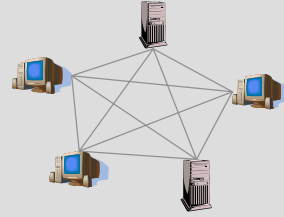


Peer-to-Peer-Netzwerke

Was bedeutet Peer-to-Peer (P2P)?

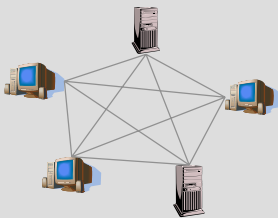
- Peer: der/die Ebenbürtige, der/die Gleichgestellte
- Peer-to-Peer: gleichberechtigte, direkte Kommunikation



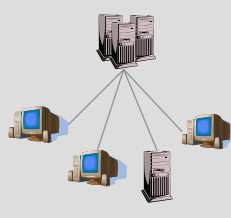
Peer-to-Peer vs. Client-Server

Peer-to-Peer

Client-Server

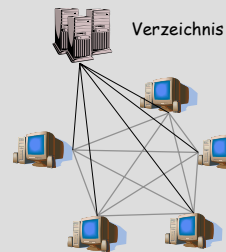


dezentrale Architektur



zentrale Architektur

Napster

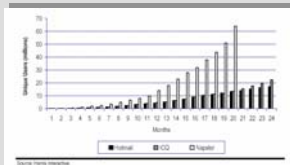


- zentrales Verzeichnis: Wer ist online und hat was gespeichert?
- Übertragung direkt von Nutzer zu Nutzer
- Konsument wird automatisch zum Verbreiter

- Kombination zentraler mit dezentraler Architektur
- *kein* reines P2P-Netzwerk

Napsters Erfolg

- 65 Millionen Nutzer innerhalb von 20 Monaten
- Netscape Navigator: 38 Millionen Installation innerhalb von 20 Monaten
- Januar 2001:
 - ~ 3 Milliarden Musikstücke mit jeweils ~ 3MB ausgetauscht
 - = 72 000 000 GBit
 - = 28 GBit/Sekunde



Warum war Napster so erfolgreich?

- Zugriff auf Inhalte anderer Nutzer
- Netzwerkeffekt: je mehr Nutzer, desto größer der Nutzen für jeden einzelnen Nutzer
- Netzwerkeffekte auch bei Instant-Messaging: je mehr Nutzer, desto mehr potentielle Kommunikationspartner
- bei Napster werden Netzwerkeffekte jedoch unmittelbar, ohne persönliche Kommunikation realisiert

Napster kein reines P2P-Netzwerk?



- hervorragende Rolle des zentralen Verzeichnisses
- nicht alle Knoten gleichberechtigt
- Wörtliche Definition von Peer-to-Peer schließt konventionelle Server-Server-Kommunikation ein, aber Napster aus.
- Peer-to-Peer nur Etikett für eine Entwicklung, die durch Napster ausgelöst wurde
- Peer-to-Peer im wörtlichen Sinn also *nicht* geeignet diese Entwicklung zu beschreiben

Peer-to-Peer nach Clay Shirky



- **Peer-to-Peer:** Anwendungen, die Ressourcen am Rand des Internets ohne feste IP-Adressen ausnutzen
Ressourcen: Speicherkapazität, CPU-Zeit, Inhalte, menschliche Präsenz



- operieren außerhalb des DNS
- operieren mit unsteten Ressourcen
- müssen Ressourcen hohe Autonomie gewähren

Also doch!



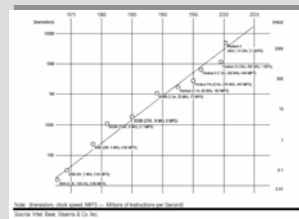
Nach Clay Shirky ist Napster Peer-to-Peer weil:

- Ressourcen am Rand des Internets ausgenutzt werden (Inhalte und Speicherkapazität)
- Napster außerhalb des DNS operiert
- Napster durch Echtzeitverzeichnis und redundante Speicherung mit unsteten Ressourcen operiert
- Ressourcen eine hohe Autonomie haben: entscheiden selbst, welche Inhalte, sie wann zur Verfügung stellen

Warum Peer-to-Peer?



- Moore (1975): alle 12-18 Monate verdoppelt sich die Transistordichte eines Chips (hat sich bestätigt)
- Prozessorgeschwindigkeit wächst genauso schnell
- Anzahl der Geräte im Internet wächst noch schneller



Kapazität wächst exponentiell

Warum Peer-to-Peer?



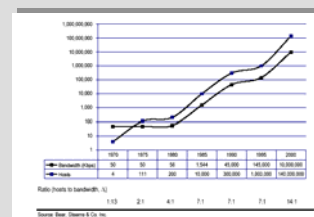
- Ressourcen des Internets wachsen also exponentiell (MBit, MIPS)
- Ressourcen am Rand des Internets bleiben ungenutzt
- Peer-to-Peer aktiviert diese ungenutzten Ressourcen

keine technologischen Vorteile, sondern Kostenvorteile

Warum Peer-to-Peer?



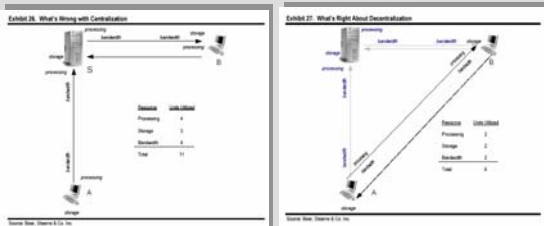
- Gilder: Bandbreite verdreifacht sich alle 12 Monate
- Anzahl der Geräte im Internet wächst seit 1975 mit gleicher Geschwindigkeit



Bandbreite bleibt Engpass!

Warum Peer-to-Peer?

- direkte Kommunikation nutzt den Engpass Bandbreite besser aus:



© Klaus Schild, 2004

13

Seminarthemen

Architekturen

- 29.4. P2P, Napster
- 6.5. dezentrale Architekturen: Gnutella, Freenet
- 13.5. Skalierbarkeit

Plattformen

- 27.5. JXTA: Einführung und Überblick
- 3.6. JXTA: Programmierung im Detail
- 10.6. Suche

aktuelle Anwendungen

- 17.6. File-Sharing, spontane Zusammenarbeit, ...

Soziale Netzwerke

- 24.6. elektronisch unterstützte soziale Netzwerke
- 1.7. Friendster, Tribe, ...
- 8.7. Propagierung von Vertrauen

© Klaus Schild, 2004

14

Architekturen

Napster

- zentrales Echtzeitverzeichnis
- Übertragung direkt von Nutzer zu Nutzer
- kopierte Datei steht allen zur Verfügung: Konsument wird automatisch zum Verbreiter

Gnutella

- vollständig dezentral
- P2P-Suche (Breitensuche)

Freenet

- vollständig dezentral
- P2P-Suche (Tiefensuche)
- Zwischenknoten speichern Suchergebnisse: automatische Verbreitung von Suchergebnissen

Welche dezentrale Architektur ist skalierbar?

© Klaus Schild, 2004

15

Plattformen

JXTA™

- Sammlung offener Protokolle für P2P-Systeme
- unabhängig von Anwendungen, wie File-Sharing oder Instant Messaging
- unabhängig von Plattformen und Transportprotokollen
- unabhängig von Geräten: Server, PCs, Handys und PDAs
- Java-Implementierung von SUN

© Klaus Schild, 2004

16

Anwendungen: aktivierte Ressourcen

File-Sharing (⇒ Napster):

Inhalte, Speicherkapazität

Mediendistribution (⇒ Flycode):

Speicherkapazität

spontane Zusammenarbeit (⇒ AIM, Groove):

menschliche Präsenz, Inhalte

Verteiltes Speichern (⇒ Scale Eight):

Speicherkapazität

Verteiltes Berechnen (⇒ SETI@home):

CPU-Zeit

© Klaus Schild, 2004

17

Soziale Netzwerke: Friendster & Co

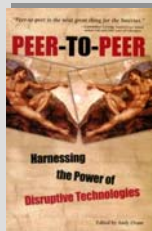
- P2P-Suche nutzt Small-World-Phänomen aus.
- Friendster nutzt dieses Phänomen auch aus: ich nenne meine Bekannte, das System macht mich mit den Bekannten meiner Bekannten bekannt...
- neue Form von Communities
- Anwendungen: Dating, Networking, Wissensmanagement, ...
- Hauptproblem: Wie wird Vertrauen propagiert?
- Implementierung mit P2P-Technologien nahe liegend

© Klaus Schild, 2004

18



- auf der Webseite des Seminars:
http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/04/S_P2PNET/index.html
- Oram (Editor), Peer-to-Peer
Harnessing the Power of
Disruptive Technologies,
Oreilly, 2001.



Architekturen

- 29.4. P2P, Napster
- 6.5. dezentrale Architekturen: Gnutella, Freenet
- 13.5. Skalierbarkeit

Plattformen

- 27.5. JXTA: Einführung und Überblick
- 3.6. JXTA: Programmierung im Detail
- 10.6. Suche

aktuelle Anwendungen

- 17.6. File-Sharing, spontane Zusammenarbeit, ...

Soziale Netzwerke

- 24.6. elektronisch unterstützte soziale Netzwerke
- 1.7. Friendster, Tribe, ...
- 8.7. Propagierung von Vertrauen