

# Bachelorarbeit: A Real-time Streaming Protocol for Large-Scale Peer-to-Peer Networks

Christopher Probst

Institut für Informatik  
Heinrich-Heine-Universität Düsseldorf

3.12.2014



## Einleitung

- Problemstellung
- Motivation
- Zielsetzung

## Umsetzung

- Theorie
- Implementierung

## Ergebnispräsentation

- Möchte man Daten von einem einzelnen Computer aus zu beliebig vielen anderen Computern transferieren, so wird häufig ein Client / Server Netzwerk genutzt.
- Diese Methode ist einfach, zuverlässig und lang erprobt. Jedoch gibt es Skalierungsprobleme:
  - ▶ Jeder Client erhöht die Last des Servers
  - ▶ Mehr Clienten benötigen mehr Server und mehr Bandbreite
  - ▶ Uploadbandbreite der Clienten bleibt genutzt

- Um ohne ein skalierbares Serversystem Daten dennoch schnell zu verbreiten, kann ein Peer-to-Peer Netzwerk benutzt werden, wo jeder Computer einen Peer darstellt und bei der Datenverbreitung hilft. Einige Anwendungsfälle sind:
  - ▶ File Sharing (BitTorrent)
  - ▶ Audio und Video Streaming (Skype)
  - ▶ DHTs (Kademlia, Chord)
- Geräte mit einer geringen Uploadbandbreite können so dennoch schnell Daten verbreiten, wie z.B. ein Live-Videostream von einem Smartphone aus.

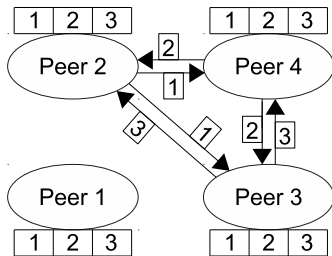
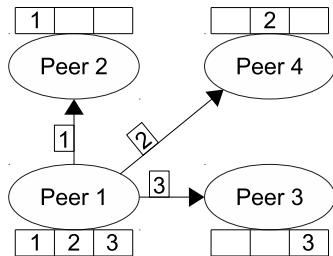
Implementierung einer Peer-to-Peer Anwendung, die Daten ausgehend von einem Peer, auch genannt Super-Peer, an beliebig viele Peers versendet. Dabei sollen folgende Bedingungen eingehalten werden:

- Möglichst hohe Auslastung der Uploadbandbreite der einzelnen Peers.
- Jeder Peer soll die Daten möglichst zur gleichen Zeit fertigstellen.
- Gesamtdauer unabhängig von der Anzahl der Peers.
- Gesamtdauer kleiner als  $2 * T_0$ .

- Die Daten werden vor dem Transfer in kleine Teile (Chunks) geteilt. Es muss mindestens so viele Chunks geben, wie es Peers im Netzwerk gibt.
- Der Super-Peer verschickt disjunkte Chunks parallel an alle übrigen Peers.
- Jeder Peer verschickt seinen Chunk an alle anderen Peers.
- Es wird vereinfacht angenommen, dass jeder Peer die gleiche Uploadbandbreite hat.

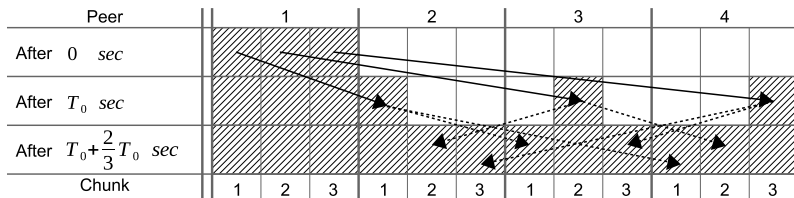
Wenn es genauso viele Chunks wie Peers gibt, kann man den Transfer in zwei Stufen betrachten. Peer 1 ist in diesem Fall der Super-Peer.

- ▶ Links: Alle Peers bekommen einen disjunkten Chunk vom Super-Peer.
- ▶ Rechts: Die Peers schicken sich ihren eigenen Chunk gegenseitig zu.



Das folgende Bild zeigt den zeitlichen Ablauf:

- ▶ In den ersten  $T_0$  Sekunden, schickt der Super-Peer einen disjunkten Chunk an jeden Peer. Jeder Chunk ist daher  $\frac{\text{Datengröße}}{\text{Anzahl}}$  groß.
- ▶ Anschließend schickt jeder Peer seinen Chunk an die anderen beiden Peers, was  $\frac{2}{3} * T_0$  Sekunden dauert.
- ▶ Nach  $T_0 + \frac{2}{3} * T_0$  Sekunden besitzt also jeder Peer die Daten.



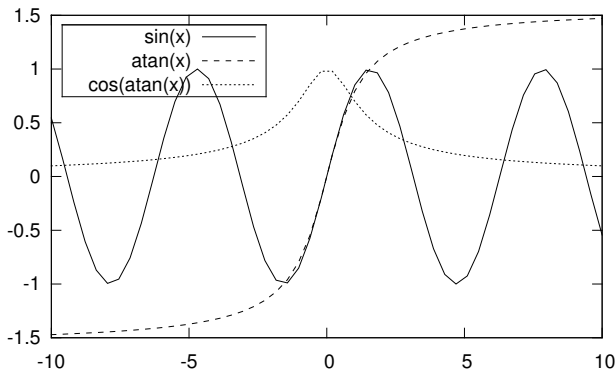


- Verdoppelt man die Anzahl der Chunks, halbiert man die Zeit zwischen  $T_0$  Sekunden und dem Ende.
- Die allgemeine Gesamtdauer lässt sich daher mit  $T(n, c) = T_0 + \frac{n}{c} * \frac{n-1}{n} * T_0 = (1 + \frac{n-1}{c}) * T_0$  berechnen, wobei  $n$  die Anzahl der Peers, jedoch ohne den Super-Peer, und  $c = n * 2^i, i \in \mathbb{N}_0$  die Anzahl der Chunks ist.
- Mit dieser Methode kann man immer unter  $2 * T_0$  Sekunden kommen. Verdoppelt man die Anzahl der Chunks beliebig oft, so kann man sogar beliebig nah an  $T_0$  Sekunden herankommen.

- Mesh Topologie: Jeder Peer ist mit jedem Peer verbunden.
- Pull-Based: Chunks werden nur auf Wunsch übertragen.
- Announcements: Jeder Peer kündigt an, welche Chunks vorliegen.
- Automatic (Re-)Connect: Peers finden andere Peers durch Super-Peer.

- Peer-Loss Detection: Super-Peer verschickt Chunks von Peers, die das Netzwerk verlassen haben, erneut.
- Jeder Peer versucht von möglichst vielen Peers gleichzeitig Chunks zu beziehen. Dies ist wichtig, da ein Pull-Based Ansatz verwendet wird.
- Implementiert in Java mit Netty5.

Grafiken werden wie gewohnt als PDF eingebunden (nur ohne figure-Umgebung):



In Diskussionen nach Präsentationen kommt es häufiger vor, dass Nachfragen gestellt werden, die mit dem regulären Material der Präsentation nicht zu beantworten sind.

Daher lohnt es sich, zusätzliche

- ▶ Grafiken
- ▶ Tabellen mit Detailinformationen
- ▶ Erklärungen

aus der schriftlichen Arbeit im Anhang unterzubringen.