

# Taming the torrent

Johannes Ranftl

Seminar Innovative Internet-Technologien und Mobilkommunikation, WS 2008/2009

Institut für Informatik, Lehrstuhl Netzarchitekturen und Netzdienste  
Technische Universität München

j.ranftl@dora-showtechnik.de

**Kurzfassung** – Das Datenvolumen im Internet wächst täglich und P2P – Tauschbörsen tragen einen Großteil dazu bei. ISP's versuchen diesen P2P Traffic zu minimieren oder zu blocken. Das Bittorrent Protokoll ist eines der bekanntesten Filesharing P2P Protokolle. Im Folgenden wird ein Tool vorgestellt, das versucht den P2P Traffic im Netz des ISP's zu halten und so die Verbindungsqualität der Peers erhöht und die Kostenstruktur der ISP's verbessert.

## Schlüsselworte

Bittorrent, Downloadgeschwindigkeit, Ono, Content Distribution Network, Azureus, ISP

## 1. Einleitung

Durch das täglich wachsende Datenvolumen und die immer komplexeren Webanwendungen steigt der Bedarf nach Bandbreite immer stärker an. Nicht nur Unternehmen wollen heute mit schnellen Breitbandanschlüssen im Internet surfen, auch immer mehr Privatanwender wechseln auf schnelle DSL-Anschlüsse um das volle Leistungsangebot des Internets nutzen können.

Vor allem durch Web 2.0, Streaming-Medien und Plattformen wie z.B. Youtube.com oder Google-Video wird es für die Internetprovider immer schwieriger die nötigen Bandbreiten zuverlässig zur Verfügung zu stellen.

Durch das stetig steigende Angebot an P2P Software, wie Voice over IP und Tauschbörsen wird der Traffic für die Internet Service Provider (ISP) immer schwerer regulierbar. Vor allem die netzübergreifenden Anfragen, vom Netz des lokalen ISP ins Netz des ISP des anderen Peers von P2P Software treiben die Kosten für Traffic der ISP's in die Höhe. Eine Regulierung ist nur schwer möglich, da der Endknoten einer Peer to Peer Verbindung einen fixen Punkt darstellt und die Pakete somit nicht über Alternativrouten geroutet werden können.

Auch Bittorrent stellt ein Peer to Peer Netzwerk dar. Es wurde entwickelt um Dateien im Peer to Peer Betrieb zu tauschen. Eines der bekanntesten Bittorrent Programme ist Azureus.

In der folgenden Arbeit wird ein Plugin für Azureus vorgestellt, das versucht, bevorzugt auf Peers innerhalb des eigenen ISP-Netzes zuzugreifen. Somit verbessert sich nicht nur die Kostenstruktur des ISPs, was ja lediglich einen Vorteil für den Provider mit sich bringt, sondern auch die Qualität der Verbindung, worüber sich vor allem die Nutzer von Azureus freuen können, da höhere Downloadraten möglich sind und die Latenzzeit zum Peer um eine vielfache reduziert werden kann.

Die Arbeit ist in 4 Teile gegliedert. Im zweiten Teil werden die Grundlagen vermittelt die nötig sind um einen Einblick in die Funktionsweise von Ono zu bekommen. Es wird speziell auf P2P

Netze, Bittorrent und Positionsbestimmung im Netzwerk eingegangen.

Der dritte Teil beschreibt die Funktionsweise und den Nutzen von Ono. Vor allem die Netzwerkbestimmung mittels CDN-Anfragen steht hier im Mittelpunkt.

Der letzte Teil ist eine persönliche Beurteilung der Software die den Nutzen und die Funktionen des Tools kritisch in Frage stellt.

## 2. Grundlagen

Um auf die Funktionsweise von Ono eingehen zu können muss erst ein gewisses Grundverständnis für einige Techniken geschaffen werden, die sich das Azureus-Plugin zu Nutze macht. Im nachfolgenden Abschnitt werden deshalb die nötigsten Grundlagen kurz erläutert.

### 2.1 P2P Netze

„In einem Peer-to-Peer-Netz sind alle Computer gleichberechtigt und können sowohl Dienste in Anspruch nehmen als auch Dienste zur Verfügung stellen. Die Computer können als Arbeitsstationen genutzt werden, aber auch Aufgaben im Netz übernehmen. Kernkomponente in einer Peer-to-Peer-Architektur ist das Overlay-Netzwerk, welches den Peers im Netzwerk die Funktionen Lookup und Suche zur Verfügung stellt.

Mit der Lookup-Operation können Peers im Netzwerk diejenigen Peers identifizieren, die für eine bestimmte Objektkennung (Object-ID) zuständig sind. In diesem Fall ist die Verantwortlichkeit für jedes einzelne Objekt mindestens einem Peer fest zugeteilt, man spricht von strukturierten Overlays. Mittels der Such-Operation können die Peers nach Objekten im Netzwerk suchen, die gewisse Kriterien erfüllen (z. B. Datei- oder Buddynamen-Übereinstimmung). In diesem Fall gibt es keine Zuordnungsstruktur für die Objekte im P2P-System, man spricht also von unstrukturierten Overlays.

Sobald die Peers in dem P2P-System identifiziert wurden, die die gesuchten Objekte halten, wird die Datei (in Dateitauschbörsen) direkt, d. h. von Peer zu Peer, übertragen. Es existieren verschiedene Verteilungsstrategien, [die entscheiden,] welche Teile der Datei von welchem Peer heruntergeladen werden sollen, z. B. BitTorrent.

In der Informationstechnik ist das Gegenteil zur Peer-to-Peer-Architektur die Client-Server-Architektur. Bei dieser bietet ein Server einen Dienst an und ein Client nutzt diesen Dienst. In Peer-to-Peer-Netzen ist diese Rollenverteilung aufgehoben. Jeder Host in einem Rechnernetz ist ein peer, denn er kann gleichzeitig Client und Server sein.“ [1]

## 2.2 Content Distribution Network

In der heutigen Zeit, in der durch Streaming-Medien und Web 2.0 das Datenvolumen im Internet kontinuierlich wächst wird es immer wichtiger, Inhalte im Internet möglichst effizient bereitzustellen und Lastspitzen auszugleichen. Das ist durch statische Client – Server Architekturen nicht mehr möglich. Um dieser Problematik vorzubeugen wurden CDN's entwickelt.

„Alle CDNs basieren auf dem gleichen Prinzip:

Die Inhalte eines Webanbieters werden auf verteilte Server an unterschiedlichen Orten repliziert, die sich üblicherweise innerhalb der Netzwerke von Internet Service Providern (ISP) und somit in unmittelbarer Nähe zu den Konsumenten befinden. Die hierbei verwendeten Knoten können dabei eigenständige Server sein, die von einem CDN-Betreiber ausschließlich für diesen Zweck installiert werden. Der Webanbieter beauftragt einen CDN-Betreiber (z. B. Akamai) mit der Verteilung seiner Inhalte und wird dadurch zu dessen Kunde (Customer). Oftmals erhalten die Kunden entsprechende (Software-)Werkzeuge, um die zu publizierenden Inhalte selbst auf den nächstgelegenen Server innerhalb des CDN hochzuladen – die Verteilung aus andere Server des CDN erfolgt automatisch. Über umfangreiche Tools hat der Anbieter Zugriff zu Aufrufstatistik, Netznutzung, usw. Beim Zugriff auf das Online-Angebot des Webanbieters erfolgt eine entsprechende ‚Umleitung‘ der Anfragen, so dass die an die Clients zurück gelieferten Inhalte überwiegend oder sogar vollständig von Servern des CDNs bedient werden. Bei der Auswahl des die Anfrage beantwortenden Servers wird immer der gerade günstigste Server ermittelt, wobei hier durch load balancing durchaus auch ein anderer als der nächstgelegene in Frage kommt, sofern dieser z. B. bereits stark ausgelastet ist.“ [2]

## 2.3 Positionsbestimmung im Netzwerk

„Um die Netzwerkdistanzen, bzw. Netzwerklatenzen, zwischen Knoten vorhersagen zu können, ohne dabei für jedes Knotenpaar die Latenz messen zu müssen, wurden Ansätze entwickelt, die meist auf der Berechnung synthetischer Koordinaten beruhen. Aus diesen Koordinaten kann dann mit einer Distanzfunktion die Latenzen zwischen zwei Knoten abgeleitet werden kann.

Im Folgenden werden grob einige gängige Verfahren skizziert:

*IDMaps* Einer der ersten Ansätze war IDMaps, ein infrastruktur-basierter Dienst zur Abschätzung der Latenzen zwischen Internetknoten. IDMaps basiert auf der Verwendung von einigen 1000 so genannter Tracer-Knoten. Alle Tracer-Knoten kennen ihre Latenzen untereinander. Außerdem gibt es eine eindeutige Zuordnung von Tracer-Knoten zu CIDR-Adresspräfixen. Die Latenzen zwischen diesen Präfixen und den so genannten Präfix-Tracer-Knoten sind bekannt. Die Berechnung der Latenz zwischen zwei Knoten  $k_1$  und  $k_2$  setzt sich nun aus den Latenzen zwischen den Präfix-Tracer-Knoten von  $k_1$  und  $k_2$  und der Latenz zwischen den beiden Tracer-Knoten zusammen [...]. Die Genauigkeit der Latenzschätzung hängt dabei von der Anzahl der Tracer-Knoten ab.

*Global Network Positioning (GNP)*: Dieser Ansatz beruht auf einer kleinen Anzahl so genannter Landmark-Knoten, deren Koordinaten allen Knoten bekannt sein müssen. Andere Knoten

müssen nun die Knoten-zu-Landmark-Latenzen via ICMP-Nachrichten messen. Basierend auf diesen Messungen werden nun die eigenen Koordinaten relativ zu denen der Landmark-Knoten errechnet [...].

*Binning*: Dieser Ansatz [...] basiert auf einer Klassifizierung der Knoten. Liegen zwei Knoten in der gleichen Klasse, sind sie im Bezug auf die Netzwerk-Latenz relativ nahe beieinander. Dieser Ansatz basiert auf der Annahme, dass es nicht notwendig ist, den jeweils optimalen Knoten zu finden, sondern dass auch eine grobe Abschätzungen durch Einteilung in verschiedene Distanzklassen ausreicht, um messbare Performanzsteigerungen zu erzielen.

*Vivaldi*: Hierbei handelt es sich um einen vollständig verteilten Algorithmus. Vivaldi [...] kommt ohne zentrale Infrastruktur aus und ist deshalb für viele P2P-Anwendungen besonders interessant. Jeder Knoten berechnet seine eigenen synthetischen Koordinaten in einem d-dimensionalen Koordinatensystem.

Dies erfolgt durch regelmäßigen Austausch der geschätzten Koordinaten und einer gemessenen Latenz zwischen zwei Knoten. Anhand dieser Daten kann nun jeder Knoten berechnen, inwiefern er die eigenen Koordinaten verschieben müsste, um die gemessene Latenz mit der geschätzten Latenz in Einklang zu bringen. Dabei wird angenommen, dass alle Knoten regelmäßig mit anderen Knoten kommunizieren und so die benötigten Daten austauschen können. Es ist ausreichend, wenn dabei jeder Knoten mit nur einem Bruchteil aller Knoten kommuniziert. Die Latenz lässt sich durch Berechnung der euklidischen Distanz der Koordinaten zweier Knoten schätzen. Hinsichtlich der Genauigkeit ist Vivaldi mit GNP vergleichbar. Die Qualität der Prognose hängt hier von der Anzahl der Kommunikationspartner und der Dimensionalität der Koordinaten ab[...].“ [3]

## 2.4 Bittorrent

„BitTorrent [...] ist ein kollaboratives Filesharing-Protokoll, das sich besonders für die schnelle Verteilung großer Datenmengen eignet. Technisch ist das Protokoll der OSI-Schicht 7, also der Anwendungsschicht, zuzuordnen und setzt auf das TCP/IP-Referenzmodell. Die Referenzimplementierung durch den Erfinder Bram Cohen erfolgte in der Programmiersprache Python. Mittlerweile steht eine Reihe alternativer Programme zur Verfügung, die das BitTorrent-Protokoll implementieren.

Im Vergleich zum herkömmlichen Herunterladen einer Datei mittels HTTP oder FTP werden bei der BitTorrent-Technik die (ansonsten ungenutzten) Upload-Kapazitäten der Downloader mitgenutzt, auch wenn sie die Datei erst unvollständig heruntergeladen haben. Dateien werden also nicht nur von einem Server verteilt, sondern auch vom Nutzer zum Nutzer (Peer-to-Peer). Das belastet den Server weniger und der Anbieter spart Kosten. Insgesamt ist die Downloadlast nicht geringer, sie wird lediglich auf die einzelnen Nutzer verlagert. Bei populären Dateien verhindert diese Technik das Zusammenbrechen des Netzes infolge des Überschreitens der Kapazitätsgrenzen des Anbieters.

Tracker

BitTorrent besteht aus zwei Teilen: Dem Server-Programm, genannt Tracker [...] und dem Client, der auf dem PC des

Anwenders als Gegenstelle fungiert. Der Tracker verwaltet Informationen zu einer oder mehreren Dateien. Der herunterladende Client erfährt vom Tracker, wer sonst noch die Datei herunterlädt und verteilt. Sobald ein Client ein Segment [...] der Datei erhalten und die Prüfsumme verifiziert hat, meldet er dies dem Tracker und kann dieses Dateistück nun schon an die anderen Clients weitergeben. Die Menge aller Clients, die am gleichen Torrent interessiert sind, nennt man Schwarm. Clients, die im Besitz des kompletten Inhalts des Torrents sind, und somit nichts von anderen Clients herunterladen, sondern lediglich Daten verteilen, nennt man Seeder [...]. Clients, die nur im Besitz einiger Teile des Torrents sind, und sowohl Daten verteilen als auch beziehen, nennt man Peers [...]. Clients, die nur herunterladen, ohne selber zu verteilen, nennt man Leecher [...].

Um eine Datei herunterladen zu können, benötigt der Client eine Torrent-Datei [...]. In dieser befindet sich die IP-Adresse (bzw. der Hostname) des Trackers sowie Dateiname, Größe und Prüfsummen der herunterzuladenden Datei. Eine Torrent-Datei kann auch Informationen über mehrere Dateien beinhalten. Torrent-Dateien sind wenige Kilobytes groß und werden üblicherweise auf der Website des Anbieters zum Herunterladen bereitgestellt. Löscht der Anbieter den Torrent aus dem Tracker

oder geht der Kontakt zum Tracker verloren, können die Clients keinen neuen Kontakt zu anderen Clients mehr aufbauen, der Austausch zwischen schon bekannten Clients kann aber fortgeführt werden. Um trotzdem die Kontaktaufnahme zu anderen Clients zu ermöglichen, wird in neueren Clients zusätzlich der trackerlose Betrieb verwendet.

Im Gegensatz zu anderen bekannten File-Sharing-Systemen werden nicht beliebige Dateien aus den Beständen der Teilnehmer ausgetauscht. Vielmehr verteilt jeder Schwarm nur die Dateien, welche der Autor der Torrent-Datei explizit zum Herunterladen vorgesehen hat. Auch der Betreiber des Trackers bestimmt selbst, welche Downloads von diesem verwaltet werden sollen. Die einzelnen Tracker stehen nicht in Verbindung zueinander, es existiert daher kein gemeinsames Netz. Jede Tracker-Datei erschafft somit ihr eigenes temporäres Netz aus beteiligten Clients. Anbieter können sich so von fremden, möglicherweise illegalen Inhalten leichter distanzieren. [4]

### 3. Taming the Torrent mit Ono

Nachdem die Grundlagen zum Verständnis von Ono geschaffen wurden kann nun genauer auf die Funktion und den Nutzen eingegangen werden.

Das Kapitel ist in 3 Teile untergliedert. Im ersten Teil wird genauer auf das Suchverfahren eingegangen, dass mit Hilfe von CDN anfragen versucht Peers innerhalb des ISP Netzes zu identifizieren und gezielt Verbindung zu diesen Peers aufzubauen.

Im zweiten Teil werden dann konkrete Messergebnisse vorgestellt und interpretiert um den Vorteil des Plugins zu visualisieren.

#### 3.1 Positionsbestimmung mittels CDN

Um zu identifizieren, welche Peers innerhalb des gleichen ISP Netzwerkes liegen nutzt Ono die Basis Funktionalität der Content-Distribution-Netzwerke. Diese beinhaltet Anfragen via DNS-Redirection zu den günstigsten Servern weiterzuleiten.

Content Distribution Netzwerke haben üblicherweise in den meisten ISP Netzwerken Server, die miteinander verbunden sind.

Ono schickt einfache Anfragen an Inhalte (Content) (siehe Abbildung 1) an das Content Distribution Netzwerk und dieses leitet die Anfrage automatisch auf die performantesten Server für den jeweiligen Peer weiter.

Abbr.	DNS Name	CDN	Description
AA	e100.g.akamaiedge.net	Akamei	Air Asia (Southeast Pacific)
CN	a1921.g.akamai.net	Akamei	CNN.com (US news site)
LM	a245.g.akamai.net	Akamei	LeMonde.com (French news site)
FN	a20.g.akamai.net	Limelight	Fox News (US news site)
Abbr.	wdig.vo.llnwd.net	Akamei	ABC Streaming Video (US television)
PW	a1756.g.akamai.net	Akamei	Popular Web Site

Abbildung 1

Anhand der Server, auf die der Peer geleitet wurde, erstellt Ono ein Verzeichnis, indem die gelieferten Server gelistet sind. Ono ermittelt anschließend die Latenzzeiten zu diesen Servern. Der dadurch entstehende „Netzwerkplan“ wird dann mit den Plänen der anderen Peers ausgetauscht und verglichen.

Zeigt sich beim Vergleich der Server, dass die Netzwerkpläne sehr ähnlich sind, so wird mit dem gefundenen Peer bevorzugt eine Verbindung aufgebaut.

Die Priorisierung der Peers findet auf Basis der Ratio Maps statt, die anhand der vom CDN zurück gelieferten Server erstellt wurde.

#### 3.2 Vorteile durch Ono

Internet Provider versuchen auf immer neuen Wegen P2P Verkehr so gut es geht einzugrenzen und zu limitieren. Vor allem der Verkehr fremder ISPs der nur durch die eigenen Netze geleitet wird ist den Providern ein Dorn im Auge, da er einen Großteil des Traffics verursacht. Verständlicher Weise wird dieser Traffic von den Providern im Netz nicht mit höchster Priorität behandelt. Deshalb ist die Latenzzeit und die Rate der Paketverluste in P2P Verbindungen über mehrer Netze hinweg oft sehr hoch. Die Qualität der Verbindung ist somit eher als schlecht anzusehen.

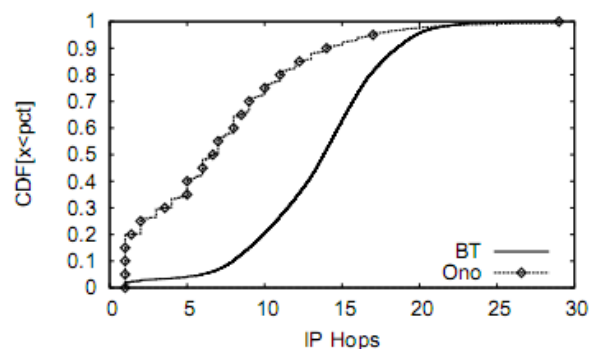


Abbildung 2 (Y- Achse Skala: 0 -100%)

Ono versucht durch den Vergleich der erstellten Netzwerkpläne möglichst viel Traffic innerhalb des ISP internen Netzes abzuwickeln. Der Vorteil liegt auf der Hand. Für den Provider sind die Kosten geringer, da er nicht so viel Traffic in fremden

Netzen produziert. Zudem steigert die Qualität der Verbindung im eigenen Netz die Zufriedenheit seiner Kunden.

Für den Azureus Nutzer ist der Nutzen jedoch nicht weniger von Bedeutung. Die stabilere Verbindung führt zu weniger Paketkollisionen und somit zu weniger Verlusten. Die Latenzzeit zum gegenüberliegenden Peer ist um ein vielfaches geringer. Dadurch lassen sich bessere Downloadraten erzielen.

In Abbildung 2 ist zu sehen, dass durch das aktive Plugin die IP-Hops um einen großen Anteil reduziert werden können. Auf der

Um diese Zahl interpretieren zu können, ist es jedoch wichtig ein Verständnis für den Begriff IP-Hops zu bekommen.

Die Anzahl der IP Hops sagt aus, über wie viele verschiedene Server (IPS) ein Paket geroutet wird, bis es seinen Zielort erreicht. Eine niedrige Zahl an IP Hops ist somit ein Zeichen dafür, dass die Gegenstelle mit hoher Wahrscheinlichkeit im gleichen Netz liegt.

Im Zusammenhang mit den IP-Hops sollten auch die AS Hops erwähnt werden. Durch Ono bleibt mehr als 30% des Verkehrs innerhalb eines abgeschlossenen Systems, kurz AS genannt.

Dies zeigt deutlich, dass dadurch der Traffic stärker innerhalb des Netzes eines ISP gehalten werden kann als ohne das aktivierte Plugin.

Abbildung 3 zeigt wie viele AS (Autonome Systeme) durchlaufen werden bis der gegenüberliegende Peer erreicht wird. Es werden beide Methoden dargestellt. Sowohl das Standard Verfahren über Bittorrent und Azureus, sowie das optimierte Verfahren mit Hilfe des Plugins Ono.

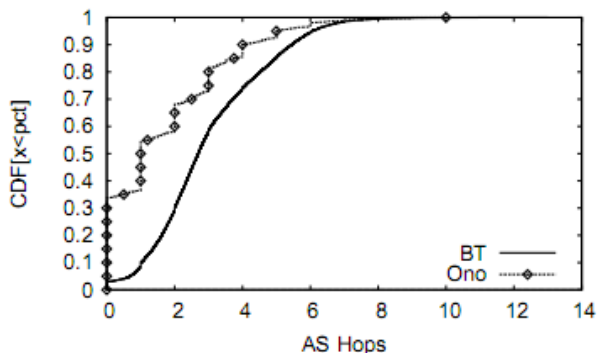


Abbildung 3 (Y- Achse Skala: 0 -100%)

Doch nicht nur die Anzahl der IP oder AS Hops sind ein Indikator für die Qualität der Verbindung.

Die Qualität hängt aber auch von der Latenzzeit ab. Die Latenzzeit, oder auch Verzögerungszeit genannt, spiegelt die Zeit wieder, die vergeht, bis eine Antwort von der Gegenstelle zurückkommt.

Die Latenzzeit wird von vielen Faktoren beeinflusst und ist ein wichtiger Wert in der Beurteilung von Netzwerkverbindungen.

In Abbildung 4 ist die Latenzzeit im Verhältnis, mit aktiviertem Ono-Plugin und ohne dargestellt. Um die Grafik richtig beurteilen zu können bedarf es jedoch einer kleinen Anmerkung. Die meisten Systeme runden Werte unter einer ms auf den Wert 1. Dadurch ergibt sich ein Wert von über 35% der Verbindungen mit einer Latenzzeit unter 1 ms. Dieser Wert ist für

Internetverbindungen, und besonders für P2P-Verbindungen außergewöhnlich gut.

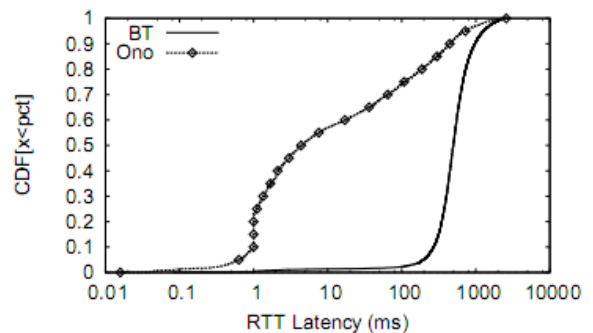


Abbildung 4 (Y- Achse Skala: 0 -100%)

Neben den IP und AS Hops, sowie der Latenzzeit bleibt ein weiteres Mittel, um die Effizienz und Qualität von TCP / IP Verbindungen beurteilen zu können.

Durch die Vielzahl an Paketen die durch das Internet geschickt werden und die große Flut an Paketdaten, die die Netzwerke und Netzwerkgeräte verarbeiten müssen, kommt es zu Paketkollisionen. Diese Kollisionen führen zum Verlust der kollidierten Pakete und können leider nicht verhindert werden, doch die Netzwerkprotokolle wissen damit umzugehen.

Durch Kontrollpakete wird erkannt ob es zu Kollisionen gekommen ist oder nicht. Wird also gerade eine Datei übertragen und das Kontrollpaket erkennt, dass es Kollisionen gegeben hat, so wird das verloren gegangene Paket erneut gesendet. Dies führt zu erhöhtem Traffic, ohne einen wirklichen Vorteil bei der Downloadgeschwindigkeit zu erreichen, da das verlorene Paket lediglich erneut gesendet wird.

Dieses Phänomen der Paketkollisionen wird durch die Loss rate, also die Verlustrate dargestellt. Je höher die Loss rate ist, desto mehr Pakete kollidieren irgendwo im Netz und gehen somit verloren. Diese müssen dann neu übertragen werden.

Abbildung 5 zeigt sehr deutlich, dass durch Ono die Verlustrate der Pakete um ein vielfaches geringer gehalten werden könnte.

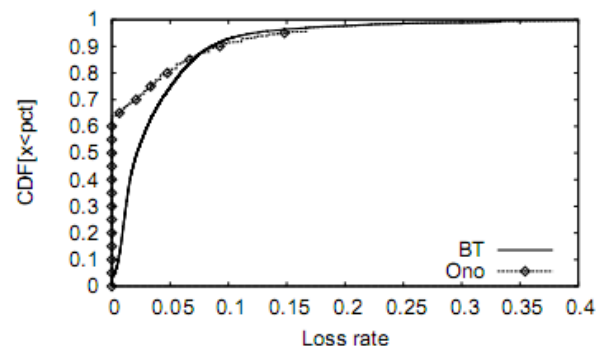


Abbildung 5 (Y- Achse Skala: 0 -100%)

Die Statistiken zeigen deutlich, dass durch Ono die Qualität der Verbindung um ein vielfaches gesteigert werden kann. Die Frage,

die sich jetzt stellt, ist, wie viele Systemressourcen benötigt werden, um dies zu erreichen.

### 3.3 Ressourcenverbrauch von Ono

Der Nutzen des Azureus Plugins wäre verschwindend gering, wenn durch die Auswahl der richtigen Peers schon ein Großteil der Bandbreite und Systemressourcen in Anspruch genommen werden würde. Daher stellt sich die Frage, wie aufwendig die gezielte Peeranalyse ist.

Der dieser Arbeit zugrunde liegende Artikel „Taming the Torrent“ stellte Messergebnisse vor, die zeigen, dass sich die verfügbare Bandbreite durch die CDN Messverfahren nicht ausschlaggebend verändert.

Die Messungen werden schon von den CDN selbst übernommen, so dass sich für die Peers lediglich die DNS Anfragen zu den Content Distribution Networks ergeben. Diese Kommunikation ist jedoch mit dem Austausch einiger weniger Pakete schon erledigt. Somit entsteht durch die Anfragen an die CDNs eine so geringe Netzwerklast, dass diese vernachlässigt werden kann.

Zur Funktionalität von Ono gehört jedoch auch noch der Austausch und der Vergleich der erstellten Netzwerk Ratio Maps, also der angelegten Netzwerkkarten mit denen der gefundenen Peers.

Dieser Prozess ist aufwendiger, als die DNS Anfrage beim CDN, doch in Relation zu den heute verfügbaren Bandbreiten immer noch verschwindend gering. Messungen haben ergeben, dass Ono pro Tag mit einem Uploadvolumen von weniger als 50 KB auskommt. Das Downloadvolumen liegt bei unter 70 KB pro Tag und ist somit ebenfalls zu vernachlässigen.

Die Vorteile liegen also auf der Hand. Der Artikel zeigt, wie die Verbindung durch den Einsatz CDN basierter Messverfahren in einem P2P Netzwerk verbessert werden kann, ohne gewaltige Messverfahren und Bandbreiten in Anspruch nehmen zu müssen. Lediglich durch verschicken kleiner DNS Anfragen wird die Auswahl der richtigen Peer ausgelöst und die Qualität der Verbindung somit deutlich gesteigert.

Ob sich diese Messergebnisse jedoch auch wirklich in Downloadraten zeigen und welche Probleme sich zusätzlich noch ergeben wird im folgenden Abschnitt erläutert.

## 4. Kritik

### 4.1 Nutzen

Nach einem ersten Blick auf die Messergebnisse ist man durchaus vom Nutzen des Plugins Ono überzeugt und sieht anhand der zahlreichen Grafiken relativ deutlich, wo der Vorteil gegenüber der normalen Peersuche liegt.

Betrachtet man die Thematik jedoch einmal genauer fängt man an sich über die Eine oder andere Problemstellung Gedanken zu machen. So stellen sich folgende Fragen:

Ist das Tool für jeden Inhalt den ich über Bittorrent downloaden möchte von Vorteil?

Findet Ono den Peer der mir am nächsten ist, oder den Peer der für mich am besten ist?

Gibt es einen Vorteil davon, den Traffic innerhalb meines ISP Netzes zu halten?

Beginnen wir mit der letzten Frage. Der Vorteil der sich für einen Endbenutzer ergibt hängt leider nicht nur von dem Faktor ab, ob meine Peergegenstelle innerhalb meines Netzes liegt und ist auch direkt mit der zweiten Frage verknüpft, denn das Ziel ist es als Azureus-Nutzer sicher nicht den nächsten Peer zu finden. Das Ziel ist es eher, den Peer zu finden, der die gewünschten Dateifragmente am schnellsten und unkompliziertesten zur Verfügung stellt. Liegt dieser Peer in einem anderen ISP Netz, so hat dies für den Azureus-Nutzer keine Nachteile, sofern der Download möglichst schnell abgeschlossen werden kann.

Hieraus gelangt man wiederum zur ersten Frage. Bringt das Tool für jeden Inhalt einen Vorteil? Nehmen wir mal an der Filesharing User möchte eine nicht sehr weit verbreitete Datei herunterladen. Ono verbindet ihn zu den verfügbaren Peers die innerhalb seines ISPs liegen.

Die beste Qualität der Verbindung, also mit geringer Latenzzeit und wenig IP-Hops ergibt sich nun bei einem Peer innerhalb des gleichen ISPs. Dieser ist nun jedoch gerade dabei eine Musiksammlung via Bittorrent aus dem Internet herunterzuladen. Seine maximale Upload-Bandbreite ist somit erschöpft, er stellt für den diesen Peer jedoch trotzdem eine Verbindung zur Verfügung.

Sicherlich ergibt sich jetzt ein Vorteil für den Provider, da dieser den Traffic innerhalb seines Netzes halten kann. Der Azureus Nutzer hätte jedoch mit einem Peer, der ein paar Knoten weiter entfernt in einem Nachbarnetz liegt, vielleicht mehr Bandbreite zur Verfügung gestellt bekommen. Dadurch wäre auch der Download schneller zu einem erfolgreichen Ende gekommen.

Somit zeigt sich in einigen Messergebnissen deutlich, wie gering sich der Vorteil von Ono in einigen Fällen zeigt.

In Abbildung 6 sieht man die Download-Geschwindigkeit mit und ohne den Einsatz von Ono in einem normalen ISP Netz. Der Unterschied ist gering.

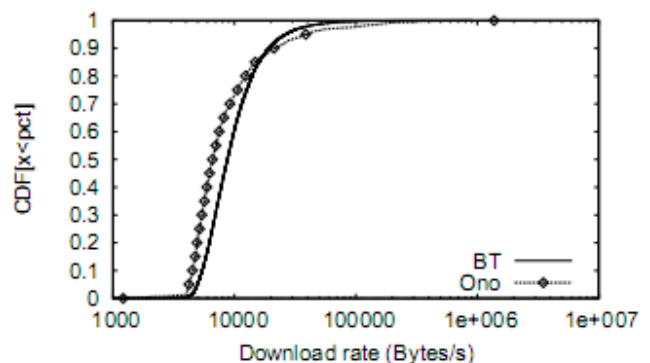


Abbildung 6 (Y-Achse Skala: 0 -100%)

Da die Anzahl der Nutzer stetig steigt und das Angebot immer größer wird ist es in 80%-90% der Fällen so, dass Azureus User ihren verfügbaren Down- und somit auch Upload fast immer zu 100% ausnutzen. Auch deshalb kann Ono hier keinen deutlichen Geschwindigkeitsvorteil erzielen.

Aus diesem Grund haben sich die Entwickler den Vorteil des Plugins mit Hilfe eines speziellen Tarifsystems eines fremden Providers gezeigt. Die Besonderheit des Tarifs liegt darin, dass für Verbindungen innerhalb des ISP Netzes mehr Bandbreite zur Verfügung gestellt wird als für Netzübergreifende Verbindungen. Abbildung 8 zeigt das Ergebnis dieser Messung.

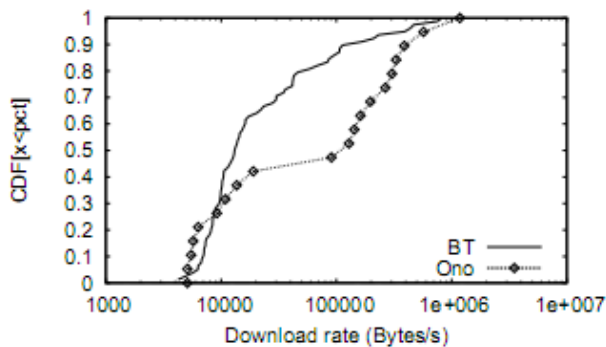


Abbildung 7 (Y- Achse Skala: 0 -100%)

Hier wird deutlich, dass die annähernde Vollausslastung der meisten Peers mit unter ein Grund für die schlechten Downloadvorteile von Ono ist.

Recherche in Online Foren hat ergeben, dass die Meinung über das Tool recht eindeutig ist. Die Aussagen könnten in folgendem Zitat zusammengefasst werden: „Ich konnte keinen Geschwindigkeitsvorteil erkennen!“

## 4.2 Resume

Abschließend könnte man sagen, die Idee des Plugins ist nicht neu. Es gibt viele verschiedene Plugins für Azureus oder andere Bittorrent Clients die von sich behaupten, die Downloadqualität und somit die Downloadraten zu steigern.

In der Praxis scheitern die meisten Tools jedoch nicht an den Messverfahren, sondern entweder an der zu geringen Verfügbarkeit der angefragten Dateien, oder an der allgemein überlasteten Bittorrent Peer to Peer Netzarchitektur. Denn die Grenzen setzen am Ende doch die Provider mit ihren Breitbandanschlüssen.

## 5. Literaturverzeichnis

- [1]<http://de.wikipedia.org/wiki/Peer-to-Peer>
- [2][http://www.net.t-labs.tu-berlin.de/teaching/ws0708/IR\\_seminar/ausarbeitungen/thomas\\_guenther.pdf](http://www.net.t-labs.tu-berlin.de/teaching/ws0708/IR_seminar/ausarbeitungen/thomas_guenther.pdf)
- [3][http://i30www.ira.uka.de/teaching/thesisdocuments/p2p/2004/tomanek\\_st\\_hybrid-overlay.pdf](http://i30www.ira.uka.de/teaching/thesisdocuments/p2p/2004/tomanek_st_hybrid-overlay.pdf)
- [4]<http://de.wikipedia.org/wiki/BitTorrent>
- [5] Alle Abbildungen wurden aus dieser Quelle entnommen: David R. Choffnes and Fabián E. Bustamante: Taming the Torrent, A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems. USA