

Seminararbeit

# Grundlagen von Content Delivery Networks

Thomas Seebauer  
15.11.2004

## Seminar Content Delivery Networks

Universität Mannheim  
Lehrstuhl Praktische Informatik IV: Prof. Dr. Wolfgang Effelsberg  
Betreuer: Dipl.-Ing. Hans Christian Liebig

# **Inhalt**

- 1 Einleitung**
- 2 Gründe für die Entwicklung von Content Delivery Networks**
  - 2.1 Inhalte Ortsnah verteilen: Die Entstehung von CDNs**
  - 2.2 Anforderungen an ein CDN**
- 3 Klassifizierung der übermittelten Inhalte**
- 4 Architektur eines CDN**
- 5 Verteilung und Verwaltung von Content Replikationen**
  - 5.1 Verteilung von Content Replikationen**
  - 5.2 Verwaltung von Content Replikationen**
- 6 Wie findet man den besten Surrogate Server?**
- 7 Request Routing Verfahren**
  - 7.1 DNS basiertes Request Routing**
  - 7.2 Transport Layer Request Routing**
  - 7.3 Application Layer Request Routing**
- 8 Peer to Peer Netzwerke**
- 9 Weiterführende Content Verarbeitung**
- 10 Zusammenfassung**

# 1 Einleitung

Ursprünglich für die Verteilung statischer HTML Seiten vornehmlich für Wissenschaftliche Institutionen gedacht, hat sich die Struktur des Internet in den letzten Jahren zunehmend überfordert gezeigt von den Anforderungen moderner Datenkommunikation.

Dies führte zu einem zunehmenden Interesse daran, neue Wege zur Verteilung von Daten über das Internet zu finden. Dabei stellte sich sehr schnell heraus, dass die vielversprechendsten Ansätze auf „Content Awareness“ beruhten – also der Anpassung der Übertragungsart und der Übertragungswege an die Art der zu übertragenden Daten. Diese Anpassung der alten inhaltsunabhängigen und zentralistischen Netzwerkstrukturen an inhaltsgebundene Architekturen schuf die „Content Delivery Networks“ (CDN).

Moderne CDNs nutzen die herkömmliche Netzwerkstruktur auf verschiedenen Levels um Daten mit Hilfe von Replizierung und Anfragemanagement Zeit- und Ortsnah an die Endbenutzer zu verteilen.

Diese Seminararbeit soll einen Überblick über die Grundlagen von CDNs geben. Nach einer Untersuchung der Gründe für Einsatz und Entwicklung eines CDN sollen die Grundlegenden Anforderungen an ein CDN festgelegt und somit dessen elementare Bestandteile identifiziert werden.

## 2 Gründe für die Entwicklung von Content Delivery Networks

Ursprünglich wurde das Internet entwickelt, um in erster Linie statische HTML Seiten mit wissenschaftlichen Informationen über die Verbindung mehrerer Computer Personen, die Räumlich mehr oder weniger weit getrennt waren für gemeinsame Arbeiten zugänglich zu machen. Anfänglich entwickelten sich mehrere solcher Netzwerke parallel und schlossen sich nach und nach zu immer größeren Verbänden zusammen. So entstand die Struktur des heutigen Internet als Netzwerk von Netzwerken.

Ein Content Provider stellt hierbei den Inhalt auf seinem Server bereit und die Endbenutzer können diesen von dort abholen, wobei die Datenpakete mittels Routingverfahren durch die verschiedenen Teil-Netze des Internet vom Absender bis zum Empfänger geschleust werden.

Diese Struktur war und ist einer der größten Vorteile des Internet, stellt gleichzeitig aber auch dessen größtes Problem auf dem Weg in die Kommunikation der Zukunft dar. Einerseits wird durch sie eine größtmögliche Erweiterbarkeit und Fehlertoleranz erreicht, andererseits kann aber keinerlei Aussage gemacht oder gar Garantie gegeben werden über die Zuverlässigkeit und Geschwindigkeit der Datenübermittlung. Doch gerade dies sind entscheidende Kriterien für eine moderne Kommunikations-Infrastruktur.

Heute geht die Nutzung des Internet weit über dessen ursprüngliche Anwendungsgebiete hinaus. Insbesondere macht die kommerzielle Nutzung inzwischen einen wesentlichen Teil des Internet aus. Hierbei ist es natürlich zunehmend erforderlich, juristisch verbindliche Aussagen über die Verlässlichkeit einer Internetanwendung zu machen oder die Geschwindigkeit einer Datenübermittlung garantieren zu können.

Neben dem Zweck der Datenübermittlung hat sich auch die Art der übermittelten Daten stark verändert. Wurden früher nur statische HTML Dateien übertragen, so werden nun zunehmend Multimediainhalte und dynamische Daten bis hin zu Live Video Streams übermittelt. Auch hierbei ist beispielsweise eine garantierte Mindestbandbreite dringend erforderlich, um Verbindungsabbrüche zu vermeiden.

Und selbst beim ureigensten Zweck des Internet – dem Übertragen von Informationen mittels des HTTP Protokolls stößt die Technologie immer häufiger an ihre Grenzen. Exponentiell wachsende Benutzerzahlen haben dafür gesorgt, dass Die Leitungen insbesondere an den Übergangspunkten zwischen den Teilnetzen des Internet (Peering- oder Routing Points) schnell überlastet sein können.

Verschärfend kommt hinzu, dass der überwiegende Teil des gesamten Datenverkehrs im Internet auf einige wenige Content Provider entfällt [6], wodurch es in Stosszeiten zusätzlich noch zu einer Überlastung des Content Servers kommen kann.

Dies führt dazu, dass die Zugriffszeiten im Internet trotz stetig steigender Bandbreiten insbesondere auf der so genannten „letzten Meile“ immer noch im Bereich von mehreren Sekunden liegen und je nach Auslastung der Content Server und des Übertragungsweges stark schwanken. Für kommerziell orientierte Real-Time Anwendungen ist dies oft nicht ausreichend und kann auch bei herkömmlichen Internetseiten dazu führen, dass sich die Benutzer einer anderen, eventuell schnelleren Konkurrenzseite zuwenden.

Der erste Ansatzpunkt zur Lösung dieser Probleme schien für viele Content- und Internet Service Provider (ISP) eine Erhöhung der Rechner- und Leitungskapazitäten zu sein. Doch es wurde schnell erkannt, dass damit den Problemen nur bedingt beizukommen war.

So konnten zwar einzelne ISP ihre Leitungskapazitäten erhöhen, doch durch die Struktur des Internet als Netz der Netze, gab es stets irgendwo entlang eines Übertragungsweges noch Schwachpunkte, auf die niemand einen Einfluss hatte, da keine zentrale Autorität vorhanden war, die kleinere ISP zum Ausbau ihrer Netze zwingen konnte.

Auch bei den Content Servern konnte man mit einer erhöhten Rechnerkapazität deren Überlastung in Stosszeiten reduzieren, doch durch die rasant wachsenden Benutzerzahlen wurden diese Aufrüstungen oft relativiert, kurz nachdem sie eingeführt waren, und schließlich stieß man unweigerlich an die Grenze dessen, was ein einzelner Rechner, wie Leistungsstark er auch sein mochte, bewältigen konnte.

Daher begann man damit, bei großen Content Providern als Server nicht mehr einzelne Rechner zu verwenden, sondern vernetzte Rechnercluster, auf die die Last der ankommenden Content Requests von einem zentralen Rechner aus gleichmäßig verteilt werden konnte.

Dieses so genannte „Load Balancing“ wird auch heute noch vielfach eingesetzt und trägt bereits alle wichtigen Merkmale moderner Content Delivery Networks.

## **2.1 Inhalte Ortsnah verteilen: Die Entstehung von Content Delivery Networks**

Nachdem sich mit dem Load Balancing eine Technologie entwickelt hatte, die es stark belasteten Content Providern ermöglichte, mit vergleichsweise geringen Kosten die Kapazität ihrer Content Server beinahe beliebig zu erweitern, blieb immer noch das strukturelle Problem des Internet. Alle Anfragen nach Inhalten wie auch die Übermittlung der Inhalte an den Endbenutzer mussten über viele Subnetze weitergeleitet werden und passierten somit viele potentielle Engpässe. Doch das Load Balancing bot auch eine Lösung für dieses Problem, hatte man doch damit schon das Konzept, mehrere Kopien des Contents anzulegen und bei Bedarf die passende Kopie abzurufen. Warum also nicht die Server mit den Kopien in physikalischer Nähe zu den Endbenutzern aufstellen.

Eine erste Umsetzung dieses Grundgedankens aller Content Delivery Networks stellte die Proxy Server Technologie dar. Hierbei laufen die Verbindungen aller Endbenutzer eines ISP zunächst einmal über einen lokalen Server, welcher alle Anfragen abfängt, den angeforderten Content vom ursprünglichen Content Server holt, im lokalen Speicher abspeichert und ihn dann an den Endbenutzer weiterleitet. Fragt nun ein anderer Endbenutzer den gleichen Content ab, wird dieser vom Proxy Server aus dessen lokalem Speicher (Cache), an den Endbenutzer weitergereicht und nicht erneut vom ursprünglichen Content Server geholt. Durch diese Proxy Caching Technik konnten die Engpässe bei den eigentlichen Content Servern und an den Peering Points zwischen den Teilnetzen des Internet schon erheblich reduziert werden. Allerdings hat diese Technik den entscheidenden Nachteil, dass die Content Provider keinen direkten Einfluss auf die Reproduktionen ihres Contents auf den Proxy Servern haben. Zudem sind Proxy Caches nicht für Echtzeitanwendungen und Dynamische Inhalte geeignet und da auch sie unabhängig von der Art der Inhalte arbeiten und somit die Endbenutzer nicht nach Prioritäten einstufen können, ist es auch mit Proxys unmöglich, verbindliche Aussagen über Verlässlichkeit und Geschwindigkeit von Datenübermittlungen, so genannte „quality of service (QoS) agreements“ zu machen. Daher muss ein modernes Content Delivery Network einige weitere Anforderungen erfüllen.

## **2.2 Anforderungen an ein CDN**

Um Daten effizient an eine große Anzahl von Endbenutzern verteilen zu können, reicht es nicht aus, nur Caching Server im Internet zu verteilen. Es muss auch dafür gesorgt werden, dass der Endbenutzer zu dem, für ihn jeweils am besten geeigneten Cache weitergeleitet wird und die Kopien des Contents auf den Caching Servern stets aktuell gehalten werden. Daneben gilt es auch noch einigen andere Anforderungen, speziell kommerzieller Anwendungen gerecht zu werden. Daher muss ein CDN mehrere wichtige Anforderungen erfüllen [1].

### **- Funktionen zum verteilen der Content Kopien**

Die Server mit den Kopien der zu verteilenden Daten (Surrogate Server) können nicht wahllos im Internet verteilt werden, sondern sollten möglichst an Strategisch günstigen Orten stehen. Dabei sollten diese Server aber nicht statisch festgelegt sein, sondern sich den Schwankungen im Datenfluss des Internet dynamisch anpassen.

### **- Funktionen zur Aktualisierung der Content Kopien**

Da die Content Provider bei einem CDN jederzeit Einfluss auf ihren Content nehmen können sollen, werden Mechanismen zur Aktualisierung der Replikationen benötigt, um nicht wie bei herkömmlichen Proxy Servern auf passive Abfragemechanismen warten zu müssen.

### **- Analysefunktionen**

Um den Anforderungen moderner Internetanwendungen zu genügen müssen verschiedenste Analysemechanismen vorhanden sein.

Einerseits muss die Aktivität des gesamten Netzwerks ständig überwacht werden, um bei einer Anfrage sofort einen passenden Surrogate Server finden zu können.

Andererseits müssen für kommerzielle Anwendungen Informationen über Anzahl und Art der Zugriffe gesammelt werden, um marktwirtschaftliche Analysen des Verhaltens der Endbenutzer zu ermöglichen.

#### **- Replikationsauswahlfunktionen**

In Abhängigkeit von den Ergebnissen der Netzwerkanalyse müssen die Endbenutzer stets zu den bestmöglich erreichbaren Surrogate Servern weitergeleitet werden.

#### **- Funktionen zur Verbindungssteuerung**

Im Falle einer, während einer laufenden Übertragung plötzlich auftretenden Überlastung in einem Teil des Netzwerkes sollten die Verbindungen der Endbenutzer möglichst ohne eine Unterbrechung des Datentransfers auf andere Surrogate Server umgeleitet werden.

### **3 Klassifizierung der übermittelten Inhalte**

Aufgrund der vielfältigen Natur der, im Internet bereitgestellten Daten, sind die Anforderungen an ein CDN je nach Art der zu übermittelnden Daten sehr variabel.

Daher soll hier ein kurzer Überblick über die, am weitesten verbreiteten Anwendungen gegeben, und diese daraufhin untersucht werden, inwieweit sie vom Einsatz eines CDN profitieren können [3].

#### **- Statische Inhalte**

Hierunter fallen alle Inhalte, die sich nicht sehr häufig ändern. Insbesondere die statischen HTML Seiten des „alten“ Internet gehören in diese Kategorie, oder auch die klassischen Anwendungsgebiete des FTP Protokolls wie beispielsweise das Herunterladen von Softwarepatches.

Diese Art von Inhalte stellt keine sehr hohen Anforderungen an ein CDN und kann mit den klassischen Proxy Technologien bearbeitet werden.

Dementsprechend führt die Nutzung eines CDN hier nur zu einer mäßigen Verbesserung der Übertragungsleistungen.

#### **- Dynamische Web Applikationen**

Die überwiegende Mehrheit der heute existierenden Internetseiten gehört in diese Kategorie. Es handelt sich dabei um, mit Hilfe von serverseitigen Scriptsprachen oder Anwendungen meist mit Datenbankanbindung erstellte HTML Seiten wie beispielsweise Online-Shops.

Da sich bei Web Applikationen die Inhalte beinahe ununterbrochen und häufig von User zu User verändern, ist es oft schwierig, überhaupt replizierbare Elemente zu finden, um diese auf den Surrogate Servern zu speichern. Dringend erforderlich für diese Art von Inhalt sind daher aktive Update Mechanismen.

Ein anderer Ansatz ist es, die ganze Applikation auf die Surrogate Server zu kopieren, was aber ebenfalls zu komplizierten Datenbankabgleichsmechanismen führt.

In jedem Fall sind insbesondere für e-Commerce Anwendungen sichere Übertragungswege und eine Transaktionssteuerung von Nöten.

Diese Kategorie von Content stellt für ein CDN die größte Herausforderung dar.

Aber vor allem kommerziell geprägte Anwendungen können sehr von den Zusatzdiensten profitieren, die ihnen ein CDN mit seinen Analysefunktionen zur Verfügung stellen kann.

### **- Streaming Media**

Bei streaming Media handelt es sich meist um Video oder Audio Übertragungen.

Man unterscheidet hierbei zwischen „Live“ und „On Demand“ Streams.

Bei On Demand Streams können annähernd dieselben Techniken zum Einsatz kommen wie bei statischen Inhalten, da hierbei eine statische Video- oder Audiodatei auf den Surrogate Servern zur Übertragung bei Bedarf (on demand) bereitgestellt wird. Der Unterschied besteht darin, dass auf dem Rechner des Endbenutzers eine Spezielle Clientsoftware installiert ist, die mit dem Abspielen der Mediendatei noch während der Übertragung beginnt. Daher muss bei diesem Vorgang natürlich eine kontinuierliche Mindestübertragungsrate garantiert sein, um ein Abreißen des Mediastreams zu verhindern.

Bei Live Streams werden die Mediendaten nicht dauerhaft auf den Surrogate Servern gespeichert. Diese dienen hier nur als Verteilerstationen, so dass der Mediastream in einer Baumstruktur verteilt werden kann und nicht jeder Endbenutzer eine eigene Verbindung zum ursprünglichen Content Server aufbaut und dessen Internetanbindung überlastet. Hierbei wird natürlich sowohl zum Ursprungsserver wie auch zu den Clients eine stetige Mindestbandbreite benötigt und die Surrogate Server müssen über eine spezielle Software verfügen, die das Aufsplitten und verteilen des Mediastreams ermöglicht.

Um den, für Streaming Media nötigen stetigen Datenfluss zu gewährleisten, verfügen die Surrogate Server außerdem häufig über zwei weitere Schutzmechanismen.

Zum einen besitzen sie die Möglichkeit auch untereinander Daten auszutauschen (Peering), für den Fall, dass bei einzelnen Surrogate Servern die Verbindung zum Ursprungsserver abreißt und zum anderen können sie auch die angeschlossenen Clients untereinander weiterreichen, falls die gleichmäßige Verteilung des Datenflusses im Netzwerk dies erfordert.

Eine Übertragung von Streaming Media über das Internet ohne CDN ist so gut wie unmöglich, wodurch diese Anwendung zu den Kernaufgaben eines CDN gehört.

## **4 Architektur eines CDN**

Ausgehend von den zuvor festgestellten Anforderungen an ein CDN und den verschiedenen Erfordernissen, die von den übermittelten Inhalten diktiert werden, kann man nun die fünf wesentlichen Komponenten eines CDN identifizieren [5][2].

### **- Surrogate Server**

Die elementarste Komponente sind natürlich die Surrogate Server. Diese ermöglichen es dem CDN, den Content Ortsnah am Endbenutzer zur Verfügung zu stellen.

### **- Distribution System**

Das Distribution System ist für die Replizierung und Verteilung des Contents zuständig. Es soll versuchen, durch Analyse der vergangenen Content Anforderungen die Content Replicas möglichst effizient auf die, zur Verfügung stehenden Surrogate Server zu verteilen und anschließend aktiv zu aktualisieren, wenn sich der Content auf dem Ursprungsserver verändert.

### **- Request Routing System**

Das Request Routing System überwacht ständig den Status des gesamten CDN. Es nimmt die Content Anforderungen der Endbenutzer entgegen und leitet diese basierend auf einer Analyse des aktuellen Netzwerkstatus an den am besten geeigneten Surrogate Server weiter. Anschließend überwacht es die Verbindungen um bei unerwartet auftretenden Problemen die Endbenutzer auf andere Surrogate Server umleiten zu können

### **- Accounting System**

Das Accounting System zeichnet sämtliche Aktivitäten des CDN auf und ermöglicht damit viele zusätzliche Anwendungen - von der Fehleranalyse über die Anfertigung von Zugriffsstatistiken bis hin zu Bezahlungsfunktionen für den Zugriff auf einzelne Teile des, im CDN angebotenen Contents.

### **- Peering System**

Das Peering System dient dem CDN zur Kommunikation mit anderen CDN. Eine inter CDN Kommunikation ermöglicht eine noch bessere Content Verteilung im gesamten Internet.

## **5 Verteilung und Verwaltung von Content Replikationen**

Bei der Verteilung und Verwaltung von Content Replikationen innerhalb des CDN – also den typischen Aufgaben des Distribution Systems - handelt es sich um die komplexesten und umstrittensten Aufgaben eines CDN. Auf diesem Gebiet findet derzeit der größte Teil an Neuentwicklungen statt und die verschiedenen Verfahren werden kontrovers Diskutiert. Eine intensive Beschäftigung mit diesem Thema würde den Rahmen dieser Seminararbeit bei weitem überschreiten. Daher sollen hier nur die Grundprobleme kurz angesprochen und einige Hinweise auf weiterführende Literatur gegeben werden.

### **5.1 Verteilung von Content Replikationen**

Hierbei handelt es sich um die äußerst komplexe Frage, wo und wie man die Kopien des zu verteilenden Contents im Internet ablegt und dabei die Wartezeiten für die Endbenutzer, die Zahl der Content Kopien und die, zur Verwaltung der Kopien benötigte Bandbreite minimiert.

Hierzu existiert eine Vielzahl von Algorithmen, von denen viele in [7] untersucht werden. Die Mehrheit dieser Verfahren hat gemeinsam, dass sie auf mindestens einem statischen Faktor beruht. Das heißt, entweder wird bei vorgegebener Netzwerkstruktur die Performance für die User optimiert, oder es wird eine optimale und kosteneffektive Netzwerkstruktur für eine vorgegebene Mindestperformance bei den Usern berechnet. Zudem wird bei diesen Verfahren das, einmal berechnete Verteilungsmuster statisch festgelegt und kann nicht ohne weiteres wieder geändert werden. Dadurch kann die Verteilung der Content Replikationen natürlich nur bedingt auf die unaufhörlichen dynamischen Veränderungen in der Struktur des Internet oder im Userverhalten angepasst werden.



Erst in jüngster Zeit beschäftigt man sich zunehmend mit Möglichkeiten, die Verteilung des Contents dynamisch an die Veränderungen anzupassen. Allerdings sind solche Algorithmen aufgrund ihrer Komplexität bislang nur auf CDN mit begrenzten Ausmaßen anwendbar. Ein solcher Ansatz wird beispielsweise in [8] vorgestellt.

## 5.2 Verwaltung von Content Replikationen

Ebenso wichtig und nicht minder kontrovers diskutiert wie die Verteilung von Content Replikationen ist die Verwaltung derselben. Sind die Replikationen erst einmal verteilt, müssen sie immer auf dem neuesten Stand gehalten werden und Zu- oder Abgänge auf dem Master Server müssen an die Surrogate Server weitergegeben werden.

Bei herkömmlichen Caching Verfahren, wie sie beispielsweise bei der Proxy Technologie eingesetzt werden, findet eine passive Aktualisierung der Daten im so genannten „Pull“ verfahren statt. Das heißt, dass der Aktualisierungsvorgang vom Surrogate Server eingeleitet wird, wenn ein bestimmtes Verfallsdatum des Contents überschritten ist. Dies hat aber den entscheidenden Nachteil, dass bei einer Aktualisierung des Content auf dem Master Server vor diesem festgelegten Verfallsdatum Inkonsistenzen auftreten können. Daher sind Caching verfahren für moderne CDN nur bedingt brauchbar.

Die logische Konsequenz ist der Einsatz eines „Push“ Verfahrens, bei dem der Master Server nach einer Aktualisierung seines Contents diesen aktiv auf alle Surrogate Server verteilt. In einem CDN übernimmt diese Aufgabe das Distribution System.

Allerdings hat auch dieses aktive Aktualisierungssystem Nachteile. Bei den großen Datenmengen die beispielsweise beim On Demand Streaming übertragen werden müssen, kann es leicht dazu kommen, dass nach einer Aktualisierung des Content auf dem Master Server alle Leitungen zu den Surrogate Servern durch den Updatevorgang ausgelastet sind und keine anderen Anwendungen mehr möglich sind.

Ein Lösungsansatz für dieses Problem besteht darin, die Updatezeiten auf, für gewöhnlich trafficarme Tageszeiten zu verlegen, um die Endbenutzer möglichst wenig zu belasten. Allerdings entsteht dadurch wiederum eine Verzögerung zwischen Aktualisierung und Verteilung der aktualisierten Daten.

Ein weiteres Problem aktiver Updatevorgänge entsteht, wenn der Content häufiger aktualisiert wird als ihn die Endbenutzer abrufen. In diesem Fall entsteht eine erhebliche Menge an eigentlich unnötigem Datenverkehr und somit unnötige Kosten.

Ein Verfahren, welches mit beiden angesprochenen Problemen fertig werden soll, ist das Versenden von Update Notifications. Hierbei wird nach einem Content Update nicht der ganze Aktualisierte Content an die Surrogate Server versendet, sondern nur eine Notiz mit dem Hinweis auf den veränderten Content. Der Surrogate Server speichert diese Notiz, holt sich den aktualisierten Content aber erst dann, wenn dieser von einem Endbenutzer angefordert wird. Doch auch dieses Verfahren ist wiederum nicht frei von Problemen, denn hierbei wird der erste Benutzer, der einen neuen Content abrufen ähnliche Verbindungseigenschaften erleben als hätte er eine direkte Verbindung zum Masterserver ausgebaut und kein CDN benutzt. Im schlimmsten fall egalisiert dieser Effekt das CDN, wenn die Intervalle der Contentanforderungen durch Endbenutzer denen der Contentaktualisierungen entsprechen.

Aufgrund dieser ungelösten Probleme gibt es auch zur Verwaltung der Content Replikationen eine Vielzahl von Lösungsansätzen. Ebenso wie bei der Content Verteilung ist auch dies eines der Hauptentwicklungsfelder im Bereich der CDN.

## 6 Wie findet man den besten Surrogate Server?

Bevor man sich mit dem, vielleicht wichtigsten Gebiet eines CDN, dem Request Routing beschäftigen kann, ist es zunächst von Nöten, die Mittel zur Bestimmung des besten Surrogate Servers zu analysieren.

Für die Auswahl des besten Surrogate Servers für eine Content Anfrage sind einige Kriterien von Bedeutung. Am einfachsten zu bestimmen sind hier bei die geographische Nähe des Surrogate Servers zum Endbenutzer und die Auslastung des Surrogate Servers.

### - geographische Nähe

Die geographische Position des Endbenutzers lässt sich meist leicht mit Hilfe von IP Header Informationen oder Bind abfragen bestimmen.

Dadurch kann man einen Surrogate Server bestimmen, der im Idealfall an denselben Point of Presence (PoP) angeschlossen ist wie der Endbenutzer.

### - Auslastung des Surrogate Servers

Die Auslastung der einzelnen Surrogate Server wird dem Request Routing meist durch regelmäßige Statusberichte mitgeteilt.

Dadurch kann vermieden werden, dass ein Endbenutzer an einen Surrogate Server weitergeleitet wird, der bereits mit voller Kapazität arbeitet.

Wesentlich schwieriger zu bestimmen sind Daten über die Auslastung des Netzwerkes. Diese sind aber für ein effektives Request Routing unerlässlich. Daher gibt es auch hier viele verschiedene Lösungsansätze, die sich aber im Wesentlichen auf zwei Grundprinzipien zurückführen lassen.

### - Passives Netzwerkmonitoring

Beim passiven Netzwerkmonitoring sammeln die Surrogate Server Daten über ihre Verbindungen zu den Endbenutzern und geben diese an das Request Routing System weiter. Dort wird daraus eine Projektion des Netzwerkverhaltens erstellt, die dabei helfen soll, passende Surrogate Server für zukünftige Content Anfragen zu finden.

Diese Methode der Informationsgewinnung ist aber problematisch, da sich das Netzwerkverhalten des Internet heute so schnell und unvorhersehbar ändern kann, dass es bei der Zuweisung von Surrogate Servern, basierend auf statistischen Daten wie es hier der Fall ist, leicht zu Fehlentscheidungen kommen kann.

### - Aktives Netzwerk Probing

Hierbei testet das Request Routing System das Verhalten des Netzwerkes mit Hilfe von aktiven Anfragen wie beispielsweise ICMP ECHO. Diese Methode ist hervorragend geeignet, um ein aktuelles Bild des Netzwerkes zu bekommen und einen optimalen Surrogate Server zu bestimmen. Allerdings sind die Verwendungsmöglichkeiten für diese Methode der Informationsgewinnung stark eingeschränkt.

Zum einen verursacht sie zusätzlichen Traffic, der natürlich möglichst gering gehalten werden muss und zum anderen werden solche aktiven Anfragen von vielen Firewalls und Routern mit NAT Funktionalität blockiert.

Und nicht zuletzt können derartige Anfragen auf fremden Intrusion Detection Systemen zu Alarmmeldungen führen, worüber die jeweiligen Systemadministratoren selten erfreut sein dürften.

Eine Übersicht über die netzwerkabhängigen Kriterien zur Festlegung eines optimalen Surrogate Servers sowie die Möglichkeiten zu deren Bestimmung soll in nachfolgender Tabelle [1] gegeben werden.

<b>Kriterium</b>	<b>Ziele</b>	<b>Bestimmung durch</b>
<b>Latenz</b>	Verbindung mit der niedrigsten Latenz finden	Probing und Monitoring
<b>Paketverluste</b>	Verbindung mit geringstmöglicher Fehlerquote	Probing und Monitoring
<b>Nähe im Netzwerk</b>	kürzester Übertragungsweg	Nur mit active Probing möglich bzw. sinnvoll
<b>Durchschn. Bandbreite</b>	besten Übertragungsweg für Streaming Media finden	
<b>Startverzögerung</b>		
<b>Frame Rate</b>		

## 7 Request Routing Verfahren

Effiziente Verfahren zur Weiterleitung der Content Requests sind das Herzstück eines jeden Content Distribution Netzwerkes. Dementsprechend viele Vorschläge und Möglichkeiten zur Umsetzung solcher Weiterleitungsrouinen gibt es.

Das Ziel jedes Request Routing Systems ist es, nach einer eingegangenen Content Anforderung durch einen Endbenutzer in möglichst kurzer Zeit einen optimalen Surrogate Server zu finden und den Benutzer auf diesen umzuleiten.

Die gebräuchlichsten Routingverfahren heutiger CDN sind das DNS Routing, das Routing auf der Transportschicht (OSI Layer 4) und das Routing auf Anwendungsebene (OSI Layer 7).

### 7.1 DNS basiertes Request Routing

Heute existierende kommerzielle CDN stützen sich meist auf das DNS basierte Request Routing. Hierbei werden einem Domain Namen einfach mehrere IP Adressen zugeordnet. Kommt nun eine DNS Abfrage eines Clients herein, sucht ein spezieller DNS Server anhand der, in Abschnitt 6 aufgezeigten Analyseverfahren einen passenden Surrogate Server und ordnet dessen IP dem Abgefragten Domain Namen zu. Dabei wird mittels der Time to Live (TTL) Direktive festgelegt, dass diese Verbindung nur temporär gilt um bei einem zukünftigen Zugriff des Endbenutzers den Surrogate Server wechseln zu können.

DNS basiertes Request Routing ist die einfachste Methode, hat aber auch einige Nachteile:

- Mit DNS Rerouting kann nur eine Auflösung aus DNS Ebene erreicht werden. Das bedeutet, dass ein Client alle Inhalte von einem einzigen Surrogate Server bekommt. Für große CDN und insbesondere bei dynamischen Web Applikationen kann es aber von Vorteil sein, auch einzelne Elemente des Contents von verschiedenen Surrogate Servern zu holen.

- Das DNS Rerouting ist auf die Mitarbeit der Clientseitigen DNS Server angewiesen. Diese unterstützen aber beispielsweise häufig keine sehr kleinen TTL Werte, um den DNS Traffic zu reduzieren, was dann schnell zu einem Ungleichgewicht in der Clientverteilung unter den Surrogate Servern führen kann.
- Beim DNS Rerouting wird zur Standortbestimmung des Endbenutzers nur dessen DNS Server verwendet, was zu einer ungenauen oder sogar fehlerhaften Standortbestimmung führen kann.
- Alle Benutzer eines Clientseitigen DNS Servers werden innerhalb eines TTL Intervalls auf denselben Surrogate Server weitergeleitet, was bei den DNS Servern großer ISP zu einer Überlastung des jeweiligen Surrogate Servers führen kann.

## 7.2 Transport Layer Request Routing

Beim Request Routing auf der Transportschicht kann eine wesentlich höhere Auflösung erreicht werden und es stehen der Request Routing Engine mit den Layer 4 Header Daten wie beispielsweise IP-Adresse oder TCP Port wesentlich genauere Informationen über den Endbenutzer zur Verfügung. Daher kann mit dieser Routing Technik meist ein passenderer Surrogate Server gefunden werden, als es mit dem DNS Routing möglich ist. Im Gegensatz zum DNS Rerouting verläuft beim Layer 4 Rerouting der Datenstrom vom Client zum CDN (Upstream) immer über das Request Routing System und nur der Datenstrom vom Surrogate Server zum Client (Downstream) nimmt den direkten Weg. Da der Downstream bei den meisten Anwendungen jedoch ohnehin um ein vielfaches höher ist als der Upstream, sind hierdurch nur selten Geschwindigkeitseinbußen zu verzeichnen. Dafür bietet sich aber durch den permanenten Kontakt der Clients zum Request Routing System die Möglichkeit einer aktiven Überwachung der Verbindung und eines Rerouting auf einen anderen Surrogate Server im Falle von Verbindungsproblemen. Da beim Layer 4 Request Routing allerdings ein gewisser Zusatztraffic entsteht, eignet sich dieses Routingverfahren jedoch primär für die Übertragung größerer Content Dateien mit langlebigen Sessions wie beispielsweise FTP Transfers oder Streaming Media Sessions.

## 7.3 Application Layer Request Routing

Mit Hilfe des Application Layer Request Routing ist ein Hochauflösendes Request Routing auf der Basis der einzelnen Objekte, aus denen Multimedia Content besteht möglich. Die meisten Application Layer Protokolle wie beispielsweise HTTP, RTSP oder SSL übertragen in ihren Header Daten genügend Informationen, um effektiv passende Surrogate Server zu einem eingehenden Content Request zu suchen. Derzeit gibt es beim Application Layer Request Routing zwei verschiedene Mechanismen, die häufig eingesetzt werden:

### - Header Redirection

Dies dürfte wohl die einfachste Form des Request Routing überhaupt sein, da hierbei Mechanismen genutzt werden, die in den meisten Application Layer Protokollen bereits vorhanden sind. Im Falle von HTTP verwendet man hier beispielsweise die 303 Header Anweisung, um den Client auf einen Surrogate Server umzuleiten.

### **- URL Rewriting**

Bei diesem Request Routing Verfahren wird durch die Request Routing Engine die URL des angeforderten Contents so verändert, dass dieser von einem passenden Surrogate Server geholt wird.

Dieses Verfahren eignet sich besonders gut für dynamische Internetseiten, die ja aus vielen Objekten zusammengesetzt sind, welche alle einzeln über URLs angesprochen und im HTML Text miteinander verbunden werden. Dadurch bietet sich hier die Möglichkeit, jedes Objekt von einem anderen Surrogate Server zu holen, sollte die Netzwerkauslastung dies erforderlich machen.

## **8 Peer to Peer Netzwerke**

Nicht unerwähnt bleiben sollen Peer to Peer Netzwerke. Diese werden oft mit zu den Content Delivery Networks gerechnet, auch wenn ihre Struktur stark von der hier besprochenen abweicht [4].

Peer to Peer Netzwerke dienen vornehmlich zum Verteilen von statischen Dateien wie etwa Video- oder Audiodateien (ohne Streamingfähigkeit) und wurden dementsprechend auch in erster Linie mit Blick auf die Erfordernisse solchen Contents entwickelt.

Es gibt mittlerweile mehrere Umsetzungen solcher Netzwerke, die in ihrer genauen Struktur mehr oder weniger voneinander abweichen, die aber alle auf einem Grundschema aufbauen.

In einem Peer to Peer Netzwerk wird jeder Endbenutzer (Peer) als Netzwerkknoten eingebunden und fungiert gleichermaßen als Client und Server.

Ein Peer kann nun eine Suchanfrage absetzen, die entweder an eine zentrale Datenbank oder im Broadcast Verfahren an andere Peers weitergegeben wird.

Wurde eine Übereinstimmung mit der Suchanfrage gefunden, kann der Benutzer den Auftrag geben, die entsprechende Datei herunterzuladen. Der Benutzer wird nun direkt mit dem Benutzer verbunden, der die entsprechende Datei anbietet und lädt diese von dort herunter.

Dabei wird in den meisten Peer to Peer Netzen eine Datei noch zusätzlich in viele kleine Abschnitte zerlegt, die dann von jeweils verschiedenen Anbietern heruntergeladen werden können.

Eine Verbesserung der Netzwerkperformance im Vergleich zum herkömmlichen Internet wird dabei in erster Linie durch die Tatsache erreicht, dass eine gewünschte Datei in der Regel bei vielen verschiedenen Peers im Netzwerk verfügbar ist und so immer eine optimale Verbindung herausgesucht werden kann.

Allerdings geschieht die Verteilung der Dateien im Netz zufällig und so kann es insbesondere bei selten gewünschten (und somit selten vorhandenen Dateien) dazu kommen, dass man eine ähnliche oder sogar schlechtere Verbindung erhält als man sie bei herkömmlichem FTP Dateitransfer hätte.

Peer to Peer Netzwerke sind derzeit noch stark in der Entwicklung und auf einigen Gebieten viel versprechend, können aber aufgrund ihrer, dem klassischen Internet ähnelnden dezentralen Struktur bei der Mehrheit des heute Auftretenden Contents insbesondere im kommerziellen Bereich mit den, in dieser Arbeit besprochenen CDN nicht konkurrieren.

## 9 Weiterführende Content Verarbeitung

Mit der Einführung eines CDN steht eine Infrastruktur zur Verfügung, die auch noch weit über die eigentliche Aufgabe der Content Verteilung hinaus genutzt werden kann. Besonders im Gespräch sind dabei heute die Content Services.

Schon heute reicht es oft nicht mehr aus, wenn ein Content Provider seine Informationen in nur einer Form wie beispielsweise einer HTML Internetseite zur Verfügung stellt. Die Benutzer wollen auf diesen Content zunehmend nicht mehr nur mit ihren Webbrowsern zugreifen, sondern diesen beispielsweise unterwegs mit einem PDA oder Handy abrufen, oder ihn mittels Web-Service geliefert bekommen, um ihn weiterzuverarbeiten. Die Aufbereitung desselben Contents für die verschiedenen Bedürfnisse kostet dabei teilweise enorme Rechnerkapazität.

In einem CDN hat man jedoch schon ein Netzwerk aus verteilten Surrogate Servern. Nutzt man diese neben dem reinen Ausliefern des Content auch für dessen Bearbeitung, so verteilt sich die anfallende Rechnerlast und ist dadurch vergleichsweise kostengünstig beherrschbar. Zudem stehen beispielsweise mit dem geographischen Standort des Endbenutzers und anderen Informationen, die ohnehin für das Request Routing benötigt werden, Daten zur Verfügung, die eine genaue Anpassung des Contents an die Bedürfnisse des Kunden ermöglichen. Beispiele für mögliche Content Services wären:

- **Anpassung des Content an technische Gegebenheiten:**
  - Anpassung des Content an verschiedenste Endgeräte.
  - Anpassung an die verfügbare Bandbreite des Endbenutzers (z.B. Streaming Media)
  
- **Anpassung des Content an den Endbenutzer:**
  - Automatische Übersetzung in die lokale Sprache
  - Einfügen regional relevanter Zusatzinformationen für den Endbenutzer
  - Personalisierung des Content anhand der Daten des Accounting Systems
  - Einfügen von, auf den Endbenutzer abgestimmter Werbung
  
- **Content Filter Dienste**
  - Virenüberprüfung des Content vor der Auslieferung
  - Zugriffskontrolle durch Contentanalyse (z.B. Alterskontrolle)
  
- **Filtern der Content Requests**
  - Zugriffskontrolle durch Anfrageanalyse
  - Anlegen von Benutzerprofilen
  - Anonymisierung von Zugriffen

Diese Dienste und noch viele weitere sind denkbar und können in einem Content Delivery Network global für alle Content Anbieter zur Verfügung gestellt und gleichzeitig lokal je nach Surrogate Server auf die Bedürfnisse der Endbenutzer abgestimmt werden

## 10 Zusammenfassung

Ein CDN ermöglicht die Ortsnahe Verteilung von Inhalten verschiedenster Natur an die Endbenutzer und kann somit die zunehmenden Probleme, die durch die Struktur des Internet in den vergangenen Jahren entstanden sind kompensieren.

Um diese Verteilung des Content effizient umsetzen zu können muss ein CDN aus vielen verschiedenen Modulen bestehen, die auf verschiedenen Ebenen der Netzwerkstruktur miteinander interagieren. Daher sind CDN sehr komplexe Systeme, deren Entwicklung noch lange nicht abgeschlossen ist.

In dieser Seminararbeit wurde ein Überblick über die verschiedenen Mechanismen und Elemente eines CDN gegeben und dessen Einsatzmöglichkeiten erörtert.

Schon heute werden CDN vielfach von Kommerziellen Anbietern eingesetzt, um Engpässe bei Rechner- sowie Leitungskapazitäten zu vermeiden. Daneben erschließt sich mit CDN aber noch ein weites Feld zusätzlicher Einsatzmöglichkeiten für die, sich stark verändernde Kommunikationslandschaft der Zukunft.

## Literaturverweise

1. N. Bartolini, E. Casalicchio, and S. Tucci - Universit`a di Roma  
*A Walk through Content Delivery Networks*
2. Markus Hofmann - Bell Labs Research / Lucent Technologies  
*Content Networking – Architecture, Protocols and Practice*
3. Balachander Krishnamurthy - AT&T Labs - Research  
*On the Use and Performance of Content Distribution Networks*
4. S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, H. M. Levy  
Department of Computer Science & Engineering - University of Washington  
*An Analysis of Internet Content Delivery Systems*
5. Gang Peng - Department of Computer Science, State University of New York  
*CDN: Content Distribution Network*
6. Sebastian Adam  
*Grundlagen webbasierter Informationssysteme*
7. M. Karlsson, C. Karamanolis, and M. Mahalingam  
*A unified framework for evaluating replica placement algorithms.*
8. N. Bartolini, F. Lo Presti, and C. Petrioli.  
*Optimal dynamic replica placement in content delivery networks.*