HUMBOLDT-UNIVERSITÄT ZU BERLIN

Peer-to-Peer-Systeme

Teil I: Organisatorisches und Einführung

Björn Scheuermann

Humboldt-Universität zu Berlin
Wintersemester 2015/16

Organisatorisches

- Die Veranstaltung besteht aus drei Teilen:
  - Vorlesung
  - Übung
  - Projekt
- Aktuelle Informationen, Folien, Übungsaufgaben, Ankündigungen, Terminänderungen,. . . :

  https://www.ti.informatik.hu-berlin.de/teaching/ws/p2p

Vorlesung

- Ziele:
  - Vermitteln von Basiswissen
  - Kennenlernen von Ideen
  - Aufzeigen von Zusammenhängen
- Termin: Do 13 Uhr in RUD26 1’306
- Teilnahme freiwillig, aber *wichtig*
- Thematischer Fokus auf Protokollen und Algorithmen. . .
- . . . manchmal aber auch ein bisschen Theorie
- Vielleicht gelegentlich etwas unkonventionell – abwarten und überraschen lassen! ;-)

Übung

- Ziele:
  - Anwenden der theoretischen Kenntnisse aus der Vorlesung
  - (selbständiges) Erarbeiten, Vertiefen und Kennenlernen anderer/alternativer Ansätze
- Termin: Do 15 Uhr in RUD26 1’306, (ungefähr) 14-tägig, erstmals am 29.10.
- Übungsblätter werden online bereitgestellt
- Erfolgreiche Übungsteilnahme ist Voraussetzung für Leistungspunkte und Prüfungszulassung
- Bedeutet hier:
  - *vor der ersten Übung* über GOYA angemeldet
  - Vorrechnen per Zufallsgenerator
  - ein „Joker“ für unentschuldigtes Fehlen („Entschuldigung“ = rechtzeitig vorher + begründet)
  - alternativ: Lösungen spätestens am Tag vor der Übung schriftlich einreichen

Projekt

- Programmierprojekt über das gesamte Semester
- Das Thema bestimmen Sie selbst!

Implementieren Sie ein Peer-to-Peer-System

- Zum Beispiel könnten Sie eine (einfache) Anwendung entwickeln oder ein existierendes Programm um eine Peer-to-Peer-basierte Komponente erweitern
- Zusammenarbeit von zwei, in Ausnahmefällen auch drei Teilnehmern an einem gemeinsamen Projekt ist möglich; der Umfang des Projektes sollte dies dann aber widerspiegeln

Kriterien für Projekte

Ihr Projekt sollte. . .

- dem Umfang von 3 Leistungspunkten (ca. 90 Stunden Arbeitsaufwand pro Person) gerecht werden,
- ohne eine zentrale Instanz funktionieren,
- die notwendigen Peer-to-Peer-Protokolle und -Algorithmen auf der Anwendungsschicht selbst implementieren (in aller Regel sollten TCP- oder UDP-Sockets als Basis verwendet werden),
- (prinzipiell) mit beliebig vielen Teilnehmern funktionieren, von denen jeder einzelne jederzeit ausfallen darf, ohne dass das System „stirbt“, und
- über das Internet funktionsfähig sein (also: kein Multicast, kein Broadcast).

(Nicht unbedingt lösen müssen Sie das Bootstrapping-Problem und NAT-Traversal – mehr dazu im Laufe der Vorlesung.)

Projektskizze

- Zunächst stecken Sie Ihre Ziele in einer *Projektskizze* ab
  - *maximal* eine DIN-A4-Seite
  - abzugeben bis *spätestens Ende November*
  - soll 1) die Zielsetzung und 2) die zentrale zu lösende Problemstellung klar benennen
- Wir diskutieren die Projektskizze in meiner Sprechstunde, vor allem in Hinblick auf Umfang und (technische) Machbarkeit

Projekt

- Sie implementieren das von Ihnen vorgeschlagene System – rechtzeitig vor Ihrem Prüfungstermin
- Am Ende des Semesters stellen Sie Ihre Ergebnisse vor
- Die besten Projekte werden prämiert – Kriterien:
  - Originalität
  - technischer Anspruch
  - Qualität der Umsetzung
  - Qualität der Präsentation

Prüfung

- Mündlich
- Termine werden am Anfang und am Ende der vorlesungsfreien Zeit angeboten
- Zulassungsvoraussetzungen beachten!

Feedback und Sprechstunde

- Fragen, Probleme, Kritik, Vorschläge, Anregungen, Ideen,. . . : Immer sehr gerne!
- Auch ich möchte gerne noch etwas dazulernen!
- Sprechstunde: nach Vereinbarung
- E-Mail: scheuermann@informatik.hu-berlin.de

Voraussetzungen

Wichtige Voraussetzungen für diese Veranstaltung sind:

- Grundlagenwissen Rechnernetze, z. B.
  - Was ist eine IP-Adresse? Eine Portnummer?
  - Was ist ein Autonomes System?
  - Wie funktioniert NAT? Wie TCP?
  - . . .
- Grundbegriffe der Komplexitätstheorie (*O*-Notation!)
- „Basiswissen“ Mathematik (auch Stochastik)
- Neugier und Spaß am Tüfteln

Hilfreich (aber nicht notwendig) sind außerdem
Verteilte Systeme, Graphentheorie, Kryptographie,. . .

Literatur

- Peter Mahlmann, Christian Schindelhauer:
  Peer-to-Peer-Netzwerke.
  Springer, 2007.
- Ralf Steinmetz, Klaus Wehrle (Hrsg.):
  Peer-to-Peer Systems and Applications.
  Springer, 2005.
- John F. Buford, Heather Yu, Eng Keong Lua:
  P2P Networking and Applications.
  Morgan Kaufmann, 2009.
- Für Grundlagen: James Kurose, Keith Ross:
  Computer Networking: A Top-Down Approach. Pearson Education.
- **viel wichtiger: Forschungsliteratur!**

Peer-to-Peer?

Beispiele für Peer-to-Peer-Anwendungen

- Filesharing (Napster, Gnutella, eDonkey,. . .)
- Internettelefonie (Skype)
- Kooperative Downloads / Dateiverteilung (BitTorrent)
- Anonymität im Internet (TOR, Freenet)
- Audio-/Video-Streaming
- E-Mail
- Verteiltes Backup
- . . .

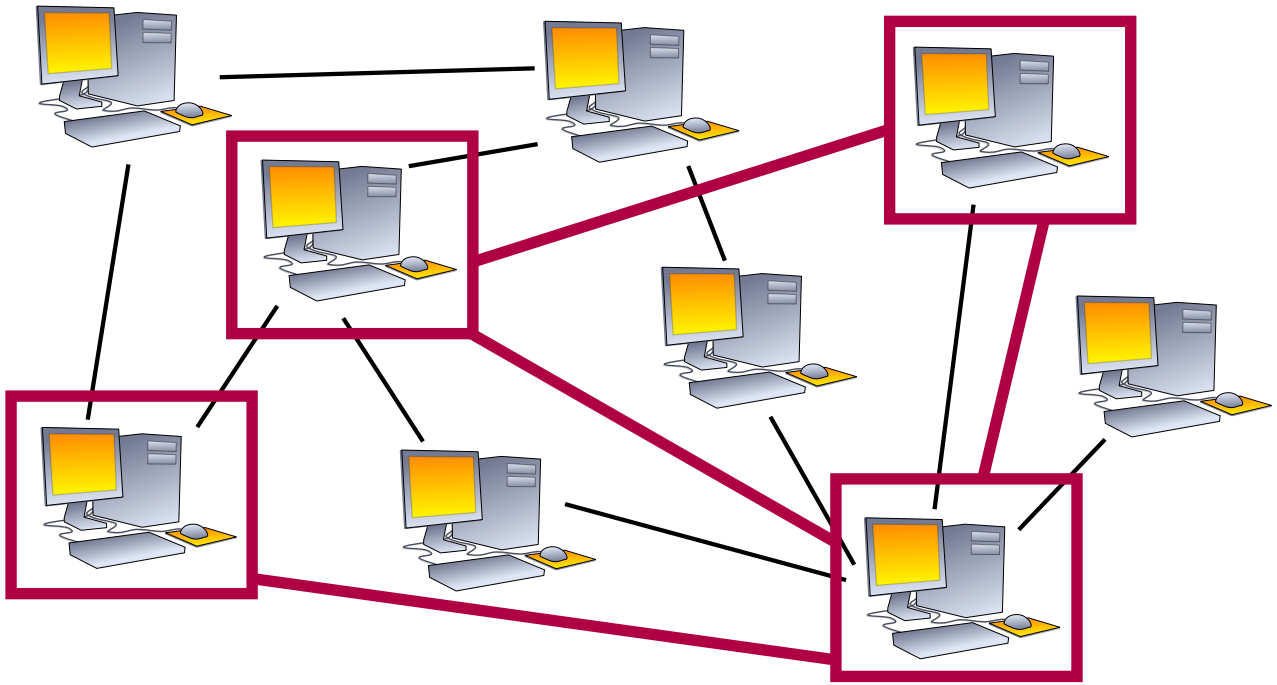
„Peer-to-Peer“ – Versuch einer Definition

- Was ist ein „Peer-to-Peer-System“?
- Engl. peer = Ebenbürtiger, Gleichgestellter (auch: Mitglied des Hochadels)
- Von lat. par = gleich
- *„Ein selbstorganisierendes System von gleichgestellten, autonomen Entitäten (Peers), das auf die gemeinsame Nutzung verteilter Ressourcen in einer Netzwerkumgebung unter Vermeidung zentralisierter Dienste abzielt.“* [Oram: Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O’Reilly, 2001; Steinmetz, Wehrle: Peer-to-Peer Networking and Computing, Informatik Spektrum 27(1), 2004]

Overlay-Netzwerke

Peer-to-Peer-Systeme sind oft *Overlay-Netzwerke*:

- Ein Netzwerk „über“ einem anderen Netzwerk
- Ein Teil der Knoten nimmt daran teil
- Diese Knoten etablieren (logische) Verbindungen
- So entsteht ein „Meta-Netzwerk“: das Overlay-Netz



Merkmale von Peer-to-Peer-Netzwerken

Einige zentrale Merkmale: Die Peers. . .

- . . . interagieren direkt miteinander und agieren somit als Client ebenso wie als Server
- . . . sind über ein Netzwerk verbunden und typischerweise geographisch weit verteilt
- . . . haben wechselnde Adressen im zugrundeliegenden Netzwerk
- . . . nutzen die Ressourcen anderer Peers (Speicherplatz, Bandbreite, Rechenzeit,. . .)
- . . . sind gleichgestellte Partner und sind autonom bezüglich der lokalen Ressourcen
- . . . sind nicht zuverlässig verfügbar

Struktur und Inhalte der Veranstaltung

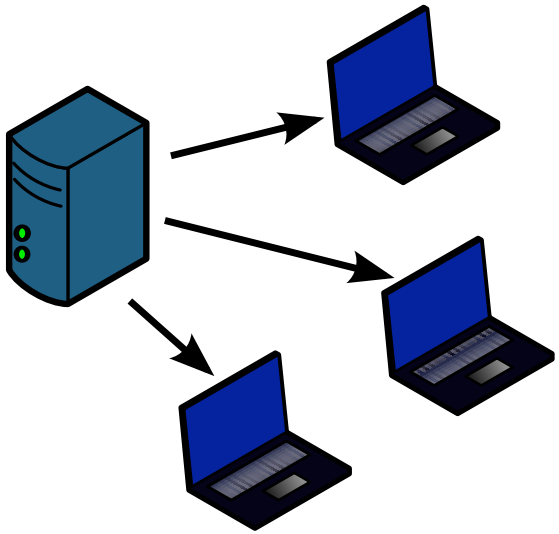
- Einführung und Motivation
- Unstrukturierte Systeme
  - Gnutella
  - Hierarchische Overlays
  - Zufallsgraphen, Small Worlds, skalenfreie Netze
- Strukturierte Overlays
  - CAN
  - Chord
  - Gradminimierte Netze
  - ...
- Praktische Aspekte
  - NAT Traversal
  - Sicherheit und Fairness
- Anwendungen

Welche Vorteile kann man durch den Einsatz von Peer-to-Peer-Techniken gewinnen?

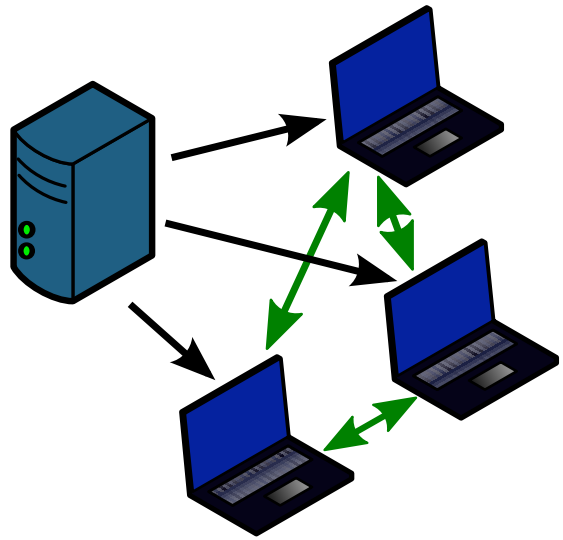
Was spricht gegen Peer-to-Peer-Lösungen?

Performance-Vorteile durch Peer-to-Peer?

- Betrachten wir als Beispiel die Dateiverteilung über ein Peer-to-Peer-Netzwerk ($\Rightarrow$ BitTorrent)
- Eine Datei der Größe F liegt auf einem Server und soll an n Rechner übertragen werden



(a) Client-Server



(b) Peer-to-Peer-unterstützt

- Frage: Wie lange dauert das Verteilen der Datei (mindestens)?

[Kumar, Ross: Peer-Assisted File Distribution: The Minimum Distribution Time, HotWeb 2006]

Mindest-Verteilungszeit Client/Server

- Dateigröße F
- Server hat Uploadbandbreite u_S
- n Clients mit Downloadbandbreiten $d_1, \ldots, d_n$
- $d_{\min} := \min_i d_i$
- Annahmen:
  - Netzwerkkern ist kein Engpass
  - Latenzen spielen keine Rolle
  - Dat(ei)en können beliebig fein unterteilt werden

Untere Schranke(n) für die Gesamtdauer D_{CS}?

Mindest-Verteilungszeit Client/Server

1. Server muss jedem Teilnehmer eine Kopie schicken:

$$D_{CS} \geq n \cdot \frac{F}{u_S}$$

2. Langsamster Client muss eine Kopie erhalten:

$$D_{CS} \geq \frac{F}{d_{min}}$$

Insgesamt:

$$D_{CS} \geq \max \left\{ n \cdot \frac{F}{u_S}, \frac{F}{d_{min}} \right\}$$

Mindest-Verteilungszeit Peer-to-Peer

- Dateigröße F
- Server hat Uploadbandbreite u_S
- n Peers mit
  - Downloadbandbreiten $d_1, \ldots, d_n$
  - Uploadbandbreiten $u_1, \ldots, u_n$
- $d_{\min} := \min_i d_i$
- Annahmen:
  - wie oben, und zusätzlich...
  - Peers können empfangene Bits sofort weiterleiten

Untere Schranke(n) für die Gesamtdauer D_{P2P}?

Mindest-Verteilungszeit Peer-to-Peer

1. Server muss die Daten min. einmal verschicken:

$$D_{\text{P2P}} \geq \frac{F}{u_S}$$

2. Langsamster Client muss eine Kopie erhalten:

$$D_{\text{P2P}} \geq \frac{F}{d_{\min}}$$

3. Gesamte Upload-Kapazität des Systems muss die Daten n-mal übertragen (Datenmenge insges. $n \cdot F$):

$$D_{\text{P2P}} \geq \frac{n \cdot F}{u_S + \sum_i u_i}$$

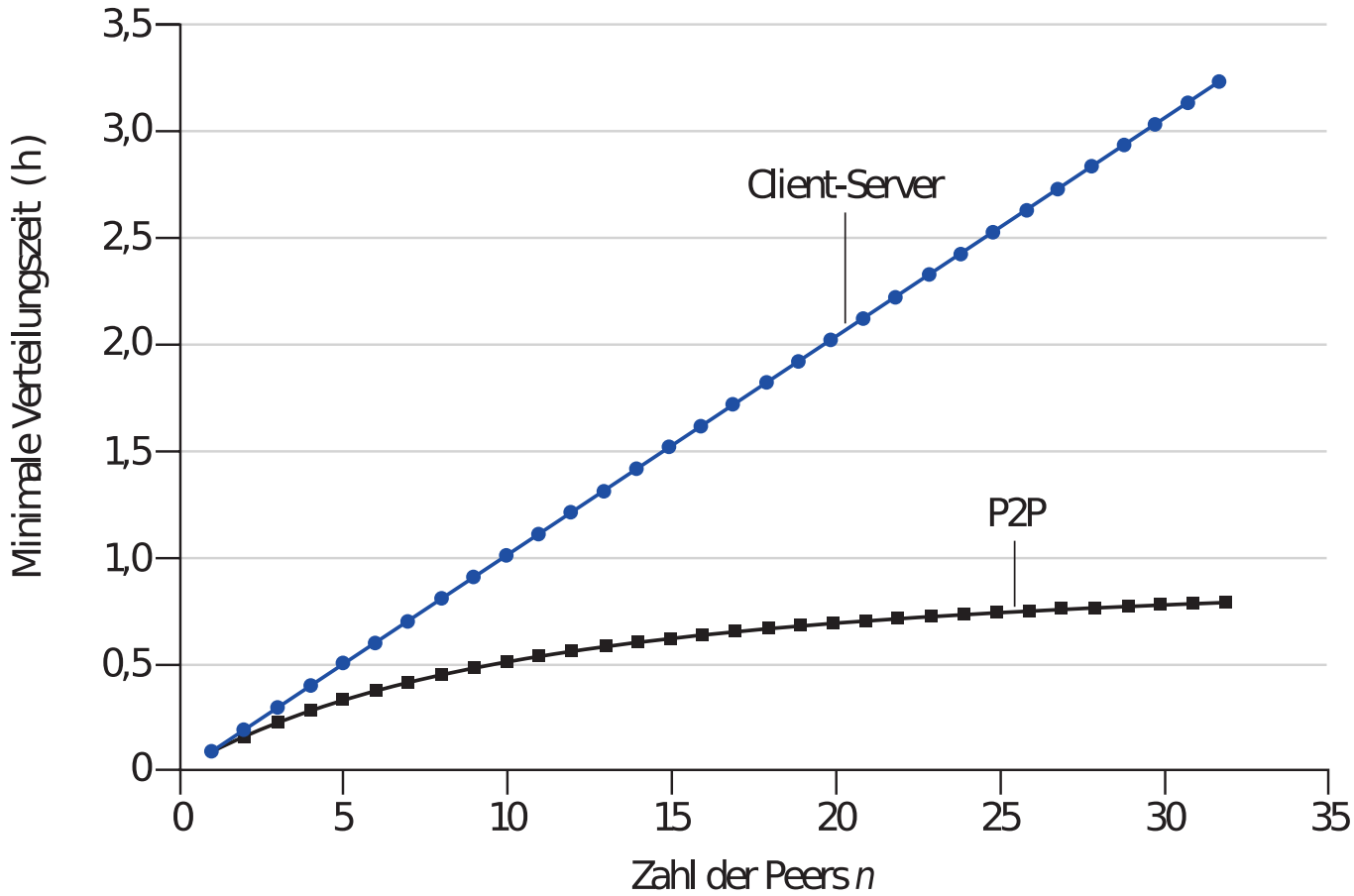
Insgesamt:

$$D_{\text{P2P}} \geq \max\left\{\frac{F}{u_S}, \frac{F}{d_{\min}}, \frac{n \cdot F}{u_S + \sum_i u_i}\right\}$$

Ist (unter o. g. Annahmen) realisierbar! [Kumar, Ross 2006]

Vergleich Mindest-Verteilungszeit

- Hier: identische Client-Uploadraten u
- $F/u = 1\,\mathrm{h}$ $u_S = 10u$ $d_{\min} \geq u_S$



(Abbildung: [Kurose, Ross 2008])

Vergleich Mindest-Verteilungszeit

$$D_{\mathrm{CS}} \geq \max\left\{n \cdot \frac{F}{u_S}, \frac{F}{d_{\min}}\right\} \qquad D_{\mathrm{P2P}} \geq \max\left\{\frac{F}{u_S}, \frac{F}{d_{\min}}, \frac{n \cdot F}{u_S + \sum_i u_i}\right\}$$

$$\begin{aligned} & \frac{F}{u_S} && \leq && n \cdot \frac{F}{u_S} \\ \text{und} \quad & \frac{n \cdot F}{u_S + \sum_i u_i} && \leq && n \cdot \frac{F}{u_S} \\ \Longrightarrow \quad & D_{\mathrm{P2P}} && \leq && D_{\mathrm{CS}} \end{aligned}$$

Peer-to-Peer-unterstützte Dateiverteilung ist immer mindestens gleich schnell wie die Client-Server-Lösung

Selbstskalierbarkeit

$$D_{\mathrm{CS}} \geq \max\left\{ n \cdot \frac{F}{u_S}, \frac{F}{d_{\min}} \right\} \qquad D_{\mathrm{P2P}} \geq \max\left\{ \frac{F}{u_S}, \frac{F}{d_{\min}}, \frac{n \cdot F}{u_S + \sum_i u_i} \right\}$$

Betrachte Grenzwerte für $n \to \infty$ (Annahme: $\forall i : u_i = u$):

$$\lim_{n\to\infty} D_{\mathrm{CS}} = \infty \qquad (\Rightarrow \text{skaliert nicht!})$$

$$\lim_{n\to\infty} D_{\mathrm{P2P}} = \max\left\{ \frac{F}{u_S}, \frac{F}{d_{\min}}, \frac{F}{u} \right\} < \infty$$

Es gibt also eine obere Schranke für die Peer-to-Peer-Verteilzeit, die für eine beliebig große Anzahl von Clients gilt!

Die verfügbaren Ressourcen in Peer-to-Peer-Systemen können mit der Zahl der Teilnehmer wachsen: **„Selbstskalierbarkeit“**