dieser Datacenter sind so gewählt, dass sie möglichst nah an den PoPs verschiedener großer Access Provider liegen. Hier besteht der Vorteil darin, dass die CDN-Anbieter mit wenig Aufwand und einem geringen Investment wenige Router-Hops von den Inhaltekonsumenten, aus verschiedenen Netzen, entfernt sind. Aufgrund der geringen Streuung der Stellvertreterserver sind die Verwaltungs- und Instandhaltungskosten für Big Datacenter CDNs geringer als bei Highly Distributed CDNs. Als nachteilig bleibt anzuführen, dass die Inhalte weiter vom Inhaltekonsumenten entfernt vorgehalten werden als bei der Highly Distributed Architektur. Dieser Ansatz wird unter anderem von der Firma Limelight Networks umgesetzt. Limelight Networks betreiben ein eigenes AS mit eigener Netzwerkinfrastruktur über das die Rechenzentren vernetzt sind. Der Übergang in die ISP-Netze erfolgt über Peering-Verbindungen.

Cloud CDN Bei der Cloud CDN Architektur werden virtualisierte Serverinfrastrukturen zur Auslieferung von Inhalten verwendet. Es können bereits bestehende Cloud-Dienste genutzt werden oder die CDN-Anbieter errichten eigene Clouds. Den Vorteil der Cloud CDNs bildet die einfache Skalierung sowie die bedarfsgebundene Abrechnung durch den Cloud-Anbieter. Diese Architektur ist durch die bedarfsgebundene Abrechnung gut geeignet, um Highly Distributed CDNs oder Big Datacenter CDNs bei der Auslieferung von Inhalten während Flash Crowds [Arlitt und Jin (2000)] zu unterstützen.

Peer2Peer-Assisted CDN Bei der Peer2Peer-Assisted Architektur werden Highly Distributed CDNs, Big Datacenter CDNs oder Cloud CDNs durch die Peer2Peer-Technologie unterstützt, Inhaltekonsumenten können also auch als eine Art Stellvertreterserver fungieren. Sie stellen die heruntergeladenen Inhalte anderen Inhaltekonsumenten zur Verfügung. Reine Peer2Peer-CDNs bieten aufgrund der asymmetrischen Zugangstechnik heutiger Breitbandanschlüsse nicht den ausreichenden Datendurchsatz in Richtung des Netzes (Upstream / Upload), um vollständig auf Peer2Peer basieren zu können [Leighton (2009)]. Hybride Ansätze, die die Peer2Peer-Assisted Architektur mit einer der drei zuvor genannten Architekturen verbinden, können helfen die Gesamtkosten der Bereitstellung von Inhalten zu reduzieren.

2.2.2 Protokolle

CDNs werden für die Auslieferung unterschiedlichster Inhalte genutzt. Hierzu gehören unter anderem Grafiken, Webseiten, Video on Demand und Live-Videostreams. Um die Übertragung dieser unterschiedlichen Inhaltstypen optimal ausführen zu können, müssen CDNs laut Kurose und Ross (2009) und Hofmann und Beaumont (2005) über das im Web übliche HTTP-Protokoll hinaus weitere Protokolle unterstützen. Hierzu gehören die im Folgenden dargestellten Protokolle.

Hypertext Transfer Protocol Das Hypertext Transfer Protocol (HTTP) ist ein von der IETF in RFC1945 (Version 1.0) [Berners-Lee u. a. (1996)] und in RFC2616 (Version 1.1) [Fielding u. a. (1999)] standardisiertes Protokoll zur Übertragung von Daten. Hauptsächlich wird HTTP zur Übertragung von Webseiteninhalten verwendet. Im Folgenden werden einige der Charakteristika des HTTP-Protokolls aufgeführt.

Anfrage-Rückantwort Bei HTTP handelt es sich um ein Request-Response Protokoll. Der Client stellt eine Anfrage an einen Webserver, der diese Anfrage verarbeitet und daraufhin den angeforderten Inhalt ausliefert. Es ist nicht möglich, dass der Server unaufgefordert Inhalte an den Client ausliefert. Eine Inhaltsauslieferung besteht aus dem angeforderten Inhalt sowie einem Header, über den der Server mithilfe von Statuscodes den Erfolg, den Misserfolg oder weitere Informationen zur Inhaltsanfrage zurück liefert.

Zustandslosigkeit Nachdem ein Webserver auf die HTTP-Anfrage eines Clients geantwortet hat, verwaltet er keine weiteren Informationen über die Transaktion. Alle Anfragen geschehen unabhängig von vorherigen Anfragen. Dies wird als Zustandslosigkeit bezeichnet. Hierdurch ist das HTTP-Protokoll relativ einfach zu implementieren. Diese Eigenschaft birgt jedoch auch Probleme. Bei der Verwendung von personalisierten Webseiteninhalten, wie sie beispielsweise beim Online-Banking oder E-Commerce zum Einsatz kommen, ist es wichtig, die bereits getätigten Interaktionen mit dem Webserver einem Benutzer zuordnen zu können.

Persistente Verbindung Version 1.1 des HTTP-Protokolls unterstützt persistente Verbindungen. Hierbei können mehrere Objekte über eine TCP-Verbindung angefordert und übertragen werden. In Version 1.0 muss für jede Webseite und jedes in sie eingebettete Objekt noch eine eigene TCP-Verbindung auf- und wieder abgebaut werden, wodurch bei nicht-persistenten Verbindungen ein großer Mehraufwand für die Verwaltung der Verbindungen entsteht. Ferner kann es zu gesteigerten Übertragungslatenzen kommen. Mit persistenten Verbindungen ist es möglich Request Pipelining zu nutzen. Hierbei werden Inhaltsanfragen nacheinander an den HTTP-Server gesendet, ohne auf eine Antwort auf die vorherige Anfrage zu warten. Dies verringert die Gesamtübertragungszeit.

Segmentierte Codierung Chunked Transfer Encoding ist ein Mechanismus, der es HTTP-Servern erlaubt, auf Inhaltsanfragen zu antworten, bei denen die genaue Größe der Antwort noch nicht feststeht. Die Inhaltsauslieferung wird in kleine Segmente aufgeteilt (chunks), die dann mit einem Header, in dem die Größe des jeweiligen Teils vermerkt ist, übertragen wird. Hierdurch ist es möglich, Streaminginhalte unter Verwendung des HTTP-Protokolls zu übertragen.

Real-time Transport Protocol Das Real-Time Transport Protocol (RTP) ist ein von der IETF in RFC 3550 [Schulzrinne u. a. (2003)] standardisiertes Protokoll zur Übertragung von Echtzeitdaten. TCP und UDP sind als Transportschicht Protokolle für RTP standardisiert [Lazzaro (2006)].
In den meisten Fällen kommt UDP zum Einsatz. UDP bietet den Vorteil, weniger Übertragungsaufwand zu erfordern, was der Echtzeitübertragung von Audio- / Videodaten zugutekommt. Allerdings ist die wiederholte Übertragung von Audio- / Video-Echtzeitdaten in der Regel nicht praktikabel, da die verspätete Ankunft nicht mehr benötigter Pakete Bandbreite verschwendet. Bei der Verwendung von TCP kommt es zum Head-of-Line Blocking. Bereits eingetroffene Daten werden hierbei nicht an die Applikation weitergereicht, da zuvor versendete Daten während der Übertragung verloren gegangen sind. Erst nach Eintreffen der zuvor verloren gegangenen Pakete werden alle Daten bis zum nächsten noch nicht eingetroffenen Paket ausgeliefert. Dies führt dazu, dass bereits erfolgreich übertragene Daten zu spät an die Applikation ausgeliefert werden und somit die Qualität des Services gemindert wird.
RTP sorgt dafür, dass Pakete, die nicht in der richtigen Reihenfolge beim Client ankommen, sortiert und gegebenenfalls verworfen werden.
RTP ist für die Unicast- sowie die Multicastübertagung geeignet. Es unterstützt unterschiedliche Nutzlastformate, die im RFC 3551 [Schulzrinne und Casner (2003)] in einzelnen RTP Audio Video Profilen (RTP/AVP) beschrieben sind. Für die Übertragung jedes einzelnen Streams wird eine eigene RTP-Verbindung aufgebaut. Für einen Videostream, zu dem ein zugehöriger Audiostream vorliegt, ergeben sich somit zwei RTP-Sitzungen. Für diese einzeln transportierten Streams stellt RTP Mittel zur Synchronisierung zur Verfügung.

Real-Time Transport Control Protocol Das Real-Time Transport Control Protocol (RTCP) ist ein ebenfalls in RFC 3550 [Schulzrinne u. a. (2003)] standardisiertes Protokoll, welches zusammen mit RTP verwendet wird.
RTCP ist für die Identifizierung von Sitzungsteilnehmern und dem Einsammeln von sitzungsbezogenen Statistiken zuständig. Die zu übermittelnden Nutzdaten werden jedoch mithilfe von RTP übertragen. Sowohl der Server als auch die Clients, die an einer RTP-Sitzung teilnehmen, generieren RTCP-Pakete. Diese periodisch versendeten Pakete umfassen folgende Statistiken:

- Anzahl gesendeter Pakete
- Anzahl verlorener Pakete
- Informationen zu Jittern
- Informationen zur Round Trip Time (RTT)

Ob und wie die hierdurch gewonnenen Informationen genutzt werden, wird in RFC

3550 nicht spezifiziert. Der Sender des Streams kann diese Informationen beispielsweise verwenden, um die Qualität des ausgesendeten Streams anzupassen [Schulzrinne u. a. (2003)].

Real Time Streaming Protocol Das Real-Time Streaming Protocol (RTSP) spezifiziert im RFC 2326 [Schulzrinne u. a. (1998)] standardisiert Wiedergabe-Kontrollfunktionen. RTSP definiert ein Protokoll mit dem Streaming-Server gesteuert werden können. Die Funktionen umfassen unter anderem Abspielen, Pausieren, Vorspringen, Zurückspringen und Stoppen der Multimediainhalte, also Funktionen wie sie auch DVD-Player oder Videorekorder bieten (siehe Abbildung 2.6).
RTSP-Pakete werden unabhängig von den Nutzdaten in einer eigenen Verbindung ausgetauscht. Für diese Verbindung kommt entweder das TCP- oder das UDP-Protokoll zum Einsatz. Die Verwendung beider Protokolle ist in Schulzrinne u. a. (1998) standardisiert.

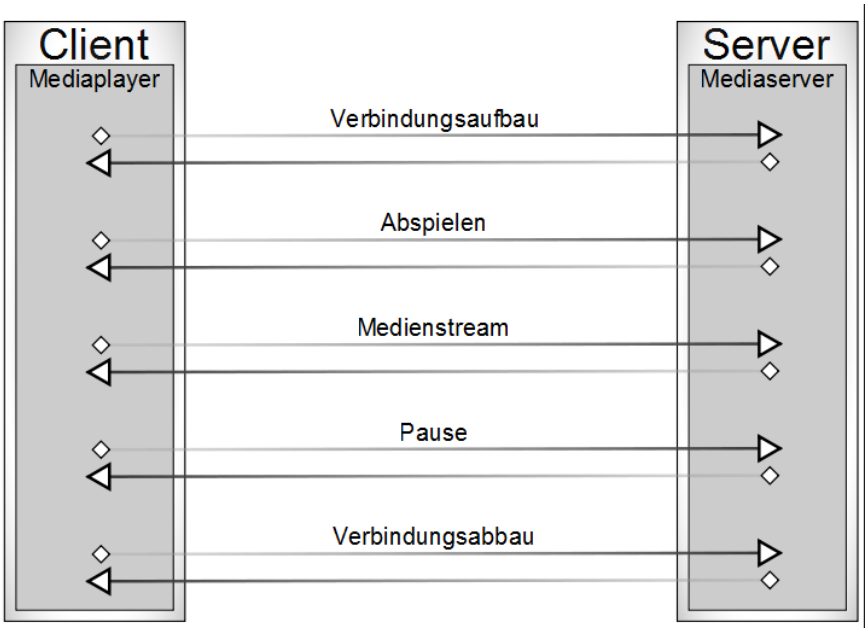


Abbildung 2.6: Interaktion zwischen Client und Server mittels RTSP nach O'Driscoll (2008)

Real Time Messaging Protocol Bei dem Real-Time Messaging Protocol (RTMP) [Adobe Systems (2009)] handelt es sich um ein proprietäres Protokoll der Firma Adobe Systems Inc. Adobe machte die Spezifikation des Protokolls jedoch im April 2009 öffentlich zugänglich. RTMP wird vor allem zur Übertragung von Adobe Flash Videos eingesetzt. Als Transportschichtprotokoll kommt TCP zum Einsatz. Im Gegensatz zu RTP / RTCP

werden alle RTMP-Pakete in einer für die Gesamtdauer der Übertragung bestehenden TCP-Verbindung gebündelt. Hierzu werden Kanal-IDs herangezogen, die für jedes Paket identifizieren, zu welchem Kanal es gehört. Es werden unterschiedliche Kanäle genutzt:

- RTMP Remote Procedure Calls
- Videostream-Daten
- Audiostream-Daten
- Kontrollnachrichten

Es existieren fünf Varianten des RTMP-Protokolls [Adobe Systems]:

plain RTMP RTMP-Pakete werden über eine TCP-Verbindung unter Nutzung des Ports 1935 übertragen.

RTMPT RTMP-Pakete werden in validen HTTP-Paketen transportiert. Die RTMP-Pakete können so durch Firewalls getunnelt werden.

RTMPS RTMP-Pakete werden über eine SSL gesicherte Verbindung übertragen.

RTMPE RTMPE ist eine verbesserte Variante zur verschlüsselten Übertragung, bei der zusätzlich kein Zertifikatemanagement verwendet wird, wie es bei SSL der Fall ist.

RTMPTE RTMPE-Pakete werden mithilfe von HTTP getunnelt.

2.3 Distribution System

Das Distribution System ist für die effiziente Verteilung der Inhalte innerhalb eines CDNs sowie für die Konsistenz der zwischengespeicherten Inhalte zuständig. Nachdem der Inhalteanbieter die Inhalte identifiziert hat, die durch das CDN verfügbar gemacht werden sollen (siehe Kapitel 2.3.1), werden diese in das CDN eingebracht und auf geeignete Stellvertreterserver repliziert (siehe Kapitel 2.3.2).

2.3.1 Auswahl von Inhalten

Der Inhalteanbieter muss festlegen, welche Inhalte aus dem CDN an den Inhaltekonsument ausgeliefert werden sollen und welche Inhaltsanfragen vom Herkunftsserver bedient werden sollen.
Eine Webseite kann komplett aus einem CDN heraus zur Verfügung gestellt werden, was den Betrieb eines eigenständigen Herkunftsservers überflüssig macht. Es können auch ausgewählte Teile einer Webseite durch ein CDN zur Verfügung gestellt werden, so dass

dynamisch generierte Inhalte vom Herkunftsserver und die statischen Inhalte, wie beispielsweise Grafiken, aus einem CDN ausgeliefert werden [Pathan und Buyya (2007)].
Wenn eine Webseite vollständig aus einem CDN heraus betrieben werden soll, konfiguriert der Inhalteanbieter seine DNS-Server so, dass die Anfragen direkt an den Request-Routing DNS-Server des CDNs geleitet werden. Dieser leitet den Client an einen Stellvertreterserver, der die Inhaltsanfragen bedient. Dieses Vorgehen stellt eine einfache Methode zum Hosting von statischen Inhalten zur Verfügung.
Nachteile birgt dieser Ansatz, wenn es um die Auslieferung dynamischer Inhalte geht. Die Schwierigkeiten bestehen darin, auf den Stellvertreterserver die jeweils aktuellste Version der Inhalte vorzuhalten. In vielen Fällen ist dies nicht möglich oder nicht gewollt, so dass die Daten von einem zentralen Server ausgeliefert werden müssen. Als Beispiel sind diesbezüglich Anwendungen mit gesteigerten Sicherheitsanforderungen, wie sie Onlinebanking oder E-Commerce darstellen, zu nennen.
Beim Ansatz, nur Teile einer Website aus einem CDN heraus zur Verfügung zu stellen, wird die HTML-Seite vom Herkunftsserver generiert und ausgeliefert. Eingebettete Objekte wie Grafiken und Videos werden durch das CDN zur Verfügung gestellt. Dieses verringert die Last auf dem Herkunftsserver und ist praktikabel, da sich eingebettete Objekte nicht so häufig ändern wie die Webseiten, in die sie eingebettet sind [Pathan und Buyya (2007)].
Zur Auswahl der Inhalte, die aus dem CDN heraus ausgeliefert werden, gibt es unterschiedliche Strategien. Hierzu gehört die erfahrungswertbasierte-Strategie, bei der Webseitenadministratoren die Objekte manuell und aufgrund eigener gesammelter Erfahrungen auswählen. Bei der popularitätsbasierten Strategie werden die beliebtesten Objekte auf die Stellvertreterserver übertragen. Nachteilig hierbei ist, dass Statistiken gepflegt werden müssen. Da die Beliebtheit von Inhalten schwankt, müssen die Statistiken periodisch geprüft und aktualisiert werden. Dies kann z. B. durch das Auswerten von Logdateien passieren, hierzu ist jedoch zusätzliche Rechenkapazität notwendig.

2.3.2 Verteilung von Inhalten

Die Inhalte können wie in Pathan und Buyya (2007) und Buyya u. a. (2006) beschrieben über drei verschiedene Mechanismen auf die Stellvertreterserver gelangen.
Zum einen rufen die Stellvertreterserver selber die Inhalte vom Herkunftsserver ab. Dabei interagieren sie nicht miteinander. Bekommt ein Server eine Anfrage, die er nicht aus dem lokalen Cache beantworten kann, wendet er sich an den Herkunftsserver und fordert bei diesem die benötigten Inhalte an. Dieses Verhalten spiegelt die Arbeitsweise eines klassischen Proxyservers wieder.
Ein weiterer Mechanismus verhält sich wie der zuvor beschriebene, die Inhalte werden jedoch, wenn sie nicht im lokalen Cache des Stellvertreterservers vorliegen, von einem nahe gelegenen Stellvertreterserver geladen.

Bei dem dritten Mechanismus können die Inhalte vom Herkunftsserver aus auf die Stellvertreterserver ausgebracht (gepushed) werden. Die Stellvertreterserver interagieren, um sowohl die Replikations- als auch die Update-Kosten möglichst gering zu halten.

2.4 Accounting System

Die Accounting Infrastruktur sammelt Informationen des Request-Routing, Delivery und des Distribution Systems. Die Informationen werden in Logdateien auf den einzelnen Systemen geschrieben und vom Accounting System zur Auswertung aggregiert. Zur Erhebung der Informationen werden bekannte Protokolle wie FTP und SNMP eingesetzt.
Die Informationen werden zur Abrechnung der zur Verfügung gestellten Dienstleistung verwendet. CDN-Administratoren sind mit Hilfe des Accounting Systems in der Lage, einen Überblick über den Status des CDNs zu bekommen. Des Weiteren werden diese Informationen den Inhalteanbietern zur Verfügung gestellt. Sie haben auf diese Weise Zugriff auf die Verkehrsberichte und die Zugriffsstatistiken ihrer Inhalte [Vakali und Pallis (2003)].

Kapitel 3

Video on Demand

Video on Demand beschreibt ein Konzept, das Videoinhalte auf Anfrage ausliefert. Es gibt verschiedene Ausprägungen von VoD (siehe Kapitel 3.2). Des Weiteren existieren unterschiedliche Arten der Übertragung von Videodaten, die in Kapitel 3.1 dargestellt werden.

3.1 Übertragung

Es gibt verschiedene Methoden, um Videoinhalte in einem IP-Netzwerk zum Client bzw. dem Wiedergabegerät zu übertragen. Die Methoden unterscheiden sich in der Effizienz der Datenübertragung in bestimmten Einsatzkontexten. In den folgenden Abschnitten werden einige der möglichen Übertragungsmethoden erläutert.

Unicast Bei der Unicastübertragung wird der Videostream an jedes VoD-Endgerät einzeln übertragen (One-to-One). Die Bandbreitenbelastung des Streamingservers steigt proportional mit der Anzahl der Clients.

Broadcast Bei der Broadcastübertragung wird der Videostream an alle an das Netzwerk angeschlossenen Geräte übertragen. Hierbei ist es irrelevant, ob der Stream von diesem Gerät angefordert wurde oder nicht. Dieser Umstand führt in Netzwerken zu Problemen, da die Ressourcen der angeschlossenen Geräte durch nicht benötigte Pakete stark belastet werden. Diese Belastung kann auch zu Überlastungen und somit zu einem Denial of Service einzelner wenig performanter Geräte führen.
Ein weiteres Problem besteht in der Belastung der Netzwerkinfrastruktur, deren Kapazität bei der Übertragung an alle angeschlossenen Gerät verschwendet wird.

Multicast Multicasting [Deering (1989)] ist eine Technik für die One-To-Many Datenübertragung in IP-Netzwerken. Bei Multicasting werden die zu versendenden Daten lediglich einmal vom Sender übertragen. Auf dem Weg durch das Netz werden die IP-Pakete dupliziert. Hierzu wird ein Shortest Path Tree (SPT) aufgebaut an dessen Wurzel sich

die Multicastquelle befindet. Pakete, die an die Multicastempfänger gesendet werden, werden an den Gabelungspunkten des Baumes dupliziert und an die Clients weitergeleitet (siehe Abbildung 3.1).

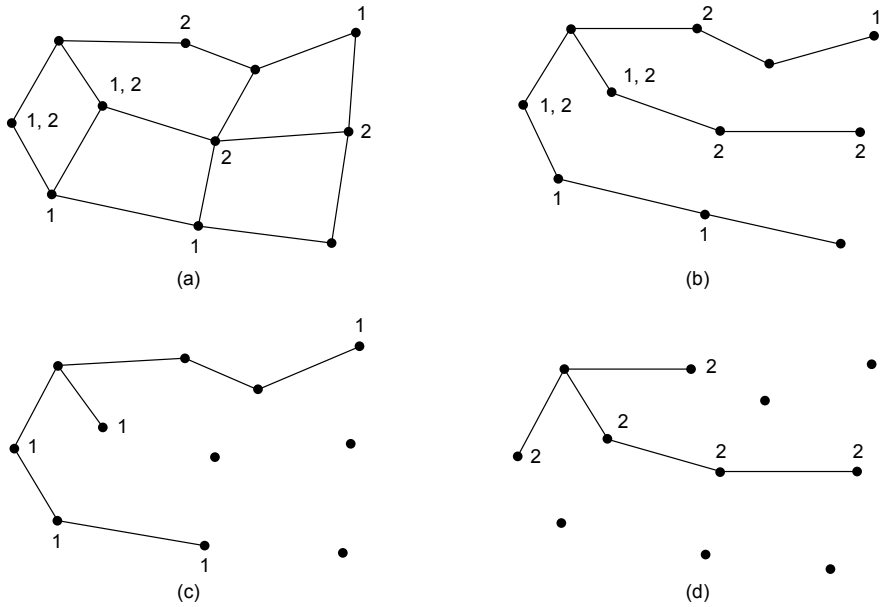


Abbildung 3.1: (a) Netzwerk (b) Spannbaum vom linken Router aus (c) SPT für Multicastgruppe 1 (d) SPT für Multicastgruppe 2 [Tanenbaum (2002)]

3.2 Ausprägungen

Die Auslieferung von Videoinhalten kann auf unterschiedliche Art und Weise geschehen, die ebenso unterschiedliche Eigenschaften, Vorzüge und Nachteile besitzen. Die geläufigsten Ausprägungen der VoD-Auslieferung werden nachfolgend aufgeführt.

Near-VoD Bei Near-VoD startet ein Videostream auf verschiedenen Kanälen zu unterschiedlichen Zeitpunkten. Ein Inhaltekonsument, der das VoD-Angebot ansehen möchte, tritt einem Kanal, der sogenannten Multicastgruppe, bei und kann somit den ausgesendeten Stream empfangen. Dies bietet dem Inhalteanbieter die Möglichkeit, VoD ressourcenschonend anzubieten, da mithilfe der Multicastübertragung nicht für jeden Inhaltekonsument ein eigener Stream übertragen werden muss.

Internet-VoD Bei Internet-VoD wird das öffentliche Internet für die Verbreitung der Videoinhalten genutzt. Der Videostream kommt hierbei aus Netzen, die außerhalb der Autorität des lokalen Netzanbieters liegen. Hierdurch kann allerdings die Quality of Service

(QoS) nicht sichergestellt werden. Der lokale Netzanbieter überträgt den Datenverkehr nicht mit einer erhöhten Priorität, so dass es zu störenden Jittern und einer erhöhten Latenz kommen kann. Beispiele für Internet-VoD Angebote sind Videoportale wie YouTube oder Maxdome.

Movie on Demand Bei Movie on Demand wird der Videoinhalt auf Anfrage an den Client gestreamt. Jeder Stream wird einzeln an die Inhaltekonsumenten übertragen. Es werden somit große Ansprüche an die Netzwerk- sowie Serverinfrastruktur des Inhalteanbieters gestellt. Die Multicastübertragung lässt sich in diesem Fall nicht nutzen, da Inhaltekonsumenten zu jedem beliebigen Zeitpunkt mit dem Empfang des Streams beginnen und ihn jederzeit pausieren oder spulen können. Ein Beispiel für Movie on Demand Angebote ist das Produkt T-Home der Deutschen Telekom.

Push-VoD Bei Push-VoD werden die Videoinhalte im Vorfeld vom VoD-Server auf VoD-Endgeräte übertragen. Die Idee hinter diesem Konzept ist die Vermeidung von Bandbreitenengpässen. Der Videoinhalt wird zu Zeiten übertragen, zu denen das Netz des Anbieters relativ unbelastet ist. Dies führt beim Netzanbieter zu einer besseren Auslastung der Netzwerkinfrastruktur. Des Weiteren kann zur Übertragung der Videoinhalte in den Speicher des VoD-Endgerätes die Multicastübertragung verwendet werden. Hierdurch werden die Netzwerkbelastung sowie die Belastung des VoD-Servers zusätzlich verringert, da die Videodatei vom Videoserver nur ein einziges Mal versendet werden muss, um von allen mit Strom versorgten und ans Netz angeschlossenen VoD-Endgeräten empfangen zu werden. Die Inhaltekonsumenten können direkt mit der Wiedergabe der Inhalte beginnen, da die Videoinhalte sofort verfügbar sind.
Der Umfang der auf diese Weise zur Verfügung gestellten Inhalte ist unter anderem durch die Größe des internen Speichers der VoD-Endgeräte beschränkt. Um diese Limitierung zu umgehen, ist es möglich, die aktuell populären Videoinhalte auf den VoD-Endgeräten vorzuhalten und für weniger populäre Videoinhalte die Movie on Demand Auslieferungsmethode zu verwenden. So kann dem Inhaltekonsumenten in jeder Situation der volle Dienstumfang des VoD-Angebots zur Verfügung gestellt werden.

Kapitel 4

Problematisierung & Konzept

ISPs müssen ständig steigende Bandbreitenanforderungen befriedigen. Die durch die Inhalteanbieter bereitgestellten Inhalte vervielfachen ihren Umfang und ihr Volumen, da die Auflösung von Videos und Bildern durch die zunehmende Verbreitung von High Definition (HD)-Inhalten steigen. Die Inhaltekonsumenten erwarten, dass ihre ISPs ihnen ein gutes Nutzungserlebnis garantieren. Die geforderten Eigenschaften umfassen die Verfügbarkeit des Anschlusses, hohe Übertragungsrate und kurze Reaktionszeiten aller im Internet verfügbaren Inhalte. Langes Puffern und Unterbrechungen von Videoinhalten werden zunehmend inakzeptabel. ISPs müssen ihre Netze somit ständig ausbauen und optimieren, um den steigenden Anforderungen nachzukommen.
Wie in Tabelle 4.1 dargestellt, bieten die Internetdienste (Breitband Internetanschluss, Transitverkehr) die geringsten Erlöse pro übertragenem Megabyte. Zu erklären ist dieser Umstand unter anderem durch die Einführung von Pauschaltarifen, sogenannten Flat-Rates, bei der Bereitstellung von Internetzugängen. Durch die voranschreitende Verbilligung solcher Angebote im Zusammenhang mit steigenden Übertragungsraten erwarten Kunden, dass ihre Anbindung an das Internet schneller und gleichzeitig billiger wird.
Da durch das einfache Bereitstellen von Internetanschlüssen keine Kundenbindung entsteht, sind Kunden nicht an einen bestimmten ISP gebunden und können ohne Nachteile zu einem anderen, billigeren Anbieter von Internetanschlüssen wechseln. Jeder beliebige ISP ist in der Lage, die Standardleistung, eine Anbindung an das Internet, in gleichem Maße zu erbringen. Um die Kunden stärker an sich zu binden, ist der ISP daran interessiert, sich vom Wettbewerb abzusetzen.
Als einer dieser sogenannten Mehrwertdienste ist es dem ISP möglich seinen Kunden ein VoD-Angebot zur Verfügung stellen. Kunden erhalten Zugriff auf eine Onlinevideothek, aus der sie sich auf Anfrage, ohne Wartezeit und ohne das Haus verlassen zu müssen, Videofilme „ausleihen“ und ansehen können. Details zum Konzept eines vom ISP betriebenen VoD-Angebots werden in Kapitel 4.1 dargestellt.
Des Weiteren transportieren ISPs Inhalte für CDN- und Inhalteanbieter zu ihren Kunden. Sie

Dienstleistung	Erlös pro MB
SMS	1.000,00 $
Mobile Anrufe	1,00 $
Festnetz Anrufe	0,10 $
Breitband Internetanschluss	0,01 $
Transitverkehr	0,0001 $

Tabelle 4.1: Erlös pro übertragenem Megabyte [Odlyzko (2009)]

stellen also durch das Transportieren der Inhalte allen Beteiligten eine Leistung zur Verfügung. Von ihren Kunden werden sie für die Bereitstellung eines Internetanschlusses und den Transport der von ihnen angeforderten oder versendeten Inhalte bezahlt.
Die Inhalteanbieter nutzen die Infrastruktur der ISPs, um ihre Inhalte an ihre Kunden, die Inhaltekonsumenten, auszuliefern. Hierzu werden im Falle eines Big Datacenter CDNs Peerings zwischen dem Netz des Inhalteanbieters und dem ISP eingerichtet. Faratin u. a. (2007) beschreiben für diesen Fall zwei spezielle Arten des Peerings. Zum einen die Partial Transit Methode, bei der der ISP dem Inhalteanbieter Zugang zu ausgewählten Netzbereichen bzw. den Inhaltekonsumenten der entsprechenden Netzbereichen gewährt. Zum anderen das Paid Peering, bei dem der Inhalteanbieter für die über die Peering-Verbindung versendeten Daten zahlt. Um die über die Peering-Verbindung ausgetauschten Inhalte im Netz transportieren zu können, ist es für den ISP nötig, seine Netzwerkinfrastruktur stetig auszubauen. Dies stellt für die ISPs einen erheblichen Investitionsaufwand in Form von Hardware sowie Betriebs- und Instandhaltungskosten dar. Die steigenden Inhaltsgrößen bergen nicht nur Anforderungen an die Backboneinfrastrukturen der ISPs. Die Transit- und Peeringschnittstellen zu anderen ISPs und Carriern müssen ebenso ausgebaut werden. Hierzu muss in die Anbindung sowie in benötigte Stellfläche für Netzwerkequipment in den Internet Exchange Points (IXPs) investiert werden.
Die Belastung der Backbones und IXPs können ISPs durch die Einrichtung von Stellvertreterserver in unterschiedlichen Lokationen ihrer Netze verringern. Die Stellvertreterserver fungieren als Zwischenspeicher für die abgerufenen Inhalte. Hierdurch können, je nach Platzierung der Stellvertreterserver, Mehrfachübertragungen desselben Inhalts über die IXPs und / oder den Backbone vermieden werden. Details zu dieser Verkehrsoptimierung werden in Kapitel 4.2 dargestellt.

4.1 Mehrwertdienste

ISPs sind in der Lage, ihren Kunden durch die Bereitstellung eines VoD-Angebots einen Mehrwertdienst zur Verfügung zu stellen. Kunden können Videos direkt auf Anfrage anschauen. Sie können die Videoinhalte rund um die Uhr abrufen und sind somit nicht an

Öffnungszeiten oder die Auswahl der lokalen Videotheken gebunden. Des Weiteren entfallen für den Inhaltekonsumenten die Wege zur Videothek und das Risiko, das gewünschte Titel vergriffen sind. Bei VoD-Angeboten ist es ebenso ausgeschlossen, dass Kunden Überziehungsgebühren zahlen müssen oder für den Defekt von Datenträgern haftbar gemacht werden, da kein physikalisches Medium verliehen wird, sondern lediglich Zugriffsrechte gewährt werden. Inhaltekonsumenten wählen aus dem VoD-Angebot des ISPs ein Video, welches sie anschauen möchten. Dieses Video wird unmittelbar per Stream an den Kunden übertragen, der dieses direkt wiedergeben kann.
ISPs können den Vorteil nutzen, dass die Inhalte in ihrem eigenen Netz ausgeliefert werden. Sie sind hierdurch in der Lage, die Qualität der Datenübertragung positiv zu beeinflussen, wodurch sie den Inhaltekonsumenten eine höhere Servicequalität garantieren können. Zur Priorisierung der Datenübertragung dienen dem ISP die im Netzwerkumfeld bereits etablierten QoS-Mechanismen (siehe Marchese (2007)). Die Nutzung solch priorisierter Datenübertragungen bleibt den einfachen Inhalteanbietern und globalen CDN-Providern im Normalfall verwehrt, da diese durch die administrative Autorität der jeweiligen Netze gewährt werden muss. Standardmäßig werden alle QoS-Parameter von eintreffenden Paketen an der Netzwerkgrenze überschrieben, sie können jedoch auch, abhängig von den Vereinbarungen und Verträgen der beiden beteiligten Peering-Partner, beibehalten oder umgeschrieben werden.
An dieser Stelle bleibt jedoch anzumerken, dass die unterschiedliche Behandlung der Datenübertragungen einen Einfluss auf die Netzneutralität [Crowcroft (2007)] nimmt. Unter der Netzneutralität wird eine neutrale Datenübertragung verstanden, bei der der Datenverkehr unabhängig von Herkunft und Ziel eine gleiche Behandlung erfährt.
Für die Einrichtung eines VoD-Angebots durch einen ISP ergeben sich unterschiedliche Möglichkeiten, die sich durch die Platzierung und die Übertragungsarten unterscheiden. Im Folgenden werden drei dieser Möglichkeiten detailliert dargestellt.

4.1.1 Ansätze

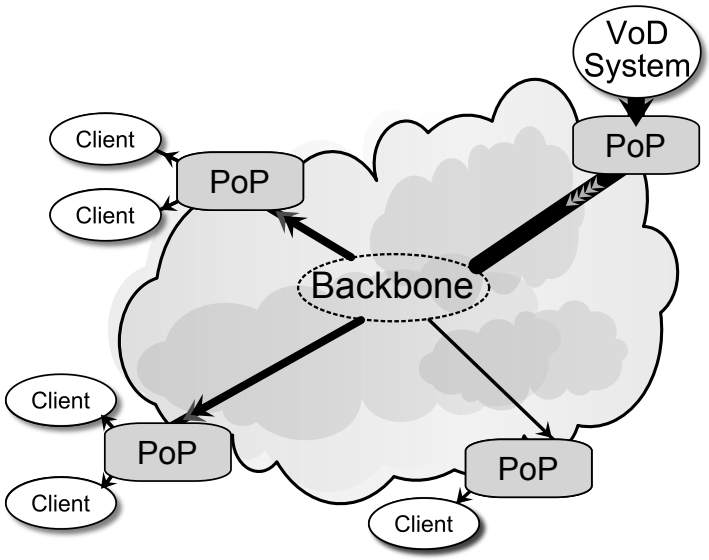
ISP near VoD Der ISP überträgt die Videostreams innerhalb seines Netzes per Multicast. Der Inhaltekonsument sucht sich zunächst das gewünschte Video und die gewünschte Startzeit aus dem verfügbaren Angebot aus. Anschließend tritt der Client der Multicastgruppe, in der das Video zur gewünschten Zeit gestreamt wird, bei. Er empfängt nun die zum Videostream gehörigen Datenpakete.
Vorteilhaft an der Multicastübertragung verglichen mit der Unicastübertragung ist der verringerte Bandbreitenbedarf bei der Versorgung mehrerer Inhaltekonsumenten mit demselben Inhalt.
Nachteilig ist anzuführen, dass den Inhaltekonsumenten durch die Verwendung der Multicastübertragung nicht die Möglichkeit geboten wird, das Video zu pausieren oder

zu spulen. Dies liegt darin begründet, dass die Quelle nur einen Videostream für alle Inhaltekonsumenten aussendet. Des Weiteren ist es nötig, dass alle Netzwerkelemente auf dem Weg von der Quelle zum Client des Inhaltekonsumenten multicastfähig sind.

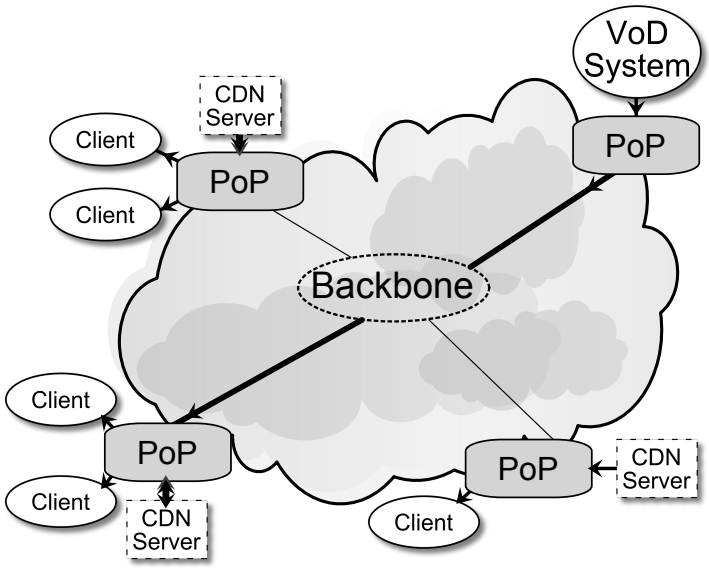
Zentralisiertes ISP Movie on Demand Der ISP überträgt Videostreams in seinem Netz per Unicast. Die Inhaltekonsumenten können aus dem VoD-Angebot die gewünschten Videoinhalte auswählen. Diese Videos werden anschließend per Unicast an jeden Inhaltekonsument einzeln versendet. Vorgehalten werden die Videoinhalte an einer zentralen Stelle im ISP Netz, der VoD-Bibliothek.
Zu den Implikationen dieses Ansatzes gehört, dass alle Streams von diesem zentralen Punkt aus den Backbone traversieren müssen, um zu den Clients zu gelangen. Es ist hervorzuheben, dass sich der zentralisierte Ansatz somit erheblich auf den Bandbreitenbedarf im Backbone des ISP-Netzes auswirkt. Da oft dieselben Videos, wenn auch zeitversetzt, von den Clients angefordert werden, wird die Backbonekapazität durch mehrfache Übertragung derselben Inhalte belegt. Vorteilhaft für die Kunden ist bei der Unicastübertragung, dass der Videostream zu jedem beliebigen Zeitpunkt starten kann. Sie haben hierdurch die Möglichkeit, den Stream an jeder beliebigen Stelle zu pausieren oder zu spulen, ohne das andere Inhaltekonsumenten hiervon betroffen sind. Der ISP benötigt für diese Art der Übertragung keine multicastfähige Netzwerkinfrastruktur.

CDN basiertes ISP Movie on Demand Der ISP überträgt die Videostreams zu den Clients per Unicast. Wie beim Zentralisierten ISP Movie on Demand können die Inhaltekonsumenten die gewünschten Videos aus dem VoD-Angebot wählen, der Videostream wird dann umgehend per Unicast an den Client ausgeliefert. Der Unterschied zwischen den beiden Ansätzen besteht in der Lokation, von der aus die Videoinhalte an den Inhaltekonsumenten ausgeliefert werden. Beim CDN basierten ISP Movie on Demand werden die Videostreams nicht von einer zentralen Stelle ausgeliefert. Es werden Stellvertreterserver in geeigneten Lokationen genutzt, auf die die stärker frequentierten Inhalte repliziert werden. Hierdurch wird vermieden, dass nahe der zentralen VoD-Bibliothek Bandbreitenengpässe entstehen können. Die Verteilung der Stellvertreterserver im Netz folgt dem Highly Distributed CDN Ansatz (siehe Kapitel 2.2.1). Hierdurch können die Inhalte nah am Client vorgehalten werden, was die Latenz der Verbindung gering hält und die Backbonebandbreite schont.

Abbildung 4.1 soll den unterschiedlichen Bandbreitenbedarf der drei Ansätze verdeutlichen. Darüber hinaus werden in Tabelle 4.2 die drei Ansätze ISP near VoD, Zentralisiertes ISP Movie on Demand und CDN basiertes ISP Movie on Demand noch einmal zusammengefasst und bezüglich ihrer weiteren Eigenschaften bewertet. Hieraus ist zu entnehmen, dass der CDN basierte ISP Movie on Demand Ansatz insgesamt am besten abschneidet. Die



(a) ISP near VoD / Zentralisiertes ISP Movie on Demand



(b) CDN basiertes ISP Movie on Demand

Abbildung 4.1: Bandbreitenbedarf im Backbone bei unterschiedlichen VoD-Ansätzen

Backbonebandbreite wird geschont, da die Videostreams aus den nahe gelegenen PoPs ausgeliefert werden. Die hieraus resultierende verringerte Anzahl der an der Übertragung beteiligten Netzwerkelemente beeinflusst die Qualität des Dienstes positiv, da die Pakete durch eine geringere Anzahl an Übertragungsqueues wandern. Es wird nicht nur die Latenz

	ISP near VoD	Zentralisiertes ISP Movie on Demand	CDN basiertes ISP Movie on Demand
Backbone Belastung	+	-	++
Jitter, Delay & Paketverlust Anfälligkeit	-	-	++
On-Demand Fähigkeit	-	++	++
Übertragungsart	Multicast	Unicast	Unicast
Skalierbarkeit (Anzahl Clients)	++	-	++
Geringer Managementaufwand	+	++	-

Tabelle 4.2: Bewertung der vorgestellten VoD-Ansätze

verringert, auch die Möglichkeit des Auftretens von Jittern und der Verlust von Paketen wird hierdurch gemindert. Durch die Unicastübertragung wird keine multicastfähige Netzwerkinfrastruktur benötigt. Die Inhaltekonsumenten sind in der Lage, die Videostreams zu pausieren oder zu spulen. Durch die verteilte CDN-Struktur ist es relativ einfach möglich, weitere Stellvertreterserver in Betrieb zu nehmen. Dies unterstützt die Skalierbarkeit des Ansatzes. Die Betriebs- und Anschaffungskosten steigen jedoch bei der Einbindung vieler verteilter Stellvertreterserver. Im Gegenzug verringert sich die Last im Backbone, was wiederum zu Investitionseinsparungen oder dem Hinauszögern eines Backboneausbaus führt.

4.1.2 Design

Um einen VoD-Dienst auf Basis eines CDNs einzurichten, sind die einzelnen Komponenten eines CDNs zu berücksichtigen und möglichst effizient im Gesamtsystem zu platzieren. Des Weiteren muss für jede Aufgabe die passende Technologie bzw. der am besten geeignete Ansatz gewählt werden. In den folgenden Abschnitten werden die einzelnen Technologien und Ansätze zum Aufbau eines ISP betriebenen VoD-Systems dargestellt.

Elemente

VoD-Katalog Der VoD-Katalog stellt dem Inhaltekonsumenten eine Übersicht der durch den VoD-Dienst verfügbaren Videos zur Verfügung. Die Inhaltekonsumenten sind in der Lage, Informationen wie sie in Programmzeitschriften oder im Elektronischen Programmführer (EPG) enthalten sind, einzusehen. Die im VoD-Katalog angebotenen Informationen können diese informationen in ihrer Quantität noch übertreffen. Es stehen beispielsweise Videos mit Filmvorschauen und detaillierte Informationen zu den Schauspielern zur Verfügung.
Über den VoD-Katalog wird der vom Inhaltekonsumenten gewünschte Videoinhalt ausgewählt und das Zugriffsrecht hierfür wird erworben. Um keine Ausfälle

des VoD-Katalogdienstes hinnehmen zu müssen, wird dieser als SLB-Cluster zur Verfügung gestellt, wodurch der Service auch bei Ausfall einzelner VoD-Katalog-Nodes noch in vollem Umfang zur Verfügung steht.

Stellvertreterserver Die Stellvertreterserver sind in den einzelnen PoPs des ISPs untergebracht. Sie sind für die Auslieferung der populärsten Filme und Filmvorschauen zuständig, da die Speicherkapazität dieser Systeme geringer ist als sie die VoD-Bibliothek aufweist. Alle anderen, aktuell nicht stark angeforderten Videos, werden aus den VoD-Bibliotheken ausgeliefert.

VoD-Bibliothek In der VoD-Bibliothek werden alle im VoD-Katalog verfügbaren Videoinhalte vorgehalten. Es besteht eine hohe Anforderung an die Verfügbarkeit der VoD-Bibliothek, da bei einem Ausfall der VoD-Bibliothek das gesamte VoD-Angebot beeinträchtigt ist. Aus diesem Grund wird die Bibliothek durch die Verwendung eines SLB-Clusters realisiert. Dies führt dazu, dass Ausfälle einzelner Bibliotheks-Nodes keine Auswirkung auf die Gesamtverfügbarkeit des VoD-Angebots haben. In einer weiteren Ausbaustufe ist die Verwendung des GSLB-Ansatzes (siehe Kapitel 2.1.3) möglich, wodurch die Backbonebelastung aufgeteilt und die Ausfallsicherheit weiter erhöht wird.
Die VoD-Bibliothek ist wie die Stellvertreterserver in der Lage, die Inhaltsanfragen der Inhaltekonsumenten zu bedienen. Sie ist für die Auslieferung aller Videoinhalte zuständig, die aktuell einer geringen Anfragehäufigkeit unterliegen. Stark nachgefragte Videos werden hingegen an die Stellvertreterserver übertragen und dort vorgehalten.

Request-Routing System Als Request-Routing Mechanismus kommt das URL Rewriting Verfahren zum Einsatz. Hierdurch ist die IP-Adresse eines jeden Clients bekannt. Beim DNS-based Redirect wird bei der rekursiven Namensauflösung die IP-Adresse des Clients durch die des Caching DNS Servers „maskiert“ (siehe Kapitel 2.1.3). Auch beim GSLB-Ansatz wird das DNS-based Redirection eingesetzt (siehe Kapitel 2.1.3), somit ist auch dieser Ansatz kein geeignetes Mittel für das Request-Routing. Die Umleitung mittels HTTP-Headern (HTTP-Redirection) wird nicht transparent durchgeführt und indiziert einen weiteren Request / Response Zyklus (siehe Kapitel 2.1.3).
All dies lässt sich mit dem URLs Rewriting Ansatz vermeiden, da die URLs hier bereits bei der ersten Auslieferung an den Client angepasst werden (siehe Kapitel 2.1.3). Der VoD-Katalog liefert somit jedem Client direkt die URL, mithilfe der der Client die Videoinhalte von dem am besten geeigneten Server abrufen kann. Somit stellt das URL Rewriting für die Bereitstellung eines ISP betriebenen VoD-Dienstes die beste Option dar.

Der Ablauf des Request-Routings findet wie folgt statt. Die VoD-Katalog-SLB-Cluster haben eine Verbindung zum Request-Routing System, über die Informationen des am besten geeigneten Stellvertreterservers oder den am besten erreichbaren VoD-Bibliotheks-SLB-Cluster abgefragt werden. Hierzu übertragen die VoD-Katalog-Server die IP-Adresse des anfordernden Clients und einen eindeutigen Identifikator für das angeforderte Video. Mithilfe dieser Informationen ist das Request-Routing System in der Lage, eine Entscheidung über die Auswahl des Stellvertreterservers oder des VoD-Bibliothek-SLB-Clusters für die Übertragung zu treffen. Das Request-Routing System trifft die Entscheidungen zunächst anhand von Proximitätsinformationen, um die am nächsten liegenden Server ausfindig zu machen. Anschließend werden die Netzwerklast- und die Server load-Metrik ausgewertet. Die Netzwerklast Metrik ist für das Request-Routing System relativ einfach zu erheben, da es sich um das eigene Netz mit den Komponenten des ISPs handelt, bei denen dem Request-Routing System Zugriff auf die Leistungsindikatoren der Netzwerkelemente gewährt werden kann. Somit ist es nicht nötig, diese Werte durch aktives Network Probing zu ermitteln und der zusätzliche Datenverkehr für die Messungen entfällt.
Bei der Server load-Metrik werden mithilfe von Agenten, die auf den Stellvertreterservern ausgeführt werden, Gesundheitsinformationen ermittelt.

Accounting System Das Accounting System speichert Daten zum Erwerb der Zugriffsrechte durch die Inhaltekonsumenten. Sobald sich Kunden für ein Video aus dem VoD-Katalog entschieden und dem Erwerb des Zugriffsrechts zugestimmt haben, wird dies durch das hinzufügen eines Datensatzes im Accounting System vermerkt.
Je nach Modell kann die Gültigkeitsdauer des Eintrags von wenigen Stunden bis zum Ende des VoD-Dienstes variieren. Dies bedeutet, dass der Inhaltekonsument das Video in dieser Zeitspanne so oft konsumieren kann wie er möchte. Sobald die Gültigkeit des Zugriffsrechts abgelaufen ist, muss das Recht für einen erneuten Zugriff ein weiteres Mal erworben werden.
Sowohl die Stellvertreterserver als auch die VoD-Bibliothek-Server wenden sich an das Accounting System mit der angeforderten URL sowie der IP-Adresse des anfragenden Clients. Das Accounting System ist unter Berücksichtigung dieser Informationen in der Lage, Auskunft darüber zu erteilen, ob es sich um einen legitimen Zugriff handelt oder nicht.
Die Daten des Accounting Systems werden für die Abrechnung des VoD-Dienstes verwendet. Dies geschieht gegenüber dem Inhaltekonsumenten und gegenüber den Inhalteanbietern, wenn diese pro verkauftem Zugriffsrecht abgerechnet werden.

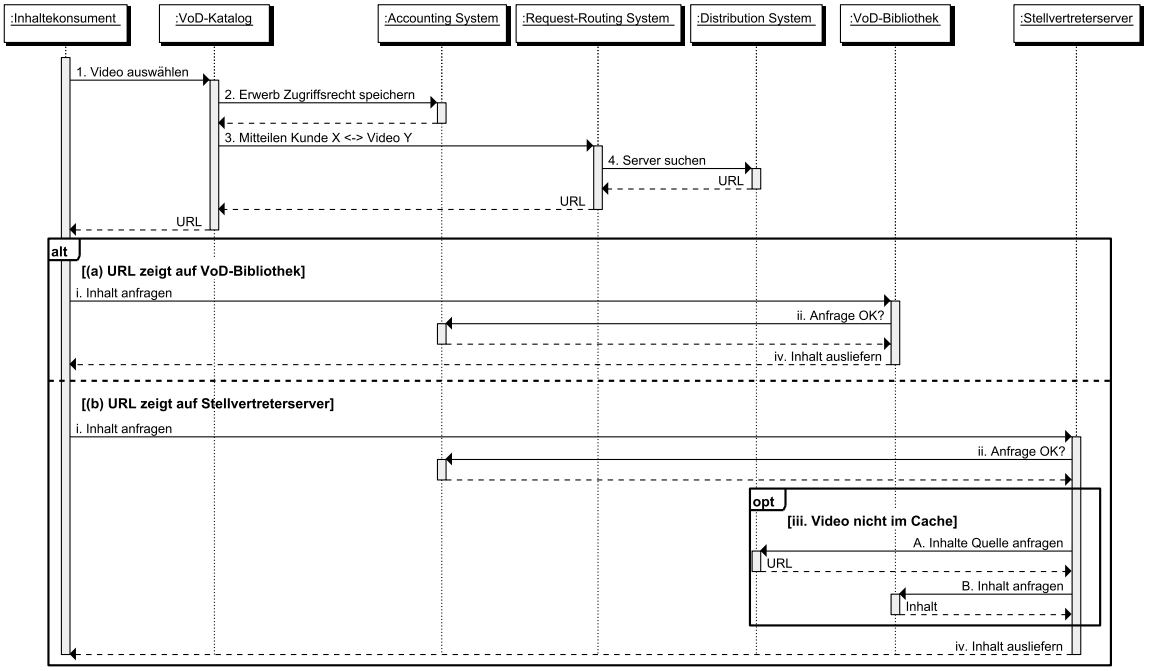


Abbildung 4.2: Ablauf der VoD-Inhaltsanfrage

Distribution System Das Distribution System ist für die Ausbringung der populären Inhalte aus den zentralen VoD-Bibliotheken auf die Stellvertreterserver zuständig. Es kommuniziert mit dem Accounting System, von dem es Informationen über die Popularität von Inhalten erhält. Anhand dieser Informationen entscheidet es, welche Inhalte auf den Stellvertreterservern vorgehalten werden. Wird ein Video von einem Stellvertreterserver angefordert, versucht dieser mithilfe des Distribution Systems einen anderen möglichst nahe gelegenen Stellvertreterserver zu finden, von dem er den Videoinhalt abrufen kann. Sollte noch kein geeigneter Stellvertreterserver das Video im Cache vorhalten, so wird dieses aus der am nächsten liegenden VoD-Bibliothek abgerufen und in den Cache aufgenommen.

Ablauf einer VoD-Inhaltsanfrage

1. Inhaltekonsument durchsucht den VoD-Katalog.
   - Der Kunde wählt ein Video aus, das er anschauen möchte.
2. Der VoD-Katalog-Server wendet sich an das Accounting System, um diesem mitzuteilen, welches Video von welchem Kunden angefordert wurde.
   - Das Accounting System speichert die Assoziation Kunde zu Video (Zugriffsrecht).

3. Der VoD-Katalog-Server wendet sich an das Request-Routing System und teilt diesem mit, welcher Kunde welches Video anschauen möchte.
4. Das Request-Routing System greift auf den Datenbestand des Distribution Systems zu und sucht einen geeigneten Stellvertreterserver, wenn es sich um populäre Videoinhalte handelt. Handelt es sich um weniger populäre Videoinhalte sucht es einen geeigneten VoD-Bibliothekscluster.
   Die ermittelten Daten werden an den VoD-Katalog zurückgeliefert.
5. Der VoD-Katalog-Server sendet dem Inhaltekonsumenten die durch das Request-Routing System zurückgegebene URL zu. Diese wird unter Zuhilfenahme der URL Rewriting Methode in die Antwort eingebracht.
6. Der Client benutzt die URL, um die Videoinhalte abzurufen. Hierbei gibt es zwei Möglichkeiten.
   (a) URL verweist auf einen VoD-Bibliothekscluster.
      i. Die VoD-Bibliothek nimmt die Anfrage entgegen.
      ii. Die VoD-Bibliothek stellt eine Anfrage an das Accounting System, ob der Kunde das angeforderte Video sehen darf.
      iii. Die VoD-Bibliothek überträgt den Videoinhalt an den Client des Inhaltekonsumenten.
   (b) URL verweist auf einen Stellvertreterserver
      i. Der Stellvertreterserver nimmt die Anfrage entgegen.
      ii. Der Stellvertreterserver stellt eine Anfrage an das Accounting System, ob der Kunde das angeforderte Video sehen darf.
      iii. Nun gibt es zwei Möglichkeiten:
      - Das Video ist im Cache vorhanden.
        A. In diesem Fall ist keine besondere Aktion durchzuführen.
      - Das Video ist nicht im Cache vorhanden.
        A. Der Stellvertreterserver wendet sich an das Distribution System, um zu erfahren, von welchem anderen Stellvertreterserver oder von welchem VoD-Bibliothekscluster die Inhalte abgerufen werden können.
        B. Der Stellvertreterserver ruft den Inhalt von der zurückgelieferten Adresse ab und speichert ihn zwischen.
      iv. Der Videoinhalt wird an den Client des Inhaltekonsumenten übertragen.

4.2 Verkehrsoptimierung

ISPs können die Verringerung der Übertragungskosten durch das Zwischenspeichern häufig angefragter Inhalte realisieren. In diesem Fall sind Inhalte gemeint, die nicht primär aus dem CDN des ISPs zur Verfügung gestellt werden. Diese Inhalte werden beim Anfragen durch einen Inhaltekonsumenten durch den Stellvertreterserver des ISPs abgerufen und auf diesem zwischengespeichert. Weitere Anfragen, die diesen Inhalt betreffen, werden aus dem Cache bedient. Hierbei spielt es keine Rolle, ob der Inhalt vom gleichen oder einem anderen Inhaltekonsumenten angefragt wird. Durch die Verteilung der Stellvertreterserver über die verschiendenen PoPs sind die Inhalte in der Nähe des Inhaltekonsumenten zwischengespeichert. Inhaltekonsumenten profitieren somit beim erneuten Anfragen der Inhalte durch die Proximität zu den Stellvertreterservern. Dies trägt dazu bei, dass die Inhalteübertragung im Bezug auf Jitter, Latenz und den Verlust von Paketen bessere Eigenschaften aufweist als diese bei der Übertragung vom Herkunftsserver zu erwarten sind. Für den ISP ist es von Vorteil, dass die Transitkosten und die Belastung des Backbones verringert werden.
Dieser Ansatz birgt jedoch auch Probleme. Das CDN muss wissen, welche Inhalte zwischengespeichert werden dürfen und welche nicht, welche Inhalte dynamisch generiert werden und sich bei jedem Aufruf ändern. Besonders kritisch ist dies in Fällen, in denen die Inhalte personalisiert sind, wie dies beispielsweise bei Warenkörben in Onlineshops der Fall ist.
Im Folgenden werden unterschiedliche Ansätze zur Realisierung der Verkehrsoptimierung dargestellt.

4.2.1 Ansätze

Für das Zwischenspeichern von Webseiteninhalten werden Proxyserver eingesetzt. Sie nehmen die Inhaltsanfragen der Clients entgegen und bearbeiten diese. Hierzu rufen sie die vom Client angefragten Inhalte vom Herkunftsserver ab und liefern sie anschließend an den Client des Inhaltekonsumenten aus. Zusätzlich zu dieser einfachen Auslieferung der Inhalte sind sie in der Lage, Inhalte zu verändern, zwischenzuspeichern, nach Viren zu scannen sowie die Last von Inhaltsanfragen auf mehrere Server zu verteilen. Aufgrund des breiten Anwendungsgebiets existieren verschiedene Typen von Proxyservern [Rabinovich und Spatscheck (2001); Wessels (2001)], die in unterschiedlichen Szenarien eingesetzt werden. Im Folgenden werden drei Typen von Proxyservern näher dargestellt.

Forward Proxy Forward Proxyserver werden von Clients genutzt, um mit ihrer Hilfe die Inhalte verschiedenster Webseiten abzurufen. Die Verwendung des Proxyservers wird explizit durch den Inhaltekonsumenten oder einen Administrator auf dem Client konfiguriert. Alle Anfragen des Clients werden zunächst an den Proxyserver gesendet, der dann die Inhalte vom Herkunftsserver abruft oder ggf. aus dem Cache an den Client ausliefert. Forward Proxyserver werden häufig eingesetzt, um bestimmte Inhal-

te aus Webseiten herauszufiltern, Virenscans durchzuführen oder angefragte Inhalte zwischenzuspeichern und bei weiteren Anfragen aus dem Cache auszuliefern. Sie werden oft in Einrichtungen eingesetzt, in denen eine zentrale Autorität die Benutzung des Proxyservers vorschreibt und die direkte Datenkommunikation der Clients ins Netz unterbindet. Ausschließlich der Proxyserver ist in diesen Umgebungen imstande, Webseiteninhalte direkt aus dem Internet abzurufen.

Reverse Proxy Reverse Proxyserver werden von Webseitenbetreibern vor ihren Webservern platziert. Alle Anfragen von Clients, die an die Webseite gestellt werden, werden zunächst durch den Reverse Proxy geleitet. Dieser kann Teile der Webseite, z. B. Bilder oder Videos, zwischenspeichern. Durch diese Art der Verwendung wird ein Teil der Last, die auf die Webserver einwirkt, durch die Proxyserver abgefangen. Der Webserver kümmert sich somit hauptsächlich um die Generierung und Auslieferung der dynamischen Inhalte.

Reverse Proxyserver werden auch eingesetzt, um Webseiten Secure Sockets Layer (SSL) oder Transport Layer Security (TLS) verschlüsselt auszuliefern. Hierzu werden die Clientanfragen vom Reverse Proxy angenommen und an die Webserver weitergeleitet. Die Webserver übertragen die Inhalte unverschlüsselt an den Proxyserver, auf dem das zur Verschlüsselung benötigte digitale Zertifikat installiert ist. Der Proxyserver verschlüsselt die Daten und übermittelt sie so in gesicherter Form an den Client. Dieses Vorgehen bringt die Vorteile, dass nicht jeder Server ein digitales Zertifikat für die Verschlüsselung benötigt und die Last, die durch das Verschlüsseln der Inhalte entsteht, von den Servern zum Proxyserver verlagert wird. Dies ist vor allem dann sinnvoll, wenn der Proxyserver speziell für die Verschlüsslung von Inhalten optimiert ist. Somit trägt dieser Ansatz dazu bei, die Anschaffungs- und Betriebskosten eines Webangebots gering und die Infrastruktur gleichzeitig skalierbar zu halten.

Interception Proxy Bei der Interception Technik werden die Clientverbindungen von einem Netzwerkelement, welches in den Übertragungsweg vom Client zum Herkunftsserver integriert ist, an einen Proxyserver weitergeleitet. Diese Umleitung der Datenpakete findet ohne das Wissen und die explizite Konfiguration des Browsers durch den Inhaltekonsumenten statt. Um Interception Proxys in die Datenübertragung einzubinden, werden sie entweder direkt oder mithilfe eines weiteren Geräts (Interceptor) in die Upstreamverbindung eingebunden. Es ist somit möglich, die Verwendung eines Proxyservers auch dann durchzusetzen, wenn der Proxyserverbetreiber keine administrative Gewalt über die verwendeten Clients bzw. Browser besitzt [Fielding u. a. (1999)].

Die erzwungene Verwendung eines Proxyservers auf diese Art birgt folgende Probleme. Die Datenpakete müssen zunächst erkannt, aus dem gesamten Datenverkehr herausgefiltert und an den Proxyserver umgeleitet werden. Anschließend können die

Pakete vom Proxyserver verarbeitet werden. Für die Umleitung müssen die Pakete unter umständen bis in die Applikationsschicht untersucht werden, um eine Entscheidung treffen zu können, ob der Proxyserver für die Verarbeitung des entsprechenden Pakets zuständig ist oder nicht.
Pakete, die an den Proxyserver umgeleitet werden sollen, müssen diesem zugestellt werden. Hierzu gibt es zwei Ansätze, die entweder auf der Sicherungsschicht (Layer 2) oder auf der Vermittlungsschicht (Layer 3) agieren. Im folgenden Abschnitt werden diese beiden Techniken detailliert dargestellt.

Interception Pakettransport

Layer 2 Umleitung Das Intercepting Netzwerkelement sendet die Pakete an die Sicherungsschicht-Adresse (MAC-Adresse) des Proxyservers. Auf diese Weise bleiben die Informationen der Vermittlungsschicht (IP-Adressen etc.) in den umgeleiteten Paketen unverändert erhalten [Cieslak u. a. (2001)].
Dieser Ansatz zur Umleitung ist sehr effizient, da nur die Empfänger-MAC-Adresse umgeschrieben werden muss. Es wird jedoch vorausgesetzt, dass das Intercepting Netzwerkelement und der Proxyserver an dasselbe Sicherungsschicht-Netzwerk angeschlossen sind, da die Pakete noch immer die IP-Adresse des originalen Webservers tragen (siehe Abbildung 4.3). Würden diese Pakete auf dem Weg zum Proxyserver einen weiteren Router passieren, würde er diese anhand seiner IP-Adresse wieder in Richtung des Original-Webservers transportieren.
Dieser Ansatz erfordert des Weiteren Veränderungen am Protokollstack des Proxyservers. Dieser muss so angepasst werden, dass alle ankommenden Pakete, unabhängig von der im Paket angegebenen Empfänger-IP-Adresse, an die Anwendungsschicht des Proxyservers weitergereicht werden.

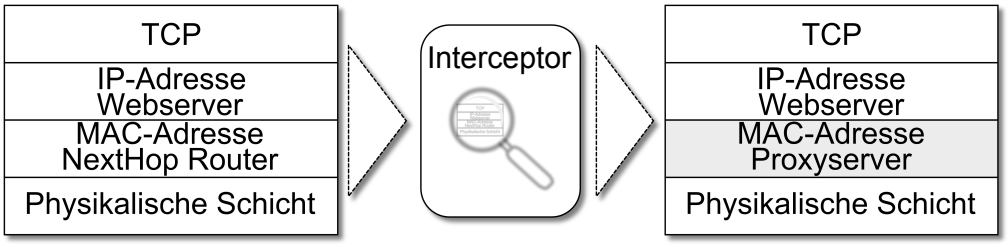


Abbildung 4.3: Layer 2 Umleitung - MAC-rewrite

Layer 3 Umleitung Das Intercepting Netzwerkelement verpackt jedes empfangene Paket in einem neuen Paket und adressiert dieses an den Proxyserver. Bei dieser IP-in-IP Kapselung [Farinacci u. a. (2000)] wird, wie in Abbildung 4.4 zu sehen, der Layer 2 Header des Originalpakets verworfen. Dem verbleibenden Paket werden neue Layer 2 und 3 Informationen sowie ein GRE-Header vorangestellt. Die Empfänger-IP-Adresse des äußeren Vermittlungsschicht-Headers enthält die IP-Adresse des Proxyservers, die Pakete können hierdurch zum Proxyserver geroutet werden. Die Limitierung, das Proxyserver und der Interceptor im selben Layer 2 Netzwerk residieren müssen, entfällt somit für diesen Ansatz. Des Weiteren ist es nicht nötig, den Protokollstack des Proxyservers anzupassen, da die IP-Adresse des Originalservers erst in der Applikationsschicht des neuen Pakets sichtbar wird. Nachteilig ist der größere Aufwand für die Enkapsulierung des Originalpakets.

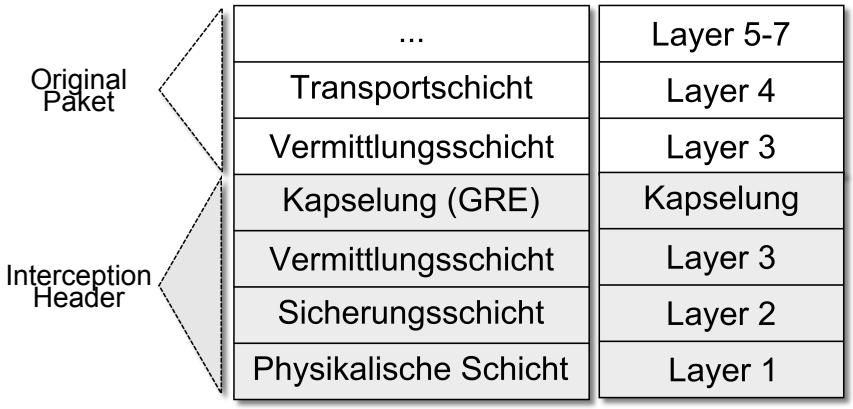
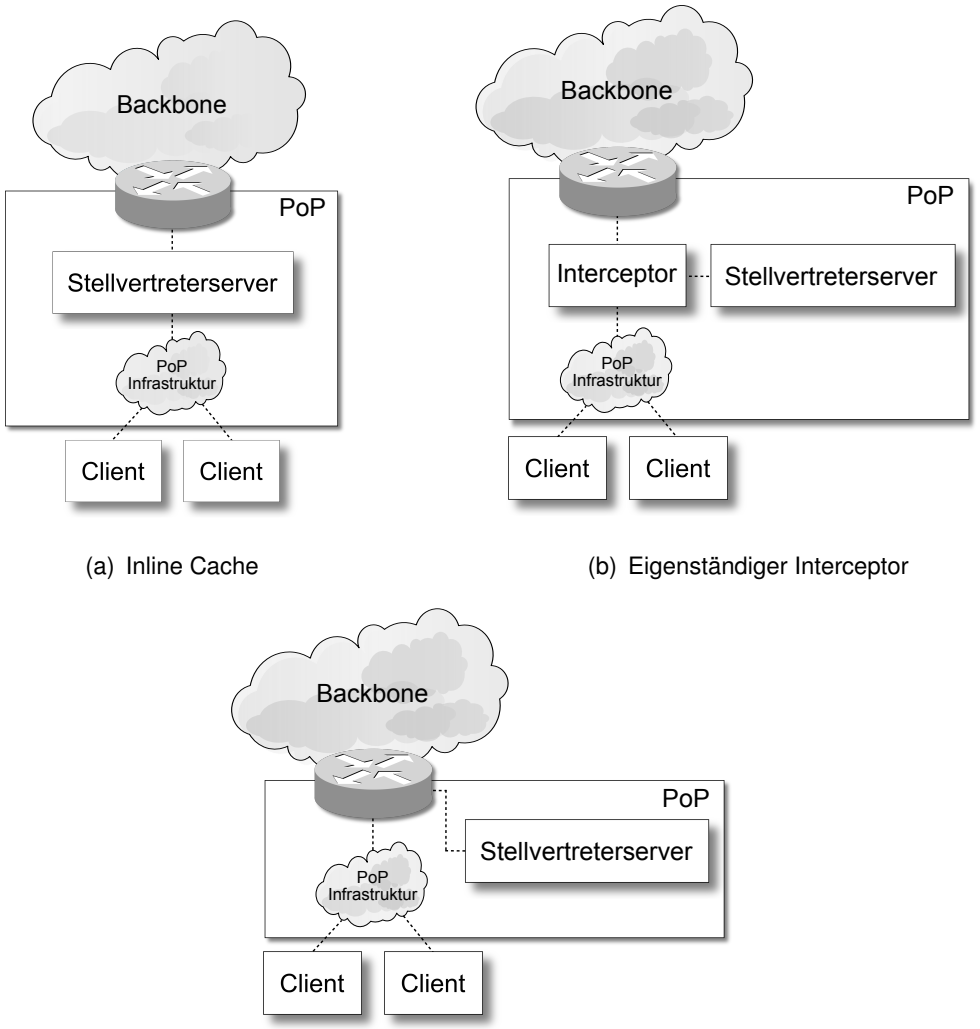


Abbildung 4.4: Layer 3 Umleitung - IP-in-IP Kapselung

Interception Methoden

Die Unterbrechung der Inhaltsanfragen kann auf unterschiedliche Weise geschehen. Nachfolgend sind drei Interception Methoden dargestellt.

Inline Cache Bei der Verwendung von Stellvertreterservern als Inline Element (siehe Abbildung 4.5) wird der gesamte Datenverkehr, der von dem angeschlossenen Client initiiert wird, durch dieses geleitet. Hierdurch ist der Stellvertreterserver in der Lage, die Inhaltsanfragen aus dem Gesamtstrom der Daten zu erkennen, die Weiterleitung zu unterbrechen und die Inhaltsanfrage selbst zu bearbeiten. Nachteilig bei diesem Ansatz ist die hohe Performanceanforderung an den Stellvertreterserver, da der gesamten Datenverkehr des Netzwerksegments durch diesen geleitet wird. Des Weiteren ist



(a) Inline Cache

(b) Eigenständiger Interceptor

(c) Interception Router - Policy Based Routing

Abbildung 4.5: Cache Ansätze

es notwendig, die Stellvertreterserver redundant auszulegen, um bei einem Ausfall nicht das gesamte Netzsegment vom Backbone abzuschneiden.

Interceptor Bei der Verwendung eines Interceptors werden die Stellvertreterserver nicht direkt in den Übertragungsweg der Datenpakete integriert. Die Datenpakete werden durch den Interceptor inspiziert und entweder direkt an den Herkunftsserver weiter- oder an den Proxyserver umgeleitet. Interceptoren sind in der Lage, den Status angeschlossener Netzwerkelemente zu prüfen und aufgrund dieser Prüfungen den Stellvertreterserver zu um-

gehen, sollte dieser nicht mehr ordnungsgemäß arbeiten. Grundlage für die Überprüfung der angeschlossenen Proxyserver stellen Protokolle wie WCCP [Cieslak u. a. (2001)] oder NECP [Cerpa u. a. (2000)] dar. Diese Protokolle sind für die Kommunikation zwischen Interceptor und Proxyservern zuständig und sind unter anderem auf das Erkennen von Fehlern und Ausfällen der Proxyserver ausgelegt und reagieren schnell, sollten diese nicht mehr erreichbar sein. Die Überprüfung des Proxyservers ist ebenso durch das Abrufen einer Dummy-Seite möglich. Sollte die Dummy-Webseite durch den Interceptor nicht mehr abrufbar sein, so werden keine Pakete mehr an den Proxyserver umgeleitet. Bei dieser Methode ist die Reaktionsgeschwindigkeit jedoch geringer als bei WCCP und NECP. Zunächst muss der Ablauf der TCP Wiederholungen und Timeouts abgewartet werden, bevor ein Proxyserver als nicht funktionsfähig eingestuft wird. Nachteilig bei dieser Methode ist des Weiteren, dass zwei eigenständige Netzwerkelemente betrieben werden müssen, die separat zu administrieren sind.
Positiv wirkt sich die Aufteilung der Zuständigkeiten der Netzwerkelemente aus. Das Umleiten von Paketen, sowie das Verwalten des Caches wird somit von unterschiedlichen, für die jeweilige Operation optimierten Netzwerkelementen, übernommen.
Nachfolgend werden die Arbeitsweisen von Layer 4 und Layer 7 Switches beschrieben, die die Methodiken für das Umleiten von Datenpaketen zur Verfügung stellen.

Layer 4 Switch Ein Layer 4 Switch ist in der Lage, den Vermittlungs- und Transportschicht Header jedes Pakets auszulesen. Nach Ankunft des Pakets wird zunächst im IP-Header geprüft, ob es sich um ein Paket handelt, das einer TCP-Verbindung angehört. Wenn dies zutrifft, wird im TCP-Header geprüft, ob es sich bei dem im Paketheader enthaltenen Empfängerport um Port 80, den standardisierten HTTP-Port handelt. Sobald das Paket diese Eigenschaften aufweist, wird es vom Layer 4 Switch an den Proxyserver umgeleitet. Alle weiteren Pakete folgen weiter dem normalen Datenfluss in Richtung des Empfängers.
Layer 4 Switches sind in der Lage, die Pakete, die sie umleiten, auf mehrere Proxyserver zu verteilen. Um dies effizient zu tun, wird z. B. ein Hashwert über die Empfänger-IP-Adresse des Datenpakets, in dem die Inhaltsanfrage enthalten ist, gebildet. Der Bereich an möglichen Hashwerten wird auf die zur Verfügung stehenden Proxyserver abgebildet. Es existiert eine eindeutige Abbildung, die der Layer 4 Switch nutzen kann, um die Last der Anfragen aufzuteilen. Die Verwendung einer eindeutigen Abbildung führt zu einer Steigerung der Effizienz des Proxyser-

vers, da dieselben Webseiteninhalte immer von demselben Proxyserver abgerufen werden und die Elemente immer in höchstens einem Proxyserver je Cluster vorgehalten werden.

Layer 7 Switch Ein Layer 7 Switch ist in der Lage, über den Vermittlungs- und Transportschicht-Header hinaus auch die Nutzlast eines Datenpakets auszuwerten. So wird die Quantität der Informationen, aufgrund derer die Umleitungsentscheidungen getroffen werden, weiter erhöht. Dem Layer 7 Switch ist nach der Untersuchung des Datenpakets beispielsweise bekannt, ob es sich um die Anfrage eines Bildes, eines Videos oder eines anderen Inhaltstyps handelt. Die unterschiedlichen Inhalte können nun von unterschiedlichen, für den entsprechenden Inhaltstyp optimierten Proxyservern abgerufen werden. Des Weiteren ist der Layer 7 Switch in der Lage, Anfragen von nicht zwischenspeicherbaren Inhalten direkt an den Herkunftsserver weiterzuleiten anstatt diese zunächst an den Proxyserver umzuleiten.

Stellt der Proxyserver fest, dass nicht zwischenspeicherbarer Inhalt angefragt worden ist, so wird die Inhaltsanfrage ohne weitere Bearbeitung an den Originalserver weitergeleitet. Durch das Auswerten der Inhaltsanfrage auf dem Interceptor und die damit verbundene Unterbindung des Umleitens, verringert sich die Last sowie die Latenz, die durch die unnütze Umleitung der Anfragen auf dem Proxyserver entstehen würden.

Interception Router Ein Interception Router kann wie ein Layer 4 Switch ebenfalls den Vermittlungs- und Transportschicht-Header der von ihm verarbeiteten Pakete auswerten und diese anhand zuvor definierter Regeln umleiten. Wie Layer 4 Switch sind auch Interception Router in der Lage, die Pakete mithilfe von Layer 2 oder Layer 3 Techniken an die Proxyserver umzuleiten. Hierzu verwenden sie in der Regel die Routingtabelle, in der für bestimmte Routen Weiterleitungsregeln hinterlegt sind, welche entsprechend der Umleitungstechniken spezielle Aktionen beschreiben. Diese Technik wird auch als Policy Based Routing (PBR) bezeichnet.

Interception Router unterstützen in vielen Fällen auch die Lastverteilung auf unterschiedliche Proxyserver. Da sie jedoch primär für das Routen von IP-Datagrammen zuständig und hierauf optimiert sind, bieten sie in der Regel nicht den gleichen Leistungsumfang wie ihn reine Layer 4 oder 7 Switches bieten.

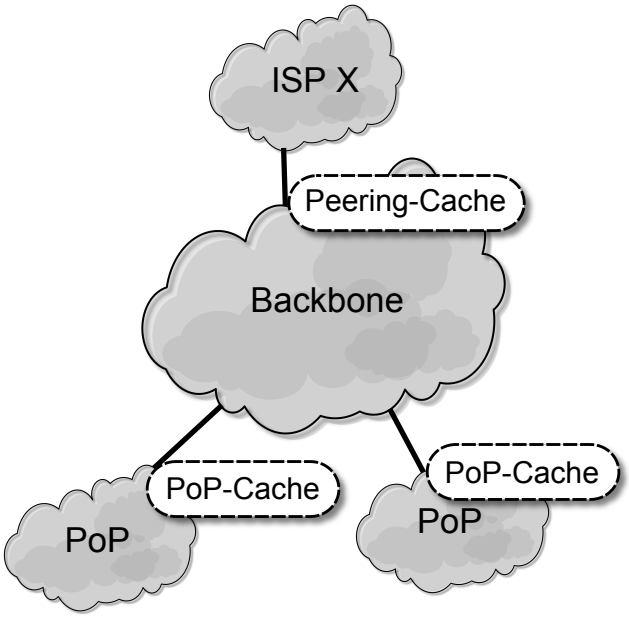


Abbildung 4.6: Cache Positionierung

4.2.2 Design

Um ein CDN für die Verringerung von Transitkosten und Backbonebelastung zu nutzen, müssen die Stellvertreterserver so platziert werden, dass sie von einem Interceptor die entsprechenden Datenpakete weitergeleitet bekommen.
Je nach Position der Stellvertreterserver werden die verschiedenen Anforderungen unterschiedlich gut erfüllt. Im Folgenden werden drei Ansätze genauer erläutert, die in Abbildung 4.6 schemenhaft dargestellt sind.

Peering-Punkt Cache Die Stellvertreterserver werden im ISP Netz hinter den Peering-Routern installiert. Hierdurch werden alle Inhaltsanfragen, die eine Transit-Verbindung passieren, durch den Stellvertreterserver geleitet und die angefragten Webinhalte können zwischengespeichert werden.
Durch das Zwischenspeichern der Inhalte im Peering-Punkt wird die über die Transit-Verbindung übertragene Datenmenge reduziert, was sich für den ISP in verringerten Transitkosten niederschlägt.
Dieser Ansatz ist nicht zur Reduzierung der Backbonebelastung des ISP Netzes geeignet, da die Inhalte weiterhin den ISP eigenen Backbone durchqueren müssen.

PoP Cache Die Stellvertreterserver werden in den PoPs platziert. Die Einbindung findet so statt, dass die gesamten Inhaltsanfragen der an diesen PoP angeschlossenen Inhaltekonsumenten an den Stellvertreterserver umgeleitet werden.

Durch das Zwischenspeichern der Inhalte im PoP verringern sich die Antwortzeiten für bereits zwischengespeicherte Inhalte, da die Latenz zum Stellvertreterserver geringer ist als zum Herkunftsserver. Des Weiteren wird die Belastung des Backbones verringert, da dieser für die Auslieferung bereits zwischengespeicherter Webinhalte nicht weiter beansprucht werden muss.
Bei der Umsetzung des PoP Cache-Ansatzes werden in jedem PoP Stellvertreterserver installiert. Hierdurch kann sich eine große Anzahl verteilter Stellvertreterserver ergeben, woraus sich erhöhte Administrationskosten ableiten lassen.
Die Performanceanforderungen der beim PoP Cache verwendeten Stellvertreterserver kann hingegen als gering bewertet werden, da die Anzahl der in einem einzelnen PoP angeschlossenen Inhaltekonsumenten geringer ist als die Gesamtanzahl der Inhaltekonsumenten, die einen Peering-Punkt Cache belasten.

Hierarchische Caches Beim hierarchischen Caching werden die beiden Ansätze PoP Cache und Peering-Punkt Cache zusammen eingesetzt. Hierdurch werden die Inhalte nah am Inhaltekonsumenten gespeichert und entlasten den Backbone. Die Peering-Punkt Caches entlasten die Transit-Verbindung und verringern die anfallenden Transitkosten.
Die Anfrage, die an den Herkunftsserver gerichtet ist, wird unterbrochen und an den PoP Cache umgeleitet. Dieser versucht die Webinhalte aus seinem Cache auszuliefern. Wenn die angeforderten Daten nicht bereits vorliegen, stellt er selbst eine Inhaltsanfrage an den Herkunftsserver. Diese Anfrage wird über den Backbone zum Peering-Punkt übertragen, wo auch diese Inhaltsanfrage unterbrochen und an den Peering-Punkt Cache umgeleitet wird. Anschließend versucht dieser die Anfrage aus seinem Cache heraus zu beantworten. Hat auch dieser die Inhalte nicht in seinem Cache, so stellt auch er eine Inhaltsanfrage an den Herkunftsserver, die über die Peering-Verbindung übertragen wird.

Kapitel 5

Praktische Umsetzung

Im Folgenden werden die in Kapitel 4 dargestellten Problemstellungen und Konzepte in einer Testumgebung umgesetzt und bewertet.
Es wird zum einen die Übertragung von Videomaterial unter Verwendung einer adapativen Bitratenanpassung, zum anderen die Zwischenspeicherung von YouTube Videos untersucht und dokumentiert.
Zur Umsetzung dieser beiden voneinander unabhängigen Testszenarien wird eine CDN-Serverlösung, der Media Flow Controller, der Firma Juniper Networks verwendet. Der Media Flow Controller (MFC) [Juniper Networks (2010)] ist für die Zwischenspeicherung und das Übertragen von Videoinhalten optimiert, bietet jedoch ebenso die Möglichkeit zur Zwischenspeicherung und Übertragung von Webseiteninhalten wie Bilder, Texte oder andere Daten.

5.1 Testumgebung

Der Testaufbau besteht aus einem Netzwerk wie in Abbildung 5.1 dargestellt, das sich aus einem Router, dem MFC, drei Clients und einem Switch zusammensetzt.
Detaillierte Informationen zu den einzelnen Komponenten sind den nachfolgenden Abschnitten 5.1.1 bis 5.1.3 zu entnehmen.

5.1.1 Router

Der für den Testaufbau verwendete Router ist in der Lage, den zu verarbeitenden Datenverkehr bis hoch zur Anwendungsschicht zu inspizieren. Hierdurch ist es möglich, die entsprechenden Pakete, die an TCP-Port 80 gesendet werden und HTTP-Anfragen enthalten, an den MFC umzuleiten. Wenn es sich nicht um HTTP-Datenverkehr handelt, werden die Pakete weiter an den Backbone in Richtung Empfänger geleitet. Alle Datenpakete, die vom MFC versendet am Router eintreffen, werden ohne weitere Untersuchung durch den

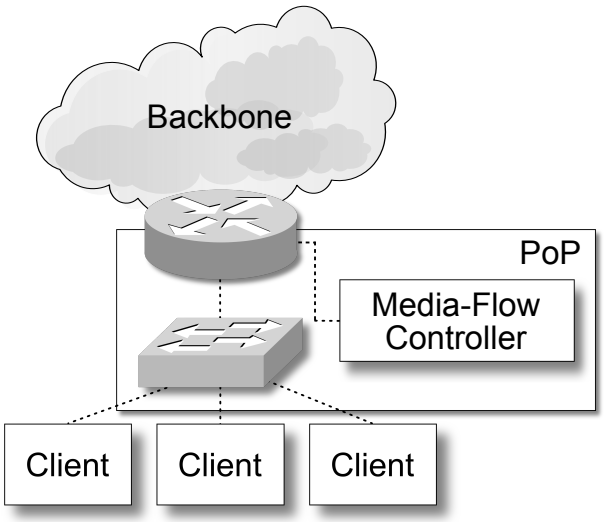


Abbildung 5.1: Testumgebung

Router an den entsprechenden Empfänger weitergeleitet.
Zur Umleitung der Pakete an den MFC werden diese in ein neues IP-Datagramm eingefasst. Die Verwendung der Layer 3 Umleitung ist durch die nicht vorhandene Unterstützung der Layer 2 Umleitung zu erklären, birgt jedoch für das Testszenario keine konkreten Vor- oder Nachteile. Das Anwendungsszenario erlaubt ebenso die Verwendung der Layer 2 Umleitung.

5.1.2 Media Flow Controller

Der MFC kann als Interception, Reverse und Forward Proxy eingesetzt werden. Der MFC wird für die Umsetzung aller in diesem Kapitel dargestellten Szenarien in der Version 2.0.4 verwendet. In dieser Version wird kein transparentes Weiterleiten von Paketen, also kein transparentes Bridging oder Routing der Datenpakete unterstützt. Hierdurch ist die Nutzung des MFCs als Inline Proxy in dieser Version nicht möglich [Juniper Networks (2010)].
Jede Inhaltsanfrage der Clients muss von den Regeln eines Namensraums abgedeckt werden. Namensräume beschreiben Eigenschaften und Aktionen, die für diese Teile oder ganze Webseiten gelten bzw. ausgeführt werden sollen. Namensräume werden durch URL-Bestandteile, d. h. Domain-, Pfadangaben oder einer Kombination aus beidem, definiert. Zu den konfigurierbaren Eigenschaften gehört unter anderem der Umgang mit den Cache-Anweisungen im HTTP-Header einer Inhaltsauslieferung. Es kann z. B. definiert werden, ob diese vom MFC beachtet oder überschrieben werden. Des Weiteren wird spezifiziert, von welchen Servern die Inhalte eines Namensraums abzurufen sind. Der MFC unterstützt die Einrichtung mehrerer Namensräume mit jeweils unterschiedlicher Konfiguration. Fallen die

an den MFC umgeleiteten Anfragen in keinen Namensraum, werden sie als nicht zulässige Anfragen eingestuft und als Resultat daraus nicht verarbeitet. Der MFC beantwortet die Anfrage dann mit der HTTP-Fehlermeldung 404 Not Found.
Da Forward Proxys in der Regel Anfragen für eine Vielzahl von Domains verarbeiten sollen, wird bei der Verwendung des MFCs als Forward Proxy bei der Angabe der Namensräume mit Wildcards gearbeitet. Die konfigurierten Namensräume werden mit Prioritäten versehen, so dass bei Überschneidungen der Namensräume die Einstellungen des zutreffenden Namensraums mit der höchsten Priorität in Kraft treten.
Somit ist es möglich, den Namensraum, der auf den gesamten Webverkehr zutrifft, so zu konfigurieren, dass keine Inhalte zwischengespeichert werden. Ein anderer Namensraum, der beispielsweise auf example.com zutrifft, also spezieller ist, wird so konfiguriert, dass alle Inhalte zwischengespeichert werden. Um sicher zu stellen, dass die Regel des spezifischeren Namensraums bei Anfragen an example.com verwendet wird, wird der Namensraum für example.com mit einer höheren Priorität versehen, wodurch er bei der Auswahl des zuständigen Namensraums vorrangig zur Anwendung kommt.
Für die Zwischenspeicherung aller vorgehaltenen Inhalte nutzt der MFC eine hierarchische Speicherstruktur. Nach dem Abrufen werden die Inhalte zunächst im Arbeitsspeicher vorgehalten. Füllt sich der Arbeitsspeicher zunehmend, werden die Inhalte gemäß des Least Recently Used (LRU)-Verfahrens in die nachfolgende Cache-Hierarchie ausgelagert. Derzeit werden drei Hierarchiestufen unterstützt: Random-Access Memory (RAM), Solide State Disk (SSD) und Hard Disk Drive (HDD). Die Aufteilung in Hierarchien liegt in der Geschwindigkeit und dem Preis der einzelnen Technologien sowie der unterschiedlichen Popularität verschiedener Inhalte begründet. Inhalte, die häufig abgerufen werden, werden im RAM, dem schnellsten der drei verfügbaren Speicher des Proxyservers gehalten. Weniger häufig abgerufene Inhalte werden über die SSD auf die HDD, den langsamsten Speicher verschoben, wo sie bei weiter abnehmender Popularität (in Relation zu den anderen Inhalten) überschrieben werden. Die Einführung der Hierarchien in die Speicherverwaltung erlaubt es, eine kosten- und zugleich performanceeffiziente Speicherung und Auslieferung der Inhalte zu realisieren.
Der MFC kann in VoD-Diensten als Stellvertreterserver eingesetzt werden. Er ist in der Lage, die Qualität des übertragenen Videomaterials mittels adaptiver Bitratenanpassung an die äußeren Gegebenheiten des Übertragungskanals anzupassen. Je nach verwendetem Client (Smartphone, Set-Top Box, PC) oder verfügbarer Bandbreite wird die Bitrate des versendeten Videomaterials angepasst. Hierdurch wird dem Inhaltekonsumenten eine möglichst hohe Quality of Experience (QoE) geboten. Unterbrechungen der Videowiedergabe aufgrund von leergelaufenem Puffer werden mithilfe dieser Funktion minimiert oder verschwinden gänzlich.
Solange genügend Bandbreite zur Verfügung steht, wird diese durch den Versand eines qualitativ hochwertigen Videomaterials mit großer Bitrate genutzt. Sobald sich die verfügbare Bandbreite ändert, werden die Bitrate und damit die Qualität des übertragenen Videomate-

rials entsprechend angepasst.
Der MFC ist des Weiteren in der Lage, Inhalte aus Videoportalen, wie beispielsweise YouTube[5] oder MyVideo[6], zwischenzuspeichern, um diese Inhalte bei wiederholten Inhaltsanfragen der Inhaltekonsumenten aus dem Zwischenspeicher auszuliefern anstatt das Videomaterial erneut vom Herkunftsserver abzurufen.

5.1.3 Client Computer

Bei den Client Computern handelt es sich um Windows 7 Arbeitsplatzrechner. Als Webbrowser kommt der Microsoft Internet Explorer in der Version 8 zum Einsatz. Das Adobe Flash Plugin ist in der Version 10.1 installiert. Die Anbindung an das Netzwerk erfolgt mithilfe eines 100 MBit/s Ethernet Adapters pro Rechner.
Die Default-Route der Client Computer zeigt auf den Backbonerouter. Das bedeutet, dass die gesamte Kommunikation mit anderen IP-Netzen über den Backbonerouter abgewickelt wird.

[5] www.youtube.com
[6] www.myvideo.de

5.2 Videoübertragung mittels adaptiver Bitratenanpassung

5.2.1 Zielsetzung

Im Testszenario „Videoübertragung mittels adaptiver Bitratenanpassung“ wird das Konzept eines VoD-Angebots (siehe Kapitel 4.1) unter Verwendung des MFCs als VoD-Stellvertreterserver ausschnittsweise umgesetzt und untersucht. Das Hauptaugenmerk des Szenarios liegt auf der Analyse des Delivery Systems, das beim MFC mittels adaptiver Bitratenanpassung einen erweiterten Funktionsumfang zu den in Kapitel 4.1 genannten Konzepten liefert. Es wird untersucht, wie der gesamte Ablauf, angefangen bei der Integration des Videos in die Bibliothek bis hin zur Übertragung des Videomaterials an den Client, durchgeführt wird. Es soll durch die praktische Umsetzung geklärt werden, wie die adaptive Bitratenanpassung im Detail funktioniert und welche Voraussetzungen hierfür ggf. eingehalten werden müssen. Hierbei wird beobachtet, wie die Adaption der Bandbreite stattfindet, wer den Wechsel auf eine andere Bandbreite veranlasst (Client oder MFC) bzw. wie dem gegenüber der Wechsel der Bandbreite signalisiert wird. Darüber hinaus wird untersucht, wie sich die Bitratenadaption bei der Änderung äußerer Gegebenheiten, wie z. B. dem Schwanken der verfügbaren Übertragungskapazität, verhält.
Hieraus resultierend soll die Fragestellung geklärt werden, inwiefern der MFC für die Bereitstellung von VoD-Angeboten in ISP-Netzen geeignet ist.

5.2.2 Testaufbau / Untersuchungsmethodik

Die Technik zur adapativen Bitratenanpassung bei der Übertragung von Videoinhalten wird bei Juniper unter dem Namen SmoothFlow geführt. Die Videodaten, die mithilfe dieser SmoothFlow-Funktionalität übertragen werden, unterliegen dabei relativ strikten Anforderungen an die verwendeten Dateiformate und Codecs. Die Videos müssen in Flash Video Containern (.flv) oder MPEG-4 Containern (.mp4) vorliegen. Als Videocodec wird die Verwendung von H.264 vorausgesetzt. Die Audiospuren müssen im MP3- oder AAC-Format vorliegen.
Um ein vorhandenes Video durch den MFC mit der SmoothFlow-Funktionalität übertragen zu können, ist es notwendig, die Videodaten durch die in Abbildung 5.2 dargestellten Schritte vorzubereiten. Der MFC unterstützt keine On-The-Fly Transkodierung des Videomaterials. Das Original-Video muss also bereits vor der Übertragung in den unterschiedlichen Bitraten, in denen es zur Verfügung stehen soll, in jeweils einzelnen Dateien auf dem Herkunftsserver vorliegen. Diese Dateien, die dasselbe Video mit unterschiedlichen Bitraten beeinhalten, werden im Weiteren als Bitraten-Videos bezeichnet. Diese Bitraten-Videos werden in einzel-

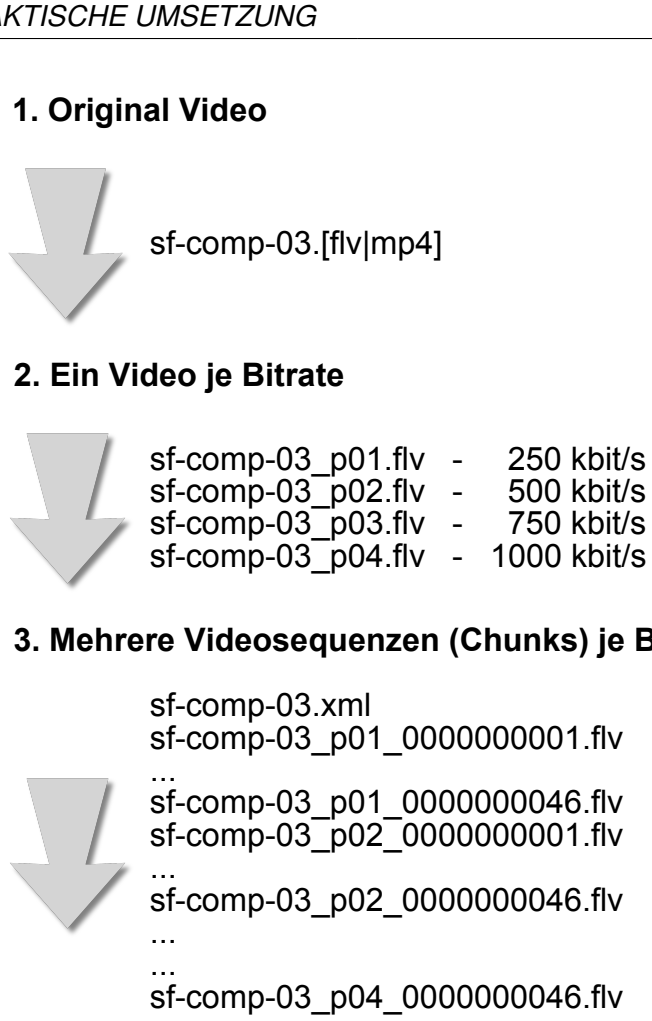


Abbildung 5.2: Schritte vom Originalvideo zum SmoothFlow-fähigen Videomaterial

ne kurze Sequenzen geteilt und als Dateien gespeichert. Von diesem Zeitpunkt an ist das Videomaterial bereit für die Übertragung mittels SmoothFlow.

Alle Bitraten-Videos müssen in derselben räumlichen Auflösung und Bildwiederholrate vorliegen, so dass das „Umschalten“ zwischen den Bitraten-Videos nur eine Änderung der Bitrate, jedoch nicht eine Änderung der dargestellten räumlichen Auflösung oder Bildwiederholrate zur Folge hat. Dies geschieht bei der Encodierung des Videomaterials durch die Anpassung der Quantisierungsmatrix, durch die der mögliche Wertebereich eines jeden Macroblocks eingeschränkt wird. Dies führt bei der anschließenden Anwendung der Entropiekodierung zu einer verringerten Datenmenge [Richardson (2003)].

Die SmoothFlow-Funktionalität bedingt das Vorhandensein von Keyframes an denselben Stellen über alle Bitraten-Videos hinweg. Hierdurch wird das Umschalten der Bitraten unterstützen.

Zur Erfüllung dieser Anforderung wird ein festes Keyframe-Intervall beim Enkodieren der Bitraten-Videos angegeben. Dies veranlasst die Enkodierungssoftware alle n Bilder ein Keyframe in das resultierende Videomaterial einzufügen, wodurch jeweils eine Group-of-Pictures (GOP) fester Länge entsteht [Richardson (2003)]. Für die Konfiguration des festen Keyframe-Intervalls führt das Administrationshandbuch des MFCs ein Intervall von zwei Sekunden als optimal an.
Zur Erstellung der Bitraten-Videos stehen laut Juniper Networks (2010) zwei Vorgehensweisen zur Verfügung. Im Folgenden wird zunächst auf die Vorbereitung mittels Cloud-Diensten und anschließend auf die eigenständige Vorbereitung eingegangen.

Enkodieren mittels Cloud-Dienst Um das vorliegende Videomaterial in die geforderten Formate und Bitraten zu konvertieren, ist im Admin-Guide des MFCs [Juniper Networks (2010)] unter anderem die Verwendung von Cloud-Diensten beschrieben. Cloud-Dienste wie encoding.com stellen bereits vorgefertigte Enkodierungs-Profile für die Verwendung des Videomaterials mit dem MFC zur Verfügung, so dass keine großen Investitionen im Bereich des Videoenkodierens getätigt oder KnowHow aufgebaut werden müssen. Die vorhandene Original-Videodatei wird auf einem Server gespeichert, auf dem ein File Transfer Protocol (FTP)-Serverdienst installiert ist. Unter Zuhilfenahme der von Juniper Networks zur Verfügung gestellten Skripte wird der Cloud-Dienst angewiesen, die Videodatei vom entsprechenden FTP-Server herunterzuladen und die gewünschten Bitraten-Videos zu erzeugen. Nach erfolgreichem Abschluss des Enkodierprozesses werden die Bitraten-Videos durch den Cloud-Dienst auf demselben FTP-Server, von dem das Original-Video stammt, abgelegt. Um das Videomaterial weiter für die Übertragung per SmoothFlow vorzubereiten, werden die Videodaten anschließend in Chunks aufgeteilt (gestückelt). Auch hierzu liefert Juniper Networks ein vorgefertigtes Skript, dem lediglich die Bitraten-Videos übergeben werden müssen. Das Skript liest die Videodateien für die einzelnen Bitraten ein und zerlegt jedes dieser Bitraten-Videos in kleine Sequenzen. Die resultierenden Chunks beginnen jeweils mit einem Keyframe und entsprechen in der Länge dem beim Enkodieren angegebenen Keyframeintervall.
Zu jedem Satz von Videos wird durch das Skript, das auch für die Stückelung der unterschiedlichen Bitraten-Videos zuständig ist, eine Extensible Markup Language (XML)-Datei generiert und im selben Verzeichnis wie die Bitraten-Videos abgelegt. Diese XML-Datei enthält Informationen über die unterschiedlichen verfügbaren Bitraten eines Videos.
Die gestückelten Videodateien werden anschließend zusammen mit der generierten XML-Datei auf dem Herkunftsserver, der für das Ausliefern von Inhalten mittels HTTP konfiguriert ist, abgelegt. Der HTTP-Server ist nicht in der Lage, die Übertragung mittels SmoothFlow zu unterstützen. Von hier ist es dem MFC jedoch möglich, auf An-

frage eines Inhaltekonsumenten das aufbereitete Videomaterial in den lokalen Zwischenspeicher zu laden und von dort auszuliefern. Darüber hinaus ist es möglich, die Videodaten mittels FTP explizit auf die MFCs auszubringen, so dass die Auslieferung bei Erstanfrage ohne Verzögerung durch weitere Interaktion mit dem Herkunftsserver geschehen kann.

Eigenständiges Enkodieren Beim eigenständigen Enkodieren ist es erforderlich, dass die Videos bereits in den unterschiedlichen Bitraten vorliegen. Das Enkodieren kann unter Zuhilfenahme jedweder Software geschehen, solange das resultierende Videomaterial die bereits erläuterten Anforderungen bezüglich Keyframes, Audio- / Videocodec etc. erfüllt.

Um das Videomaterial für die SmoothFlow-gestützte Übertragung verfügbar zu machen, ist es nötig, eine Profildatei (AssetDescription) zu erstellen, in der die verschiedenen Bitraten-Videos zu einem Profil zusammengefasst werden.

Wie in Listing 5.1 dargestellt wird zunächst über den Parameter `Profiles` die Anzahl der verfügbaren Bitraten-Videos definiert, im vorliegenden Beispiel sind dies vier. Der Parameter `Frame Rate` gibt die Anzahl der Bilder pro Sekunde an, mit der das Videomaterial enkodiert ist. Das `KeyFrameInterval` definiert, in welchem Abstand Keyframes enthalten sind. Dieser Parameter korrespondiert mit dem festen Keyframeintervall, das auch beim Enkodieren der Bitraten-Videos angegeben wird. Hierüber wird bestimmt, in welchen Intervallen auf eine andere Bitrate gewechselt werden kann. Die Angabe dieses Wertes hat in Sekunden zu erfolgen. Die Werte `Profile_X` definieren die unterschiedlichen Profile und ihre jeweilige Bitrate. Die Asset-Datei im Listing 5.1 definiert Profile mit den Bitraten 250, 500, 750 und 1000 kbit/s. Unter Zuhilfenahme der `URI` Parameter werden die URLs zu den Bitraten-Videos angegeben. Hierbei korrespondiert der Dateiname mit der zuvor getätigten Definition der Profile. Das Schema sieht wie folgt aus:

„*[Name]*_p*[Profil Nr.]*.flv“.

Der Parameter `Assured Flow rate` ist ein binärer Parameter, er kann die Werte 0 und 1 annehmen. Der Wert 0 deaktiviert, der Wert 1 aktiviert die Verwendung der AssuredFlow Funktionalität [Juniper Networks (2010)], durch die auf der Schnittstelle des MFCs die durch `AFR Threshold` definierte Bandbreite für die Übertragung des Videomaterials reserviert wird.

Die Profildatei wird nach der Erstellung zusammen mit den Bitraten-Videos auf dem im Profil definierten Herkunftsserver veröffentlicht. Nach der Ausbringung der Dateien auf dem HTTP-Herkunftsserver wird die Verarbeitung des Videomaterials durch den MFC angestoßen. Hierzu wird unter Verwendung des MFCs als Interception Proxy eine Inhaltsanfrage gestellt, in der die AssetDescription (siehe Listing 5.1) angefordert wird. In diesem Fall ist die AssetDescription unter dem Dateinamen sf-comp-03.dat in dem selben Ordner wie die Bitraten-Videos verfügbar. Um den MFC anzuweisen, die weitere Vorverarbeitung durchzuführen, ist es notwendig, die AssetDescription unter Verwendung des Parameters `sf` mit dem Wert 4 abzurufen.

```
Profiles:4
Frame Rate:24
KeyFrameInterval:2
Profile_1:250
Profile_2:500
Profile_3:750
Profile_4:1000
...
URI: http://users.informatik.haw-hamburg.de/~vahlen_m/sf-comp-03_p01.flv
URI: http://users.informatik.haw-hamburg.de/~vahlen_m/sf-comp-03_p02.flv
URI: http://users.informatik.haw-hamburg.de/~vahlen_m/sf-comp-03_p03.flv
URI: http://users.informatik.haw-hamburg.de/~vahlen_m/sf-comp-03_p04.flv
...
Assured Flow rate:1
AFR Threshold:600
```

Listing 5.1: SmoothFlow Asset (sf-comp-03.dat)

Fällt diese Inhaltsanfrage in den für die SmoothFlow-Funktionalität konfigurierten Namensraum, interpretiert der MFC diese Inhaltsanfrage als Verarbeitungsanforderung und lädt das in der AssetDescription definierte Videomaterial in seinen Zwischenspeicher. Von hier aus wird die Stückelung der einzelnen Bitraten-Videos vorgenommen. Die auf dem MFC zwischengespeicherten Dateien entsprechen dem in Abbildung 5.2 unter Punkt 3 dargestellten Schema.

5.2.2.1 Wiedergabesoftware

Um die Übertragung mittels adaptiver Bitratenanpassung zu nutzen, wird eine Wiedergabesoftware mit speziellen Eigenschaften benötigt. Die Wiedergabesoftware muss unter anderem anhand der aktuellen Übertragungsrate erkennen, ob das Videomaterial ggf. mit einer höheren Qualität übertragen werden kann oder ob diese sogar verringert werden muss.
Juniper Networks stellt einen Referenzplayer für die Verwendung der SmoothFlow-Funktionalität zur Verfügung. Hierbei handelt es sich um einen in Adobe Flash programmierten Client. Dieser kann durch die Verwendung der Flash Technologie relativ einfach in Webseiten eingebunden werden. Für diese Integration in eine Webseite ist es erforderlich, dass der Flash-Container (SWF-Datei), der den Referenzplayer enthält, referenziert wird. Darüber hinaus wird über Anweisungen in der HTML-Datei definiert, welches Video im Player wiedergegeben werden soll.

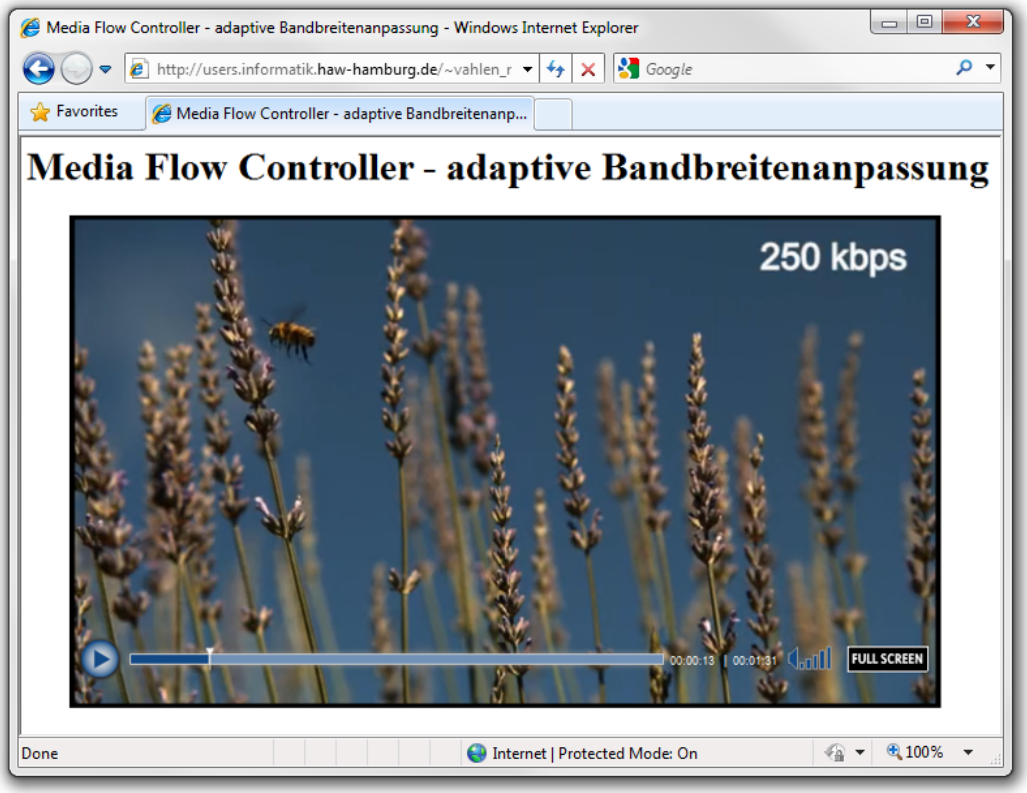


Abbildung 5.3: Webseite mit integriertem SmoothFlow-Player

Für den Testaufbau wird eine Webseite verwendet, in der lediglich der Referenzplayer und eine Überschrift angezeigt werden. Nach dem Aufrufen der Webseite werden alle eingebetteten Objekte geladen und wie in Abbildung 5.3 zu sehen im Browser dargestellt.
Nachdem der Referenzplayer auf dem Client ausgeführt wird, beginnt die SmoothFlow unterstütze Wiedergabe des Videomaterials. Die Videodaten werden an den Client versendet, vom Referenzplayer verarbeitet und dargestellt.

5.2.2.2 Netzwerkanalyse

Um die übertragenen Daten für die weiteren Analysen verfügbar zu machen, wird die Netzwerkanalysesoftware Wireshark[7] auf dem Client-Computer installiert. Bei dieser Software handelt es sich um einen Netzwerksniffer, der in der Lage ist, die gesamte Netzwerkkommunikation des Client-Computers mitzuschneiden und für die weitere Auswertung zu speichern.

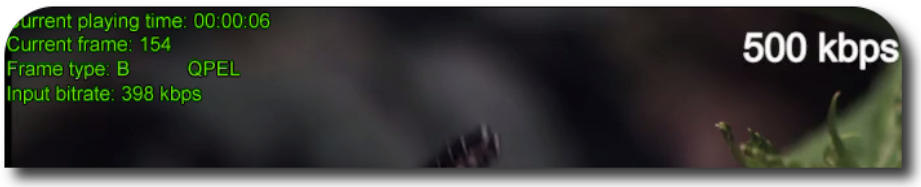
[7] http://www.wireshark.org

Zur Reduktion der mitzuschneidenden Daten wird ein Erfassungsfilter (Capture Filter) gesetzt, der nur Datenpakete mitschneidet, die an TCP-Port 80 gesendet werden oder die von diesem Port stammen. Durch diese vorgeschaltete Filterung der Pakete wird die Last, die durch das Speichern der mitgeschnittenen Kommunikation auf den Client einwirkt, verringert.
Mithilfe von Wireshark werden anschließend die übertragenen Videodaten aus der mitgeschnittenen Kommunikation extrahiert und als Videodatei für die weitere Auswertung gespeichert. Des Weiteren werden die HTTP-Anfragen, die resultierende Antwort des MFCs, die Dauer und übertragene Datenmenge der einzelnen TCP-Verbindungen erfasst.

5.2.2.3 Videoanalyse

Um die Bitraten-Videos und das mitgeschnittene Videomaterial zu analysieren, kommt die Multimediabibliothek FFmpeg[8] in Zusammenspiel mit der Wiedergabesoftware Mediaplayer Classic[9] zum Einsatz. Während des Abspielens der Videodateien im Mediaplayer Classic ist die FFmpeg Bibliothek in der Lage, unterschiedliche statistische Werte bezüglich der verarbeiteten Videodaten in eine Comma-Separated Values (CSV)-Datei zu schreiben. Für die Auswertung und den Vergleich der unterschiedlichen Videos werden die Werte Framenummer und Bitrate für den Export in diese CSV-Datei gewählt. So können nach der einmaligen Wiedergabe jedes Videos Auswertungen aus den exportierten Werten generiert werden.
Als Besonderheit der für den Test verwendeten Bitraten-Videos ist anzumerken, dass die ungefähre Bitrate der einzelnen Dateien im Videomaterial als statischer Text eingeblendet ist. Diese Information ist in Abbildung 5.3 als auch in Abbildung 5.4 in der rechten oberen Ecke des dargestellten Videos zu sehen. Je nach Bitraten-Video sind die Werte 250, 500, 750, 1000, 1250, 1500 und 2000 kbit/s abgebildet. Da es sich bei diesen Zahlen um fest in das Bildmaterial integrierte Informationen handelte, die nicht erst durch den SmoothFlow-Player errechnet und eingeblendet werden, sind diese Zahlen ebenfalls in dem mittels SmoothFlow übermittelten und mitgeschnittenen Video enthalten. In Abbildung 5.4 sind zwei Bildausschnitte der übertragenen Datei zusammengefasst dargestellt.
Die in Abbildung 5.4 dargestellten Bildausschnitte sind darüber hinaus mit weiteren Informationen angereichert, die am linken oberen Bildrand in grüner Farbe dargestellt sind. Hierbei handelt es sich um Informationen, die die Multimediabibliothek FFmpeg bei der Wiedergabe der Videodaten durch den Mediaplayer Classic in den Videostream mit einbringt und die zeigen, welcher Frame des Videos aktuell dargestellt wird (`Current Frame`), zu welcher Sekunde des Videos dieser Frame gehört (`Current playing time`) und welche durchschnittliche Bitrate das Video vom ersten bis zu dem aktuell dargestellten Frame besitzt (`Input bitrate`). Mithilfe dieser Zusatzinformationen ist es möglich, die Übergänge von einem auf einen anderes Bitraten-Video framegenau zu bestimmen.

[8] http://www.ffmpeg.org
[9] http://mpc-hc.sourceforge.net



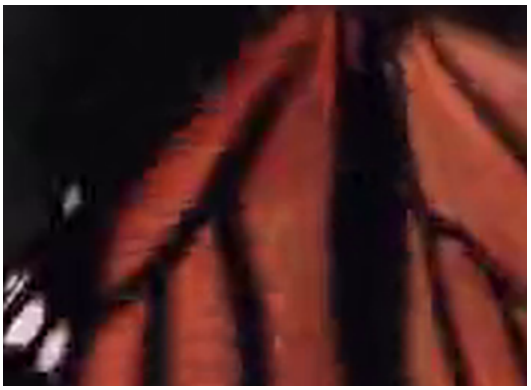
(a) Bildauschnitt von Frame 154 bei Sek. 06; 500 kbit/s

(b) Bildauschnitt von Frame 1213 bei Sek. 48; 1250 kbit/s

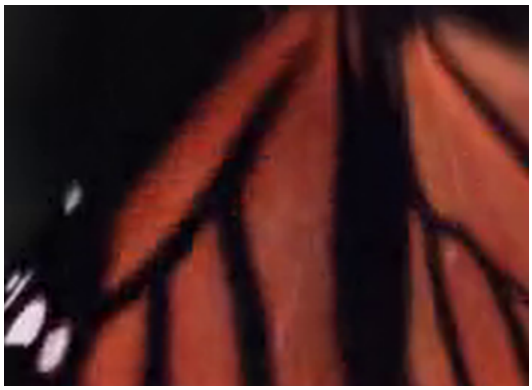
Abbildung 5.4: Bildauschnitte des übertragenen Videos
Bildauschnitte des mittels SmoothFlow übertragenen Videos, angereichert mit Informationen der FFmpeg-Bibliothek.

Abbildung 5.5 zeigt die Bildqualität der unterschiedlichen Bitraten-Videos im direkten Vergleich. Hierzu ist ein Ausschnitt desselben Frames der unterschiedlichen Bitraten-Videos vergrößert abgebildet. Unter 5.5(a) ist das gesamte Bild in der 1500 kbit/s Qualität zu sehen. Die Qualitätsunterschiede zwischen 250 kbit/s und 500 kbit/s sind deutlich zu erkennen. Zwischen dem 500 kbit/s und dem 1000 kbit/s Video sind sie ebenfalls noch gut zu erkennen. Der Qualitätsunterschied vom 1000 kbit/s zum 1500 kbit/s Ausschnitt ist bereits deutlich geringer.

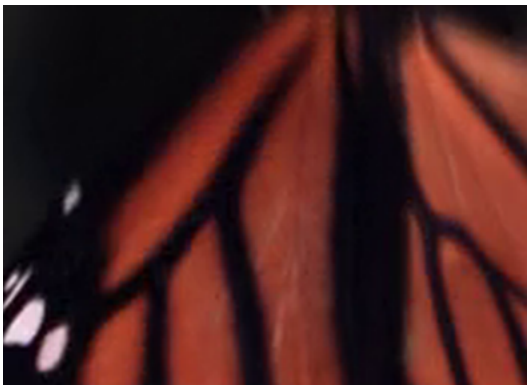
(a) Gesamtbild bei 1500 kbit/s



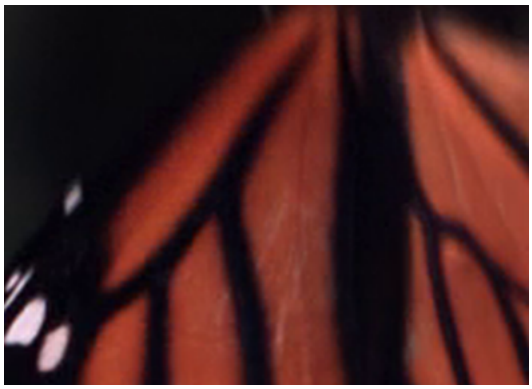
(b) Bildausschnitt 250 kbit/s



(c) Bildausschnitt 500 kbit/s



(d) Bildausschnitt 1000 kbit/s



(e) Bildausschnitt 1500 kbit/s

Abbildung 5.5: Bildqualität unterschiedlicher Bitraten-Videos im Vergleich.

```
<xml>
   <sf_version>1.0</sf_version>
   <n_profiles>7</n_profiles>
   <profile_info>
      <profile_map>
         <profile_name>p01</profile_name>
         <br>320</br>
      </profile_map>
      <profile_map>
         <profile_name>p02</profile_name>
         <br>568</br>
      </profile_map>
      <profile_map>
         <profile_name>p03</profile_name>
         <br>816</br>
      </profile_map>
      <profile_map>
         <profile_name>p04</profile_name>
         <br>1072</br>
      </profile_map>

      ...

   </profile_info>
   <afr_th>137500</afr_th>
</xml>
```

Listing 5.2: XML-Profildatei für SmoothFlow-Player (sf-comp-03.xml). Beschreibt verfügbare Videoprofile und deren Bitrate.

5.2.3 Ergebnisse

Nach dem Abrufen und Darstellen der Webseite inklusive des SmoothFlow-Referenzplayers fordert dieser umgehend die Videodatei `/sf-comp-03_p01.flv` mit den angehängten Parametern `sid`, `sf` und `pf` an. Diese Inhaltsanfrage ist in Tabelle 5.1 zum Zeitpunkt 0,0s zu sehen. Alle weiteren Anfragen, die der SmoothFlow-Player an den MFC stellt, werden unter Verwendung einer zweiten TCP-Verbindung durchgeführt, die in den Tabellen 5.1 und 5.2 als Konversation (Konv.) bezeichnet wird. Anschließend wird die XML-Datei (siehe Listing 5.2) mit der Definition der unterschiedlichen verfügbaren Profile angefordert.

In Tabelle 5.1 ist ersichtlich, dass weitere HTTP-Anfragen in der zweiten TCP-Verbindung abgesetzt werden. Bei diesen nachfolgenden Inhaltsanfragen wird immer dieselbe Datei angefragt, es ändern sich lediglich die übergebenen Parameter `nc` und `pf`.

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-length: 116
Date: Wed, 08 Dec 2010 05:54:27 GMT
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
    <HEAD>
        <TITLE>200 OK</TITLE>
    </HEAD>
    <BODY>200 OK</BODY>
</HTML>
```

Listing 5.3: Positivmeldung auf Inhaltsanfrage mit dem Parameter sf=3

Die Anfragen der ersten und der zweiten TCP-Verbindung unterscheiden sich nur in der Anzahl der übergebenen Parameter. Des Weiteren unterscheiden sich die Antworten des MFCs auf die Inhaltsanfragen. In der ersten TCP-Verbindung wird auf die gestellte Inhaltsanfrage das Video ausgeliefert, in der zweiten TCP-Verbindung wird bei jeder Inhaltsanfrage, die an die Video-URL gestellt wird, die in Abbildung 5.3 dargestellte Statusmeldung an den Client ausgeliefert.

Zeit	Konv.	Src.	Dest.	Anfrage
0,0	***1***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=1**&**pf=2**
1,8	***2***	Client	MFC	GET /sf-comp-03.xml?**sid=36939502**&**sf=2**
10,7	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=5**&nc=63117
18,9	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=4**&nc=88125
23,2	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=3**&nc=14630
31,4	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=5**&nc=31038
43,6	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=6**&nc=56150
51,6	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=5**&nc=56058
55,6	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=4**&nc=5612
63,7	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=6**&nc=64410
71,9	***2***	Client	MFC	GET /sf-comp-03_p01.flv?**sid=36939502**&**sf=3**&**pf=5**&nc=81018

Tabelle 5.1: SmoothFlow HTTP-Anfragen des Clients. Initiale Videoanfrage des Clients in Konv. 1 sowie nachfolgende Anforderungen zum Bitratenwechsel in Konv. 2.

Ausgewählte statistische Daten der Netzwerkkommunikation zwischen MFC und Client sind in Tabelle 5.2 dargestellt. Aus der Relation der Werte `Bytes A→B` zu `Bytes B→A` ist ersichtlich, dass die erste TCP-Verbindung für den Transport einer größeren Datenmenge in Richtung des Clients genutzt wurde. In der zweiten TCP-Verbindung sind mehr Daten vom Client zum MFC übertragen worden. Beide Verbindungen haben annähernd gleich lange bestanden.

Konv.	Dauer	Gerät A	TCP Port A	Gerät B	TCP Port B	Bytes A→B	Bytes A←B
1	72,25s	Client	55867	MFC	http	219.617	13.218.538
2	72,38s	Client	55877	MFC	http	7.028	4.609

Tabelle 5.2: Statistik der SmoothFlow Konversationen. In Konversation 1 wird das Video zum Client übertragen. Konversation 2 dient der Übertragung der Umschaltanweisungen an den MFC.

Im Weiteren werden die mithilfe der Multimediabibliothek FFmpeg erhobenen Daten betrachtet. Abbildung 5.6 stellt die durchschnittliche Bitrate der Bitraten-Videos `/sf-comp-03_p01.flv` bis `/sf-comp-03_p01.flv` dar. Die eingezeichneten Bitraten spiegeln für jeden Frame die durchschnittliche Bitrate der Videosequenz vom ersten Frame bis einschließlich des aktuell angezeigten Frames wieder, d. h. die Absolutgrößen jedes einzelnen vorangegangenen Frames werden aufsummiert und durch die Anzahl der dargestellten Frames geteilt. Zudem ist die durchschnittliche Bitrate des mittels SmoothFlow übertragenen Videos aufgetragen.
Anhand der Abbildung ist zu erkennen, dass sich das durch SmoothFlow übertragene Video nicht in die Struktur der zuvor erhobenen Messwerte der Bitraten-Videos einreiht. Die Durchschnittsbitrate des übertragenen Videos ist einer stärkeren Änderung unterworfen als dies bei den Bitraten-Videos der Fall ist.
Aufgrund der Einschränkungen der FFmpeg-Bibliothek, nur die Durchschnittsbitrate zur Verfügung zu stellen, sind in Abbildung 5.6 keine abrupten Sprünge der Bitrate des übertragenen Videos ersichtlich. Ein Wechsel der Bitrate ist hingegen aus der Änderung der Steigung der Kurve zu erkennen.
Die dargestellten Werte spiegeln ausschließlich die Videobitrate wieder. Die Audiobitraten wurden nicht berücksichtigt, da die unterschiedlichen Videos über dieselbe Audiospur verfügen und somit keine Änderung der Audiobitrate in Relation zu den anderen Bitraten-Videos entsteht.

5.2.4 Auswertung

Durch die Reihenfolge der Inhaltsanfragen (siehe Tabelle 5.1), bei denen zuerst das Video und anschließend die Profildatei angefragt wird, ist sichergestellt, dass die Wiedergabe des Videomaterials möglichst schnell beginnt. Es wird zu Beginn der Übertragung keine Zeit in das vorgeschaltete Laden von Verwaltungsinformationen investiert. Über den Parameter `pf` mit dem Wert 2 wird bei der initialen Anfrage des Videos zunächst das Bitraten-Video des zweiten Profils vom MFC abgerufen. Die Bitrate des zweiten Profils ist im Gegensatz zu den weiteren Profilen relativ gering. Daher füllt sich der Puffer des Clients schnell und die Wiedergabe kann möglichst früh starten.

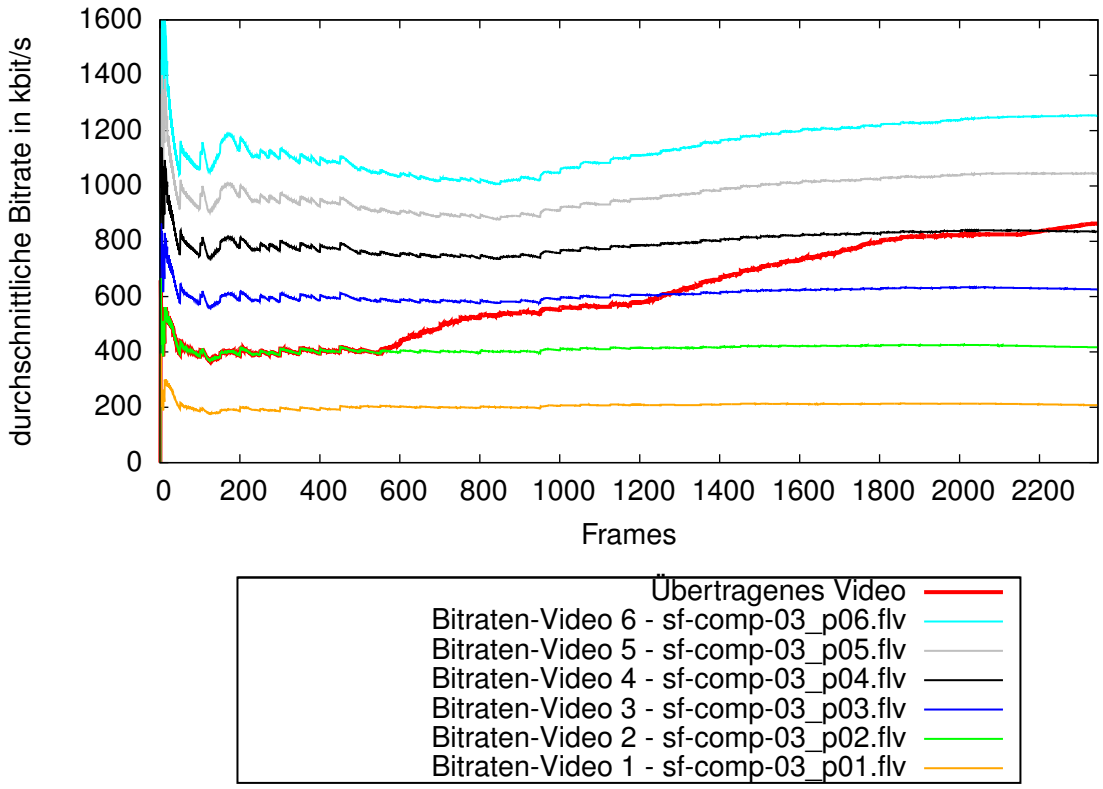


Abbildung 5.6: Durchschnittsbitrate aller Videos

Profil Parameter	Bitrate
pf=1	250 kbit/s
pf=2	500 kbit/s
pf=3	750 kbit/s
pf=4	1000 kbit/s
pf=5	1250 kbit/s
pf=6	1500 kbit/s
pf=7	2000 kbit/s

Tabelle 5.3: SmoothFlow Bitrate je Profil

Der Statistik in Tabelle 5.2 ist anhand der übertragenen Datenmenge in Richtung des Clients zu entnehmen, dass das Video in der ersten TCP-Verbindung angefordert und übertragen wird.
Die Übermittlung der XML-Datei würde bei Anfrage in derselben TCP-Verbindung erst nach dem Abschluss der erfolgreichen Übertragung des Videos beginnen, da sich die Antworten auf HTTP-Anfragen innerhalb einer TCP-Verbindung nicht überholen dürfen (HTTP-Pipelining [Fielding u. a. (1999)]). Die Anfrage und Übertragung der XML-Profildatei sowie alle weiteren Anfragen geschehen somit in der zweiten TCP-Verbindung.
Nach erfolgreichem Empfang der XML-Datei (Tabelle 5.1; Zeit 1,8s) sind dem SmoothFlow-Player alle verfügbaren Profile für das aktuell in der Wiedergabe befindliche Video bekannt. Aufgrund der in der Startphase begonnenen und fortwährend weitergeführten Ermittlung der Übertragungsrate ist der SmoothFlow-Player in der Lage, dass am besten geeignete Profil für die Übertragung zu ermitteln und beim MFC anzufordern.
Die zweite TCP-Verbindung wird im weiteren Verlauf der Videoübertragung für die Anforderung der unterschiedlichen Profile verwendet. Sobald die verwendete Bitrate von der ermittelten optimalen Bitrate abweicht, wird eine Inhaltsanfrage in der zweiten TCP-Verbindung an den MFC gesendet. Diese Inhaltsanfragen sind in Tabelle 5.1 in der Zeit zwischen 10,7s und 71,9s dargestellt. Sie enthalten die Parameter `pf`, `sid`, `nc` und `sf`. Der Parameter `pf` gibt die zu verwendende Profil-ID an. Eine Auflistung der im Testaufbau verwendeten Bitraten und der zugehörigen Profilparameter ist in Tabelle 5.3 aufgeführt.
Um zu signalisieren, dass beide TCP-Verbindungen zu einer Sitzung eines Inhaltekonsumenten gehören, wird der Parameter `sid` bei jeder Anfrage übergeben. Der Parameter `sf` ist bei der Anfrage in der ersten TCP-Verbindung auf den Wert 1 gesetzt, was bedeutet, dass auf diese Anfrage in der entsprechenden Verbindung das Video übertragen werden soll. Bei allen weiteren Anforderungen die bezüglich dieser Übertragung an den MFC gesendet werden, ist der Parameter `sf` mit dem Wert 3 versehen. Inhaltsanfragen, in denen der Parameter `sf` den Wert 3 aufweist, werden vom MFC als Steueranweisungen interpretiert. Auf diese Steueranweisungen wird nicht durch das Versenden des Videomaterials reagiert. Sie werden verarbeitet und anschließend wird mit einer Statusmeldung beantwortet, wie sie

in Listing 5.3 aufgeführt ist. Durch diesen Mechanismus wird signalisiert, dass die Anfragen vom MFC erfolgreich verarbeitet wurde. Der MFC passt die über die erste TCP-Verbindung versendeten Daten entsprechend der durch die Steueranweisungen in der zweiten TCP-Verbindung angeforderten Videodaten an.
Bei der Übertragung des Videos werden die einzelnen Teile eines Bitraten-Videos auf dem MFC nacheinander in den TCP-Datenstrom geschrieben. Das Bitraten-Video, das auf dem MFC in einzelnen, zwei Sekunden langen Sequenzen gespeichert ist, wird auf diese Weise wieder zusammengefügt und als Ganzes an den Client ausgeliefert.
Das zu übertragende Bitraten-Video bleibt gleich, solange keine neue HTTP-Anforderung für die Sitzung beim MFC eintrifft, die ein anderes Profil und somit eine andere Bitrate anfordert. Da alle Bitraten-Videos an denselben Stellen Keyframes besitzen, kann bei Auftreten eines Keyframes von einer zur anderen Bitrate gewechselt werden. Hierzu ermittelt der MFC die nächste erforderliche Videosequenz des Gesamtvideos sowie die gewünschte Bitrate und kopiert anschließend die benötigten Daten in den TCP-Datenstrom.
Ein solcher Wechsel der Bitraten wird zu Beginn eines Group-of-Picture (GOP)-Intervalls standardmäßig durch den verwendeten H.264 Codec unterstützt und benötigt somit keine Anpassung der Wiedergabesoftware. Lediglich eine Prämisse muss erfüllt sein, die Enkodierung der Bitraten-Videos muss mit einem fixed GOP-Intervall [Richardson (2010)] durchgeführt worden sein. Nach dem Abschluss der Videoübertragung wird die zweite Konversation, wie in Tabelle 5.2 an der Dauer der Konversation zu sehen, ebenfalls geschlossen. Der Steuerkanal wird nicht weiter benötigt.
In Abbildung 5.7 ist erneut die durchschnittliche Bitrate des übertragenen Videomaterials, ergänzt um Informationen, die durch das Analysieren des Videomaterials entstanden sind, eingezeichnet. Die senkrechten gestrichelten Linien kennzeichnen die Stellen im übertragenen Video, an denen ein Bitratenwechsel stattgefunden hat. Es ist klar erkennbar, welche Bitrate das übertragene Video zu jedem Zeitpunkt aufweist.

Im Weiteren wird eine Wiedergabesoftware näher betrachtet, die die adaptive Bitratenanpassung beherrscht. Da es sich bei dem SmoothFlow-Referenzplayer um eine Closed-Source Implementierung handelt, wird der durch den Entwicklungsleiter der SmoothFlow-Technologie als funktional ähnlich ausgewiesene Open Video Player (OVP) betrachtet. Dieser ist, wie auch der SmoothFlow-Player, in der Lage, Übertragungseigenschaften und Statistiken auszuwerten und diese für die Adaption der Videobitrate zu verwenden. In den Listings 5.4 - 5.8 sind die essentiellen Bestandteile des OVP[10], die zur Adaption der Bandbreite verwendete werden, mithilfe von Pseudocode dargestellt.
In Listing 5.4 ist die Funktion `prüfeRegeln()` abgebildet, die als Einstiegspunkt in die Bitratenanpassung dient. Die Funktion wird während der Übertragung des Videomaterials periodisch, alle 0,5 Sekunden durch einen Timer aufgerufen. Sie initiiert die Prüfung unter-

[10] http://openvideoplayer.sourceforge.net/

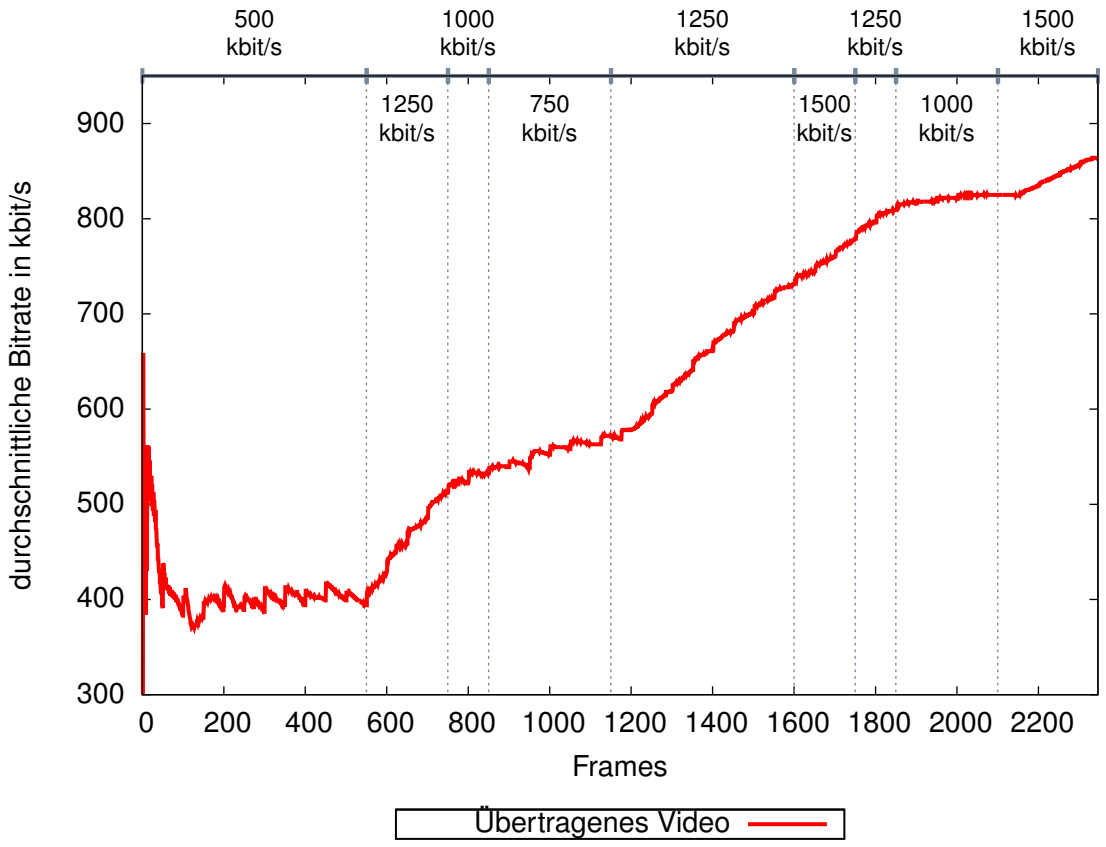


Abbildung 5.7: Durchschnittsbitrate des übertragenen Videos

```
1  FUNKTION prüfeRegeln(){
2      neuesProfil = maxProfile
3
4      FÜR_JEDE regel AUS bitratenregeln
5          profil = regel.berechneNeuesProfil()
6          WENN profil < neuesProfil UND profil != -1
7              neuesProfil = profil
8          WENN_ENDE
9      FÜR_JEDE_ENDE
10     wechseleProfil(neuesProfil)
11 }
```

Listing 5.4: Funktion prüfeRegeln()

```
12 SCHNITTSTELLE IWechselRegel{
13     FUNKTION berechneNeuesProfil(){
14         RETURN optimalesProfil
15     }
16 }
```

Listing 5.5: Schnittstelle IWechselRegel

schiedlicher Regeln zur Bitratenanpassung. Hierzu wird vor Beginn der Videoübertragung die `bitratenregeln`-Liste mit Objekten, die die `IWechselRegel`-Schnittstelle implementieren, gefüllt. Durch das Hinzufügen oder Entfernen bestimmter Regeln zu dieser Liste ist es möglich die für die Datenübertragung verwendeten Kriterien individuell anzupassen.
Das Interface `IWechselRegel` (siehe Listing 5.5) implementiert die Funktion `berechneNeuesProfil()`, mithilfe derer eine konkrete Implementierung einer Regel, das nach ihren Kriterien optimale Videoprofil bestimmen kann. Durch die Verwendung dieser Schnittstelle ist es möglich, den Algorithmus um zusätzliche Regeln zu erweitern und weitere individuelle Kriterien zu definieren.
Die Funktion `prüfeRegeln()` führt alle in der `bitratenregeln`-Liste enthaltenen Regeln aus und wählt das geringste der von den einzelnen Regeln zurückgelieferte Profil für die Übertragung aus. Dieses Profil ist verwendbar, da die Rückgabe der `berechneNeuesProfil()`-Funktion so zu interpretieren ist, dass alle Profile kleiner gleich dem zurückgegebenen Profil, die jeweilige Regel erfüllen. Somit bietet die Verwendung des niedrigsten zurückgelieferten Videoprofils die beste Qualität, die den Kriterien aller Regeln gleichzeitig entspricht.

```
17 KLASSE PufferRegel implementiert IWechselRegel{
18     pufferVoll = falsch
19
20     FUNKTION berechneNeuesProfil(){
21         WENN pufferVoll == wahr
22             RETURN -1
23         SONST
24             RETURN [kleinstes Profil]
25         WENN_ENDE
26     }
27 }
```

Listing 5.6: Klasse PufferRegel

In Listing 5.6 ist die Klasse `PufferRegel`, die die Schnittstelle `IWechselRegel` implementiert, abgebildet. Diese Regel beachtet den Pufferstand des OVP. Sollten sich keine Videodaten mehr im 3 Sekunden umfassenden Wiedergabepuffer befinden, wird das kleinste verfügbare Profil für die Wiedergabe gewählt. Dies ist eine einfache Art des Herunterregelns der Bandbreite. Diese Regel ist ausschließlich in der Lage, die Bitrate auf den geringsten Wert zu setzen. Eine Erhöhung der Bitrate wird durch diese Regel nicht geboten. Des Weiteren muss angemerkt werden, dass die Verringerung der Bitrate bei leergelaufenem Puffer für eine reibungslose Wiedergabe zu spät ist, da zu diesem Zeitpunkt keine weiteren Daten mehr zur Verfügung stehen und die Wiedergabe somit unweigerlich aussetzt. Eine solche Situation soll jedoch durch die adaptive Bitratenanpassung generell vermieden werden. Durch den radikalen Schritt das niedrigste Profils zu wählen wird zu dem Zeitpunkt in dem der Puffer leergelaufen ist, versucht den Puffer möglichst schnell wieder mit einer ausreichend langen Videosequenz zu füllen, so dass die Wiedergabe rasch wiederaufgenommen werden kann.

In Listing 5.7 ist die Klasse `VerworfeneFramesRegel`, die die Schnittstelle `IWechselRegel` implementiert, abgebildet. Die Regel verwendet zur Ermittlung des zu verwenden Videoprofils die durchschnittliche Anzahl an verworfenen Videoframes. Hierzu sind drei unterschiedliche Schwellwerte definiert. Wurden im Schnitt mehr als 24 Frames innerhalb der letzten fünf Sekunden verworfen, wählt der OVP das kleinstmögliche Profil.
Sind innerhalb dieses Zeitraums durchschnittlich 20-24 Frames verworfen worden, wird das aktuelle Profil um zwei Indizes verringert. Sind durchschnittlich zwischen 10 und 20 Frames verworfen worden, wird der angeforderte Profilindex um eins verringert. Hierdurch soll die Zeit verringert werden, die der OVP für die Verarbeitung und Darstellung der einzelnen Videoframes benötigt. Das die Wiedergabesoftware Frames verwirft hängt in der Regel mit der Performance des Gerätes zusammen, auf dem die Wiedergabe stattfindet. Können

```
28 KLASSE VerworfeneFramesRegel implementiert IWechselRegel{
29     FUNKTION berechneNeuesProfil(){
30         WENN [∅Frame-Verwerfrate] >= [24 Frames]
31             RETURN [kleinstes Profil]
32         SONST WENN [∅Frame-Verwerfrate] >= [20 Frames]
33             WENN (aktuellesProfil - 2) >= 0
34                 RETURN (aktuellesProfil - 2)
35             SONST
36                 RETURN [kleinstes Profil]
37             WENN_ENDE
38         SONST WENN [∅Frame-Verwerfrate] >= [10 Frames]
39             WENN (aktuellesProfil - 1) >= 0
40                 RETURN (aktuellesProfil - 1)
41             SONST
42                 RETURN [kleinstes Profil]
43             WENN_ENDE
44         WENN_ENDE
45         WENN_ENDE
46         WENN_ENDE
47         RETURN -1
48     }
49 }
```

Listing 5.7: Klasse VerworfeneFramesRegel

die Videodaten nicht schnell genug aus dem Arbeitsspeicher geladen werden oder ist die CPU durch andere Prozesse bzw. durch die Anwendung von Videofiltern überlastet, werden ggf. Videoframes verworfen. Geschieht dies nur vereinzelt, ist keine Verringerung der Wiedergabequalität erkennbar, häufen sich diese Ereignisse jedoch, muss die Bitrate des Videos verringert werden, um die QoE durch eine schnellere Verarbeitung und damit eine verringerte Frame-Verwurfsrate wieder zu steigern.
Sollte beim Herunterregeln des Profils ein negativer Profilindex entstehen, wir das kleinste verfügbare Profil angefordert.
Die `VerworfeneFramesRegel` ist, wie auch die `PufferRegel`, in der vorliegenden Implementierung nur in der Lage eine Verringerung der Bitrate herbeizuführen. Die beiden Regeln greifen erst wenn Ausnahmesituationen eingetreten sind, sie sind nicht in der Lage, diese frühzeitig zu erkennen und ihnen entgegen zu wirken, ganz im Gegensatz zur `BandbreitenRegel`, die in Listing 5.8 dargestellt ist. Die Klasse `BandbreitenRegel` implementiert ebenfalls die `IWechselRegel`-Schnittstelle und verwendet die durchschnittliche Übertragungsrate zur Ermittlung des nach ihren Kriterien optimalen Profils. Es wird das Profil mit der höchsten Bitrate angefordert, das nach dem Aufschlag eines

```
50 KLASSE BandbreitenRegel implementiert IWechselRegel{
51     FUNKTION berechneNeuesProfil(){
52         neuesProfil = -1
53 
54         FÜR_JEDES profil AUS verfügbareProfile
55             WENN ((profil.bitrate + 15%) < ∅Bandbreite) UND (profil > ↘
               →neuesProfil)
56                 neuesProfil = profil
57             WENN_ENDE
58         FÜR_JEDES_ENDE
59 
60         WENN (neuesProfil > aktuellesProfil)
61             # Nur hochschalten wenn Frame-Verwerfrate und Puffer OK
62             WENN (∅Frame-Verwerfrate > 2) ODER (pufferVoll == falsch)
63                 neuesProfil = aktuellesProfil
64             WENN_ENDE
65         WENN_ENDE
66 
67         RETURN neuesProfil
68     }
69 }
```

Listing 5.8: Klasse BandbreitenRegel

Sicherheitspuffers von 15% der Bitrate die aktuelle durchschnittliche Übertragungsrate nicht übersteigt. In diesem Vorgehen liegt begründet, dass sowohl ein Profil mit höherer als auch ein Profil mit geringerer Bitrate gewählt werden kann. Durch das Aufschlagen von 15% auf die in der Profilbeschreibung angegebene Bitrate des Videos soll sichergestellt werden, das die kontinuierliche Übertragung und Wiedergabe des Videomaterials auch in komplexen Videoszenen, in denen die reale Bitrate über dem Durchschnittswert des Gesamtvideos liegt, sichergestellt werden kann.
Sollte aus der vorangegangenen Berechnung eine Erhöhung des Profils resultieren, wird zusätzlich sichergestellt, dass die Rate an durchschnittlich verworfenen Frames den Wert zwei nicht übersteigt. Des Weiteren wird vorausgesetzt, dass der Videopuffer mit Daten gefüllt ist.
Ist eine der beiden Anforderungen nicht erfüllt, wird das Erhöhen des Profils unterbunden. In diesen Fällen muss zunächst der Puffer gefüllt werden oder die Erhöhung des Profils muss generell unterbunden werden, wenn das Wiedergabegerät mit dem aktuell in der Wiedergabe befindlichen Profil bereits an der Performancegrenze arbeitet.
Die Berechnung der durchschnittlichen Übertragungsrate startet nach einer Übertragungszeit von einer Sekunde. Ab diesem Zeitpunkt wird das Intervall ausgeweitet, bis die durch-

schnittliche Bitrate über ein Intervall von fünf Sekunden gebildet wird. Folgend bleibt die Länge des Intervalls gleich und alte Messwerte werden verworfen.
Es ist wichtig, das Intervall für die Durchschnittsbildung zu beschränken, um auf die aktuellen Bandbreitenverhältnisse angemessen reagieren zu können. Gleichwohl müssen kurze Schwankungen der Übertragungsrate durch die Durchschnittsbildung ausgeglichen werden und dürfen nicht unmittelbar zu einer Änderung des übertragenen Profils führen. Sollte keine Durchschnittsbildung stattfinden, kann dies zu ständig schwankenden Profilanforderungen führen, was einen negativen Einfluss auf die QoE hat. Vor allem bei geringen Bitraten sind Sprünge zwischen unterschiedlichen Bitraten deutlich erkennbar, so dass eine kontinuierliche, niedrigere Bildqualität dem ständigen Wechsel zwischen zwei Profilen vorzuziehen ist.

Abschließend kann festgehalten werden, das die Videoübertragung mittels adaptiver Bitratenanpassung unter Verwendung der SmoothFlow-Funktionalität des MFC erfolgreich umgesetzt werden konnte. Die Ziele dieses Testszenarios waren, die genaue Arbeitsweise zu analysieren sowie die Voraussetzungen und Rahmenbedingungen zu ermitteln. In diesem Zuge stellte sich heraus, dass das zu verwendende Videomaterial strengen Anforderungen an Codecs und Codec-Einstellungen unterliegt. Darüber hinaus müssen die Videos bereits in unterschiedlichen Bitraten kodiert zur Verfügung stehen, um vom MFC verarbeitet werden zu können. Die starken Restriktionen bezüglich der verwendeten Codecs bergen nur wenige Probleme, da ohnehin mehrere Enkodierungsprozesse zur Erstellung der Bitraten-Videos nötig sind, sodass alle Videos in dem entsprechenden Format enkodiert werden können. Die Verwendung mehrerer Bitraten-Videos auf den MFCs stellt bei umfangreichen Videobibliotheken ggf. ein Problem dar, da die unterschiedlichen Bitraten-Videos einen gesteigerten Speicherbedarf implizieren.
Im Test wurde das Original-Video in sieben unterschiedliche Bitraten von ca. 250, 500, 750, 1000, 1250, 1500 und 2000 kbit/s enkodiert. Im Gegensatz zu Systemen, die On-The-Fly-Video-Transkodierung unterstützen und dadurch nur das Video mit der höchsten Videoqualität speichern müssen, wächst der Speicherbedarf der SmoothFlow-Funktionalität in dieser Konfiguration rein rechnerisch um den Faktor 3,625. Da die Audiospuren ebenfalls in jedem Video erneut gespeichert werden und sich die Bitrate der Audiospuren nicht ändert, liegt der tatsächliche Faktor noch etwas höher. Bei den verwendeten Dateien lag der Wert des benötigten Speichers bei dem 3,655-fachen.
Das einmalige Enkodieren und Speichern des Videomaterials in Bitraten-Videos bringt den Vorteil, dass persistenter Speicher in der Anschaffung sowie im Betrieb relativ billig ist. Bei der On-The-Fly-Transkodierung ist die Rechenleistung der entscheidende Faktor. Die CPU nimmt unter Last mehr Leistung auf und erzeugt mehr Wärme als eine HDD, was trotz des erhöhten Speicherbedarfs für die Methodik des Vorabenkodierens und Speichern spricht.
Das Abspielen des SmoothFlow-Materials erfordert keine Veränderungen am Videocodec, was eine native Integration dieser Technik in weitere Wiedergabesoftware relativ unkompliziert ermöglicht.

Beim Übertragen der Videodaten an den Client kopiert der MFC die Teile der Bitraten-Videos, die bei der zum Zeitpunkt der Übertragung zu Verfügung stehende Bandbreite ohne Aussetzer der Wiedergabe übertragen werden können, in den TCP-Datenstrom. Der Client selbst beeinflusst die Auswahl der versendeten Bandbreite, indem er in einer zweiten TCP-Verbindung Steuersignale in Form von Inhaltsanfragen an den MFC sendet. Es ist notwendig, dass die Wiedergabesoftware die Profildatei interpretieren, die verfügbare Bandbreite ermitteln und in der zweiten TCP-Verbindung Inhaltsanfragen an den Server absetzen kann.
ISPs sind in der Lage, ihren Kunden Set-Top Boxen mit der SmoothFlow-Funktionalität zur Verfügung zu stellen, mithilfe derer die Kunden das VoD-Angebot im heimischen Wohnzimmer nutzen können. Des Weiteren ist die Verwendung der SmoothFlow-Funktionalität im Bereich der PDAs und Smartphones denkbar, bei denen die verfügbare Bandbreite von der Anzahl und dem Nutzungsverhalten anderer eingebuchter Benutzer in einer Funkzelle sowie der Empfangsleistung des mobilen Geräts abhängt. Durch diese stark schwankenden Einflussfaktoren stellt die Übertragung mittels SmoothFlow in diesen Umgebungen eine optimierte Videoübertragung zur Verfügung.

5.3 YouTube Video Caching

5.3.1 Zielsetzung

Im Testszenario „YouTube Video Caching“ wird das in Kapitel 4.2 dargestellte Konzept zur Verkehrsoptimierung unter Verwendung des MFCs untersucht. Der MFC wird hierbei als Interception Proxyserver eingesetzt, der die Inhaltsanfragen der Clients inspiziert, Inhalte stellvertretend anfordert und sie für weitere Inhaltsanfragen nah am Client verfügbar hält.
Im Speziellen wird in diesem Kapitel die Arbeitsweise des MFCs in Bezug auf die Zwischenspeicherung von YouTube-Inhalten untersucht, da es sich hierbei um eine bekannte Videoplattform mit einem großen Datenübertragungsaufkommen handelt [Gill u. a. (2007)]. In diesem Szenario werden alle von den Inhaltekonsumenten angeforderten YouTube Videos auf dem MFC zwischengespeichert, um bei weiteren Anfragen der Inhaltekonsumenten aus dem Cache des MFCs ausgeliefert zu werden.
Des Weiteren werden die Einflüsse der Zwischenspeicherung auf die Übertragungszeiten der Videoinhalte und die möglichen Bandbreiteneinsparungen betrachtet. Es wird untersucht, wie die Ladezeiten der Inhalte in Abhängigkeit vom Cache-Zustand (Video nicht im Zwischenspeicher des MFCs, Video im Zwischenspeicher des MFCs, Inhaltsübertragung ohne Verwendung des MFC, etc.) variieren.

5.3.2 Testaufbau / Untersuchungsmethodik

Für die Zwischenspeicherung von YouTube-Videos werden drei Namensräume auf dem MFC angelegt. Zum einen der Namensraum „GesamterWebVerkehr“, der auf den gesamten HTTP-Datenverkehr zutrifft und keine Einschränkungen bezüglich DNS-Namen oder URL-Bestandteilen macht. Dieser Namensraum ist mit der geringsten konfigurierbaren Priorität versehen, so dass alle weiteren Regeln mit einer höheren Priorität bevorzugt zur Verarbeitung der Inhaltsanfrage herangezogen werden. Dies ist notwendig, da alle Inhaltsanfragen in einen Namensraum fallen müssen, um verarbeitet zu werden. Der Namensraum „GesamterWebVerkehr“ ist generisch so konfiguriert, dass seine Einstellungen für alle Seiten im Web anwendbar sind. Die Richtlinie zur Zwischenspeicherung von Inhalten für diesen Namensraum wird beispielsweise so konfiguriert, dass die im HTTP-Header definierten Vorgaben beachtet werden.
Des Weiteren werden die speziellen Namensräume „YouTube1“ und „YouTube2“ angelegt. Die Verwendung von zwei unterschiedlichen Namensräumen liegt in der Art der Auslieferung der Videos begründet, für die YouTube zwei unterschiedliche URLs verwendet. Der Namensraum „YouTube1“ ist für die Bearbeitung der Anfragen mit der Zeichenfolge „/get_video“ in der URL verantwortlich, der Namensraum „Youtube2“ für die Anfragen mit der Zeichenfolge „/videoplayback“. Die Filterung der Anfragen muss auf Grundlage der beiden Zeichenfolgen geschehen, um nur auf die Videoinhalte der YouTube Webseite zu zutreffen.

```
GET http://v21.lscache3.c.youtube.com/videoplayback?ip=0.0.0.0&sparams=↘
→id%2Cexpire%2Cip%2Cipbits%2Citag%2Cratebypass%2Coc%3↘
→AU0dYRVFSVl9FSkNNOF9JTFpJ&fexp=905602&itag=37&ipbits=0&sver=3&↘
→ratebypass=yes&expire=1290178800&key=yt1&signature=96↘
→CC620B6DE6D16ADC6ADA4011C5A7CE76B08AF0.↘
→CCF5925E044BBA9219A22D75AA204009A401B88F&id=5d218157378151b9&

GET http://v24.lscache8.c.youtube.com/get_video?t=↘
→vjVQa1PpcFNFUMd60g8pSXGAiF99NtZvgnsdYvnRgfQ=&asv=3&fmt=34&video_id=↘
→BSxxO_usMTs&el=detailpage&noflv=1
```

Listing 5.9: HTTP-Request von YouTube-Videos

URL	Parameter	
	Video-ID	Format-ID
http://[HOST]*/videoplayback*?[PARAMETER-LISTE]	id	itag
http://[HOST]*/get_video*?[PARAMETER-LISTE]	video_id	fmt

Tabelle 5.4: YouTube Video URL

Jeweils ein Beispiel für die unterschiedlichen Formate der Inhaltsanfragen ist in Listing 5.9 aufgeführt. Eine Inhaltsanfrage wird unter Verwendung der „/videoplayback“-URL an den Stellvertreterserver v21.lscache3.c.youtube.com des YouTube-CDNs gerichtet, die Inhaltsanfrage unter Verwendung der „/get_video“-URL wird an den Server v24.lscache8.c.youtube.com gestellt.
Die beiden Namensräume werden so konfiguriert, dass die im HTTP-Header enthaltenen Richtlinien zur Zwischenspeicherung von Inhalten überschrieben und die Inhalte auf jeden Fall auf dem MFC zwischengespeichert werden.
Da die Inhaltsanfragen benutzerspezifische Parameter enthalten, ist es nicht möglich, die Videos unter der angeforderten URL zu speichern. Um diese bei Anfrage eines anderen Inhaltekonsumenten wieder aufzufinden und ausliefern zu können, wird die ID herangezogen, die jedes YouTube-Video eindeutig identifiziert. Diese ID wird als Parameter bei der Inhaltsanfrage übergeben (siehe Tabelle 5.4) und kann so vom MFC verarbeitet werden. Im Falle der Anforderung über die „/get_video“-URL ist die ID in dem Parameter „video_id“ enthalten. Wird das Video über die „/videoplayback“-URL abgerufen, ist die ID des Videos im Parameter namens „id“ enthalten.
Die einzelnen YouTube-Videos stehen in unterschiedlichen Videoauflösungen und Formaten zur Verfügung, so dass sich jeder Inhaltekonsument entsprechend seinem Endgerät oder seiner Internetanbindung angepasste Formate abrufen kann. Tabelle 5.5 stellt die unterschiedlichen verfügbaren Formate dar und gibt Auskunft über die für das jeweilige Format

	Standard	Medium	High	HD 720p	HD 1080p	Mobile
Format-ID	34	18	35	22	37	17
Video Container	FLV	MP4	FLV	MP4	MP4	3GPP
Auflösung	320x240 640x480	480x360 480x270	854x480	1280x720	1920x1080	176x144
Video Codec	H.264/AVC	H.264/AVC	H.264/AVC	H.264/AVC	H.264/AVC	MPEG-4 Part 2
Audio Codec	AAC	AAC	AAC	AAC	AAC	AAC

Tabelle 5.5: YouTube Video Formate & zugehörige Tags nach Juniper Networks (2010)

verwendete Format-ID, die ebenfalls beim HTTP-Request als Parameter übergeben wird. Die Format-ID ist bei beiden URL-Formaten gleich, sie unterscheiden sich lediglich durch den Parameternamen (siehe Listing 5.4). Bei der Anforderung eines Videos mittels „/video-playback“ wird die Format-ID als „itag“ übergeben, bei der Anforderung mittels „/get_video“ als „fmt“. Beispiele zu Anforderungen wie sie vom YouTube eigenen Flash Player mithilfe der beiden URL-Schemata abgesetzt werden sind in Listing 5.9 dargestellt.

Um die Videoinhalte der YouTube-Plattform zwischenzuspeichern, werden die Inhalte auf dem MFC unter einer Kombination der beiden Parameter Video-ID und Format-ID nach dem folgenden Namensschema abgelegt:
yt_video_id_[*VIDEOID*]_fmt_[*FORMATID*].
Durch diese Art der Referenzierung ist sichergestellt, dass die Videos bei nachfolgenden Abrufen im Cache aufgefunden und von dort übertragen werden können. Die URL, von der die Inhalte abgerufen werden, spielt keine Rolle, da YouTube ein CDN für die Auslieferung der Inhalte verwendet, in dem die Umleitung der Clients durch URL-rewriting geschieht. Die an den Client ausgelieferten Webseiten enthalten bereits den genauen Hostnamen des Stellvertreterservers von dem die Videos abzurufen sind (siehe Hostenamen in Listing 5.9).
YouTube bietet den Inhaltekonsumenten die Möglichkeit zu einer beliebigen Stelle in den Videos zu springen, auch wenn das Video noch nicht bis zu dieser Stelle heruntergeladen wurde. Hierzu wird bei der Anforderung des Videos die Startposition in Millisekunden in dem Parameter „begin“ übergeben (siehe Listing 5.10). Daraufhin liefert der Webserver den Videoinhalt nicht von Anfang an aus, sondern ab der vom Benutzer gewählten Position. Dieses Verhalten unterstützt die Benutzerfreundlichkeit, da ggf. lange Wartezeiten für das Herunterladen von Videodaten, die anschließend übersprung werden, vermieden werden. Des Weiteren werden hierdurch Übertragungskapazitäten geschont.
Auch der MFC unterstützt dieses Springen in den Videoinhalten. Der „begin“-Parameter wird interpretiert und das Video wird von der richtigen Position an zum Client übertragen.
Für die Übertragung von Videoinhalten stellt der MFC weitere Funktionen zur Verfügung. Durch die AssuredFlow-Funktion [Juniper Networks (2010)], sichert der MFC das Übertragen der Videoinhalte mit einer entsprechend konfigurierten Mindestbandbreite zu. Hierzu

```
GET /videoplayback?ip=0.0.0.0&sparams=id%2Cexpire%2Cip%2Cipbits%2Citag%2↘
→Calgorithm%2Cburst%2Cfactor%2Coc%3AU0dYRVJLUl9FSkNNOF9KRVZB&fexp↘
→=905602&algorithm=throttle-factor&itag=34&ipbits=0&burst=40&sver=3&↘
→expire=1290204000&key=yt1&signature=4↘
→C0F7645481045FFD305D49A26DEB0D76F4F22DE.14561↘
→EEF4A934E47D2F76A7326FFE53B14D7EE11&factor=1.25&id=5d218157378151b9&↘
→begin=300126
```

Listing 5.10: HTTP-Request eines YouTube-Videos inklusive Sprungmarke

wird auf der Schnittstelle, über die die Inhaltsanfrage am MFC eingetroffen ist, eine Bandbreitenreservierung vorgenommen. Diese Bandbreitenreservierung bezieht sich lediglich auf die Schnittstelle des MFCs, auf den Schnittstellen weiterer an der Übertragung beteiligten Netzwerkkomponneten werden keine Bandbreitenreservierungen zur Verfügung gestellt. Als Effekt dieser Reservierung werden weitere Inhaltsanfragen bei ausgebuchter Schnittstellenkapazität nicht mehr bearbeitet, neue Anfragen werden mit einer Fehlermeldung quittiert und der Client wird abgewiesen. Dies sorgt dafür, dass allen Inhaltekonsumenten die definierte Mindestbandbreite zur Verfügung steht und damit die QoE des Dienstes möglichst hoch gehalten wird. Die Übertragungsrate fällt nicht durch Überbuchung unter das zugesicherte Minimum.
Ebenfalls ist die Möglichkeit zur Konfiguration einer maximalen Übertragungsrate für die Videoinhalte anzuführen. Durch das Begrenzen der maximalen Übertragungsrate kann sich die Übertragungszeit der Videoinhalte verlängern, wenn die zur Verfügung stehende Bandbreite für die Übertragung größer ist als der konfigurierte Wert für die maximalen Übertragungsrate. Dies trägt zur Schonung der Bandbreite im Netz bei. Die Videos werden nur so schnell in den Puffer des Client geladen, dass dieser in der Lage ist, die bei der Übertragung entstehenden Jitter auszugleichen. Diese Strategie soll neben der Schonung der Bandbreite auch zur Abschwächung von Peaks bei der Datenübertragung beitragen. Die übertragene Datenmenge wird in dem Fall verringert, wenn sich der Inhaltekonsument vor dem Ende des Videos entscheidet, dieses nicht weiter anzuschauen. Sollte keine maximale Bandbreite konfiguriert sein, so hat sich der Client je nach Netzwerksituation und Anbindung bereits einen Großteil oder die vollständige Videodatei heruntergeladen. Durch dieses Verhalten ist Übertragungskapazität verschwendet worden. Erhält der Client die Daten jedoch so, dass der Inhalt ohne Aussetzer wiedergegeben werden kann, ist beim vorzeitigen Beenden nur ein sehr geringer Teil der Daten vergebens übertragen worden. Die Konfiguration einer maximal verfügbaren Bandbreite bezieht sich nur auf die Übertragung von Videoinhalten, sämtliche andere Webseitenelemente sind von dieser Einschränkung nicht betroffen, so dass der Webseitenaufbau mit der größtmöglichen Geschwindigkeit abgewickelt wird.
Um die Geschwindigkeit der Datenübertragung zu messen, wird die Netzwerkanalysesoftware Wireshark eingesetzt. Durch das Mitschneiden der Netzwerkkommunikation können

die Übertragungsgeschwindigkeit und die Übertragungsdauer ermittelt werden.
Die Software ist auf dem Client installiert, von dem die YouTube-Videos abgerufen werden. Um die Last auf den Client zu verringern, wird ein Erfassungsfilter (Capture Filter) gesetzt, durch den nur Pakete einbezogen werden für deren Übertragung der TCP-Port 80, der für HTTP verwendet wird, beteiligt ist.
Für die Übertragung der Daten vom MFC zum Client, die über ein 100Mbit/s Netz miteinander verbunden sind, wird die gespeicherte Datenmenge jedes mitgeschnittenen Paketes zudem auf 54 Byte reduziert, um nur die relevanten Header Informationen zu speichern. Die 54 Byte setzen sich zusammen aus 14 Byte Ethernet-Header, 20 Byte IP-Header und 20 Byte TCP-Header. Mit den Informationen dieser Header sind alle relevanten Informationen der Datenübertragung, die für die Auswertung benötigt werden, vorhanden.
Die mitgeschnittene Datenkommunikation wird direkt in eine Datei auf einem RAM-Disk Laufwerk geschrieben. Dies ist notwendig, um die Anzahl der von Wireshark verworfenen Pakete möglichst gering zu halten, da die Speicherung auf der Festplatte zu langsam vonstatten geht und ohne die Verwendung der RAM-Disk Teile der Kommunikation von Wireshark verworfen und somit nicht untersucht werden könnten.
Zu Beginn der Wiedergabe eines Videos ist es notwendig, dass der Puffer mit Daten gefüllt wird bevor die Wiedergabe beginnen kann. Um den Start der Wiedergabe zu beschleunigen, kann die Nutzung der Bandbreitenbeschränkung zu Beginn der Übertragung ausgesetzt werden. Bei dieser FastStart-Funktion werden die ersten Daten des Videoinhalts schnellstmöglich an den Client übertragen, so dass das initiale Befüllen des Puffers möglichst rasch geschieht.
Es werden acht Tests unter Verwendung der beschriebenen Methodik durchgeführt, jeweils vier der Tests mit demselben Video. Bei den verwendeten Videos handelt es sich um ein 9:57 Min. langes Video, das in den Auflösungen 360p (Standard Definition (SD)) und 720p (HD) vorliegt. Zunächst wird die Übertragung des Videos ohne Verwendung des MFCs durchgeführt. In dem zweiten Test wird der MFC mit geleertem Cache verwendet, so dass der Videoinhalt durch den MFC von YouTube abgerufen wird. Somit kann ermittelt werden, ob durch die Verwendung des MFCs Verzögerungen der Übertragung entstehen. Im Anschluss daran wird die Übertragung der beiden Videos aus dem Cache des MFC durchgeführt, so dass das volle Potential der 100 Mbit/s-Verbindung ausgeschöpft werden kann. Bei der letzten Übertragungsart wird die Übertragungsrate der Videos auf 500 KByte/s begrenzt und die FastStart-Funktion so konfiguriert, dass die ersten 8000 KByte ohne dieses Bandbreitenlimit übertragen werden.

5.3.3 Ergebnisse

Im Folgenden sind die bei der Übertragung der beiden Videos (SD, HD) aus den unterschiedlichen Quellen und den unterschiedlichen aktivierten Funktionen erhobenen Messwerte dargestellt.

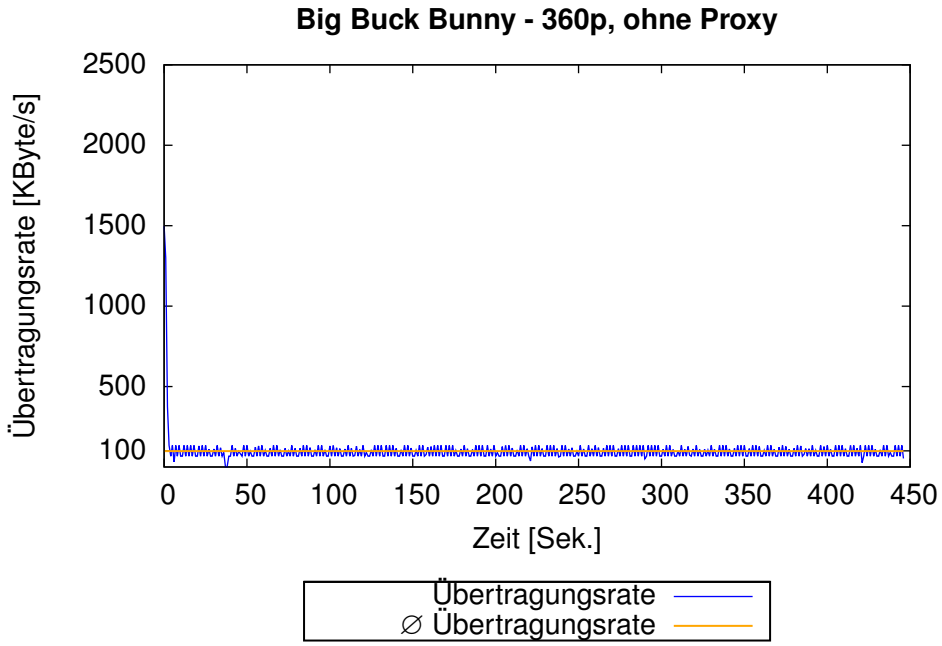


Abbildung 5.8: Übertragungsrate Big Buck Bunny - 360p, ohne Proxy

Abbildung 5.8 zeigt die Bandbreite der Übertragung des Videos in SD-Qualität ohne Verwendung des MFCs. Das Video wurde direkt aus dem YouTube-CDN abgerufen. Hierbei ist zu erkennen, dass die Übertragungsrate um die 100 KByte/s schwankt und die daraus resultierende durchschnittliche Übertragungsrate relativ genau 100 KByte/s beträgt.

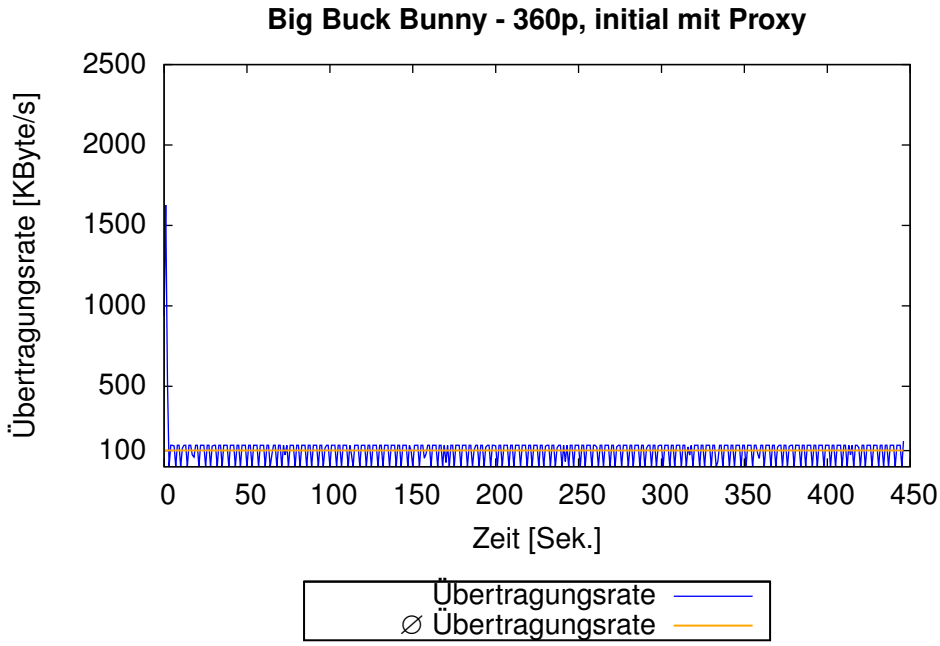


Abbildung 5.9: Übertragungsrate Big Buck Bunny - 360p, initial mit Proxy

Abbildung 5.9 zeigt die Bandbreite der Übertragung des Videos in SD-Qualität beim erstmaligen Download unter Verwendung des MFCs. Im Gegensatz zur Übertragung ohne Proxy schlägt die Kurve weiter nach unten aus, die durchschnittliche Übertragungsrate liegt dennoch weiter bei 100 KByte/s.

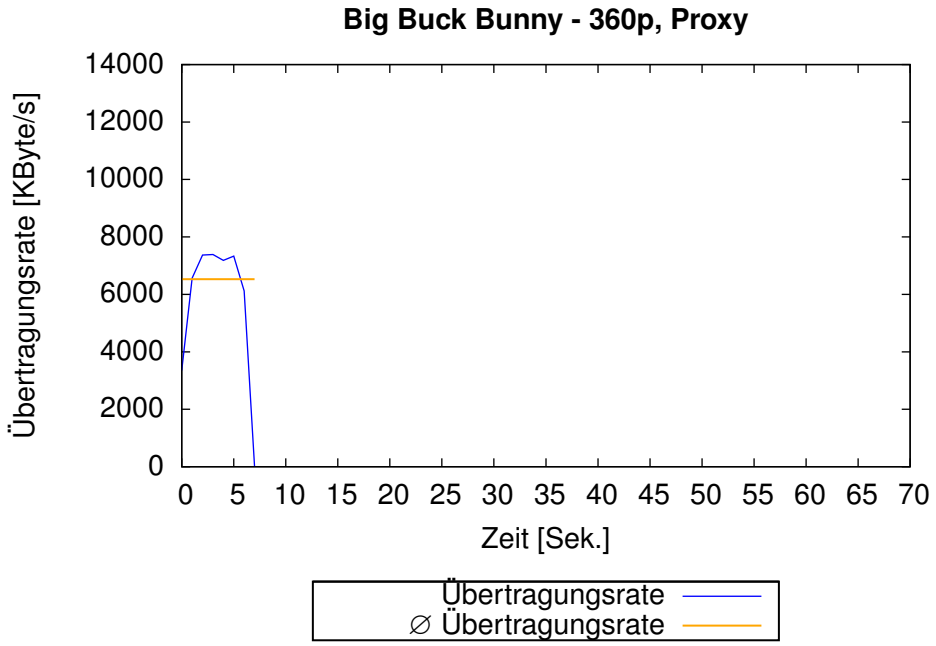


Abbildung 5.10: Übertragungsrate Big Buck Bunny - 360p, Proxy

Abbildung 5.10 zeigt die Bandbreite der Übertragung des Videos in SD-Qualität unter Verwendung des MFCs, auf dem das Video bereits vollständig im Zwischenspeicher verfügbar ist. Die Übertragung des Videomaterials findet nun mit einer durchschnittlichen Übertragungsrate von ca. 6500 KByte/s statt.

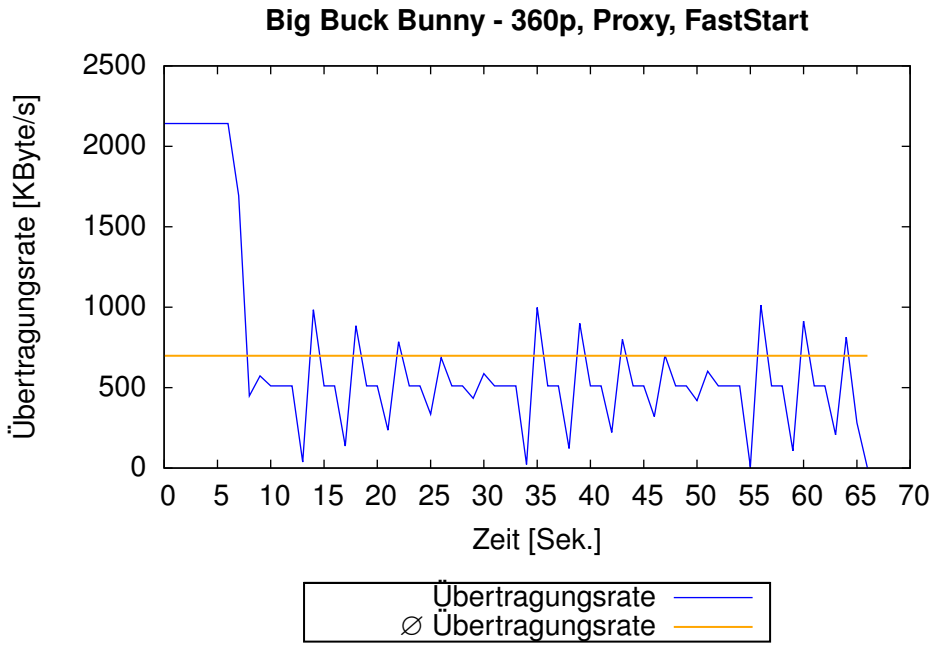


Abbildung 5.11: Übertragungsrate Big Buck Bunny - 360p, Proxy, FastStart

Abbildung 5.11 zeigt die Bandbreite der Übertragung des Videos in SD-Qualität unter Verwendung des MFCs, auf dem das Video bereits vollständig im Zwischenspeicher verfügbar ist. Die maximale Bandbreite wurde hierbei auf 500 KByte/s begrenzt, ausgenommen der ersten 8000 KByte, die durch die FastStart-Funktion mit maximaler Übertragungsrate versendet wurden.
An dem Verlauf der Kurve ist zu sehen, dass die Übertragung in den ersten 7-8 Sekunden mit einer hohen Übertragungsrate abgewickelt wurde. Anschließend pendelte sich die Übertragungsrate bei 500 KByte/s ein. Es ist ein periodischer Verlauf der Kurve zu erkennen, der sich alle 31 Sekunden wiederholt.

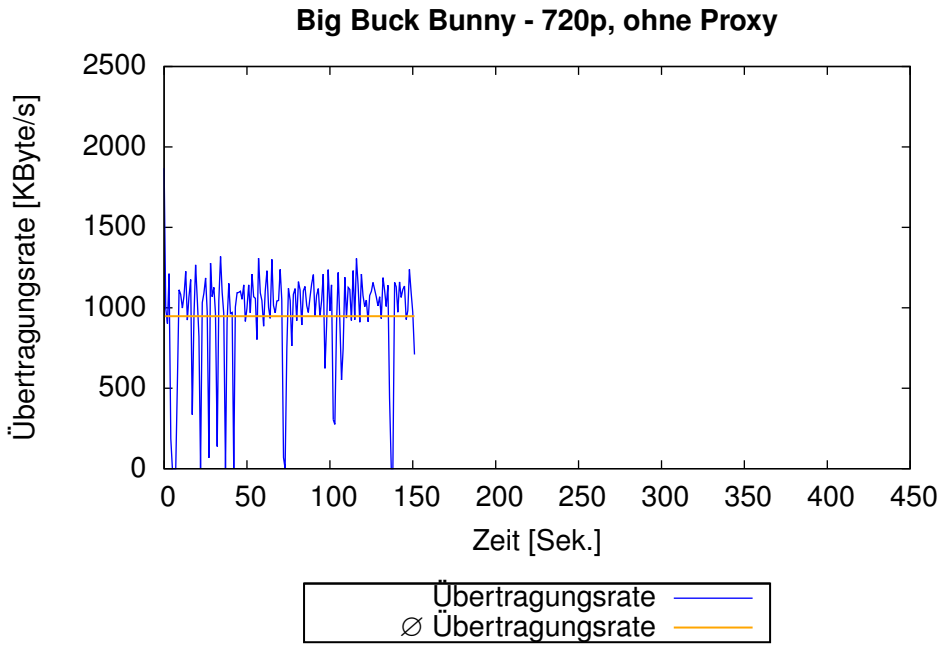


Abbildung 5.12: Übertragungsrate Big Buck Bunny - 720p, ohne Proxy

Abbildung 5.12 zeigt die Bandbreite der Übertragung des Videos in HD-Qualität, ohne die Verwendung des MFCs. Das Video wurde direkt aus dem YouTube-CDN abgerufen. Die durchschnittliche Übertragungsrate des Videomaterials liegt bei knapp 1000 KByte/s.

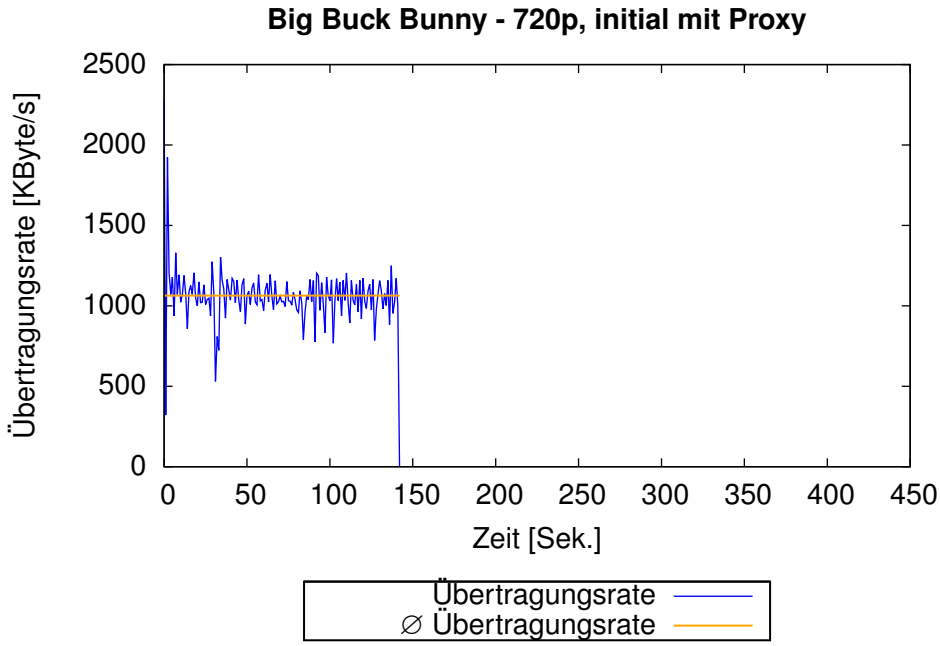


Abbildung 5.13: Übertragungsrate Big Buck Bunny - 720p, initial mit Proxy

Abbildung 5.13 zeigt die Bandbreite der Übertragung des Videos in HD-Qualität beim erstmaligen Download unter Verwendung des MFCs. Hierbei liegt die durchschnittliche Übertragungsrate leicht über 1000 KByte/s. Die Ausschläge der Kurve reichen nicht so weit nach unten, wie dies bei der Übertragung ohne Proxy der Fall ist.

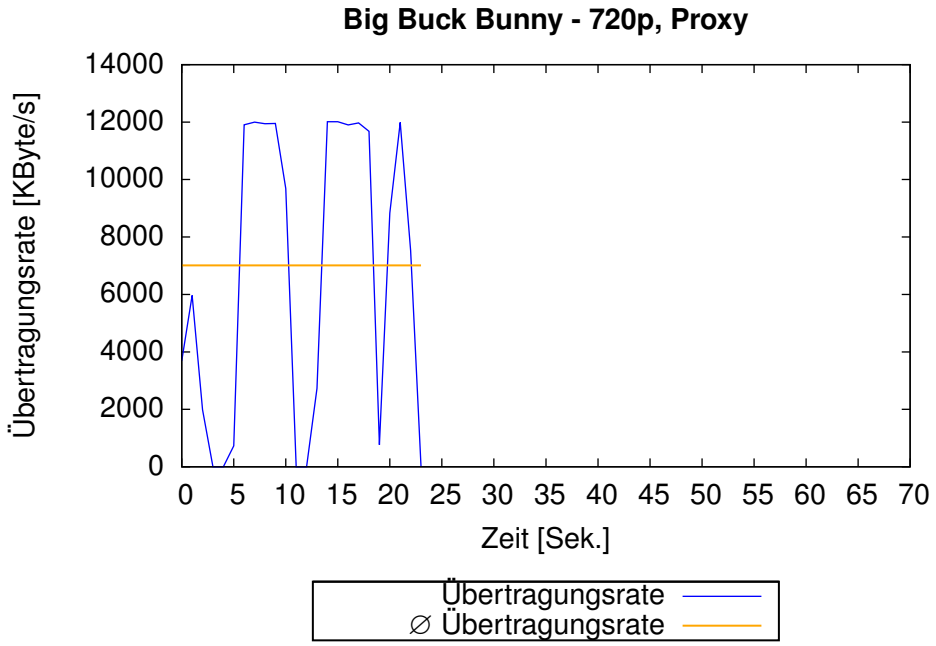


Abbildung 5.14: Übertragungsrate Big Buck Bunny - 720p, Proxy

Abbildung 5.14 zeigt die Bandbreite der Übertragung des Videos in HD-Qualität unter Verwendung des MFCs, auf dem das Video bereits vollständig im Zwischenspeicher verfügbar ist. Hierbei ist zu sehen, dass die maximale Übertragungsrate der Netzwerkkarte, die bei 100 Mbit/s = 12800 KByte/s liegt, zeitweise vollständig für die Übertragung des Videos verwendet wurde. Die durchschnittliche Übertragungsrate liegt bei ca. 7000 KByte/s.

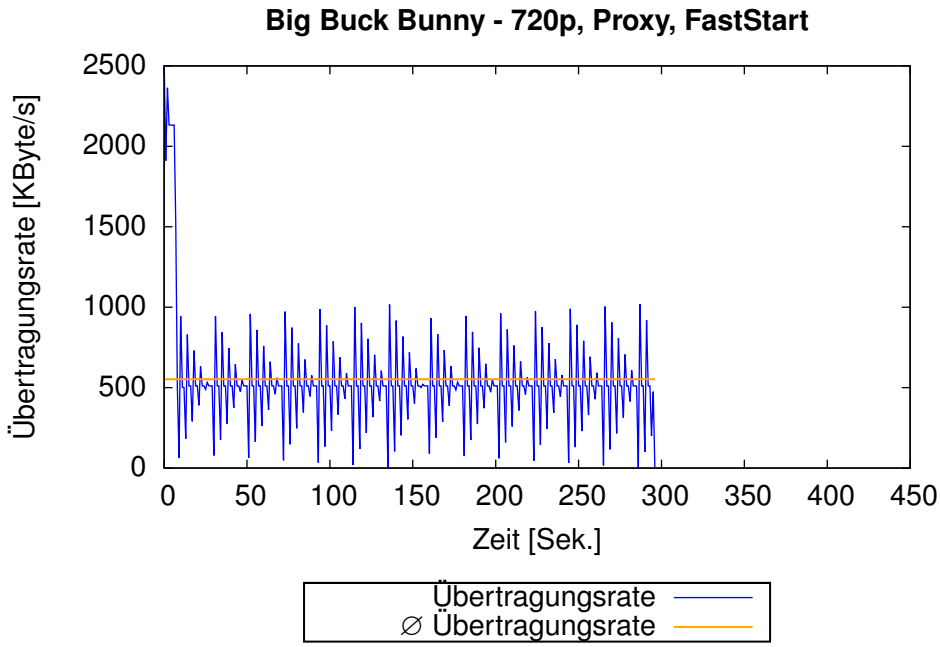


Abbildung 5.15: Übertragungsrate Big Buck Bunny - 720p, Proxy, FastStart

Abbildung 5.15 zeigt die Bandbreite der Übertragung des Videos in HD-Qualität unter Verwendung des MFCs, auf dem das Video bereits vollständig im Zwischenspeicher verfügbar ist. Die maximale Bandbreite wurde hierbei, wie auch beim SD-Video, auf 500 KByte/s begrenzt, ausgenommen der ersten 8000 KByte, die durch die FastStart-Funktion mit maximaler Übertragungsrate versendet wurden.
Auch hier startet die Übertragung mit einer hohen Übertragungsrate, um sich anschließend bei ca. 500 KByte/s einzupendeln. Die durchschnittliche Übertragungsrate liegt etwas höher als 500 KByte/s, was durch die erhöhte Übertragungsrate zu Beginn der Übertragung zu erklären ist.

Es sind weitere Messungen durchgeführt worden, mithilfe derer die mittlere Übertragungszeit sowie die Streuung der Messergebnisse ermittelt wurde. Die gemessenen Daten sind in Tabelle 5.6 für das SD-Video und in Tabelle 5.7 für das HD-Video dargestellt.
Die Streuung liegt bei der Messung der Übertragung des SD-Videos zwischen 0,009 und 0,510 Sekunden, was für eine stabile Übertragungsrate auch über mehrere Messungen hinweg spricht. Bei der Übertragung des HD-Videos liegt die Streuung zwischen 0,094 und 17,797 Sekunden, was einen erheblichen Unterschied zur Übertragung des SD-Videos darstellt.

Video	**Mittlere Übertragungszeit *[Sek.]***	**Streuung *[Sek.]***
SD; Proxy	5,91	0,139
SD; mit Proxy; FastStart	67,83	0,510
SD; ohne Proxy	443,07	0,009
SD; initial Proxy	443,74	0,045

Tabelle 5.6: Mittlere Übertragungszeiten und Streuung bei der Übertragung des SD-Videos

Video	**Mittlere Übertragungszeit *[Sek.]***	**Streuung *[Sek.]***
HD; Proxy	22,36	0,094
HD; ohne Proxy	144,32	6,116
HD; initial Proxy	149,89	17,797
HD; mit Proxy; FastStart	304,87	10,254

Tabelle 5.7: Mittlere Übertragungszeiten und Streuung bei der Übertragung des HD-Videos

5.3.4 Auswertung

Aus den gewonnenen Ergebnissen der praktischen Umsetzung der Verkehrsoptimierung unter Verwendung des Juniper Networks Media Flow Controllers lassen sich folgende Ergebnisse ableiten. Aufgrund der schwankenden Übertragungsrate um 100 KByte/s bei der Übertragung des SD-Videos lässt sich schließen, dass YouTube für die Übertragung des Videos eine maximale Übertragungsrate von 100 KByte/s verwendet (siehe Abbildung 5.8). Dies wird durch die Messergebnisse der initialen Übertragung unter Verwendung des MFCs (siehe Abbildung 5.9) unterstützt. Die Verwendung des MFCs verlangsamt oder beeinträchtigt die Geschwindigkeit der Datenübertragung nicht. Die Übertragung fällt zeitweise bis auf 0 KByte/s ab, die durchschnittliche Übertragungsrate von ca. 100 KByte/s bleibt aber erhalten, da sie zu anderen Zeitpunkten über 100 KByte/s liegt. Die Tatsache, dass die Übertragungsrate bei der initialen Übertragung zeitweise auf 0 KByte/s abfällt, ist durch die Terminierung der TCP-Verbindung des Clients auf dem MFC zu erklären, da dieser die eintreffenden Datenpakete nicht kontinuierlich weiterleitet. Die Weiterleitung an den Client geschieht erst, wenn die zur Übertragung verfügbare Datenmenge einen gewissen Schwellwert überschreitet.
Die Übertragung des SD-Videomaterials konnte durch die Übertragung aus dem Zwischenspeicher des MFC (siehe Abbildung 5.10) um das 65-fache beschleunigt werden. Die Daten wurden im Durchschnitt mit knapp 6500 KByte/s übertragen.
Bei der Übertragung des HD-Videos ohne Verwendung des MFCs (siehe Abbildung 5.12) ist eine durchschnittliche Übertragungsrate von knapp unter 1000 KByte/s ersichtlich. Bei der initialen Verwendung des MFCs (siehe Abbildung 5.13) liegt dieser Wert nur geringfügig höher, was auf die unterschiedliche Belastung der zwischenliegenden Netzwerkkomponenten zurückzuführen ist. Es ist auch hier erkennbar, das YouTube bei der Übertragung des HD-Videos eine Limitierung der maximalen Übertragungsrate vornimmt, die in diesem Fall bei 1000 KByte/s liegt. Die Übertragung des HD-Videos aus dem Zwischenspeicher (siehe Abbildung 5.14) des MFCs wird mit einer durchschnittlichen Übertragungsrate von ca. 7000 KByte/s durchgeführt, was dem 7-fachen der Übertragungsrate von den YouTube-Servern entspricht. Dabei wird zeitweise die gesamte zur Verfügung stehende Schnittstellenkapazität, die sich im Testaufbau auf 100 Mbit/s beläuft, für die Übertragung des Videomaterials genutzt.
In Abbildung 5.11 und 5.15 sind die Übertragung des SD- sowie des HD-Videos unter Verwendung der FastStart-Funktionalität in Zusammenspiel mit einer konfigurierten maximalen Übertragungsrate von 500 KByte/s pro Anfrage dargestellt. Die FastStart-Funktionalität ist so konfiguriert, dass die ersten 8000 KByte ohne Beschränkung übertragen werden, bevor die maximale Übertragungsrate anschließend eingeschränkt wird. Den beiden Abbildungen ist zu entnehmen, dass die Übertragung zunächst bei ca. 2100 KByte/s beginnt, bevor sie auf das konfigurierte Maximum von 500 KByte/s abfällt. Hierin liegt begründet, warum die durchschnittliche Übertragungsrate über den konfigurierten 500 KByte/s liegt.

Die Daten werden nicht konstant mit der konfigurierten Übertagungsrate an den Client gesendet. Der Verlauf der Kurven, die unter Verwendung der Konfiguration der maximalen Übertragungsrate gemessen wurden, lassen aufgrund ihres periodischen Verlaufs den Schluss zu, dass die Übertragung durch einen Algorithmus gesteuert wird, der periodisch arbeitet und die Übertragungsrate je Periode neu an den konfigurierten Wert annähert.
Die Konfiguration der maximalen Downloadrate pro Verbindung verhindert, dass die verwendete Netzwerkbandbreite starken Schwankungen unterliegt. Die Videodaten werden mit einer ausreichenden Bandbreite an den Client übertragen, so dass es nicht zu Unterbrechungen während der Wiedergabe kommt, gleichzeitig werden Übertragungsspitzen vermieden. Das Vermeiden bzw. Glätten solcher Spitzen kann helfen, die QoE für gleichzeitig abgewickelten Echtzeitdatenverkehr, wie dieser durch Videotelefonie oder online Spiele entsteht, möglichst hoch zu halten. Für die Echtzeitanwendungen würde die Verringerung der freien Übertragungskapazitäten und die bei der Überlastung einer Verbindung entstehenden Überläufe des Pufferspeichers in den Netzwerkgeräten einen Anstieg des Jitters und somit eine Verschlechterung der Service Qualität bedeuten. Darüber hinaus können beim vorzeitigen Abbruch der Wiedergabe Übertragungskapazitäten eingespart werden, wenn der Download des Videos bei dem Abbruch durch den Benutzer noch nicht vollständig an den Client übertragen worden ist.
Aus den Abbildungen 5.16 und 5.17 ist ersichtlich, dass die Konfiguration einer einheitlichen maximalen Übertragungsrate dieser Art die Bandbreiteneinsparung nicht zufriedenstellend unterstützt, da sich die Bitraten der SD- und HD-Videos deutlich unterscheiden und somit die Konfiguration gesonderter maximaler Übertragungsraten notwendig ist.
Das SD-Video ist bei einer Übertragungsrate von 500 KByte/s im Durchschnitt in 67 Sekunden übertragen. Beim HD-Video wird die Übertragungsrate im Vergleich zum YouTube-CDN soweit gedrosselt, dass das Video ohne Unterbrechungen angesehen werden kann, die unnötig übertragene Datenmenge bei Abbruch der Wiedergabe durch den Inhaltekonsumenten aber möglichst gering bleibt. Die maximale Downloadrate generell mit einem Wert für alle Videos zu konfigurieren, steht der Banbreiteneinsparung im Backbone bei vorzeitigem Abbruch der Wiedergabe durch den Benutzer im Wege. Im Providernetz sollten auf dem MFC weitere Namensräume konfiguriert werden, die abhängig vom Video-Format-Parameter aktiv werden und in denen unterschiedliche maximale Übertragungsraten definiert sind.
Die AssuredFlow-Funktion des MFCs reserviert auf der Netzwerkschnittstelle, an der eine Inhaltsanfrage eintrifft, einen Teil der Schnittstellenbandbreite für die Übertragung der angeforderten Inhalte. Diese Funktion hilft ebenfalls die QoS des Dienstes möglichst hoch zu halten, da bei Überbuchung der Schnittstelle der neu anfragende Client abgewiesen wird. Allen Clients, deren Übertragung bereits begonnen hat, steht weiterhin mindestens die per AssuredFlow zugesicherte Bandbreite zur Verfügung.
Aus den gewonnenen Ergebnissen der praktischen Umsetzung der Verkehrsoptimierung lassen sich des Weiteren die in Abbildung 5.16 und 5.17 zusammengefassten mittleren Übertragungszeiten ablesen. Hieraus ist zu erkennen, dass der MFC beim initialen Download und

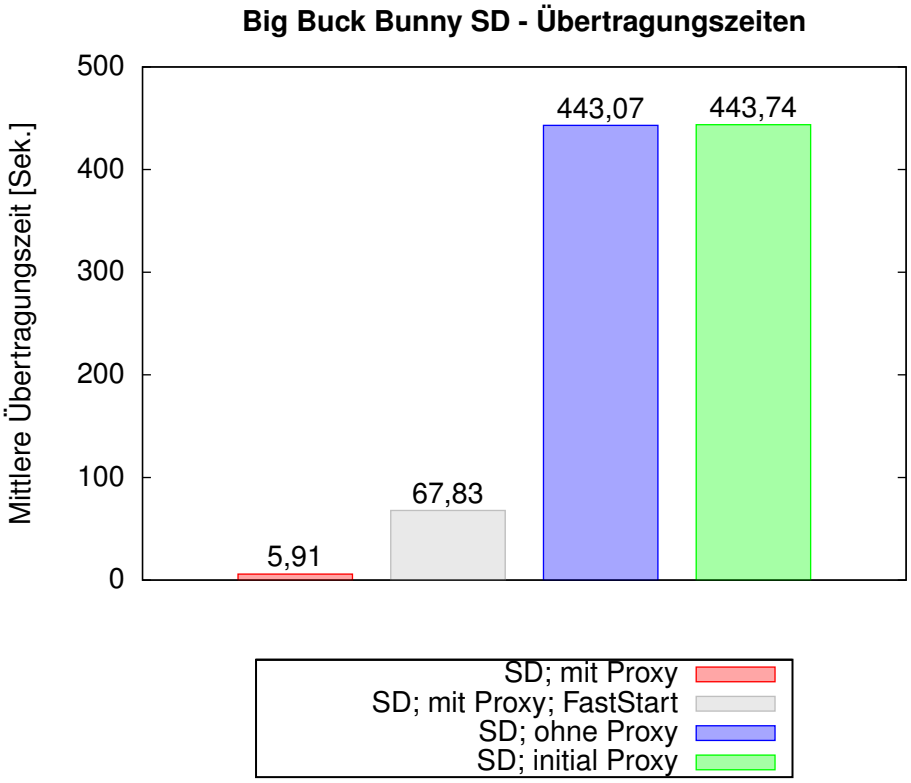


Abbildung 5.16: Mittlere Übertragungszeiten SD-Video

der Auslieferung der Videoinhalte an den Client nur leichte Verzögerungen in die Videoübertragung einbringt. Bei der Auslieferung aus dem Zwischenspeicher ist der MFC in der Lage, die Inhalte schnellstmöglich an die Clients zu übertragen. Dies kann wie zuvor erläutert durch die Konfiguration der maximalen Übertragungsgeschwindigkeit wieder verlangsamt werden. Schlussendlich ist zu klären, wie die Inhaltsanfragen auf die verfügbaren Videos der YouTube Plattform verteilt sind. Hierzu wurden in der Veröffentlichung „Youtube traffic characterization: a view from the edge“ [Gill u. a. (2007)] aus dem Jahr 2007 über drei Monate hinweg alle Anfragen, die aus dem Netzwerk der Universität von Calgary, Kanada, an die CDN-Server der YouTube-Plattform gestellt wurden, mitgeschnitten. Diese Daten wurden anschließend untersucht und ausgewertet. Dabei stellte sich heraus, dass über 50% aller Inhaltsanfragen Folgeanfragen für Videos, die zuvor schon einmal aus dem YouTube CDN abgerufen wurden, darstellten. Hingegen sind 68,1% der erstmalig angefragten Videos auch nur ein einziges Mal abgerufen worden. Dies entspricht 35% der gesamten Videoanfragen und macht 13,6% der gesamten übertragenen Menge an Videodaten aus. Zudem stellte sich heraus, dass 24% der angefragten Inhalte nicht zu Ende angesehen worden sind, was laut den Autoren auf zwei Gründe zurückzuführen ist. Zum einen wird eine schlechte Performance der

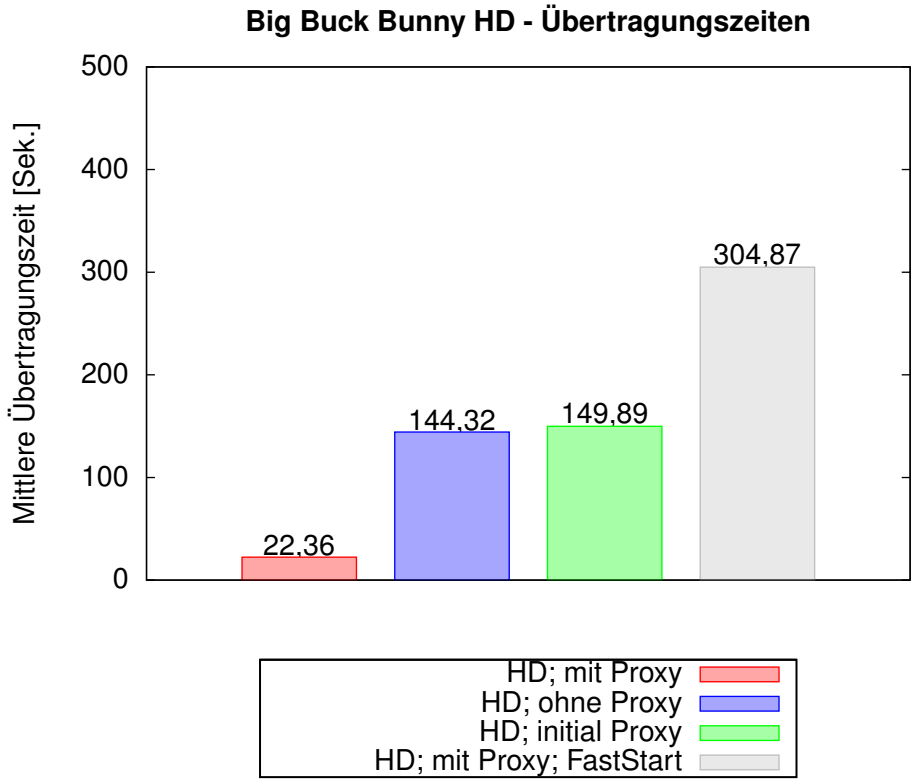


Abbildung 5.17: Mittlere Übertragungszeiten HD-Video

Datenübertragung genannt, zum anderen konnten die Inhalte selbst als Grund ausgemacht werden, da sie den Inhaltekonsumenten in den entsprechenden Fällen nicht angesprochen haben.
Interessant für die Effektivität einer Cache-Lösung, die auf YouTube Videoinhalte abzielt, ist die Verteilung der Abrufe auf die Videos. Die Untersuchung zeigte, dass die Anzahl der Downloads der einzelnen Videos der Zipf-Verteilung folgt. Das Zipfsche Gesetz besagt, dass bei der Sortierung von Objekten nach der Häufigkeit ihres Auftretens (F), das häufigste an erster Stelle, das zweithäufigste an zweiter Stelle usw., die Anzahl des Auftretens jeden Objekts aus dessen Rang (R) ableitbar ist.

$$F \sim R^{-\beta}$$

Im Fall der Untersuchungen am Campus der Universität von Calgary haben Gill u. a. eine Zipf-Verteilung mit dem β-Wert von 0,56 ermittelt. Die ermittelten Werte sind in Abbildung 5.18 dargestellt. Der Abbildung ist zu entnehmen, dass es auf der einen Seite wenige Videos gibt, die häufig abgerufen und auf der anderen Seite viele Videos die selten abgerufen werden. Das Zwischenspeichern der Inhalte bringt dem ISP nur bei den Videos einen Vorteil, die häufiger abgerufen werden, da an dieser Stelle Bandbreite im Backbone oder bei

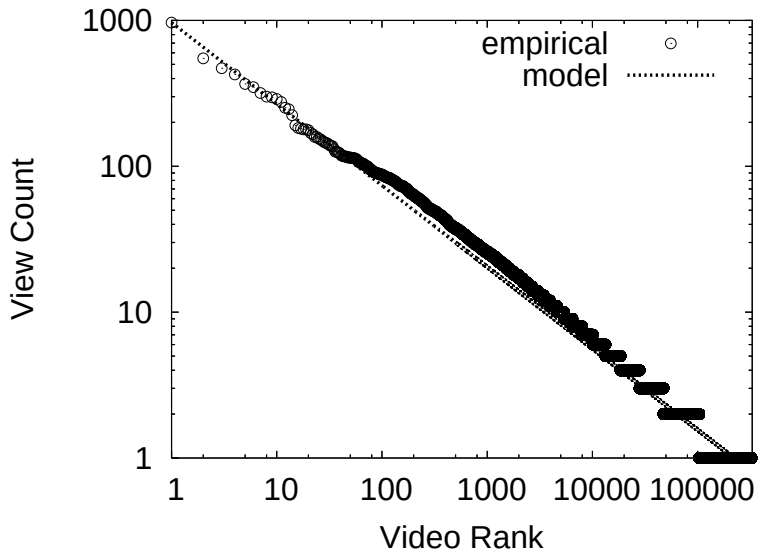


Abbildung 5.18: Rangfolge der Videos sortiert nach ihrer Anzeigehäufigkeit [Gill u. a. (2007)]

Peering-Verbindung eingespart werden kann.
Mithilfe des MFCs ist es dem ISP also möglich, bis zu 50% der durch die Übertragung von YouTube Videos entstehenden Bandbreitenbelastung des Backbones bzw. der Transit-Verbindungen, die durch die Folgeübertragungen entstanden sind, einzusparen.

Kapitel 6

Zusammenfassung & Ausblick

Im Folgenden werden die erarbeiteten Ergebnisse zusammengefasst und ein Ausblick auf weitere Forschungsgebiete und Entwicklungen im Bereich von CDNs gegeben.

6.1 Zusammenfassung

Das Ziel dieser Arbeit war es, Konzepte zur Umsetzung von ISP-betriebenen CDNs sowie die damit verbundenen Chancen zu analysieren. Es hat sich gezeigt, dass ISP-betriebene CDNs nicht unmittelbar mit den globalen CDNs von Firmen wie Akamai Technologies oder Limelight Networks verglichen werden können und somit nicht direkt mit diesen konkurrieren. ISP-betriebene CDNs besitzen andere Eigenschaften bezüglich der Anzahl erreichbarer Konsumenten, der möglichen Qualitätsbeeinflussung der Übertragung und des Einsatzgebietes.
In Kapitel 5 wurde gezeigt, wie ISPs CDNs zum Aufbau eines eigenen VoD-Angebots nutzen können, um Kunden durch die Schaffung eines Mehrwertes an sich zu binden. Ein weiterer Verwendungszweck für ISP-betriebene CDNs wurde mit der Nutzung zur Verkehrsoptimierung dargestellt, durch die ISPs in der Lage sind, die Bandbreitenauslastung im Backbone und die Auslastung der Transit- oder Peering-Verbindungen zu schonen.
Zur Analyse der Einflüsse auf die Datenübertragung sind beide Szenarien ausschnittsweise unter Verwendung des Media Flow Controllers der Firma Juniper Networks praktisch umgesetzt und analysiert worden. Bei der Umsetzung eines auf CDN basierenden VoD-Dienstes ist der MFC als Komponente des Delivery Systems in der Lage, die Bitrate des übertragenen Videomaterials an die verfügbare Bandbreite des Übertragungswegs zu jedem einzelnen Inhaltekonsumenten individuell anzupassen. Hierzu muss das Videomaterial bereits zuvor in die unterschiedlichen Bitraten konvertiert und dem MFC zur Verfügung gestellt werden. Der MFC bereitet die Videos für die Übertragung mittels SmoothFlow vor, indem sie in kurze Sequenzen geteilt werden. An diesen Übergängen besitzt der MFC die Möglichkeit, die Bitrate zu wechseln. Die SmoothFlow-Wiedergabesoftware ermittelt während der Übertragung die

verfügbare Bandbreite und fordert den MFC bei Bedarf auf, die Bitrate des versendeten Videos an die aktuellen Gegebenheiten im Netz anzupassen.
Für den Inhaltekonsumenten bedeutet dies eine gesteigerte QoE, da er sich keine Gedanken um die Qualität des Videos machen muss und nicht durch Unterbrechungen der Wiedergabe gestört wird. Der MFC passt die Qualität des übertragenen Videomaterials automatisch an.
Durch die Zwischenspeicherung der Videoinhalte von Videoplattformen wie YouTube ist es dem ISP möglich, die Belastung seiner Backboneinfrastruktur und die Belastung der Verbindung zu anderen Netzen zu reduzieren. Nahezu die Hälfte der angeforderten Videos können unter Verwendung eines Proxyservers aus dessen Zwischenspeicher ausgeliefert werden. Durch die Konfiguration einer maximalen Übertragungsrate, die sich ausschließlich auf die Videoinhalte auswirkt, ist es dem ISP möglich, die Zuverlässigkeit und die Qualität der Datenübertragung zu erhöhen. Echtzeitanwendungen, wie beispielsweise Voice-over-IP, werden durch die Vollauslastung des Übertragungskanals nicht negativ beeinflusst. Die Übertragung der Videoinhalte findet jedoch weiterhin so statt, dass die Inhalte ohne Aussetzer oder andere Einschränkungen abgespielt werden können. Es hat sich gezeigt, dass bei der Konfiguration einer maximalen Bandbreite für die Übertragung der tatsächliche Wert nicht konstant bleibt, er schwankt periodisch um das Mittel der konfigurierten Übertragungsrate.
Die praktische Umsetzung hat gezeigt, dass die beiden unterschiedlichen skizzierten Szenarien mithilfe des MFCs realisierbar sind. Es bleibt jedoch offen, wie sich der MFC unter steigender Last in der Produktivumgebung eines ISPs verhält.

6.2 Ausblick

6.2.1 CDN-Zusammenschlüsse

Wie in Kapitel 5 dargestellt, sind Inhalteanbieter gezwungen, viele Verträge mit unterschiedlichen ISPs abzuschließen, um bei der Verwendung von ISP-betriebenen CDNs die gleiche Leistung in Anspruch nehmen zu können wie sie ihnen auch durch globale CDN-Provider geboten wird. Ein weiteres Problem besteht in der Auswertung der Zugriffsstatistiken der einzelnen Inhalte. Der Inhalteanbieter ist gezwungen, die Statistiken der unterschiedlichen ISPs einzeln abzurufen und zu aggregieren, bevor Auswertungen aus den Gesamtdaten erstellt werden können. Im schlechtesten Fall ist es zudem notwendig, die unterschiedlichen Daten der CDNs zunächst in ein einheitliches Format zu bringen.
Wie zu erkennen ist, entstehen aus den dargestellten Eigenschaften ein erhöhter Arbeitsaufwand und zusätzliche Kosten für den Inhalteanbieter, wenn er dieselbe Anzahl an Inhaltekonsumenten, die er durch ein globales CDN erreicht, auch durch ISP-betriebene CDNs erreichen möchte.
Um einen annähernd so guten Service wie globale CDNs bieten zu können, müssten

verschiedene ISPs ihre CDNs zusammenschließen [Vakali und Pallis (2003)], so dass die Inhalteanbieter eine einzige Anlaufstelle haben, mit der sie Verträge abschließen, von der sie ihre Zugriffsstatistiken abrufen und bei der sie ihre gesamten Inhalte zentral ausbringen können.
Um die Kommunikation zwischen den CDNs zu gewährleisten, müssen einheitliche Schnittstellen vom Request-Routing-, Accounting-, sowie vom Distribution System implementiert werden [Day u. a. (2003)]. Die konkrete Implementierung der Systeme selbst kann dabei in jedem CDN unterschiedlich sein.
Das Request-Routing System muss in der Lage sein, über die eigenen Stellvertreterserver hinaus Informationen anderer CDNs zu verwalten. Die Accounting Systeme müssen Berichte über die verwendeten Ressourcen und die damit in Anspruch genommenen Leistungen auszutauschen. Diese Informationen dienen der Abrechnung sowohl der CDNs untereinander als auch gegenüber dem Inhalteanbieter. Das Distribution System muss die durch das CDN zur Verfügung gestellten Inhalte bei Bedarf auch in das angeschlossene CDN übertragen, so dass die Inhalte von dessen Stellvertreterservern ausgeliefert werden können.
Diese Zusammenschlüsse von CDNs werden weiter zunehmen müssen, um das Potential der ISP-betriebenen CDNs weiter zu steigern. Dies belegen auch Publikationen des Unternehmens Jet-Stream[11].

6.2.2 Application-Layer Traffic Optimization

Die Application-Layer Traffic Optimization (ALTO) Arbeitsgruppe der IETF strebt die Standardisierung eines Protokolls an, mithilfe dessen die Auswahl zu verwendender Peers in Peer2Peer-Netzwerken bezüglich der entstehenden Übertragungswege verbessert werden soll.
Das ALTO-Protokoll selbst verwendet die Client-Server Architektur. Es ist darauf ausgelegt, Informationen über die vorliegende Netztopologie zentral in Servern zu verwalten. Auf Anfrage eines ALTO-Clients verarbeitet der ALTO-Server die übermittelten Daten unter Zuhilfenahme der ihm bekannten Informationen über das Netz und liefert eine sortierte Liste der am besten geeigneten Peers zurück [Seedorf und Burger (2010)].
Der ALTO-Server liefert keine expliziten Topologieinformationen an die ALTO-Clients aus. So ist es den ISPs beispielsweise möglich, die Peerauswahl zu beeinflussen, indem die Topologie und weitere Eigenschaften des Netzes und der Transit-Verbindungen zu anderen Netzen berücksichtigt werden. Durch die Sortierung und das Zurückliefern der Liste von Peer-IP-Adressen kann ein ISP den ALTO-Dienst zur Verfügung stellen, ohne explizit Informationen über die Topologie oder zu Übergängen zu anderen Netzen öffentlich zugänglich zu machen.

[11] http://jet-stream.nl/blog/2010/02/cdn-federation-standards/

Das ALTO-Protokoll, das primär für die Verwendung in Peer2Peer-Netzwerken ausgelegt ist, kann jedoch auch bei der Ermittlung des am besten geeigneten Stellvertreterservers verwendet werden, um auch in diesen Szenarien die Backbone- und IXP-Belastung möglichst gering zu halten. Penno u. a. (2010) beschreiben in ihrem Draft wie der ALTO-Client in das Request-Routing System des CDNs integriert werden kann. Es wird mittels ALTO versucht, die Auswahl des Stellvertreterservers im Gegensatz zur randomisierten Serverauswahl durch Einbeziehung topologischer Rahmenbedingungen zu verbessern.
ALTO versucht hingegen nicht, schnell vergängliche Informationen vorzuhalten und in den Auswahlprozess einfließen zu lassen. Ein Beispiel für die nicht verwendeten Informationen sind kurzzeitige Überlastsituationen einzelner Verbindungen, um die sich im Falle von CDNs das Request-Routing System gesondert kümmern muss.

Glossar

A

AS
: AS. *siehe* Autonomous System

AssuredFlow
: Funktionalität des Juniper Networks Inc. MFCs zur Reservierung von Bandbreite auf der Netzwerkschnittstelle, über die das zu übertragene Video an den Inhaltekonsumenten ausgeliefert wird.

Autonomous System
: Abstrakte Aggregationsstufe von IP-Netzen. Eine Menge von IP-Netzen wird unter Verwendung einer AS-Nummer in Border Gateway Protocol (BGP) repräsentiert. Ein AS wird durch eine einheitliche Autorität administriert.

B

Backbone
: Menge der Hauptstränge eines Netzwerks. Verbindet beispielsweise die PoPs eines ISPs untereinander.

BGP
: BGP. *siehe* Border Gateway Protocol

Bitraten-Videos
: Videos unterschiedlicher Bitrate, die aus einem einzelnen Original-Video erstellt wurden und für die SmoothFlow-Funktionalität genutzt werden. Alle Bitraten-Videos, die zu einem Original-Video gehören, zeigen denselben Videoinhalt in unterschiedlicher Qualität.

Border Gateway Protocol
: Routing Protokoll zum Austausch von Routinginformationen zwischen Autonomen Systemen (AS).

C

Carrier
Unternehmen, das die Übertragung von Signalen oder das Durchschalten von Verbindungen als Dienstleister für einen oder mehrere Kunden gegen Entgeld übernimmt.

CDN
CDN. *siehe* Content Delivery Network

Client
Programm, das Inhaltsanfragen sendet und die zugehörige Inhalteauslieferungen empfängt.

Cloud
Virtualisierungsverbund von über das Netzwerk erreichbarer IT-Infrastruktur,die Speicher, Rechenkapazität etc. dynamisch nach Bedarf zur Verfügung stellt.

Content Delivery Network
Kommunikationsnetzwerk zur effizienten Auslieferung von digitalen Inhalten an die Inhaltekonsumenten.

Content Switch
Gerät, das die Last von HTTP-Anfragen in Kommunikationsnetzwerken auf unterschiedliche Webserver verteilt.

D

Datacenter
Rechenzentrum; Gebäude in dem die zentrale IT-Infrastruktur untergebracht ist.

Denial of Service
Angriff auf Netzwerksegmente oder einzelne Computerum einen Ausfall der angebotenen Dienste herbeizuführen.

DoS
DoS. *siehe* Denial of Service

E

Elektronischer Programmführer
System zur Darstellung aktueller sowie folgender Sendungen für Sender. Eine Art Programmzeitschrift.

EPG
EPG. *siehe* Elektronischer Programmführer

Extensible Markup Language
Datenformat für die Beschreibung und den Austausch hierarchisch strukturierter Daten.

F

File Transfer Protocol
Protokoll zur Übertragung von Dateien zwischen Systemen.

Flash Crowd
Sprunghafter und unerwarteter Anstieg von Inhaltsanfragen.

FTP
FTP. *siehe* File Transfer Protocol

G

Generic Routing Encapsulation
Technik zur Übermittelung von IP-Datagrammen innerhalb von IP-Datagrammen (IP-in-IP Kapselung).

Global Server Load Balancing
Mechanismus zur Verteilung von Lasten zwischen verschiedenen SLB-Nodes mithilfe von DNS.

GRE
GRE. *siehe* Generic Routing Encapsulation

GSLB
GSLB. *siehe* Global Server Load Balancing

H

Hausanschluss
Anschlussbereich zwischen der Vermittlungsstelle des ISP-Netzes und dem hausinternen TK-Anschluss.

HD
HD. *siehe* High Definition

Herkunftsserver
Server, der die Master-Kopie eines Inhalts vorhält.

High Definition
Videomaterial, das eine Auflösung zwischen 1280x720 und 1920x1080 Bildpunkten aufweist.

HTTP
HTTP. *siehe* Hypertext Transfer Protocol

Hypertext Transfer Protocol
Protokoll zur Übertragung von Daten. Wird hauptsächlich eingesetzt, um Webseiteninhalte an die Inhaltekonsumenten auszuliefern.

I

ICMP
ICMP. *siehe* Internet Control Message Protocol

IDS
IDS. *siehe* Intrusion Detection System

IETF
IETF. *siehe* Internet Engineering Task Force

Inhalteanbieter
Unternehmen oder Personen, die eigene oder fremde Inhalte für die Nutzung durch Dritte anbietet.

Inhaltekonsument
Konsument von digitalen Inhalten.

Inhaltsanfrage
Nachricht mit der bestimmte Inhalte zur Übermittlung angefordert werden.

Inhaltsauslieferung
Nachricht, die einen bestimmten Inhalt enthält, der zuvor in einer Inhaltsanfrage angefordert wurde.

Internet Control Message Protocol
Vermittlungsschicht-Protokoll, das dem Versand von Kontrollinformationen sowie Fehlernachrichten dient.

Internet Engineering Task Force
Organisation, die Internetstandards entwickelt (http://www.ietf.org/).

Internet Exchange Point
Austauschpunkt für Internetdatenverkehr, an dem Service Provider und große Unternehmen ihre Netze miteinander verbinden.

Internet Protokoll
Am weitesten verbreitetes Vermittlungsschicht-Protokoll, mithilfe dessen alle an ein Netz angeschlossenen Geräte adressiert werden. Stellt die Grundlage für die Kommunikation der Geräte untereinander dar.

Internet Service Provider
Unternehmen, das für seine Kunden den Zugang zum globalen Internet bereitstellt.

Intrusion Detection System
Sicherheitssystem, das Angriffe auf Netzwerke oder einzelne Computer anhand von Verhaltensmustern erkennt und meldet.

IP
IP. *siehe* Internet Protokoll

ISP
ISP. *siehe* Internet Service Provider

IXP
IXP. *siehe* Internet Exchange Point

J

Jitter
Varianz der Laufzeiten von Datenpaketen bei der Übertragung vom Sender zum Empfänger.

K

Keyframe
Vollständiges Einzelbild, das keine Referenzen auf andere Bilder besitzt.

L

Latenz
Zeit, die ein Datenpaket für die Übertragung von einem Punk zu einem anderen benötigt.

Least Recently Used
Deutsch: „Am längsten nicht verwendet“; Strategie zur Organisation von Caches, bei der die am längsten nicht verwendeten Objekte überschrieben werden, wenn neue Objekte in den vollen Cache geladen werden.

LRU
LRU. *siehe* Least Recently Used

M

Media Flow Controller
CDN-Serverprodukt der Firma Juniper Networks Inc.

MFC
MFC. *siehe* Media Flow Controller

Multicast
Gruppenkommunikation, bei der ein Sender die Daten durch einmaliges Aussenden gleichzeitig an viele Empfänger übertragen kann. Die Empfänger treten hierzu selbstständig einer Multicastgruppe bei.

O

Open Video Player
Open-Source Videowiedergabesoftware, die die adaptive Bitratenanpassung des Videomaterials unterstützt.

OVP
OVP. *siehe* Open Video Player

P

PBR
PBR. *siehe* Policy Based Routing

Peer2Peer
Kommunikationsmodell der Informatik in dem alle Teilnehmer gleichgestellt sind. Sie agieren sowohl als Server als auch als Client.

Peering
Kommunikationsverbindung zwischen BGP-Routern unterschiedlicher ASs zum Austausch von Routinginformationen und dem dadurch gerouteten Netzwerkverkehr. Der Austausch der Daten wird von beiden AS-Betreibern unentgeltlich vorgenommen.

Point of Presence
Durch ISPs verwalteter Ort, an dem die Kunden über Router mit dem ISP-Backbone verbunden sind.

Policy Based Routing
Technik zum Richtlinien-basierten Treffen von Routing-Entscheidungen.

PoP
PoP. *siehe* Point of Presence

Proximität
Nähe.

Proxyserver
Stellvertreter, der auf der einen Seite Anfragen entgegen nimmt und sie dann als seine Anfrage erneut an den Herkunftsserver stellt. Die Antworten werden anschließend an das initial anfragende Gerät übertragen.

Q

QoE
QoE. *siehe* Quality of Experience

QoS
QoS. *siehe* Quality of Service

Quality of Experience
Subjektiv wahrgenommene Qualität eines Dienstes oder einer Dienstleistung.

Quality of Service
Dienstgüte; in Netzwerken wird durch die Zuordnung unterschiedlicher Prioritäten zu Paketen eine bestimmte Dienstgüte bereitgestellt.

R

Real-Time Messaging Protocol
Properitäres Netzwerkprotokoll zur Übertragung von Audio- und Videodaten an einen Adobe Flash-Player.

Real-Time Streaming Protocol
Protokoll zur Steuerung und Kontrolle der Datenübertragung mittels RTP.

Real-Time Transport Control Protocol
Protokoll zur Übertragung von statistischen Informationen einer RTP-Verbindung.

Real-Time Transport Protocol
Protokoll zur Übermittlung von Echtzeitdaten.

Request-Routing
Prozess des Steuerns von Inhaltsanfragen an einen geeigneten Stellvertreterserver.

Round Trip Time
Zeit, die ein Datenpaket von der Quelle zum Ziel und zurück benötigt.

Round-Robin
Rundlauf-Verfahren; die verwalteten Ressourcen werden nacheinander den anfragenden Clients zugewiesen, hierdurch entsteht eine Verteilung der Last, die durch die Anfragen entsteht.

Router
Netzwerkelement, das für die Weiterleitung von Datenpaketen auf der Vermittlungsschicht zuständig ist.

RTCP
RTCP. *siehe* Real-Time Transport Control Protocol

RTMP
RTMP. *siehe* Real-Time Messaging Protocol

RTP
RTP. *siehe* Real-Time Transport Protocol

RTSP
RTSP. *siehe* Real-Time Streaming Protocol

RTT
RTT. *siehe* Round Trip Time

S

SD
SD. *siehe* Standard Definition

Secure Sockets Layer
Protokoll zur Authentifizierung und Verschlüsselung bei der Übertragung von Daten.

Server Load Balancing
Methode zur Lastverteilung in Kommunikationsnetzwerken.

SLB
SLB. *siehe* Server Load Balancing

SLB-Cluster
Verbund von Geräten, die beim Load-Balancing eine Einheit darstellen.

SmoothFlow
Funktionalität des Juniper Networks Inc. MFCs zur adapativen Bitratenanpassung bei der Übertragung von Videoinhalten.

SSL
SSL. *siehe* Secure Sockets Layer

Standard Definition
Videomaterial, das typischerweise in einer Auflösung von 640x480 (NTSC) oder 768x576 (PAL) Bildpunkten vorliegt.

Stellvertreterserver
Server, der Inhaltsanfragen empfängt und zugehörige Inhalte im Namen des Herkunftsservers ausliefert.

T

TCP
TCP. *siehe* Transmission Control Protocol

TLS
TLS. *siehe* Transport Layer Security

Transit
Datenverkehr, der zwischen unterschiedlichen ASs ausgetauscht wird. Aufgrund des asymmetrischen Datenflusses und Nutzens für beide AS-Betreiber stellt ein AS-Betreiber dem anderen, dem der Transport einen Mehrwert bietet, das übertragene Datenvolumen in Rechnung. In der Regel ist das Transit-AS nicht das endgültige Ziel der Pakete.

Transmission Control Protocol
Transportschicht-Protokoll, das verbindungsorientiert, reihenfolgenerhaltend und zuverlässig arbeitet.

Transport Layer Security
Protokoll zur Authentifizierung und Verschlüsselung der Übertragung von Daten. Weiterentwicklung des SSL-Protokolls.

U

UDP
UDP. *siehe* User Datagram Protocol

Unicast
Kommunikationsform mit 2 Teilnehmern. Ein Teilnehmer fungiert als Sender, der andere als Empfänger.

Uniform Resource Locator
Einheitlicher Ressourcen-Bezeichner, der angibt, wo eine Ressource verfügbar ist und mit welchem Mechanismus auf sie zugegriffen werden kann.

URL
URL. *siehe* Uniform Resource Locator

User Datagram Protocol
Transportschicht-Protokoll, das verbindungslos, nicht reihenfolgenerhaltend und nicht zuverlässig arbeitet.

V

Video on Demand
Methode der Auslieferung von Videoinhalten, die auf Anfrage des Konsumenten gestartet wird.

VoD
VoD. *siehe* Video on Demand

VoD-Bibliothek
Server oder Serververbund, der das gesamte Videomaterial eines VoD-Dienstes für die Übertragung an die Inhaltekonsumenten vorhält.

VoD-Katalog
Server oder Serververbund, der Informationen über die im VoD-Dienst verfügbaren Inhalte vorhält und an die Inhaltekonsumenten ausliefert.

X

XML
XML. *siehe* Extensible Markup Language

Literaturverzeichnis

[Adobe Systems] ADOBE SYSTEMS: *Overview of streaming with Flash Media Server 3*. http://www.adobe.com/devnet/flashmediaserver/articles/overview_streaming_fms3_-02.html

[Adobe Systems 2009] ADOBE SYSTEMS: *Real Time Messaging Chunk Stream Protocol*. http://www.adobe.com/devnet/rtmp/pdf/rtmp_specification_1.0.pdf. 2009

[Arlitt und Jin 2000] ARLITT, M. ; JIN, T.: A workload characterization study of the 1998 World Cup Web site. In: *Network, IEEE* 14 (2000), Mai, Nr. 3, S. 30–37. – ISSN 0890-8044

[Baker 1995] BAKER, F.: Requirements for IP Version 4 Routers / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc1812.txt`, Juni 1995 (1812). – RFC. Updated by RFC 2644

[Barbir u. a. 2003] BARBIR, A. ; CAIN, B. ; NAIR, R. ; SPATSCHECK, O.: Known Content Network (CN) Request-Routing Mechanisms / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc3568.txt`, Juli 2003 (3568). – RFC

[Berners-Lee u. a. 1996] BERNERS-LEE, T. ; FIELDING, R. ; FRYSTYK, H.: Hypertext Transfer Protocol – HTTP/1.0 / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc1945.txt`, Mai 1996 (1945). – RFC

[Buyya u. a. 2006] BUYYA, Rajkumar ; PATHAN, Al-Mukaddim K. ; BROBERG, James ; TARI, Zahir: A Case for Peering of Content Delivery Networks. In: *Distributed Systems Online, IEEE* 7 (2006), Oktober, Nr. 10, S. 3. – ISSN 1541-4922

[Cerpa u. a. 2000] CERPA, A. ; ELSON, J. ; BEHESHTI, H. ; CHANKHUNTHOD, A. ; DANZIG, P. ; JALAN, R. ; NEERDAELS, C. ; SCHROEDER, T. ; TOMLINSON, G.: *NECP the Network Element Control Protocol*. Februar 2000. – URL `http://tools.ietf.org/html/draft-cerpa-necp-02.txt`

[Cieslak u. a. 2001] CIESLAK, M. ; FORSTER, D. ; TIWANA, G. ; WILSON, R.: Web Cache Communication Protocol V2.0 / RFC-Editor. URL `http://tools.ietf.org/id/`

draft-wilson-wrec-wccp-v2-01.txt, Oktober 2001 (draft-wilson-wrec-wccp-v2-01). – Internet-Draft

[Crowcroft 2007] CROWCROFT, Jon: Net neutrality: the technical side of the debate: a white paper. In: *SIGCOMM Comput. Commun. Rev.* 37 (2007), Januar, S. 49–56. – URL http://doi.acm.org/10.1145/1198255.1198263. – ISSN 0146-4833

[Day u. a. 2003] DAY, M. ; CAIN, B. ; TOMLINSON, G. ; RZEWSKI, P.: A Model for Content Internetworking (CDI) / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc3466.txt, Februar 2003 (3466). – RFC

[Deering 1989] DEERING, S.E.: Host extensions for IP multicasting / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc1112.txt, August 1989 (1112). – RFC. Updated by RFC 2236

[Douglis und Kaashoek 2001] DOUGLIS, F. ; KAASHOEK, M.F.: Scalable internet services. In: *Internet Computing, IEEE* 5 (2001), Juli, Nr. 4, S. 36–37. – ISSN 1089-7801

[Faratin u. a. 2007] FARATIN, P. ; CLARK, D. ; GILMORE, P. ; BERGER, A.: Complexity of Internet Interconnections: Technology, Incentives and Implications for Policy. In: *35th Annual Telecommunications Policy Research Conference* (2007), September

[Farinacci u. a. 2000] FARINACCI, D. ; LI, T. ; HANKS, S. ; MEYER, D. ; TRAINA, P.: Generic Routing Encapsulation (GRE) / RFC-Editor. IETF, März 2000 (2784). – RFC. – URL http://www.rfc-editor.org/rfc/rfc2784.txt. Updated by RFC 2890

[Fei u. a. 1998] FEI, Z.-M. ; BHATTACHARJEE, S. ; ZEGURA, E.W. ; AMMAR, M.H.: A novel server selection technique for improving the response time of a replicated service, März 1998, S. 783 –791. – ISSN 0743-166X

[Fielding u. a. 1999] FIELDING, R. ; GETTYS, J. ; MOGUL, J. ; FRYSTYK, H. ; MASINTER, L. ; LEACH, P. ; BERNERS-LEE, T.: Hypertext Transfer Protocol – HTTP/1.1 / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc2616.txt, Juni 1999 (2616). – RFC. Updated by RFCs 2817, 5785

[Francis u. a. 2001] FRANCIS, P. ; JAMIN, S. ; JIN, Cheng ; JIN, Yixin ; RAZ, D. ; SHAVITT, Y. ; ZHANG, L.: IDMaps: a global Internet host distance estimation service. In: *Networking, IEEE/ACM Transactions on* 9 (2001), Oktober, Nr. 5, S. 525–540. – ISSN 1063-6692

[Gill u. a. 2007] GILL, Phillipa ; ARLITT, Martin ; LI, Zongpeng ; MAHANTI, Anirban: Youtube traffic characterization: a view from the edge. In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA : ACM, 2007 (IMC '07), S. 15–28. – URL http://doi.acm.org/10.1145/1298306.1298310. – ISBN 978-1-59593-908-1

[Hofmann und Beaumont 2005] HOFMANN, Markus ; BEAUMONT, Leland R.: *Content Networking: Architecture, Protocols, and Practice (The Morgan Kaufmann Series in Networking)*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2005. – ISBN 1-55860-834-6

[Hyun u. a. 2003] HYUN, Young ; BROIDO, Andre ; CLAFFY, K.: *Traceroute and BGP AS Path Incongruities*. 2003

[Juniper Networks 2010] JUNIPER NETWORKS, Inc.: *Media Flow Controller^{TM}-Administrator's Guide and CLI Command Reference v.2.0.4*. September 2010. – URL `http://www.juniper.net/techpubs/software/management/media-flow/media-flow2.0.4/mfc-admin-guide.pdf`

[Krishnamurthy u. a. 2001] KRISHNAMURTHY, Balachander ; WILLS, Craig ; ZHANG, Yin: On the use and performance of content distribution networks. In: *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. New York, NY, USA : ACM, 2001, S. 169–182. – ISBN 1-58113-435-5

[Kurose und Ross 2009] KUROSE, James F. ; ROSS, Keith W.: *Computer Networking: A Top-Down Approach*. Bd. 5. Prentice Hall International, 2009. – 864 S. – ISBN 0-13-136548-7

[Lazzaro 2006] LAZZARO, J.: Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc4571.txt`, Juli 2006 (4571). – RFC

[Leighton 2009] LEIGHTON, Tom: Improving performance on the internet. In: *Commun. ACM* 52 (2009), Nr. 2, S. 44–51. – ISSN 0001-0782

[Marchese 2007] MARCHESE, Mario: *QoS Over Heterogeneous Networks*. John Wiley & Sons, April 2007. – ISBN 978-0-470-01752-4

[Merola 2006] MEROLA, Antonio: Verwendung und Missbrauch von ICMP. In: *hakin9* 2 (2006), S. 45–57

[Mockapetris 1987] MOCKAPETRIS, P.V.: Domain names - concepts and facilities / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc1034.txt`, November 1987 (1034). – RFC. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936

[Obraczka und Silva 2000] OBRACZKA, K. ; SILVA, F.: Network latency metrics for server proximity, 2000, S. 421–427

[Odlyzko 2009] ODLYZKO, Andrew: Network Neutrality, Search Neutrality, and the Never-ending Conflict between Efficiency and Fairness in Markets. In: *Review of Network Economics* 8 (2009), Nr. 1. – URL http://www.dtc.umn.edu/~odlyzko/doc/rne81.pdf

[O'Driscoll 2008] O'DRISCOLL, Gerard: *Next Generation IPTV Services and Technologies*. Bd. 1. John Wiley & Sons, 2008. – ISBN 978-0-470-16372-6

[Partridge u. a. 1993] PARTRIDGE, C. ; MENDEZ, T. ; MILLIKEN, W.: Host Anycasting Service / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc1546.txt, November 1993 (1546). – RFC

[Pathan und Buyya 2007] PATHAN, Mukaddim ; BUYYA, Rajkumar: *A Taxonomy and Survey of Content Delivery Networks*. 2007. – URL http://www.gridbus.org/reports/CDN-Taxonomy.pdf

[Paxson 1997] PAXSON, Vern: End-to-end Internet packet dynamics. In: *SIGCOMM Comput. Commun. Rev.* 27 (1997), Nr. 4, S. 139–152. – ISSN 0146-4833

[Penno u. a. 2010] PENNO, Reinaldo ; RAGHUNATH, Satish ; MEDVED, Jan ; BAKSHI, Mayuresh ; ALIMI, Richard ; PREVIDI, Stefano: ALTO and Content Delivery Networks / RFC-Editor. URL http://tools.ietf.org/id/draft-penno-alto-cdn-01.txt, 2010 (draft-penno-alto-cdn-01). – Internet Draft

[Postel 1981] POSTEL, J.: Internet Control Message Protocol / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc792.txt, September 1981 (792). – RFC. Updated by RFCs 950, 4884

[Rabinovich und Spatscheck 2001] RABINOVICH, Michael ; SPATSCHECK, Oliver: *Web Caching and Replication*. Addison-Wesley Professional, Dezember 2001 (1). – ISBN 0201615703

[Rekhter u. a. 2006] REKHTER, Y. ; LI, T. ; HARES, S.: A Border Gateway Protocol 4 (BGP-4) / RFC-Editor. URL http://www.rfc-editor.org/rfc/rfc4271.txt, Januar 2006 (4271). – RFC

[Richardson 2003] RICHARDSON, Iain E. G.: *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. New York, NY, USA : John Wiley & Sons, Inc., 2003

[Richardson 2010] RICHARDSON, Iain E. G.: *The H.264 Advanced Video Compression Standard*. Bd. 2. John Wiley & Sons, Inc., August 2010. – ISBN 0470516925

[Schulzrinne und Casner 2003] SCHULZRINNE, H. ; CASNER, S.: RTP Profile for Audio and Video Conferences with Minimal Control / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc3551.txt`, Juli 2003 (3551). – RFC. Updated by RFC 5761

[Schulzrinne u. a. 2003] SCHULZRINNE, H. ; CASNER, S. ; FREDERICK, R. ; JACOBSON, V.: RTP: A Transport Protocol for Real-Time Applications / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc3550.txt`, Juli 2003 (3550). – RFC. Updated by RFCs 5506, 5761

[Schulzrinne u. a. 1998] SCHULZRINNE, H. ; RAO, A. ; LANPHIER, R.: Real Time Streaming Protocol (RTSP) / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc2326.txt`, April 1998 (2326). – RFC

[Seedorf und Burger 2010] SEEDORF, J. ; BURGER, E.: Application-Layer Traffic Optimization (ALTO) Problem Statement / RFC-Editor. URL `http://www.rfc-editor.org/rfc/rfc5693.txt`, Oktober 2010 (5693). – RFC

[Sivasubramanian u. a. 2004] SIVASUBRAMANIAN, Swaminathan ; SZYMANIAK, Michal ; PIERRE, Guillaume ; STEEN, Maarten v.: Replication for web hosting systems. In: *ACM Comput. Surv.* 36 (2004), Nr. 3, S. 291–334. – ISSN 0360-0300

[Szymaniak u. a. 2003] SZYMANIAK, Michal ; PIERRE, Guillaume ; STEEN, Maarten van: NetAirt: A DNS-based Redirection System for Apache. In: *Proceedings of the IADIS International Conference on WWW/Internet*. Algarve, Portugal, November 2003. – URL `http://www.globule.org/publi/NDBRSA_icwi2003.pdf`

[Tanenbaum 2002] TANENBAUM, Andrew S.: *Computer Networks*. Bd. 4. Prentice Hall Professional Technical Reference, 2002. – ISBN 0-13-066102-3

[Vakali und Pallis 2003] VAKALI, A. ; PALLIS, G.: Content delivery networks: status and trends. In: *Internet Computing, IEEE* 7 (2003), November, Nr. 6, S. 68–74. – ISSN 1089-7801

[Wessels 2001] WESSELS, Duane: *Web Caching*. O'Reilly Media, Juni 2001 (O'Reilly Internet Series). – ISBN 978-1-56592-536-6

[Yin u. a. 2010] YIN, Hao ; LIU, Xuening ; MIN, Geyong ; LIN, Chuang: Content delivery networks: A bridge between emerging applications and future IP networks. In: *Network, IEEE* 24 (2010), Juli, Nr. 4, S. 52–56. – ISSN 0890-8044

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw.§24(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 29. März 2011 Markus Vahlenkamp