

Verkehrsmenge und Caching von Videos

Adrian Schnell

Betreuer: Dr. Heiko Niedermayer

Hauptseminar - Innovative Internettechnologien und Mobilkommunikation, SS2011

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: schnell@in.tum.de

KURZFASSUNG

Das Web 2.0, speziell Videodienste, erzeugen einen unvorstellbar viel Datenverkehr. Um die Inhalte trotzdem schnell und kostengünstig an Benutzer ausliefern zu können, werden drei verschiedene Caching Verfahren vorgestellt. YouTube selbst setzt ein CDN¹ (Limelight Networks) ein, um die Informationsflut den Nutzern bereitstellen zu können.

Schlüsselworte

Verkehrsmenge, Caching, Streaming, Videos, YouTube, content distribution network

1. EINLEITUNG

Durch die zunehmende Verbreitung und Nutzung des Web 2.0 und der dadurch stark wachsende benutzergenerierte Inhaltsflut durch soziale Netzwerke wie Facebook, Flickr, My Space, Twitter und YouTube entstehen neue Probleme. Zum einen müssen diese Daten gespeichert werden, aber auch wieder den Nutzern zur Verfügung gestellt werden. Dabei entstehen jeweils Kosten, die man versucht möglichst gering zu halten. Im Folgenden werden die Probleme durch den wachsenden Datenverkehr sowie Lösungen besprochen.

Besonders soziale Netzwerke, wie Facebook oder Google Plus, fördern stark die Verbreitung von Medieninhalten wie Videos und Bildern. Im Januar 2011 waren 600 Millionen aktive Benutzer weltweit auf Facebook zu verzeichnen [9], die untereinander stark vernetzt sind und Nachrichten, Internetseiten, Fotos, Blogs und auch Videos teilen. Durch dieses sogenannte „friend-casting“ hat es Facebook inzwischen sogar geschafft, den Internetgiganten Google vom Thron der meist aufgerufenen Internetseite zu stoßen [7].

Dieser Artikel ist wie folgend aufgebaut. Kapitel 2 geht auf näher auf YouTube, dessen Entstehung und technische Funktionsweise ein. In Kapitel 3 wird ein Versuch unternommen, die gespeicherte Datenmenge von YouTube zu schätzen. Diese Ergebnisse werden in Kapitel 4 verwendet, um die Verkehrsmenge von YouTube grob einzuordnen. Kapitel 5 führt verschiedener Cachingstrategien auf und Kapitel 6 erläutert den Einsatz eines CDN. Beide haben die Aufgabe, die genutzte Bandbreite von Dienstleister und ISP² zu reduzieren sowie die Daten schneller an den Benutzer auszuliefern. Kapitel 7 fasst die Ergebnisse zusammen und liefert einen Ausblick.

¹content distribution network

²internet service provider

2. YOUTUBE

2.1 Entstehung

YouTube ist eine der größten und erfolgreichsten Internetseiten, laut dem Ranking von Alexa liegt YouTube auf Platz zwei der am schnellsten wachsenden Websites im Internet [2]. Laut einer älteren Studie von Nielsen Netratings von 2006 wuchs YouTube alleine in einer Juliwoche 2006 um 75% von 7,3 auf 12,8 Millionen Usern [10].

Bei YouTube handelt es sich um eine Webcommunity zur Bereitstellung von Kurzvideos und deren Bewertung und Kommentierung. Gegründet wurde YouTube 2005 in San Bruno, Kalifornien. Aufgekauft wurde YouTube bereits 2006 von Google für umgerechnet etwa 1,3 Milliarden Euro [14].

Bereits März 2008 stellte das Marktforschungsinstitut *Hitwise* fest, dass 73% aller Besucher von Videoportalen in den USA YouTube zuzuordnen sind [14].

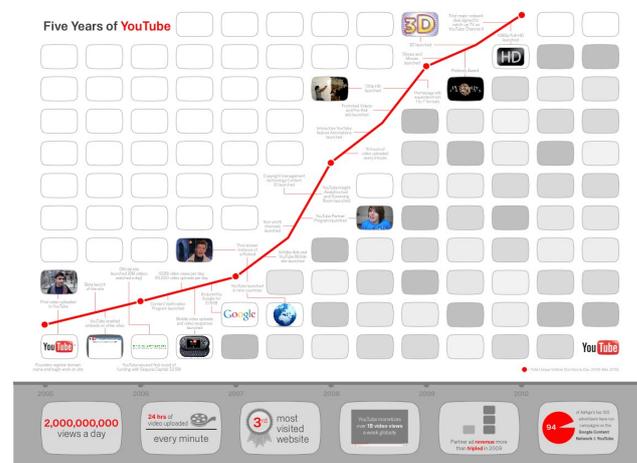


Abbildung 1: Weiterentwicklung von YouTube anhand einer Zeitachse, aus [15]

In Abbildung 1 sind einige der wichtigsten Eckpunkte in der zeitlichen Entwicklung von YouTube festgehalten. Unter anderem, dass 2008 jede Minute etwa 15 Stunden Filmmaterial von den Nutzern hochgeladen wurde [15]. Im Jahr 2010 ist diese Zahl bereits auf 24 Stunden pro Minute angewachsen und verzeichnet täglich bis zu zwei Milliarden Seitenaufrufe [16]. Aktuell im Jahr 2011 sind diese Zahlen erneut drastisch gewachsen. Inzwischen können drei Milliarden Seitenaufrufe

täglich verzeichnet werden sowie 48 Stunden Videodaten je Minute [17]. Mit dazu beigetragen hat sicherlich auch das Einbetten von Videos in andere Internetseiten und soziale Netzwerke. Die Verteilung der Videos an letztere kann der Nutzer auch vollautomatisch nach dem Hochladen neuer Videodateien von YouTube übernehmen lassen.

Im Dezember 2008 startete der erste HD Dienst mit 720 Pixeln. Etwa ein Jahr später, November 2009, startete der Full HD Dienst, der Videos mit 1020 Pixeln zur Verfügung stellt [16]. Die Benutzer haben bei jedem Video die Möglichkeit, die gewünschte Darstellungsqualität selbst festzulegen. Wenn beispielsweise ein Video mit 1080 Pixeln angeboten wird, besteht zusätzlich die Auswahl von 240, 360, 480 und 720 Pixeln Auflösung.

Erst seit kurzem ist es unter <http://www.youtube.com/movies> sogar möglich, Spielfilme in voller Länge und auch teilweise in HD anzuschauen. Die Nutzung dieses Dienstes ist nach derzeit noch zum Großteil kostenfrei.

2.2 Funktionsweise

YouTube ist ein internetbasierter Dienst, auf dem Nutzer ihre Videos, für die allgemeine Öffentlichkeit oder auch nur für Freunde, veröffentlichen können.

Dabei können alle heute gängigen Videoformate wie .WMV, .AVI, .MPG, .MP4, .FLV, .MKV und .MOV bis zu einer Abspieldauer von 15 Minuten und 2 GB Datengröße hochgeladen werden.

Bisher wurden die Videos ausschließlich über den Adobe Flash Player ausgeliefert, wobei der Sorenson Spark H.263 Video Codec verwendet wurde [2]. Dies hatte den Vorteil, dass jeder Nutzer die Videos anschauen konnte, unabhängig vom verwendeten Betriebssystem oder Browser, solange das Flash Plugin installiert war. Es wird davon ausgegangen, dass über 90% der Nutzer dieses installiert haben [4].

Seit Januar 2010 experimentiert YouTube allerdings auch mit den Audio/Video Tags von HTML5. Damit lassen sich die Videos ohne das Flash Plugin abspielen und auch mobile Geräte, wie zum Beispiel iOS Geräte³ wie das iPhone beziehungsweise iPad, sind in der Lage, HTML5 Videos abzuspielen.

Über eine Suchfunktion auf der Website, beziehungsweise Software auf mobilen Geräten, können diese dann gefunden und gestreamt werden [18]. Das Herunterladen beziehungsweise persistente Speichern der Daten ist nach YouTube Nutzerbestimmungen untersagt, sollte kein spezieller Link dazu auf der Internetseite vorhanden sein.

Um sich den Ablauf genauer vorstellen zu können, wie genau das Übertragen der Videodaten funktioniert, betrachten wir Abbildung 2. Hier ist dargestellt, wie die Kommunikation via HTTP Befehlen zwischen dem Nutzer, dem YouTube Server und dem CDN abläuft.

Jedes Video hat eine eindeutige und einzigartige 11 stellige ID zugewiesen. Hat sich der Nutzer festgelegt, welches Video

³mobile Geräte vom Hersteller *Apple*

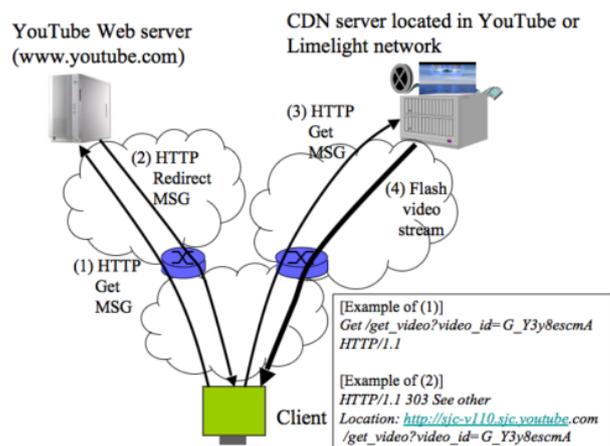


Abbildung 2: Kommunikation zwischen Nutzern, YouTube Server und CDN, aus [18]

er sehen möchte, schickt er eine HTTP Get Anfrage an den Server:

```
GET /get_video?video_id=G_Y3y8escmA
```

In dieser Nachricht fragt der Nutzer nach einem speziellen Video und überträgt dazu die Video ID, in diesem Beispiel lautet diese *G_Y3y8escmA*.

Der YouTube Server antwortet auf diese Anfrage mit einer HTTP 303 Nachricht⁴ und der Umleitungsinformation:

Location:

```
http://sjc-v110.sjc.youtube.com/get_video?video_id=G_Y3y8escmA
```

Durch dieses Header Feld wird der Nutzer auf den entsprechenden Videoserver umgeleitet, von wo dann das gewünschte Video geladen werden kann.

Auf welchen Server man umgeleitet wird, entscheidet der YouTube Webserver, um damit eine Lastverteilung im System zu erreichen. Dazu benötigt er Kenntnis über den Zustand aller CDN Server, wie zum Beispiel CPU Auslastung und aktuelle Verkehrsbelastung [18]. Darauf wird in Kapitel 6 detaillierter eingegangen.

3. GESPEICHERTE DATENMENGE

Leider gibt YouTube, beziehungsweise Google, selbst keine Statistiken und Daten frei. Daher beruhen alle hier gezeigten Zahlen auf Schätzungen oder Hochrechnungen, die auf umfangreichen Versuchen und Analysen beruhen.

In einer Untersuchung der Simon Fraser Universität (Kanada) von Xu Cheng, Cameron Dale sowie Jiagchuan Liu wurde ein Datensammler⁵ eingesetzt, der in den Rubriken „Recently Featured“, „Most Viewed“, „Top Rated“ und „Most Discussed“, über die Zeiträume „Heute“, „diese Woche“, „dieser Monat“ sowie „gesamter Zeitraum“ am ersten Tag der Testreihe Videodaten gesammelt hat. Ausgelesen wurden die Metadaten über die von YouTube bereitgestellt

⁴die Fehlermeldung HTTP 303 meldet, dass die geforderten Ressourcen vorübergehend unter der im Location Feld angegebenen URL erreichbar sind [13]

⁵englisch: crawler

Entwickler API [2].

ID	Y3y8escmA
Uploader	Alkarin
Date Added	May 19, 2007
Category	Entertainment
Video Length	268 seconds
Number of Views	596.272
Rating	4,83
Number of Ratings	1.227
Number of Comments	1.475
Related Videos	wX7B2WyqhMU, ...

Tabelle 1: Die meistverwendeten Anwendungen im Inter-Domain Verkehr zwischen July 2007 und 2009 basierend auf den Protokoll Klassifizierungen, aus [6]

Anschließend wurden bei jedem der dabei gefundenen Videos die ersten 20 Vorschläge von YouTube weiter verfolgt, bis zu einer Tiefe von vier Ebenen. Dieser Vorgang wurde die darauffolgenden Wochen etwa alle zwei bis drei Tage wiederholt. Dabei wurden insgesamt 3.269.030 verschiedenen Videos zwischen dem 22.02.2007 und 18.05.2007 indiziert.

Festgehalten wurden dabei jeweils die in Tabelle 1 aufgeführten Informationen. Dabei fand man heraus, dass 97,9% aller Videos kürzer als 600 Sekunden sind und 99,1% kürzer als 700 Sekunden. Weitaus interessanter ist dabei jedoch, dass 98,3% aller Videos kleiner als 25 MB sind sowie die Durchschnittsgröße 8,4 MB beträgt [2].

Eine Wildcard Suche, also einer Suchanfrage mit „*“, ergab 2007 noch 77,1 Millionen Videos insgesamt, wodurch 650 TB Daten bereitgehalten wurden.



Abbildung 3: Bildschirmfoto einer Wildcard-Testsuche zur Ermittlung der Gesamtzahl der von YouTube gespeicherten Videos am 20.05.2011

Eine Wiederholung dieser Suche am 20.05.2011 ergab 218 Millionen Videos (siehe Abbildung 3). Davon ausgehend, dass sich die durchschnittliche Größe der Videos in den letzten Jahren nicht verändert hat, müsste YouTube inzwischen 1746 TB gespeichert haben. Jedoch ist von weitaus größeren Zahlen auszugehen, da inzwischen Videos in HD mit 720 Pixeln sowie Full HD mit 1080 Pixeln angeboten werden, welche weitaus mehr Speicherplatz benötigen. Genaue Zahlen dazu stehen derzeit nicht zur Verfügung.

4. VERKEHRSMENGE

In Abbildung 2 ist zu erkennen, dass im Jahr 2009 52% des Internetverkehrs auf Webanwendungen anzurechnen ist. Interessant ist an dieser Abbildung, dass Videos 2,64% ausmachen, wodurch sie in dieser Einstufung Platz zwei belegen.

Protokoll- und Portanalysen geben allerdings keine volle Einsicht in die Internetnutzung, wodurch 2009 insgesamt 37% des Verkehrs nicht eindeutig zugeordnet werden konnten.

Diese Ergebnisse von den Arbor Networks sowie der Universität Michigan decken sich mit denen anderer Untersuchungen [6].

Interne Messungen bei diesen Untersuchungen ergaben, dass HTTP Videoübertragungen 25-40% des HTTP Verkehrs ausmachen. YouTube als größte aller Videoseiten nutzt zunehmend HTTP Videos und kann daher als einer der Verantwortlichen für diesen riesigen Anteil an HTTP Videos gemacht werden [6].

Dies deckt sich auch mit einer Veröffentlichung einer Studie von 2007 von Ellacoya Networks, wobei 10% des gesamten HTTP Verkehrs durch YouTube verursacht werden sollen [5].

Rank	Application	2007	2009	Change
1	Web	41,68	52,00	+10,31
2	Video	1,58	2,64	+1,05
3	VPN	1,04	1,41	+0,38
4	eMail	1,41	1,38	-0,03
5	News	1,75	0,97	-0,78
6	P2P	2,96	0,85	-2,11
7	Games	0,38	0,49	+0,12
8	SSH	0,19	0,28	-0,08
9	DNS	0,20	0,17	-0,04
10	FTP	0,21	0,14	-0,07
	Other	2,56	2,67	+0,11
	Unclassified	46,03	37,00	-9,03

Tabelle 2: Die meistverwendeten Anwendungen im Inter-Domain Verkehr zwischen July 2007 und 2009 basierend auf den Protokoll Klassifizierungen, aus [6]

Item	Image	Text	Application	Video
Responses	13.217.499	2.020.436	1.828.486	556.353
Bytes (GB)	37,58	18,59	28,93	5.787,05
% Requests	75,00	11,46	10,38	3,16
% Bytes	0,64	0,32	0,49	98,55
File Size				
Mean (KB)	3,18	18,62	5,84	10.110,72
Median (KB)	3,17	25,76	0,22	8.215,00
COV	0,29	2,31	0,66	0,97
Transfer Size				
Mean (KB)	3,08	9,60	15,97	10.332,44
Median (KB)	3,24	7,26	21,99	8.364,00
COV	0,51	1,26	0,65	0,99

Tabelle 3: korrekt übertragene Daten (HTTP Status 200) zwischen YouTube und einem Campus Netzwerk, aus [4]

In Tabelle 3 wurde zusammengestellt, welche Daten genau übertragen werden und in welchem Verhältnis sie stehen. Den größten Teil der HTTP 200 Antworten machen Bilder und Text mit 86% aus. Anwendungen wie XML und Java Script machen immerhin zusammen 10%, Videos dagegen nur 3%. Im Gegensatz dazu steht, dass Videos für etwa 98,6% der übertragenen Datenmenge verantwortlich sind [4].

Um festzustellen, was für eine Verkehrsmenge das gesamte Internet benötigt wurde eine Studie unter 110 unabhängigen großen Providern angefertigt. Die Ergebnisse des ASN Verkehrs der größten 12 dieser Anbieter wurde in Abbildung 4 dargestellt und extrapoliert. Die dabei entstandene Gerade hat eine Steigung von 2,51, was bedeutet, dass 2,51% des Inter-Domain Verkehrs etwa 1 Tbps darstellt [6]. Damit lässt sich das gesamte Verkehrsaufkommen mit $\frac{1}{2,51} = 39,8$ Tbps im Juli 2009 berechnen [6]. Auf den Monat gerechnet ergeben das etwa 106.600.320 TB an Datenverkehr.

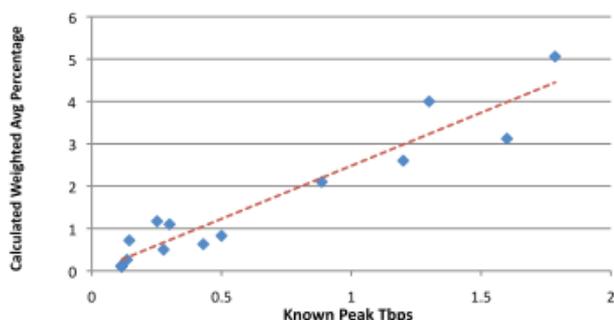


Abbildung 4: Inter-Domain Verkehr aus ASN Analysen, aus [6]

5. CACHING

Um die in Kapitel 4 aufgeführten Datenmengen bewältigen zu können und dabei die Performance aufrecht zu erhalten, müssen technische Lösungen gefunden werden.

In den folgenden Abschnitten werden verschiedene Caching Strategien aufgeführt, die dieses Problem lösen sollen. Bezug genommen wird dabei auf Simulationen der Universität Massachusetts, die den Datenverkehr zwischen Universitätsnetzwerk und YouTube protokolliert und ausgewertet haben.

Caching Strategien sind allgemein immer dann sinnvoll, wenn ein Benutzer beziehungsweise mehrere Benutzer in einem Netzwerk die selben Daten mehr als einmal aufrufen.

5.1 Lokales Caching auf Benutzerseite

Das Caching auf Benutzerseite macht dann Sinn, wenn ein Nutzer das selbe Video öfters als einmal aufruft. Eine Studie hat ergeben, dass allerdings lediglich 25% aller Videos öfter als einmal aufgerufen werden [18]. Wenn weitere Informationen mit in die Cachingstrategie einfließen, wie beispielsweise wie beliebt das jeweilige Video ist, kann die Effizienz gesteigert werden.

Um das zu testen wurde in einem Experiment von einem Nutzer mehrfach das selbe Video aufgerufen. Dabei stellte man fest, dass bei jeder Anfrage ein neuer Datenstrom vom CDN gesendet wurde. Als Schlussfolgerung kann man daher feststellen, dass weder der Browser noch der Flashplayer die Videodaten gespeichert hat [18].

Die Standardeinstellung der Cachegröße der meisten Browser liegt bei 50 MB. Wenn wir weiterhin von 7 MB je Video ausgehen, könnten etwa 7 Videos komplett im Browsercache zwischengespeichert werden. Der Nutzer würde dadurch von

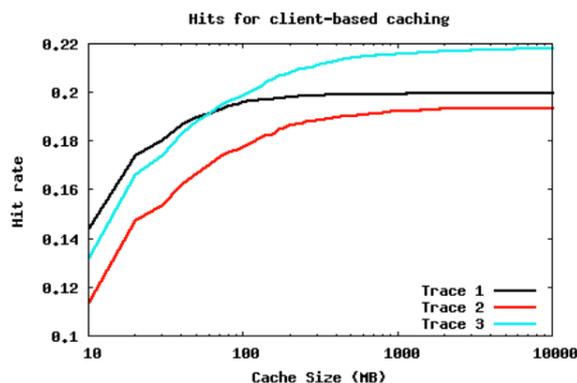


Abbildung 5: erfolgreiche Aufrufe aus dem Browser-cache des Benutzers, aus [18]

einer schnelleren Startzeit der Videos sowie einer unterbrechungsfreien Wiedergabe profitieren [18].

In einer Simulation des lokalen Cachings wurde von einem Cache von 50 MB sowie einer Videogröße von 7 MB ausgegangen. Sollte der Speicher nicht ausreichen, wurde nach dem FIFO Prinzip ⁶ Speicher freigeräumt. Abbildung 5 zeigt die Effizienz dieser Strategie. Bereits ein kleiner Browsercache zeigt im Vergleich zu einem System ohne Cache, dass 3283 von 3899 Videos mit mehreren Aufrufen eines Benutzers von dem lokalen Cache gespeichert werden könnten. [18]

Diesen Trend haben bereits große Browserentwickler entdeckt. So wird Apple mit seinem Betriebssystem OSX Lion eine neue Version des eigenen Browsers Safari freigeben, der das Caching von Video- und Audiodaten erlauben wird [1].

5.2 Peer-to-Peer Caching

Peer-to-Peer Caching ist eine Variante von der in Kapitel 5.1 vorgestellten Caching Strategie. Bei Peer-to-Peer Caching wird auch lokal gespeichert. Wird ein Video aufgerufen, wird überprüft ob das gewünschte Video bereits lokal zwischengespeichert ist. Wenn dies nicht der Fall ist, wird innerhalb des Peer-to-Peer Netzwerkes, zum Beispiel mit Hilfe einer Hash-Tabelle oder einer Datenbank, nachgeschaut ob ein anderer Nutzer dieses Video in seinem Cache hat. Es ist jedoch möglich, dass ein Cache-Treffer im Peer-to-Peer Netzwerk erzielt wird, der entsprechende Nutzer allerdings derzeit nicht online ist. In diesem Fall wird das Video von einem der YouTube Server geladen [18].

Für die Simulation dieser Strategie wird davon ausgegangen, dass der Benutzer online ist. Dies kann festgestellt werden, indem ein Protokoll aller eingehenden und ausgehenden TCP Header angelegt wird. Bei jeder Suchanfrage für ein Video wird nachgeschaut, ob dieses Video bereits im Cache eines anderen Nutzers liegt. Diese Nutzer-Netzwerk-Aktivität wird mit einem Zeitfenster von beispielsweise 30 Minuten analysiert. Es kann dabei davon ausgegangen werden, dass der Nutzer 15 Minuten vor und nach dem Netzwerkzugriff selbst aktiv war.

⁶first in first out

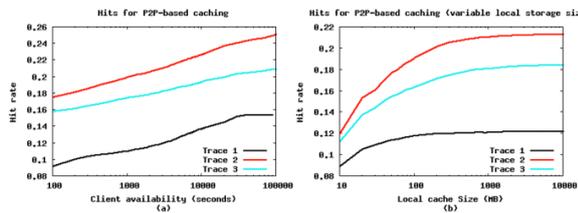


Abbildung 6: Cache Treffer bei Peer-to-Peer Caching, aus [18]

Es wurden für diese Simulation zwei verschiedenen Varianten durchgeführt. In Abbildung 6.a zeigt die X-Achse das jeweilige Zeitfenster und die Y-Achse die Cache-Treffer. In der Variante, die in Abbildung 6.b dargestellt ist, ist das Zeitfenster auf 30 Minuten festgelegt, die X-Achse zeigt hier die Cachegröße zwischen 10 MB und 10 GB an. Dabei ist festzustellen, dass ab einer Cachegröße von etwa 1 GB nur noch geringfügige Verbesserungen zu sehen sind.

5.3 Proxy Caching

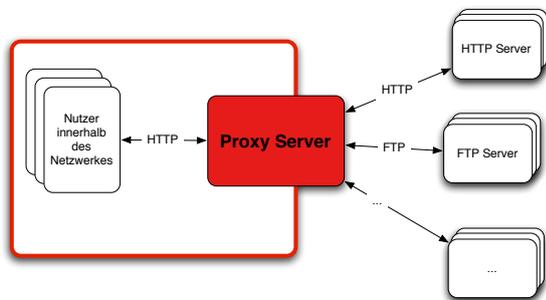


Abbildung 7: Proxy Server als Dienstbringer, nach [3]

Die dritte und letzte Caching Strategie in dieser Arbeit ist Proxy Caching. Das lokale Netzwerk, beziehungsweise der ISP, muss dabei über einen Proxy Server verfügen.

Abbildung 7 zeigt die allgemeine Funktionsweise eines Proxy Servers. Nutzer innerhalb eines geschlossenen Netzwerkes, meist hinter einer Firewall, haben nur Kontakt mit einem Proxy Server. Alle Anfragen, die von den Nutzern gemacht werden, müssen daher zunächst an den Proxy Server gesendet werden, der diese beispielsweise an FTP oder HTTP Server weiterleitet. Die Ergebnisse der Anfragen werden dann an den Nutzer zurück geleitet. [3]

Proxy Caching ist eine effizientere Caching Strategie als beispielsweise lokales Caching, da nur eine Kopie der Daten gehalten wird und damit Festplattenspeicher eingespart werden kann.

Wenn nun ein Nutzer aus diesem Netzwerk eine Anfrage an YouTube senden möchte, muss er sich zunächst an den Proxy wenden. Dieser muss feststellen können, ob dieses Video bereits im Cache liegt. Dazu können ähnlich wie bei Peer-to-Peer Caching Hash-Tabellen oder Datenbanken verwendet werden. Tritt ein Cache-Treffer ein, wird direkt an

den Nutzer übertragen - im Idealfall mit der vollen verfügbaren Bandbreite.

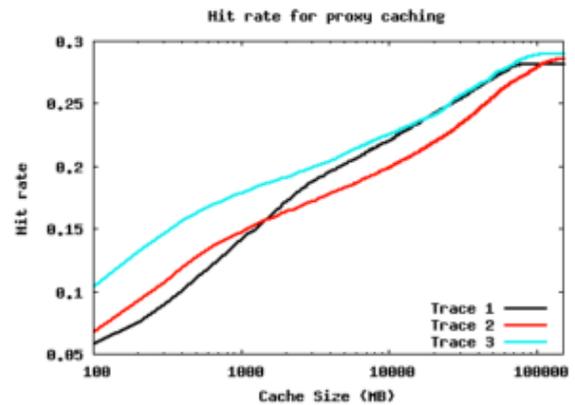


Abbildung 8: Cache Treffer bei Proxy Caching, aus [18]

Andernfalls, also falls das Video nicht im Cache liegt, gibt es zwei Entscheidungsmöglichkeiten für den Proxy Server. Zum einen gibt es die Möglichkeit, dass das gewünschte Video im Cache abgelegt werden soll. Dafür wird die Anfrage des Nutzers vom Proxy an YouTube weitergeleitet und das vom Server gesendete Video zunächst zwischengespeichert und dann direkt an den Nutzer weitergeleitet.

Die zweite Möglichkeit ist, dass sich der Proxy Server dazu entschließt, das Video nicht zu cachen. In diesem Fall wird die Benutzeranfrage nur an den YouTube Server weitergeleitet und der Nutzer erhält direkt das Video, ohne weiteres Caching [18].

Die Simulation dieses Verfahrens ist in Abbildung 8 zu sehen. Auf der X-Achse ist die Cachegröße des Proxy Server verzeichnet, die Y-Achse gibt die Cache-Treffer an.

Bereits kleine Änderungen an der Cachegröße haben dabei einen enormen Einfluss auf die Trefferquote. Bereits der Schritt von 100 MB zu 1 GB steigert die Chance auf einen Cache-Treffer um 10%. Auch eine Erweiterung des Speichers bis zu 100 GB verbessert noch das Ergebnis auf bis nahezu 25%. Wie wir in Kapitel 5.1 festgestellt haben, werden nur 25% aller Videos mehr als einmal angesehen. Dadurch haben wir bei Proxy Caching mit 100 GB Cachegröße nahezu das erreichbare Maximum erreicht.

Diese Simulation bestätigt unsere Annahme, dass Proxy Caching eine sehr effektive und kostengünstige Möglichkeit ist.

Als weiteren Vorteil könnte man auch betrachten, dass Netzwerke zeitweise nicht verfügbar sind. Sollte die bereitgestellten Daten allerdings bereits im Cache des Proxy Server vorliegen, hätten die Nutzer in diesem Netzwerk trotzdem die Möglichkeit, darauf zuzugreifen [3].

Doch bei all diesen Vorteilen bietet Proxy Caching unter Umständen auch Nachteile. Nehmen wir an, wir sind in einem Netzwerk, in dem nur wenig Nutzer immer auf die selben Dienste zugreifen. In diesem Fall würden die zwischengespeicherten Daten durch den Cache-Manager, je nach ein-

gesetzter Verdrängungsstrategie, kontinuierlich durch neue ausgetauscht. Der Proxy Server würde daher nur als Gateway zwischen Nutzer und dem Netzdienst fungieren [3].

Weiterhin wäre denkbar, dass die gespeicherten Daten zum Zeitpunkt des Abrufes durch den Nutzer bereits veraltet sind. Es gibt verschiedene Ansätze, um diesem Problem entgegen zu wirken. Es kann jedoch nicht garantiert werden, dass immer die aktuellste Version ausgeliefert wird.

5.4 Vergleich der Cachingstrategien

Alle drei hier aufgeführten Cachingstrategien sollen Netzverkehr vermindern und Vorteile für die Benutzer bringen wie verkürzte Startzeiten der Videos sowie eine unterbrechungsfreie Wiedergabe. Aber auch der Netzbetreiber wie zum Beispiel eine Universität oder ein ISP profitieren von sinnvollen Caching Strategien, es können je nach Region 40-60% Bandbreite [8] eingespart werden.

Wenn wir uns die Simulationen genauer betrachten, stellen wir schnell fest, dass sich die Verfahren in ihrer Effizienz stark unterscheiden.

Während Peer-to-Peer Caching lediglich eine leichte Verbesserung zum lokalen Caching darstellt, ist es mit Proxy Caching möglich, nahezu alle mehrfach angesehenen Videos zwischenspeichern und anderen Nutzern aus dem selben Netzwerk auszuliefern.

6. CONTENT DISTRIBUTION NETWORK (CDN)

YouTube untersagt laut Nutzungsbestimmungen das (persistente) Speichern der geladenen Videos. Caching dagegen, wie in Kapitel 5 vorgestellt, ist allerdings nicht persistent. In einem Cache werden nur die Datenblöcke gehalten, auf die oft zugegriffen wird.

Da der Datenverkehr von YouTube sehr groß ist, werden die Videos nicht nur von einem Server bereitgestellt. Aus diesem Grund werden die Videos von einem CDN (Limelight Networks [4]) ausgeliefert.

Bei einem CDN handelt es sich um ein stark verteiltes Netz von Servern, die über das Internet miteinander verbunden sind. Ziel dieses Netzwerkes ist es, (meist) Medieninhalte an Nutzer auszuliefern.

Ein CDN besteht aus einem Ursprungsserver, auf dem der Diensteanbieter seine Inhalte ablegt, einem Distributions-system, das die Inhalte auf zahlreiche Replica-Server verteilt, die Kopien der Mediendaten speichern. Die Nutzeranfragen werden von dem Request-Routing-Server auf die Replica-Server weitergeleitet. Auf welchen Server weitergeleitet wird, bestimmt eine Kennzahl, die von dem Accounting-System festgelegt wird. Kriterien für die Kennzahl sind unter anderem CPU Auslastung sowie die Anzahl der aktuell aktiven Verbindungen sowie die geographische Entfernung, um die Daten möglichst nahe am Benutzer auszuliefern. In seltenen Fällen kann es auch relevant sein, ob der Nutzer für die gewünschten Dienste bezahlt hat oder nicht [12]. In Abbildung 9 sehen wir eine vereinfachte Darstellung eines CDN. Um es übersichtlich zu halten, werden hier nur die Replica-Server gezeigt.

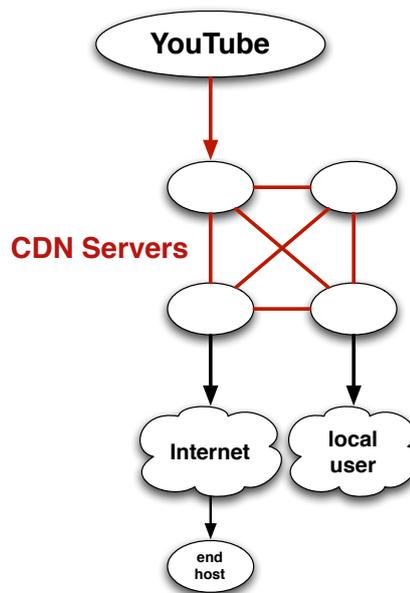


Abbildung 9: Content Distribution Network, nach [11]

Sollte ein oder mehrere der Replica Server eines solchen CDN innerhalb eines großen Netzwerkes sein, wie bisher von einer großen Universität, hätte dies den Vorteil, dass keine Internetverkehr erzeugt wird sondern die Mediadaten innerhalb des Netzwerkes verteilt werden.

Ein CDN hat also besonders dann einen großen Vorteil, wenn der Nutzer geographisch möglichst nahe an einem der Replica Servern ist.

7. ZUSAMMENFASSUNG

Wir haben gesehen, dass der Internetverkehr sehr groß ist. Im Jahr 2009 war von geschätzten 40 TB je Sekunde die Rede. Einen großen Teil davon nehmen Web 2.0 Anwendungen wie YouTube ein, die sehr große Datenmengen bereithalten. Um diese Daten kostengünstig, schnell und sicher an den Benutzer zu senden, haben wir uns verschiedene Caching Strategien angeschaut, wobei wir zu dem Ergebnis gekommen sind, dass die effektivste Möglichkeit die Daten zu Cachen der Einsatz eines Proxy Servers ist, wie er heute auch von vielen ISP betrieben wird.

Allerdings ist fraglich, in wie fern die Ergebnisse der hier aufgeführten Studien verallgemeinert werden können. Die Versuche wurden in einem großen, lokalen Netzwerk von Universitäten durchgeführt. Es ist davon auszugehen, dass Studenten ein anderes Verhalten aufweisen als der Großteil der Bevölkerung, da sie zum einen recht jung sind und daher mit der Materie Internet eher vertraut sind und das Web 2.0 vermutlich daher auch intensiver nutzen. Zum anderen sind die Gruppierungen innerhalb der Studentengemeinschaft recht groß, wodurch das „friend-casting“, also das Teilen von interessanten Inhalten schneller und in größerem Umfang statt findet.

Weiterhin fraglich ist auch, in wie fern sich die Ergebnisse aus Kapitel 3 von der heutigen, realen Datenmenge unterscheiden. Die aufgeführten Zahlen stammen von 2007, wurden also bereits zwei Jahre vor Einführung von HD Videos auf YouTube ermittelt. Aktuell ist es möglich, Videos mit einer Größe von bis zu 2 GB auf die YouTube Server zu übertragen. In wie fern diese noch komprimiert werden war in Recherchen zu diesem Artikel nicht heraus findbar. Man kann allerdings davon ausgehen, dass die damalige Durchschnittsgröße eines Videos von 8,4 MB heute weit nicht mehr ausreichen wird.

Interessant wäre eine Kombination der hier vorgestellten Möglichkeiten zur Auslieferung der Daten und Vermindert des Datenverkehrs. Vorstellbar wäre weiterhin der Einsatz eines CDN, im Idealfall mit jeweils mindestens einem Replica Server in jedem großem Netzwerk wie Universitäten und Firmen. Wenn zusätzlich noch ein Proxy Server eingesetzt wird, könnten die Replica Server zusätzlich entlastet werden, wodurch die Performance für Nutzer außerhalb dieses Netzwerkes vermutlich steigen würde.

8. LITERATUR

- [1] Apple. Osx lion. <http://www.apple.com/de/macosex/whats-new/features.html#safari>, 2011.
- [2] X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of YouTube Videos. *2008 16th International Workshop on Quality of Service*, June 2008.
- [3] J. Elkner. Wissenswertes über proxy caches. <http://www.linofee.org/jel/proxy/Knowledge/german.shtml>.
- [4] P. Gill, M. Arlitt, and Z. Li. YouTube Traffic Characterization : A View From the Edge. *Technical*, 2007.
- [5] Google Blog. YouTube verursacht 10% des HTTP-Traffics. <http://www.googlewatchblog.de/2007/06/youtube-verursacht-10-des-http-traffics/>, Juni 2007.
- [6] C. Labovitz, S. Iekel-johnson, A. Arbor, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. *Communication*, 2008.
- [7] Online Medienbeobachtung. Social media monitoring und empfehlungsmarketing. <http://www.medienbeobachtung-blog.de/tag/friend-casting/>, März 2010.
- [8] Opteq. Isp web caching & netcache clusters. <http://www.opteqint.net/content/isp-web-caching-netcache-clusters>, Februar 2009.
- [9] P. Roth. Neuer rekord: 600 millionen aktive facebook nutzer. <http://allfacebook.de/news/neuer-rekord-600-millionen-aktive-facebook-nutzer>, Januar 2011.
- [10] B. Suzy and H. Leilani. Youtube U.S. Web Traffic grows 75 percent week over week , according to NIELSEN // Netratings. (408), Juni 2006.
- [11] P. D. Tsang. Content distribution backbone network. http://mwnet.cse.ust.hk/p2pstream/research_cdbn.html, 2008.
- [12] Wikipedia. Content distribution network. http://de.wikipedia.org/wiki/Content_Distribution_Network, April 2011.
- [13] Wikipedia. Http-statuscode. <http://de.wikipedia.org/wiki/HTTP-Statuscode>, Mai 2011.
- [14] Wikipedia. Youtube. <http://de.wikipedia.org/wiki/YouTube>, Mai 2011.
- [15] Youtube. At five years, two billion views per day and counting. <http://youtube-global.blogspot.com/2010/05/at-five-years-two-billion-views-per-day.html>, Mai 2010.
- [16] Youtube. Zeitachse. http://www.youtube.com/t/press_timeline, Mai 2010.
- [17] Youtube. Thanks, youtube community, for two big gifts on our sixth birthday! <http://youtube-global.blogspot.com/2011/05/thanks-youtube-community-for-two-big.html>, 2011.
- [18] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch Global , Cache Local : YouTube Network Traffic at a Campus Network - Measurements and Implications. *Network*, 2010.