

# Peer-to-Peer Netze

Seminar: Content Delivery Networks

Prof. Dr. W. Effelsberg / Christian Liebig

Lehrstuhl für Praktische Informatik IV

Universität Mannheim

Wintersemester 2004/05

25. Januar 2005

**Thomas Butter**

butter@uni-mannheim.de

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Ziele von P2P Systemen</b>	<b>2</b>
<b>3</b>	<b>Aufbau von P2P Netzen</b>	<b>3</b>
3.1	Reine P2P Systeme . . . . .	3
3.2	Zentralisierte Systeme . . . . .	4
3.3	Hybride Systeme . . . . .	4
<b>4</b>	<b>Tragedy of the Commons</b>	<b>6</b>
4.1	Netznutzung . . . . .	6
4.2	Tausch-Bereitschaft . . . . .	7
4.2.1	Lokale Limitierung . . . . .	8
4.2.2	Peer Credits . . . . .	8
4.2.3	Tit for Tat . . . . .	8
4.2.4	Koordinierte Uploadraten . . . . .	8
<b>5</b>	<b>Verteilung der Inhalte</b>	<b>10</b>
5.1	Swarms . . . . .	10
5.2	Hash-Ring . . . . .	11
5.3	Koordiniert . . . . .	12
5.4	Streaming . . . . .	13
5.5	NAT Traversal . . . . .	13
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>15</b>

# 1 Einleitung

Peer-to-Peer (P2P) Systeme werden von Clay Shirky folgendermaßen definiert: "P2P is a class of applications that takes advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet." [1].

In dieser Arbeit werden P2P Netze nur als Mittel zur Content Delivery betrachtet. Im besonderen wird hier auf die unterschiedlichen Ziele von P2P Netzen im Vergleich zu anderen Content Delivery Networks (CDN), die Probleme der sozialen Netzwerke und deren technischen Lösungen, sowie die unterschiedlichen Arten der Verteilung der Inhalte eingegangen. In einem solchen P2P Netz können alle Nutzer gleichzeitig Server und Client sein. Jeder Knoten empfängt Inhalte und verteilt diese weiter.

Auf andere Arten von P2P Netzen wie Voice-over-IP (VoIP), Instant Messaging und auf P2P Systeme zum verteilten Rechnen wird hier nicht eingegangen. Auch die Suche nach Dateinamen oder Stichworten in P2P Netzen wird hier nicht betrachtet.

# 2 Ziele von P2P Systemen

In den meisten Content Delivery Networks wird versucht die Qualität, im Sinne von Verfügbarkeit, Darstellungsqualität und Geschwindigkeit, für den Benutzer zu erhöhen. Hierfür werden höhere Infrastruktur Kosten auf der Seite der Content-Anbieter, z.B. durch den Aufbau einer Proxy/Caching Infrastruktur, akzeptiert. Im Gegensatz hierzu versucht man durch P2P Netze die Last der Inhalte Anbieter zu minimieren und auf die Nutzer zu verteilen. Wichtige QoS Merkmale wie die Geschwindigkeit oder Reaktionszeiten werden oft nicht optimiert.

P2P Lösungen ermöglichen allerdings kleinen Anbietern große Datenmengen an eine große Zahl von Nutzern zu verteilen und erlauben so völlig neue Wege der Verbreitung von Open Source Software, Independent Movies [2], aber auch illegal verbreiteter Filme, Musik, Software und neue, günstige Wege des Vertriebs von Digital Rights Management (DRM) geschützten Inhalten [3].

Einige P2P Systeme sind speziell auf die Verteilung illegaler oder politisch unerwünschter Inhalte optimiert, indem weitere Ziele wie Anonymität oder das Auffinden

bestimmter Dateien in großen verteilten Archiven verfolgt werden. Einige andere Systeme versuchen die Verfügbarkeit der Inhalte auch beim Ausfall einer großen Zahl von Nodes zu erreichen um ein nicht zensierbares Netz zu schaffen (z.B. GNUNet [4] und FreeNet [5]). Andere Systeme orientieren sich mehr an den klassischen Zielen von CDNs und sind mehr auf die Zuverlässigkeit und Geschwindigkeit zur Verteilung einer großen Datei optimiert.

### 3 Aufbau von P2P Netzen

Heutige P2P Systeme lassen sich vom Aufbau in reine P2P Systeme, zentralisierte Systeme und Hybride Systeme einteilen. Diese Formen besitzen alle bestimmte Vor- und Nachteile bezüglich des Koordinationsaufwandes für den Erhalt eines stabilen Netzes.

#### 3.1 Reine P2P Systeme

Ein reines P2P Netz besteht nur aus gleichberechtigten Nodes, die zu jeder Zeit dem Netz beitreten können oder es verlassen können. Jeder Knoten übernimmt in diesem Fall gleichzeitig Server und Client Funktionen. Für diese Art der P2P Netze muss ein Bootstrapping Algorithmus existieren, damit ein neuer Knoten mindestens einen Teilnehmer des Netzes findet. Meistens sind dies vorgegebene IP Adressen, die eine kurze Liste von Teilnehmern bereitstellen. In bestimmten Fällen sind auch Discovery Mechanismen über Multicast oder in lokalen Netzen sogar Broadcast Pakete möglich.

Diese Form von P2P Netzen ist einem mobilen Ad-Hoc Netz sehr ähnlich und setzt oftmals ähnliche Algorithmen zum Aufbau des Netzes ein. Dies erfordert einen hohen Koordinationsaufwand zwischen den Knoten, da jeder Knoten einen Teil der Netztopologie kennen muss, um im Fall des Ausfalls eines Nachbarn reagieren zu können. Beispiele für diese Form der Organisation von P2P Netzen sind Gnutella [6] und Freenet [5].

### 3.2 Zentralisierte Systeme

Zentralisierte Systeme werden oft als P2P Netze der ersten Generation bezeichnet. Hier wird ein zentraler Server eingesetzt, der die Anfragen der Nutzer koordiniert und globales Wissen über die Verteilung der Inhalte auf den Nodes besitzt. Alle Knoten bauen zuerst zu diesem Server eine Verbindung auf und nutzen dann Informationen von dort um, je nach Verteilungsverfahren, zu anderen Knoten Verbindungen aufzubauen.

Derartige Systeme besitzen durch die zentralisierte Struktur allerdings eine Schwachstelle, die durch einen Ausfall das gesamte Netz lahmlegt. Andererseits ist der Koordinationsaufwand sehr viel geringer. Das globale Wissen im zentralen Server erspart den ständigen Austausch von Topologie Informationen zwischen den benachbarten Peers. Bekannte Vertreter dieser Gruppe sind Napster [3] und Bittorrent [7].

### 3.3 Hybride Systeme

Die dritte Gruppen sind hybride Systeme. Sie besitzen keinen zentralen, vorgegebenen Server sondern bestehen nur aus Knoten, die auch jederzeit dem Netz beitreten können. Einige dieser Knoten bekommen allerdings dynamisch eine Sonderrolle zugewiesen und nehmen dann besondere Koordinationsfunktionen ein. Die Auswahl der Knoten kann nach verschiedenen Auswahl-Kriterien geschehen:

- *Topologie*: In diesem Fall wird versucht Knoten, die eine zentrale Position im Netz einnehmen zu wählen. Ziel dieser Auswahl ist die mittlere Distanz eines Knotens zum nächsten Koordinationsknoten minimal zu halten und gleichzeitig die Zahl der Koordinationsknoten niedrig zu halten.
- *Verfügbare Bandbreite*: Hier wird die verfügbare Bandbreite der Knoten gemessen und Knoten mit einer hohen verfügbaren Bandbreite werden zu einem Koordinationsknoten. So soll gewährleistet werden, dass Koordinationsknoten nicht durch den zusätzlichen Netzverkehr überlastet werden. Zusätzlich wird bei diesem Ansatz versucht danach die Topologie anzupassen, um die Distanz der Knoten zu den Koordinationsknoten zu verringern.

- *Rechenleistung/Speicher*: Einige P2P Systeme benötigen in den Koordinationsknoten viel Speicher und wählen deshalb diese Knoten nach dem verfügbarem Speicher aus. Diese Auswahl hat den Vorteil der einfachen lokalen Entscheidbarkeit. Bei allen anderen Methoden ist viel Kommunikation mit den anderen Knoten nötig, bevor eine Einteilung möglich wird.

Diese Systeme vereinen viele Vorteile reiner P2P Systeme und der zentralisierten Systeme, aber verlangen von einigen Teilnehmern einen größeren Beitrag. Bekannte Vertreter dieser Kategorie sind Gia [8] und Kaazaa [9].

## 4 Tragedy of the Commons

”Tragedy of the Commons” ist ein Begriff aus den Wirtschafts- und Sozialwissenschaften [10]. Er beschreibt die Probleme der kostenlosen Nutzung einer begrenzten Ressource durch eine Gruppe. Als Beispiel ist hier die Nutzung einer Wiese, die im Mittelalter von allen Dorfbewohnern und Ihren Tieren genutzt werden konnte (”Commons”) genannt. Die Wiese könnte von 50 Kühen dauerhaft genutzt werden, bei mehr Tieren würde die Wiese nicht mehr schnell genug nachwachsen. Für keinen einzelnen Bewohner gibt es einen Grund grade seine Tiere nicht grasen zu lassen. Ohne eine Steuerung der Nutzung, welche Anreize schafft die Wiese nicht zu nutzen (oder Strafen bei zu starker Nutzung durch einen Einzelnen) würde der Gesamtnutzen der ”Commons” für alle sinken.

Ähnliche Probleme treten bei der P2P Content Verteilung auf. Die Probleme beziehen sich hauptsächlich auf zwei Bereiche: Die Nutzung der globalen Internet-Infrastruktur und die Bereitschaft zur Weitergabe des Content. Diese Probleme und zur Zeit eingesetzte Lösungsstrategien werden in den folgenden Abschnitten erläutert.

### 4.1 Netznutzung

Die Belastung des Netzes hängt von den übertragenen Datenmengen und der Anzahl der Router ab durch welche die Daten übertragen werden. Große Content-Anbieter, z.B. Google oder Akamai, versuchen die Anzahl der Hops zum Endkunden zu minimieren. Hierfür existieren für die Content-Anbieter Anreize wie die verbesserten Quality-of-Service Merkmale für den Nutzer. Außerdem können durch Peerings mit Internetprovidern Transitkosten gespart werden. Daraus folgt, dass die durchschnittliche Entfernung zwischen zwei beliebigen DSL-Benutzern weit höher als die Entfernung zwischen einem DSL-Benutzer und einem großen Content Anbieter ist. Dies führt wiederum zu einer erhöhten Netzbelastung durch P2P Programme.

Einige P2P Applikationen versuchen diesen Punkt durch zwei Verfahren zu verbessern:

- *Round-Trip Zeiten (RTT)*: Es wird die Round-Trip Zeit von Paketen zu einem Peer gemessen. Bei der Auswahl eines Peers wird der Peer mit der kürzesten RTT

gewählt.

- *Time-to-Live (TTL)*: Jedes IP Paket enthält einen TTL Wert. Dieser Wert wird von jedem durchlaufenen Router decrementiert. Falls jeder Peer den gleichen Startwert beim Senden der Pakete verwenden würde, entspricht der höchste TTL Wert der kürzesten Route.

Da es allerdings auch ohne diese Kriterien relativ aufwendig ist eine Quelle zu finden, wird meist darauf verzichtet die Auswahl weiter durch diese Kriterien einzuschränken. Außerdem können die TTL Werte leicht gefälscht werden, indem der Startwert höher gesetzt wird. So könnte ein Nutzer eine erhöhte Chance zu einer Verbindung bekommen.

Diese unterschiedliche Art der Netznutzung macht eine sinnvolle Messung des Datenaufkommens von P2P Netzen nur sehr schwer möglich, da der durch ein P2P System verursachte Traffic pro Router nicht unbedingt dem gleichen übertragenem Datenvolumen wie ähnlicher Traffic wie z.B. von Akamai entspricht. Die meisten Studien über das Datenaufkommen von P2P Netzen (z.B. CacheLogic [11]) lassen diese unterschiedliche Nutzung allerdings außer acht und setzen Traffic auf Routern dem übertragenen Datenverkehr gleich.

## 4.2 Tausch-Bereitschaft

Da heutige Peer-to-Peer Systeme im wesentlichen mit Unicast Paketen arbeiten entspricht die Summe der Downloads der Summe der Uploads des Gesamtsystems. Falls nicht künstlich Anreize zum Upload geschaffen werden, bringt ein Upload für einen Benutzer keinen direkten Vorteil. Die Entscheidung für einen Upload entspricht spieltheoretisch dem Prisoner's Dilemma [12]. Die dominante Strategie der Nutzer ist also nicht zu tauschen, sondern nur die Uploads der anderen zu Nutzen [13]. Dies würde (bei Ausschluss von Altruismus) die Nutzung von P2P Systemen unmöglich machen [14], weshalb von einigen Systemen Anreize zum Upload geschaffen werden. Auf die vier häufigsten wird im Folgenden eingegangen.



### 4.2.1 Lokale Limitierung

Die einfachste Lösung ist die maximale Downloadgeschwindigkeit softwareseitig in Abhängigkeit von der Uploadgeschwindigkeit zu begrenzen. Der Anreiz für Benutzer liegt in der höheren Downloadgeschwindigkeit.

Allerdings sind diese Beschränkungen durch kleine Änderungen in der Software leicht zu umgehen. Bei offenen Protokollen und mehreren Client-Implementierungen ist eine solche Lösung nicht durchsetzbar. In vielen Systemen wird allerdings dieses Verfahren zumindest als Teil-Lösung eingesetzt (z.B. edonkey [15]).

### 4.2.2 Peer Credits

Im Fall von Peer Credits speichert jeder Peer für empfangene Daten einen Credit für den Sender. Das P2P System kann nun diese Credits abfragen und so Peers, die viel uploaden, eine höhere Priorität für Downloads geben.

Die Verteilung der Credits und die Kontrolle sind durch die Ad-Hoc Natur von P2P Netzen nur sehr schwer erreichbar. In den meisten Systemen ist es möglich durch einen zweiten vorgetäuschten Client einem anderen beliebig viele Credits zu geben.

### 4.2.3 Tit for Tat

Eine weitere häufig genutzte Methode ist "Tit-for-Tat". Die Auswahl der Knoten die Daten von einem anderen Knoten empfangen wird priorisiert. Ein Peer erhält mit höherer Wahrscheinlichkeit Daten von einem anderen, falls er hierfür als Gegenleistung auch etwas bekommt.

Eine der ersten derartigen Implementierungen gab es in EMule. Jeder Peer besitzt eine Warteschlange für Downloader. Die Position in der Schlange ist abhängig von der Wartezeit und ob dieser Downloader gleichzeitig etwas sendet.

### 4.2.4 Koordinierte Uploadraten

Dieses Verfahren funktioniert nur bei zentralisierten Topologien. Ähnlich wie bei Peer Credits werden für jeden Upload Punkte für den Uploader gutgeschrieben. Diese Punkte

werden allerdings zentral verwaltet und es kann so leichter ein Betrugsversuch bemerkt werden. Außerdem ist der Koordinationsaufwand bei Peer Credits sehr viel höher.

In diesem Fall ist es auch möglich genaue Statistiken über Upload und Downloads der einzelnen Benutzer zu erhalten. In vielen Systemen ist dies aus rechtlichen Gründen, sei es aus Gründen der Beweislast beim Tausch illegaler Inhalte oder aus Datenschutzgründen, nicht erwünscht.

## 5 Verteilung der Inhalte

In P2P Systemen gibt es einige Verfahren welche für die Verteilung der Inhalte zwischen den Peers genutzt werden. Einige dieser Verfahren sind abhängig vom Aufbau des Netzes, andere können miteinander kombiniert werden. Im folgenden werden einige Ansätze, die in heutigen Systemen vorkommen erläutert.

### 5.1 Swarms

Im Gegensatz zu den meisten anderen P2P Systemen wird bei Bittorrent ein Netzwerk aufgebaut, das genau eine Datei austauscht. Durch diese Beschränkung kann der Aufwand für die Suche nach der Datei gespart werden.

Bei Bittorrent werden Dateien in viele kleine Teil-Stücke von ca. 1MB aufgeteilt. Von jedem dieser Teil-Stücke wird in einer Steuerdatei eine Prüfsumme vermerkt. Außerdem wird in dieser Steuerdatei die Adresse eines Trackers vermerkt. Alle Peers kümmern sich nun um die Verteilung genau dieser einen Datei und teilen sich die Arbeit in viele kleine Teilstücke, was zur Namensgebung dieses Verfahrens führt. Das Verfahren ist vergleichbar mit einem Bienenschwarm, welcher durch starke Arbeitsteilung ein großes gemeinsames Ziel verfolgt.

Zum Start eines Downloads muss die .torrent Steuerdatei auf eine herkömmliche, nicht P2P Art heruntergeladen werden. Der Client verbindet sich dann zum Tracker aus der Steuerdatei. Zum Tracker wird die Liste der vorhandenen Teilstücke und die Menge der Up-/Downloads gesendet. Der Tracker sendet eine Liste von Peers zurück, die nicht vorhandene Teilstücke vollständig anbieten.

Die Peers tauschen dann untereinander Teilstücke aus. Sobald ein Stück vollständig empfangen ist, kann die Prüfsumme geprüft werden und im Erfolgs-Fall das Stück weiterverteilt werden.

Durch die schnelle Prüfung von Teilstücken und den Verzicht auf eine Suche nach Dateinamen lassen sich mit Bittorrent sehr hohe Übertragungsraten erreichen. Die kleinen Teilstücke ermöglichen außerdem, dass nur vollständige und geprüfte Teile weiterverbreitet werden. Dies führt zu einer höheren Zuverlässigkeit und Geschwindigkeit.

Ein weiterer Vorteil ist die gute Reaktion auf so genannte "Flashcrowds" [16]. Man

bezeichnet damit den starken Anstieg von Usern in sehr kurzer Zeit. In diesem Fall wollen sehr viele Nutzer gleichzeitig etwas downloaden, haben aber noch nichts anzubieten. Durch die Aufteilung in kleine Teilstücke bekommen die neuen Nutzer relativ schnell ein erstes Teil und können sich so an der Weiterverteilung beteiligen. Die Uploadkapazitäten können so recht früh genutzt werden.

Eine interaktive Simulation der Verteilung von Daten in Bittorrent Netzen gibt es von [17].

## 5.2 Hash-Ring

Einen anderen Ansatz wählt z.B. FreeNet [7]. Freenet verfolgt zu den von anderen P2P bekannten Zielen noch das Ziel der Anonymität und der Unmöglichkeit einen bestimmten Inhalt zu zensieren.

Freenet hat, wie auch bei den Swarm-Ansätzen, keine Funktion um Dateien anhand Ihres Namens zu finden. Jede Datei wird durch einen Hash eindeutig identifiziert. Um eine Datei zu verbreiten muss der Hash über andere Wege, wie e-mail, web oder auch Telefon, verbreitet werden.

Jeder Knoten im Freenet Netz bekommt durch ein verteiltes Verfahren eine global eindeutige, zufällige ID zugewiesen. Dieses verteilte Verfahren kann durch keinen einzelnen Teilnehmer stark beeinflusst werden. Jeder Knoten ist nun für Dateien deren Hash "in der Nähe" der eigenen ID liegen zuständig. Die Größe des Bereichs ist von der Menge des vorhandenen Speicherplatzes abhängig, wobei es auch Überschneidungen dieser Zuständigkeitsbereiche gibt.

Um eine bestimmte Datei aus dem Netz zu bekommen fragt ein Peer bei benachbarten Knoten, deren ID näher am Hash liegt als die eigene. Diese wiederholen diesen Schritt bis Sie bei einem zuständigen Knoten ankommen. Die Datei wird dann, im Gegensatz zum Vorgehen in anderen P2P Systemen, über den gleichen Weg wie die Anfrage bis zum Anfragesteller weitergeleitet. Jeder Knoten auf dem Weg speichert alle übertragenen Daten solange er genügend Platz im Zwischenspeicher hat oder sich im Zwischenspeicher Daten befinden, welche einen weiter entfernten Hash haben.

Zur Veröffentlichung einer Datei werden den Nodes welche näher am Hash sind

mitgeteilt wo sich die Datei befindet. Durch die Art des Abrufs wird die Datei relativ schnell auch von Nodes mit ähnlicher ID gespeichert werden. So verkürzen sich die Wege bis zu jeder Datei und jede Datei wird im Netz mehrmals vorgehalten.

Das Routing der Dateien auf dem gleichen Weg wie das Routing der Informationen über die Dateien verschleiert den Ursprung. Eine Node kann nicht sagen, ob der Nachbar selbst die Datei zur Verfügung gestellt hat oder die Datei nur weitergeleitet wird. Dadurch wird die Anonymität im Netz sichergestellt. Um eine Datei sicher zu verfolgen müsste man alle Teilnehmer des Netzes überwachen.

Das Zwischenspeichern der Dateien auf den Wegen führt dazu, dass man eine bestimmte Datei nicht zensieren kann. Es ist nicht vorhersehbar welche Knoten die Datei gespeichert haben. Ein Knoten speichert eine Datei auch zwischen ohne die explizite Anforderung durch den Nutzer. Dies ist wichtig aus Gründen der Haftung für den Inhalt des Zwischenspeichers.

Diese Vorteile werden leider durch eine geringere Geschwindigkeit erkaufte. Eine Datei muss mehrmals versendet werden um beim endgültigen Empfänger anzukommen. Statt Uploadgeschwindigkeit = Downloadgeschwindigkeit (siehe 4.2) gilt hier nur noch  $\text{Uploadgeschwindigkeit} = k \cdot \text{Downloadgeschwindigkeit}$ , wobei  $k$  die durchschnittliche Zahl der Knoten auf dem Weg zur Datei ist (und damit  $\geq 1$ ).

Die Reaktion auf Flashcrowds ist sehr viel schlechter. Durch das Spiegeln der Dateien, auch in nicht interessierten Knoten wird die Verteilung einer Datei nur sehr langsam besser. Es dauert recht lange bis die Kapazitäten der neuen Nutzer optimal genutzt werden können. Allerdings ist Freenet mehr auf eine langzeitige Nutzung und auf die langzeitige Erhaltung von Content spezialisiert.

### 5.3 Koordiniert

Die koordinierte Übertragung kann nur zusammen mit einem zentralisierten P2P System benutzt werden. Hierbei werden von einem zentralen Server die Übertragungen zwischen den Peers festgelegt. Dieses Verfahren kann auch sehr leicht koordinierte Übertragungsraten nutzen.

Um den Koordinationsaufwand gering zu halten sollte eine Datei möglichst voll-

ständig von einem Peer zum anderen übertragen werden. Aufgrund dieser Tatsache eignet sich das Verfahren hauptsächlich für die Übertragung relativ kleiner Dateien. Die prominentesten Vertreter dieser Verteilungsart waren Napster [3] und AudioGalaxy.

### 5.4 Streaming

Inzwischen gibt es erste Ansätze für Streaming in P2P Netzen. Jeder Client verteilt hier den empfangenen Stream weiter, so entsteht weniger Traffic auf der Seite des Anbieters. Die bisherigen Ansätze (z.B. peercast [18]) entsprechen einem (Ad-hoc) Overlay-Multicast Netzwerk.

Die Herausforderung für ein solches Netzwerk besteht hauptsächlich in der schnellen Anpassung des Multicast Baumes an die ändernde Topologie. Die Streams sind, je nach Anzahl der hinzukommenden/verlassenden Clients, allerdings noch nicht sehr stabil.

### 5.5 NAT Traversal

Zur direkten Kommunikation zwischen 2 Peers muss mindestens einer von beiden eine vom anderen erreichbare IP Adresse besitzen. Im heutigen IPv4 Internet sind Network Address Translators (NAT [19]) allerdings weit verbreitet. Ein NAT ermöglicht durch ändern der IP Header mehreren Rechnern eine global nutzbare IP Adresse zu teilen. Der NAT hält eine Tabelle aller offenen Verbindungen jeweils mit der Port/IP Kombination des internen und des externen Netzes vor. Erhält er nun ein Paket ändert er diese Kombination jeweils in die andere und leitet das Paket dann weiter. Dieses Verfahren lässt nur Verbindungen von innen nach außen zu. NATs sind besonders häufig in privaten Haushalten zu finden, da meist nur eine IP Adresse zur Verfügung gestellt wird und mehrere Geräte genutzt werden sollen.

Befinden sich 2 Peers jeweils hinter einem NAT ist es keinem möglich zum anderen eine Verbindung aufzubauen. Es gibt allerdings eine NAT Traversal genannte Technik verbindungslose UDP Pakete durch zwei NAT zu übertragen. Die "Simple Traversal of User Datagram Protocol Through Network Address Translators" (STUN) Technik ist im RFC 3489 [20] spezifiziert.

Bei STUN handelt es sich um ein mehrstufiges Verfahren. Im ersten Schritt wird ein Paket zu einem direkt erreichbarem STUN Server geschickt. Dieser Antwort dann mit unterschiedlichen Port und IP Absendern. Aus den empfangenen Paketen kann der Client dann die genaue Funktionsweise des NAT schließen. Wenn nun zwei Clients die genaue Funktionsweise ihres NATs kennen und über eine dritte Partei mitgeteilt bekommen, dass sie eine Verbindung aufbauen sollen können sie durch gegenseitige UDP Pakete auf beiden NATs einen entsprechenden Eintrag erzeugen und so auch eingehende Pakete erlauben.

## 6 Zusammenfassung und Ausblick

Die Arbeit zeigt einige wichtige Verfahren in P2P Systemen, die zur Content Delivery eingesetzt werden können. Nur wenige der heutigen Systeme sind für eine zuverlässige Verteilung von großen Inhalten geeignet. Besonders hervorzuheben ist hier Bittorrent, das zur Verteilung sehr großer Dateien optimiert ist und die Auslieferung einer ganz bestimmten Datei garantieren kann.

Eine sehr interessante Entwicklung auf diesem Gebiet ist eXeem [21]. EXeem benutzt die bekannten Ideen von Bittorrent verteilt allerdings die Steuerdateien auch über dieses Netzwerk. Über die genaue Funktionsweise dieser Software ist allerdings noch nicht viel bekannt.

Durch die Einführung von IPv6, die damit verbesserte Ende-zu-Ende Erreichbarkeit und die Abschaffung von NATs werden in den nächsten Jahren P2P Systeme noch bessere Ergebnisse erzielen können.

Aber auch neue, noch nicht sehr weit verbreitete Techniken wie NAT Traversal und verbesserte Nutzung der Netzinfrastruktur durch RTT Messung haben das Potential die Verteilung von Inhalten mit P2P noch weiter zu verbessern.

Ein weiterer wichtiger Faktor für die Performanz von P2P Systemen sind Suchverfahren. Hier gibt es viele neue Techniken, die bisher nur teilweise praktisch eingesetzt werden.

Auch die rechtlichen Rahmenbedingungen sind sehr entscheidend für die Verbreitung von P2P Systemen, z.B. wird z.Zt. in Kalifornien ein Verbot der Verbreitung von P2P Software diskutiert, falls die Software den Tausch von geschützten Inhalten nicht unterbindet. Dies würde automatisch alle reinen P2P Systeme verbieten.



## Literatur

- [1] Clay Shirky. What Is P2P... And what Isn't. <http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html>.
- [2] Stephan Kluge. Route 66 ein amerikanischer albTraum. <http://www.route66.de/>.
- [3] Napster. <http://www.napster.com/>.
- [4] FSF Foundation. GNUNet. <http://www.gnu.org/software/gnunet/>.
- [5] Ian Clarke. The freenet project. <http://freenet.sf.net/>.
- [6] Lime Wire. Limewire - A Gnutella Client. <http://www.limewire.com/>.
- [7] Ian Clarke, Scott G. Miller, Theodore W. Hong, Oskar Sandberg, and Brandon Wiley. Protecting Free Expression Online with Freenet. In *IEEE Internet Computing*, 2002.
- [8] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making Gnutella-like P2P Systems Scalable. In *SIGCOMM2003*, 2002.
- [9] Kazaa . <http://www.kazaa.com/>.
- [10] Garret Hardin. The Tragedy of the Commons. In *Science*, pages 1243–1248, 1968.
- [11] CacheLogic. The True Picture of Peer-to-Peer Filesharing . <http://www.cachelogic.com/research/slide3.php>.
- [12] Robert Axelrod and William D. Hamilton. The Evolution of Cooperation. In *Science*, pages 1390–1396, 1981.
- [13] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster. To share or not to share? an analysis of incentives to contribute in file sharing environments, 2003.
- [14] E. Adar and B. Huberman. Free riding on gnutella, 2000.
- [15] Edonkey2000 . <http://www.edonkey2000.com/>.

## LITERATUR

---

- [16] Johan Pouwelse. The BitTorrent P2P file-sharing system. In *The Register*, 2004.
- [17] aphid. Bittorrent Simulation. <http://aphid.org/btsim/>.
- [18] Peercast.org - p2p broadcasting for everyone . <http://www.peercast.org/>.
- [19] K. Egevang and P. Francis. The IP Network Address Translator (NAT) .
- [20] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) .
- [21] eXeem . [http://www.exeem.com /](http://www.exeem.com/).