

Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Institut für Praktische Informatik

Hauptseminar Telematik
Sommersemester 2004

Betrachtungen zur Skalierbarkeit kooperativer Streaming-Systeme

Was ist eigentlich Skalierbarkeit und wo und warum spielt sie eine so große Rolle?

Betreuer: Thorsten Strufe

Bearbeiter: Matthias Eck

Matrikel-Nr.: 22979

Termin: 7. Juni 2004

Inhaltsverzeichnis

1	Einleitung.....	3
2	Kooperative Streamingsysteme	3
2.1	Live Streaming.....	3
2.2	OnDemand-Streaming.....	4
2.3	Streaming Media	4
2.4	Streaming System	5
3	Skalierbarkeit.....	6
3.1	Streaming, Formate.....	6
3.2	Streaming Systeme	7
3.2.1	Client – Server.....	8
3.2.2	Peer – to – Peer	9
4	Ausblick auf mögliche Lösungen.....	13
5	Zusammenfassung.....	14
6	Literaturverzeichnis	16
7	Abkürzungsverzeichnis	18
8	Abbildungsverzeichnis.....	19

1 Einleitung

Streaming-Systeme erfreuen sich wachsender Beachtung im Internet. Traditionelle Client- / Server-Systeme sind für die Verbreitung von Multimedia-Daten ungeeignet, da diese die hohen Anforderungen an Server- und Netzwerkressourcen für eine große Anzahl simultaner Nutzer nicht erfüllen. Eine Möglichkeit die Skalierung von Streaming-Systemen zu verbessern ist die Nutzung von Peer-to-Peer (P2P) Netzwerken. Der einfache und bequeme Zugriff auf die freigegebenen Daten von mehreren Tausend angeschlossenen Clients ist verantwortlich für die Erfolgsgeschichte von z.B. Napster & Co. Jedoch belegen einige Studien, dass gerade die reinen Peer-to-Peer Netzwerke schlecht skalieren und eine hohe Belastung für die darunterliegenden Transportnetzwerke darstellen [Ritter], [Aberer, Hauswirth]. Die Kritiker prognostizieren ein exponentielles Wachstum des Signalverkehrs und damit eine unwirtschaftliche Ausnutzung der Datennetze. Es existieren aber auch Beiträge, die versuchen das Gegenteil zu beweisen [Schollmeier]. Dies zeigt, dass dieses Themengebiet zurzeit stark diskutiert wird und im zentralen Interesse steht [Vasconcelos et al], [Ley].

2 Kooperative Streamingsysteme

Sebastian Rieger von der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) versteht unter Streaming-Media den Einsatz von Verfahren, die Audio und Video im Internet zur Verfügung stellen. Neben Audio- und Videodaten können theoretisch aber auch andere Informationsarten übertragen werden [Rieger]. Die Media Streaming Technologien werden in zwei Hauptarten gegliedert, dies wären live Streaming Systeme und OnDemand Streaming Systeme [Götz].

2.1 Live Streaming

Beim Live-Streaming werden Audio- und Videodaten in Echtzeit in das Datennetz eingespeist. Eine vorgelagerte Aufnahmetechnik erzeugt Audio- und Videodaten. Diese Daten werden mittels Encoder in das geforderte Format encodiert. Entscheidend für die Auswahl des Kompressionsverfahrens ist die Zielgruppe, für

welche die Daten bereitgestellt werden sollen. Ein MPEG-2 Datenstrom hat eine sehr gute Qualität, kann aber nicht im ISDN oder xDSL Bereich eingesetzt werden, da die erforderlichen Datenraten zwischen 3-12 Mbit/s liegen. Die Nutzer mit diesen Anschlüssen können mit einem Real-Sure Stream versorgt werden, der sich entsprechend des Mediums selbst skaliert. Unter Umständen müssen verschiedene Streams erzeugt werden, um den unterschiedlichen Anwendern entsprechend ihres Kommunikationsmediums die optimale Qualität bereitzustellen. Problematisch ist bei Live-Streams über einen Streaming-Server, dass Delayzeiten von bis zu 30s, bedingt durch die Pufferung des Klienten entstehen. Live Streaming-Systeme sind unidirektionale Übertragungen ohne Rückkanal, d.h. der Empfänger kann nicht in das Geschehen eingreifen. In Abbildung 1 wird ein Streaming-System dargestellt. Um die Daten im Netzwerk für die Nutzer bereitzustellen, wird zwischen zwei Arten der Verteilung unterschieden, Multicast und Unicast. Die Zweckmäßigkeit dieser Verfahren für Streaming Systeme wird in Kapitel 3 näher erläutert [Götz].

2.2 OnDemand-Streaming

Beim OnDemand-Streaming liegen die Audio- und Videodaten auf einem Server. Durch entsprechende Programme kann sich der Nutzer die gewünschten Daten an sein Endsystem streamen lassen. Dabei ist es für ihn uninteressant, auf welchem Server die Daten liegen, das Programm stellt die Verbindung zu dem entsprechenden z.B. Videosever her.

Die einzelnen Audio- und Video-Daten können, müssen aber nicht indiziert werden. Der Vorteil einer Indexierung, ist die bessere gezieltere Recherche, die Gewinnung von zusätzlichen Informationen aus den Daten und die Verbesserung der Gesamtübersicht über die Datenbestände.

OnDemand Streaming ermöglicht die Vorschau bei vorhandener Indexierung, weiterhin ist Vor- und Rückspulen möglich [Götz].

2.3 Streaming Media

Rieger bezeichnet Streaming Media als eine gleichzeitige Übertragung von verschiedenen Medien (Audio, Video, Daten), die einen kontinuierlichen Medienstrom in Echtzeit bilden, wobei die einzelnen übertragenen Medien dem Anwender sofort präsentiert werden. Dabei fällt für ihn die Übertragung von

Multimedia-Inhalten genau dann unter den Begriff Streaming Media, wenn mindestens ein kontinuierliches Medium (wie Audio oder Video) darin enthalten ist.

Streaming Media zeichnet sich dabei durch folgende Eigenschaften aus:

- der übertragene Medienstrom ist kontinuierlich,
- der Medienstrom wird unmittelbar (während des Empfangs) wiedergegeben,
- der Medienstrom wird zunächst nicht komplett heruntergeladen,
- der Medienstrom ist kontrollierbar, z.B. um vor- bzw. zurückzuspulen. [Rieger, S. 8]

2.4 Streaming System

Die in Abbildung 1 gezeigte logische Struktur stellt ein Streaming System dar. Das System beschreibt die an der Übertragung des Streams beteiligten Funktionskomponenten vom Sender zum Empfänger.

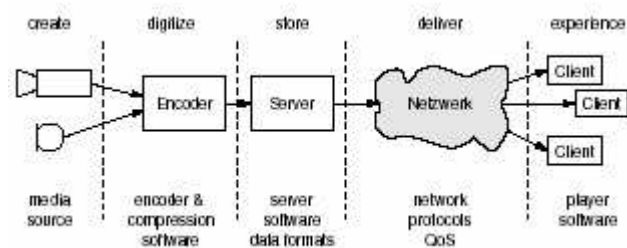


Abbildung 1: Streaming System [Hendrich, S. 2]

Die Eingabedaten (Audio, Video, etc.) werden zunächst vom Encoder digitalisiert. Danach werden sie komprimiert und in ein Audio- bzw. Video-Format eingebettet. Dabei stellt dieses Format einen standardisierten Container dar, der es ermöglicht, dass auch Media-Player sowie –Encoder unterschiedlicher Hersteller den Stream verarbeiten können.

Vom Sender wird das Format schließlich in IP-Pakete verpackt und unter Verwendung eines Streaming-Protokolls über das Netzwerk an den Empfänger (Media-Player) gesendet, wodurch ein Medienstrom in Form von mehreren IP-Paketen in Richtung des Empfängers entsteht. Vom Empfänger werden die empfangenen Pakete zunächst in einen Empfangspuffer eingereiht, wodurch er aus

den einzelnen Paketen das vom Sender verwendete Format zurück gewinnt [Rieger, S. 17 ff.].

3 Skalierbarkeit

Skalierbarkeit beschreibt die Eigenschaft einer Hard- oder Software-Lösung, sich flexibel erweitern zu lassen. Beispielsweise erlaubt eine skalierbare Software, die Anzahl der Benutzer zu erhöhen, ohne dass dadurch die Performance der Anwendung leidet [Aulie]. Die Leistung eines nicht skalierbaren Systems nimmt bei Wachstum bestimmter Parameterwerte stark ab, u.U. bis zur Funktionsunfähigkeit [Bless, K. 2].

Dr. Bless von der Universität Karlsruhe (TH) nennt folgende Aspekte der Skalierbarkeit [Bless, K. 2]:

- Skalierbarkeit bezieht sich immer auf bestimmte Systemparameter.
- Bei der Betrachtung sollte man insbesondere die Systemleistung beim Anwachsen der Systemparameter um mehrere Größenordnungen betrachten.
- Selbst lineares Wachstum bestimmter Parameter kann u.U. bereits zu einer nicht mehr handhabbaren Gesamtkomplexität führen.
- Beispiele für Parameter:
 - Anzahl der Endsysteme
 - Anzahl der Einträge in Routingtabellen
 - Anzahl der autonomen Systeme
 - Bandbreite

Im diesem Kapitel wird zum einen die Skalierbarkeit der Streaming Formate und zum anderen die der Streaming Systeme betrachtet.

3.1 Streaming, Formate

Um das Abspielen von z.B. Videostreams auch auf langsamer Hardware oder über langsamere Verbindungen zu ermöglichen, bestehen bei einigen Videoformaten, z.B. MPEG-4, wie schon bei seinem Vorgänger MPEG-2, die Möglichkeit der geschichteten Bildcodierung. Dabei kann der Videostream zeitlich, räumlich, qualitativ und vom Komplexitätsgrad skaliert werden.

MPEG-4 bietet folgende Skalierungstechniken bzgl. Textur-, Bild- oder Videostream:

- Komplexitätsskalierbarkeit (encoderseitig)
Erzeugen von Bitstreams verschiedener Größe
- Komplexitätsskalierbarkeit (decoderseitig)
Abspielen von festen, gegebenen Bitstreams in verschiedenen Komplexitätsstufen. Dies ermöglicht es, auf schwachen Rechnern nur einen Teil des Bitstreams zu decodieren.
- Räumliche Skalierbarkeit
Ermöglicht dem Decoder, einzelne Mediaobjekte in reduzierter Auflösung abzuspielen. MPEG-4 bietet für unbewegte Bilder elf Auflösungsstufen, für Videos drei.
- Zeitliche Skalierbarkeit
Ermöglicht dem Decoder, nur einen Teil der Frames eines Videos abzuspielen. Bei MPEG-4 stehen dabei drei verschiedene Stufen zur Verfügung.
- Qualitätsskalierbarkeit
Der Bitstream wird in verschiedene Ebenen aufgeteilt, so dass bei beliebiger Kombination der Layer trotzdem ein sinnvolles Signal übertragen wird. Die Aufteilung erfolgt entweder bei der Übertragung oder beim Decodieren.

Die Kombination einer räumlichen mit einer qualitativen Skalierung ist bis jetzt noch nicht möglich, ansonsten können sowohl die räumliche als auch die zeitliche Skalierung mit der Kodierung beliebiger Formen kombiniert werden [Pach, S. 36 f.].

3.2 Streaming Systeme

Bei der Untersuchung der Skalierbarkeit von Streaming Systemen sind die Nachrichtenkomplexität, Speicherkomplexität und Zeitkomplexität von zentralem Interesse. Audio- und Videodaten stellen relativ hohe Anforderungen an Speicher- und Übertragungskapazitäten. Aufgrund der Verteilung bei Peer – to – Peer Architekturen sind die Anforderungen an die Speicherkapazität bei Client-Server-Architekturen (OnDemand Streaming) ungleich schwerwiegender. Ein weiterer wichtiger Aspekt ist die Zeitkomplexität, da bei Übertragung von Audio- und Videostreams geringe Verzögerungszeiten notwendig sind. In diesem Kapitel wird sich bei der Betrachtung der Skalierbarkeit von Streaming Systemen auf die

Untersuchung des Signalverkehrs (Nachrichtenkomplexität) beschränkt. Die eigentliche Streamübertragung bleibt außen vor.

3.2.1 Client – Server

Zentrale Systeme (Client-Server-Architektur) bilden die bekannteste Topologie, Funktionalität und gespeicherte Daten sind zentral in einem Server. Die Clients können sich direkt zu diesem Server verbinden, um die benötigten Informationen anzufordern.

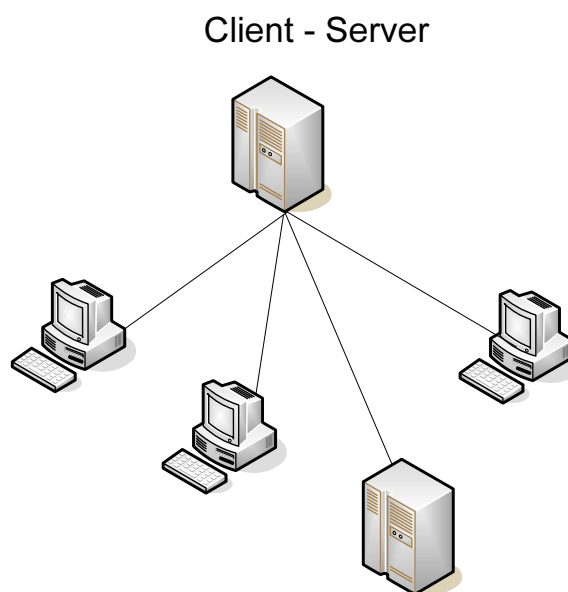


Abbildung 2: Client - Server Architektur

Jedoch kann ein einzelner, zentraler Server nicht beliebig viele Clients bedienen. Für eine steigende Anzahl von Clients, kann man einen „Cluster“ an Servern verwenden. Solche Cluster bilden fast immer einen Ring, d.h. jeder Server bildet dabei ein Knoten. Benachbarte Knoten kommunizieren miteinander um die gespeicherten Daten oder deren Status zu synchronisieren. Vorteile bei solchen Systemen sind deren Ausfallsicherheit und die Möglichkeit der Lastverteilungen, um eine größere Skalierbarkeit im Vergleich zu reinen Client – Server Systemen zu erreichen [Freismuth, S. 13].

3.2.2 Peer – to – Peer

Bei dezentralen Systemen existieren gleichberechtigte Partner, sogenannte „Peers“. Bei „reinen“ Peer – to – Peer Netzwerken bietet jeder Peer Dienste an und nimmt welche in Anspruch, d.h. er ist Server und Client in einem. Die Aufgabe der Peers ist u.a. die Speicherung und Bereitstellung von großen Datenmengen, verteilt auf viele Teilnehmer. Häufig wechselnde Peers, wenig dauerhaft verfügbare Datenquellen und eine hohe Heterogenität (Netzanbindung, Latenzzeiten, Datenbestand) sprechen für eine hohe Dynamik in P2P Netzwerken.

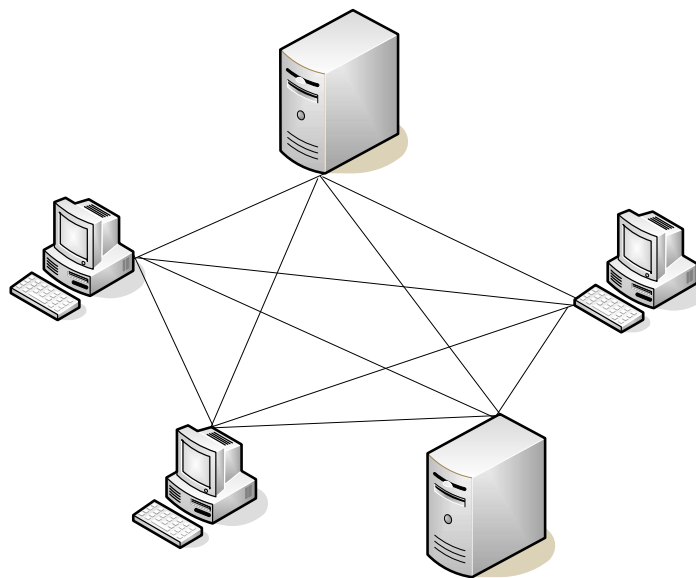


Abbildung 3: Peer - to – Peer Architektur

Peer – to

Durch die Interaktion von Endsystemen ergeben sich minimale Anforderungen an die Infrastruktur (Netzwerk, Dienste), da keine Nutzung zentraler Dienste erforderlich ist. Dadurch werden keine aufwendigen, zentralen Server mit Skalierbarkeitsproblemen benötigt. Es erfolgt sozusagen eine Umgehung des Flaschenhalses einer zentralen Realisierung [Bless, K. 10].

Peer – to – Peer Netzwerke und vor allem der bequeme Datenaustausch über solche Netze sind seit einigen Jahren immer populärer geworden. Bei aktuellen Untersuchungen steht besonders die Skalierbarkeit der verwendeten Protokolle im Interesse der Forschung. Beispielhaft für „reine“ P2P Netzwerke beschäftigen sich viele Veröffentlichungen mit dem „normalen“ Gnutella (dezentrales Filesharing). Diese Untersuchungen sind auch für die Betrachtung der Skalierbarkeit von

kooperativen Streaming Systemen nachhaltig, da sich die anfallenden technischen Probleme gleichen. Die meisten Arbeiten beschränken sich auf die Untersuchung des Signalverkehrs, die eigentliche Datenübertragung bleibt außen vor [Ley, S. 2 ff.]. Bei Gnutella sind die Peers über TCP-Verbindungen verbunden, die Peers heißen hier Servents (Kombination aus Server und Client). Anfragen werden über das Gnutella-Netz geflutet (TCP Broadcast von Ping- und Query-Nachrichten). Abbildung 4 zeigt ein Beispiel, wie ein Ping-Deskriptor ausgehend vom Servent A geroutet wird und ein zugehöriger Pong-Deskriptor als Antwort darauf von B versandt wird. Query- und QueryHit-Deskriptoren verhalten sich genauso wie die Ping- und Pong-Deskriptoren.

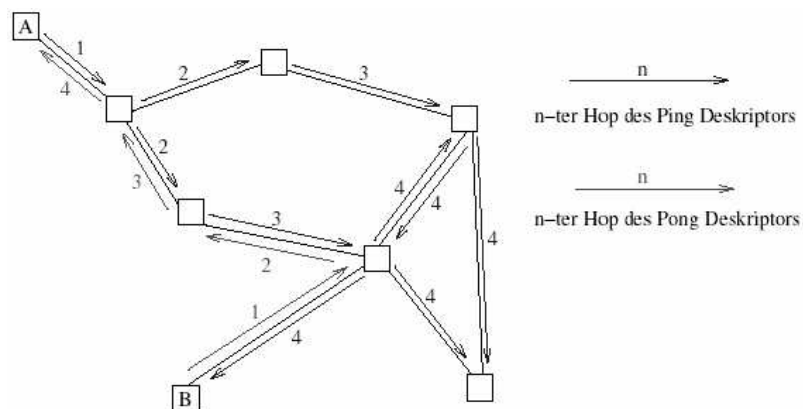


Abbildung 4: Routing im Gnutella Netzwerk [Ley, S. 8]

Das wesentliche Problem der Breitensuche (Fluten, wie bei Gnutella) ist, dass sehr viel redundanter Verkehr produziert wird, es wird viel Bandbreite und Rechenzeit verbraucht, allerdings werden auch in sehr kurzer Zeit alle Peers erreicht. Unter Umständen erzeugt aber die Breitensuche eine zu hohe Netzbelastung, so dass das Netz zum Erliegen kommt. Peers mit Schmalbandverbindungen sind oft komplett ausgelastet mit der Beantwortung von Suchanfragen. Aufgrund dieser Faktoren wird in der Literatur davon gesprochen, dass die Breitensuche nicht skaliert [Bless, K. 10]. Um Aufschlüsse über die Topologie von P2P Netzwerken zu erhalten wurden einige Verfahren entwickelt und in der Literatur diskutiert. Abbildung 5 zeigt eine Momentaufnahme des Gnutella Netzwerks aus dem Jahr 2000 [Janvanovic et al].

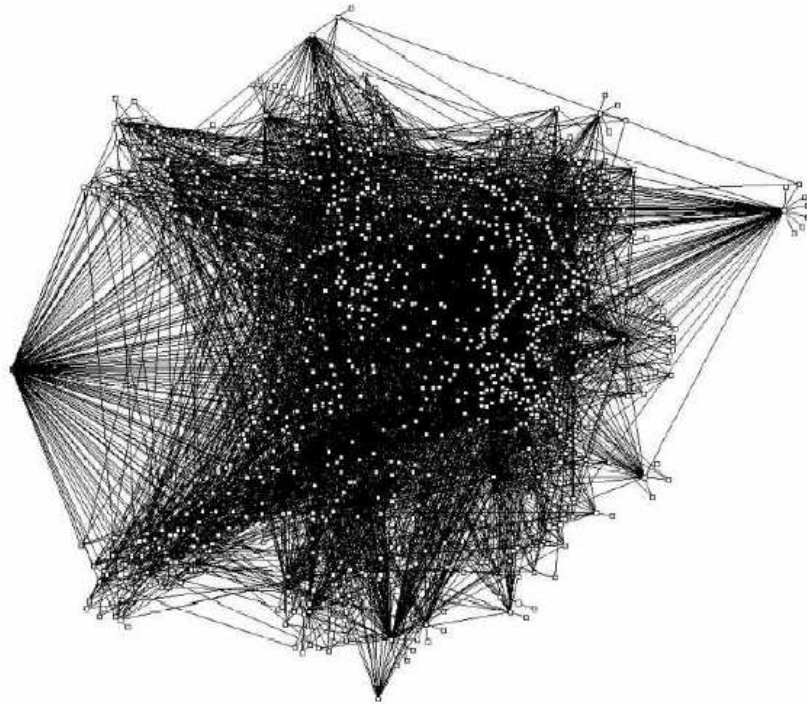


Abbildung 5: Momentaufnahme eines Teils des Gnutella Netzwerks [Ley, S. 18]

Das Gnutella-Netz der 1. Generation ist im August 2000 zusammengebrochen. Schmalbandige Peers konnten die Last nicht bewältigen. Es erfolgte eine Fragmentierung des Netzes. Eine Anpassung der Verbindungen eines Peers an deren Bandbreite, sowie der Abbruch von Verbindungen zu überlasteten Peer hilft hier nur bedingt weiter. Die Skalierung ist dadurch immer noch nicht zufriedenstellend. In der 3. Generation wurden schließlich Hierarchien eingeführt. Für schmalbandig angebundene Peers übernehmen „Super-Peers“ die Netzlast. Die Architektur entspricht nun nicht mehr einem reinen P2P Netzwerk [Bless, K. 10].

Ein Hybrid P2P Protokoll zeichnet sich durch die Existenz einer zentralen Instanz aus, über die der Signalverkehr abgewickelt wird. Ein reines Peer – to – Peer Protokoll sieht eine solche Instanz nicht vor, sämtliche Protokollfunktionen müssen von den Netzwerkteilnehmer verteilt ausgeführt werden. Es liegt die Vermutung nahe, dass der Signalverkehr bei den reinen Protokollen größer ist als bei den Hybrid-Protokollen. Allerdings haben reine Protokolle auch ihre Vorteile. Zentrale Server müssen leistungsfähig sein und über gute Netzwerkverbindungen verfügen. Weiterhin bedeutet ein Ausfall meist einen sofortigen Zusammenbruch des gesamten Netzwerks. [Ley, S. 5] Schollmeier und Schollmeier prägen mit der Veröffentlichung aus dem Jahr 2002 einen anderen Ansatz. Es soll gezeigt werden, dass die Vorteile der reinen P2P Netzwerke nicht durch einen inakzeptablen Anstieg des

Signalverkehrs erkauf werden. Die Untersuchungen beziehen sich detailliert auf das Gnutella Protokoll und gehen von der Annahme aus, dass das Gnutella Netzwerk als unendlicher Baum aufgebaut ist und aus nur endlich vielen Servents besteht. Im Ergebnis kommen sie zu dem Schluss, dass die Kritik am Gnutella Protokoll unberechtigt ist. Eine exponentielle Verbreitung von Deskriptoren kann sich nur über Bäume mit großer Tiefe entwickeln und diese sind nach deren Untersuchungen sehr unwahrscheinlich im Gnutella Netzwerk zu finden [Schollmeier] [Ley, S. 6 ff.].

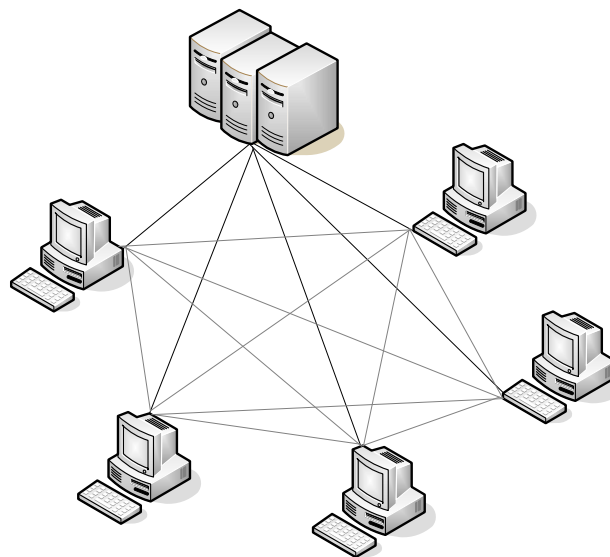


Abbildung 6: zentrales Verzeichnis

Abbildung 6 stellt ein hybrides P2P-Netz da. Dedizierte Server übernehmen Aufgaben wie Indizierung, wodurch ein schnelles Suchen ermöglicht wird.

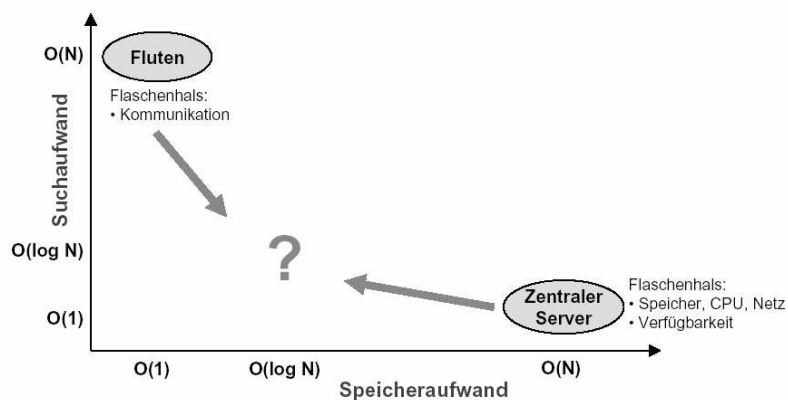


Abbildung 7: Suche nach Inhalten - Such- vs. Speicheraufwand [Bless, K. 10]

Dr. Bless von der Universität Karlsruhe (TH), sieht den Kompromiss der Suche zwischen Suchaufwand (reinen P2P Netzwerken) und Speicheraufwand (Client-Server Netzwerk) in verteilten Hash-Tabellen, dies ist in der Abbildung 7 dargestellt. Auf eine ausführliche Darstellung der verteilten Hash-Tabellen wird an dieser Stelle verzichtet [Bless, K. 10].

Bei der Konzeptionierung neuer Streaming Systeme kann man aus den Erfahrungen mit Napster, Gnutella und Co. lernen. Was die Speicher- und Nachrichtenkomplexität betrifft, sind die technischen Anforderungen vergleichbar. Bei Streaming Systemen spielt die Zeitkomplexität eine weitere grundlegende Rolle.

4 Ausblick auf mögliche Lösungen

Beim Streaming-Media werden identische, große Datenmengen in kurzen Zeiträumen an sehr viele Clients verschickt. Nutzt man dabei ausschließlich Unicast, um diese Clients zu erreichen, so wächst die benötigte Bandbreite am jeweiligen Server und den dazwischen geschalteten Routern proportional zur Anzahl der ihnen untergeordneten Teilnehmer. Dagegen bleibt die benötigte Übertragungsbandbreite beim Multicasting sowohl für den Server als auch für alle Router auf dem Weg zum Client konstant. [Rieger, S. 62]

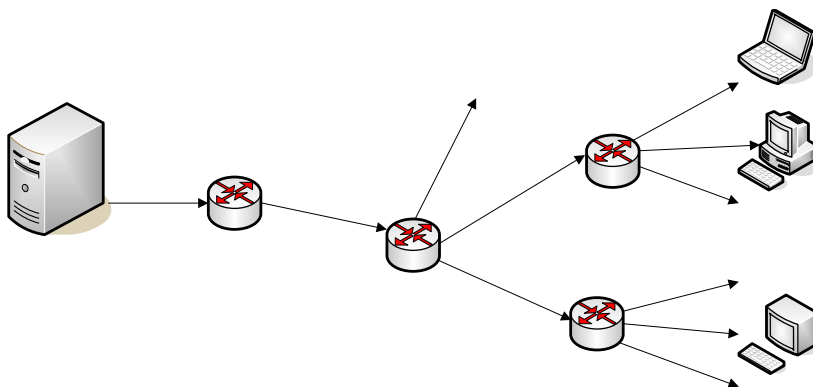


Abbildung 8: Stream Merging (RSVP)

Unterstützt wird diese Technik durch die Nutzung der Protokolle: RSVP und RTP / RTCP. Der Transfer von Multimedia-Daten findet mit Hilfe des Realtime Streaming Protocol (RTSP) statt. Es dient z.B. zur Anforderung einer Multimedia-Ressource von einem Server und der Aushandlung dessen Formates. Dabei verwendet RTSP zum Transport der Daten üblicherweise RTP über UDP. Das Realtime Transport Protocol

(RTP) stellt Funktionen zur Synchronisation und Sequenzierung von Datenströmen zur Verfügung, ist jedoch kein Transportprotokoll wie TCP oder UDP. Üblicherweise wird UDP verwendet, da Multimedia-Daten im allgemeinen einen gewissen Verlust an Paketen tolerieren können und die korrekte Ordnung der Pakete bereits durch RTP ermöglicht wird. Über das RTP Control Protocol (RTCP) erhält nun die Anwendung die Möglichkeit, die tatsächlich erbrachte QoS bei der Übertragung zu überwachen, und auf Schwankungen zu reagieren. Hierdurch könnte die Anwendung z.B. die Kodierung eines Audiostreams verändern. Es ist sinnvoll, den Netzwerkknoten, die an der Übertragung beteiligt sind, die Wünsche der Anwendung bzgl. der QoS der Übertragung mitzuteilen. Für diese Signalisierung der QoS- und damit Reservierungswünsche einer Anwendung kann z.B. das Resource Reservation Protocol (RSVP) verwendet werden. Das RSVP-Protokoll ermöglicht eine Übermittlung beliebiger QoS-Spezifikationen für Teilmengen eines Streams zwischen Sender und Empfängergruppe. RSVP stellt eine Methode zur robusten, skalierbaren und flexiblen Installation von heterogenen Reservierungen für Ressourcen bereit, die Multicast-Streams unterstützt. Dadurch skaliert die Übertragung auch für sehr viele Clients. [Furche].

5 Zusammenfassung

Durch immer neue Entwicklungen der Internet-Zugangstechnologien wird sich der bisherige Flaschenhals der Übertragung vom Netzwerkzugang hin zu den Kernnetzen und Datenservern verschieben. Die vermehrte Nutzung von breitbandigen Diensten, wie Multimedia Streaming werden die involvierten Server massiv belasten. Ein Media-on-Demand System bestehend aus Server, Netzwerk und Empfängern steht allgemein vor dem Problem, die vorhandenen Ressourcen möglichst effizient auf alle Empfängeranfragen aufzuteilen. Derzeit noch verwendete Unicast Systeme müssen wegen ihrer mangelnden Skalierbarkeit auf eine maximale Anfragerate dimensioniert werden, wenn für den Empfänger unliebsame Dienstaussfälle vermieden werden sollen [Fuhrmann]. Bei Live Streaming können IP-Multicast-Technologien eine kostengünstige Möglichkeit sein, diesem Engpass zu begegnen. Ermöglicht man dem Nutzer interaktive Funktionalitäten (Media on Demand), wie Pause, schneller Vor- und Rücklauf, so scheint man um einen dedizierten Datenstrom pro Nutzer (Unicast) nicht umhin zu kommen [Gebhard, S. 1].

Zum Abschluss möchte ich auf die Dissertation vom Michael Zink verweisen, der sich ausgiebig mit skalierbaren Video-on-Demand Systemen beschäftigt hat [Zink].

6 Literaturverzeichnis

Aberer, K.; Hauswirth M. : An Overview on Peer-to-Peer Information Systems.

<http://citeseer.nj.nec.com/aberer02overview.html>

Aulie, N.; Honegger, B. D. : Thin Client Systemen an Schulen / ETH Zürich.

<http://www.educeth.ch/informatik/berichte/thinclients/docs/thinclients.pdf>

Bless, R. : Peer-To-Peer-Netze / Universität Karlsruhe (TH).

http://www.tm.uka.de/lehre/ss04/vorlesungen/V_NGI_Unterlagen

Fuhrmann, T. : Experimentelle und analytische Untersuchung von Kommunikationsprotokollen / Universität Karlsruhe (TH).

<http://www.tm.uka.de/lehre/ws0304/vl/Protokollanalyse.html>

Furche, T. : Verhandlung, Reservierung und Überwachung von Dienstgütemerkmalen / TU München.

http://www.hegering.informatik.tu-muenchen.de/Events/Sarntal/Sarntal2000/timfu/#sec_RSVP

Gebhard, H. : Effizientes Echtzeit Media-on-Demand Verfahren in IP-Netzwerken / Universität Dortmund.

http://www-kt.e-technik.uni-dortmund.de/m_gh/folder/ISP_gebhard.pdf

Götz, Olaf : Media-Streaming, Technologien und Konzepte / Medienzentrum. Jena: Friedrich Schiller Universität Jena. Net.Objectdays 2002

Hendrich, N. : Digitale Audioverarbeitung / Universität Hamburg.

<http://tams-www.informatik.uni-hamburg.de/lehre/ws2000/vl-audioverarbeitung/audioverarbeitung-ws2000.pdf>

Jovanovic, M. A.; Annexstein, F. S.; Berman, K. A. : Scalability Issues in Large Peer-to-Peer Networks – A Case Study of Gnutella.

<http://www.eces.uc.edu/~mjovanov/Research/paper.html>

Ley, S. : Skalierbar oder nicht? Die Topologie des Gnutella Netzwerks / Lehrstuhl für Informatik IV. Aachen. Rheinisch-Westfälische Technische Hochschule Aachen
http://www-i4.informatik.rwth-aachen.de/content/teaching/seminars/sub/2003_2004_ws_docs/ley.pdf

Pach, M. : MPEG-4-Video Standard / Eberhard Karls Universität Tübingen.
<http://www.mpeg4video.de>

Rieger, S. : Streaming-Media und Multicasting in drahtlosen Netzwerken, Untersuchen von Realisierungs- und Anwendungsmöglichkeiten / Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, GWDG-Bericht Nr. 61
<http://www.gwdg.de/forschung/publikationen/gwdg-berichte/gwdg-bericht-61.pdf>

Ritter, J. : Why Gnutella Can't Scale. No, Really.
<http://www.darkridge.com/~jpr5/doc/nutella.html>

Schollmeier, R.; Schollmeier, G. : An Analysis of P2P Traffic Patterns, Proceedings of the Second International Conference on Peer-to-Peer Computing

Vasconcelos, Rocha, Santos, Ismael, Mata, Almeida, Meira, Almeida : Qualitative Analysis of Strategies for Streaming Media Distribution / Department of Computer Science. Belo Horizonte, Brazil. Federal University of Minas Gerais

Zink, M. : Scalable Internet Video-on-Demand Systems / TU Darmstadt.
http://elib.tu-darmstadt.de/diss/000378/diss_zink.pdf

7 Abkürzungsverzeichnis

IP	Internet Protocol
MPEG	Motion Picture Experts Group
P2P	Peer-to-Peer
QoS	Quality of Service
RTCP	RTP Control Protocol
RTP	Realtime Transport Protocol
RTSP	Realtime Streaming Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

8 Abbildungsverzeichnis

Abbildung 1: Streaming System [Hendrich, S. 2]	5
Abbildung 2: Client - Server Architektur	8
Abbildung 3: Peer - to – Peer Architektur	9
Abbildung 4: Routing im Gnutella Netzwerk [Ley, S. 8].....	10
Abbildung 5: Momentaufnahme eines Teils des Gnutella Netzwerks [Ley, S. 18]....	11
Abbildung 6: zentrales Verzeichnis	12
Abbildung 7: Suche nach Inhalten - Such- vs. Speicheraufwand [Bless, K. 10]	12
Abbildung 8: Stream Merging (RSVP).....	13