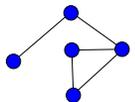


Gliederung

Peer-to-peer Systeme und Datenbanken(SS07)

- Kapitel 1: Einführung
- Kapitel 2: Beispiele
- Kapitel 3: Routing
- Kapitel 4: Schemabasierte p2p-Netzwerke
- Kapitel 5: Integrationsprobleme
 - Teil 1:
 - Teil 2:
 - Teil 3:
- Kapitel 6: Anonymität, Authentifikation
- Kapitel 7: Dienstgüte

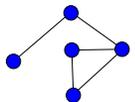
Version vom 19. April 2007



Kapitel 2

Gliederung:

- Napster (hybride Architektur)
- Gnutella (reines P2P-System)
- KaZaA (hybrid, hierarchisch)
- Freenet (Anonymität)
- Echolink (Authentifizierung, kein Download)



Beispiel 1: Napster

● Geschichte:

Mai 1999: Napster Inc. file sharing service gegründet ...

März 2001: ... erfüllt Richterspruch bezügl. Rechteverletzung

Soll bis 7 TB Daten verwaltet haben.

● Architektur:

hybrider Ansatz, Server hält Index der verfügbaren Dateien

Client muß sich registrieren und bei jeder Nutzung authentifizieren

Client darf Index durchsuchen - Information, welche anderen Clients die gesuchten Dateien haben, dann direkter Datentransfer zw. Clients.

Weitere Services integriert: chat-room, hot list, instant messaging ...

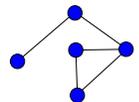
● Wertung:

- Serverausfall = Systemausfall,

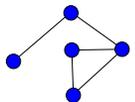
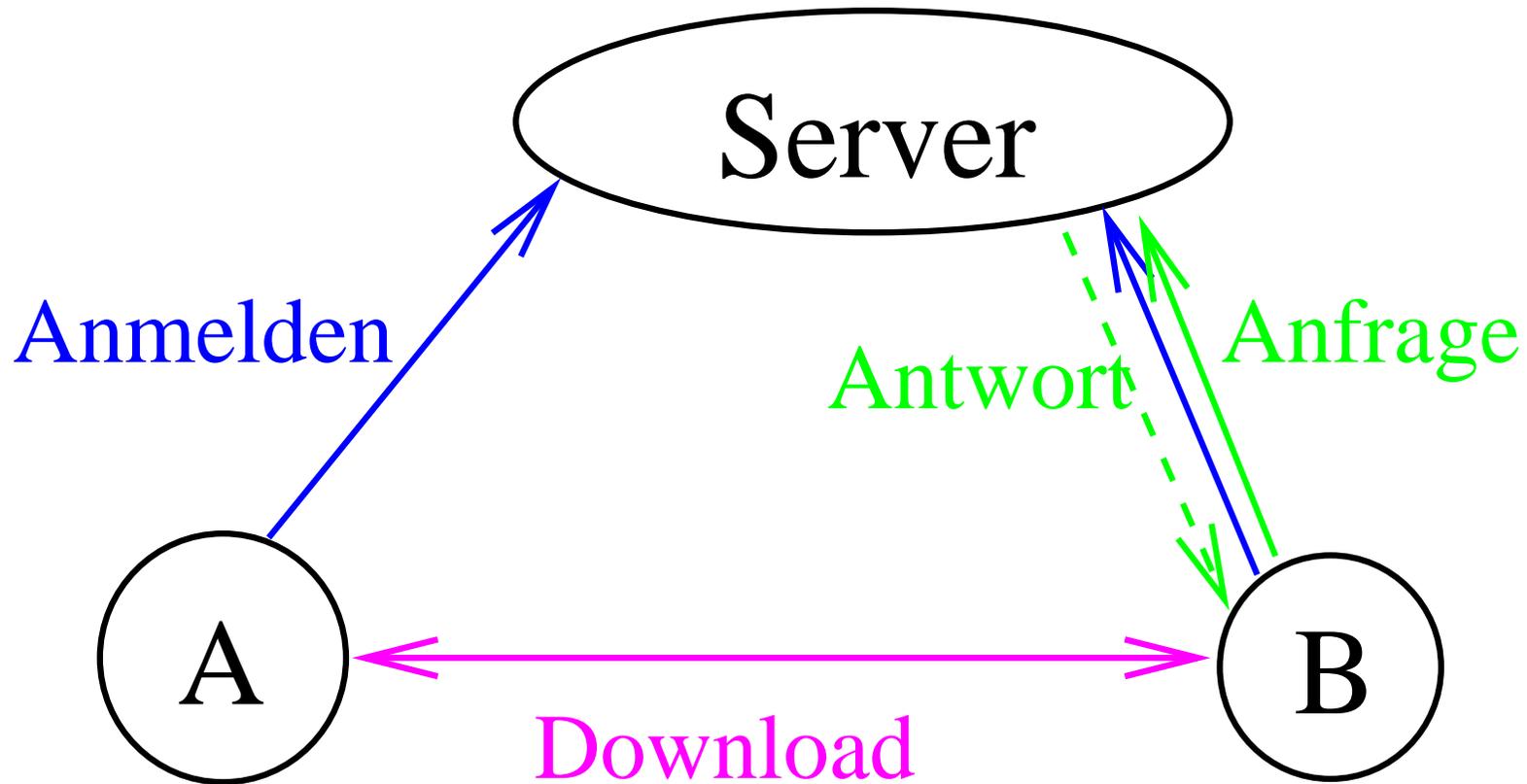
- keine Reputation der Peers,

+ Anfragen schnell, Antwort vollständig, da das P2P-Netz dem Server bekannt

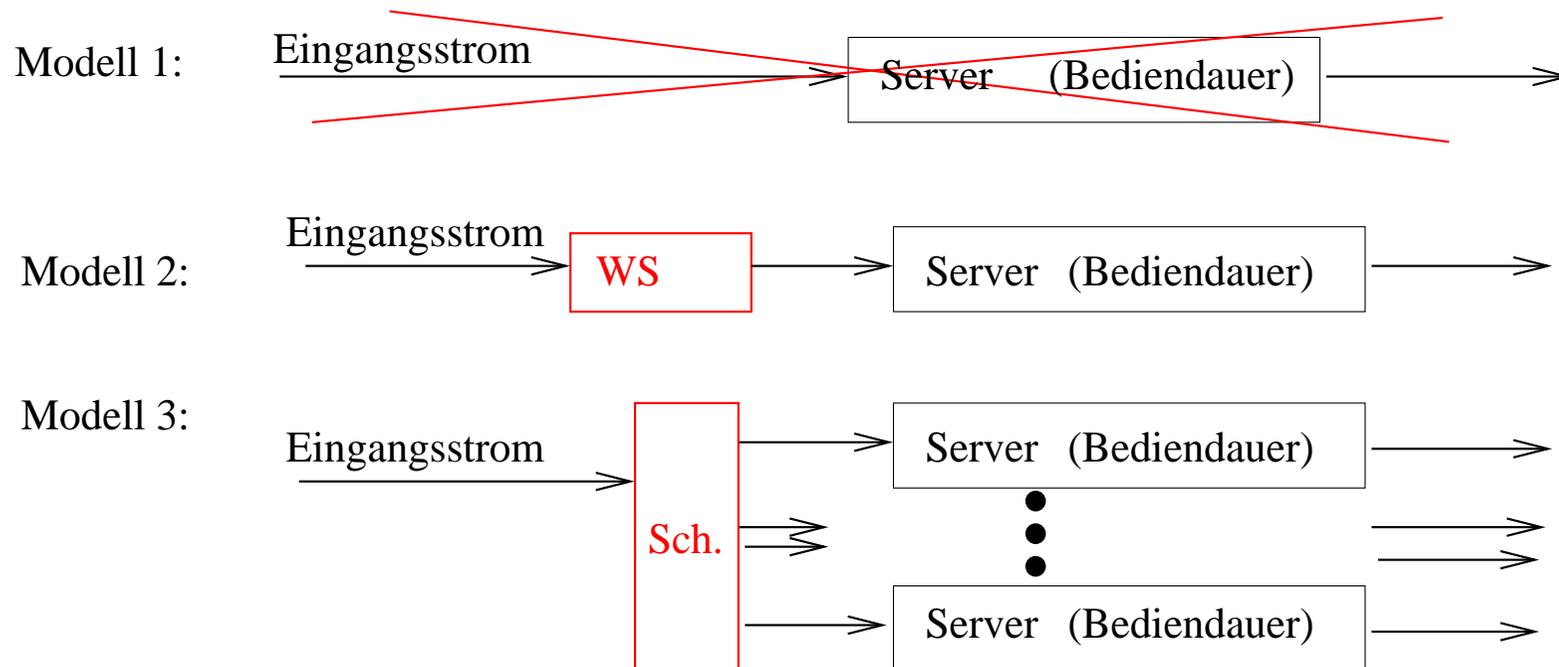
+ vom Nutzer akzeptierte Zusatzdienste



Napster - Kommunikation

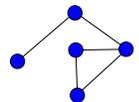


Server-Leistung



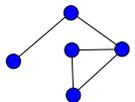
Parameter: $\lambda = 100$; 3 Sekunden mittl. Wartezeit (Mod.2,3).

- Modell 1: $\nu_B = 2,15 \times 10^8$
- Modell 2: $\nu_B \geq \lambda (= 100)$
- Modell 3: $\nu_B = 4,8$; $M = 21$, $\hat{M} = 46$



Beispiel 2: Gnutella

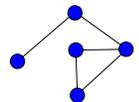
- „quick hack“, Nullsoft, Kochrezepttausch
- war unter GPL nur Stunden auf dem Nullsoft Webserver
- Protokoll im Quelltext analysiert ← Klienten von anderen Anbietern
- Kein zentraler Server, Pakete an Nachbarn weitergeleitet (typ. 4), life-time für Pakete (typ. 7), PaketID zur Schleifenerkennung. Problem des Erstkontakts durch eine feste Adresse gelöst: gnutellahosts.com:6346



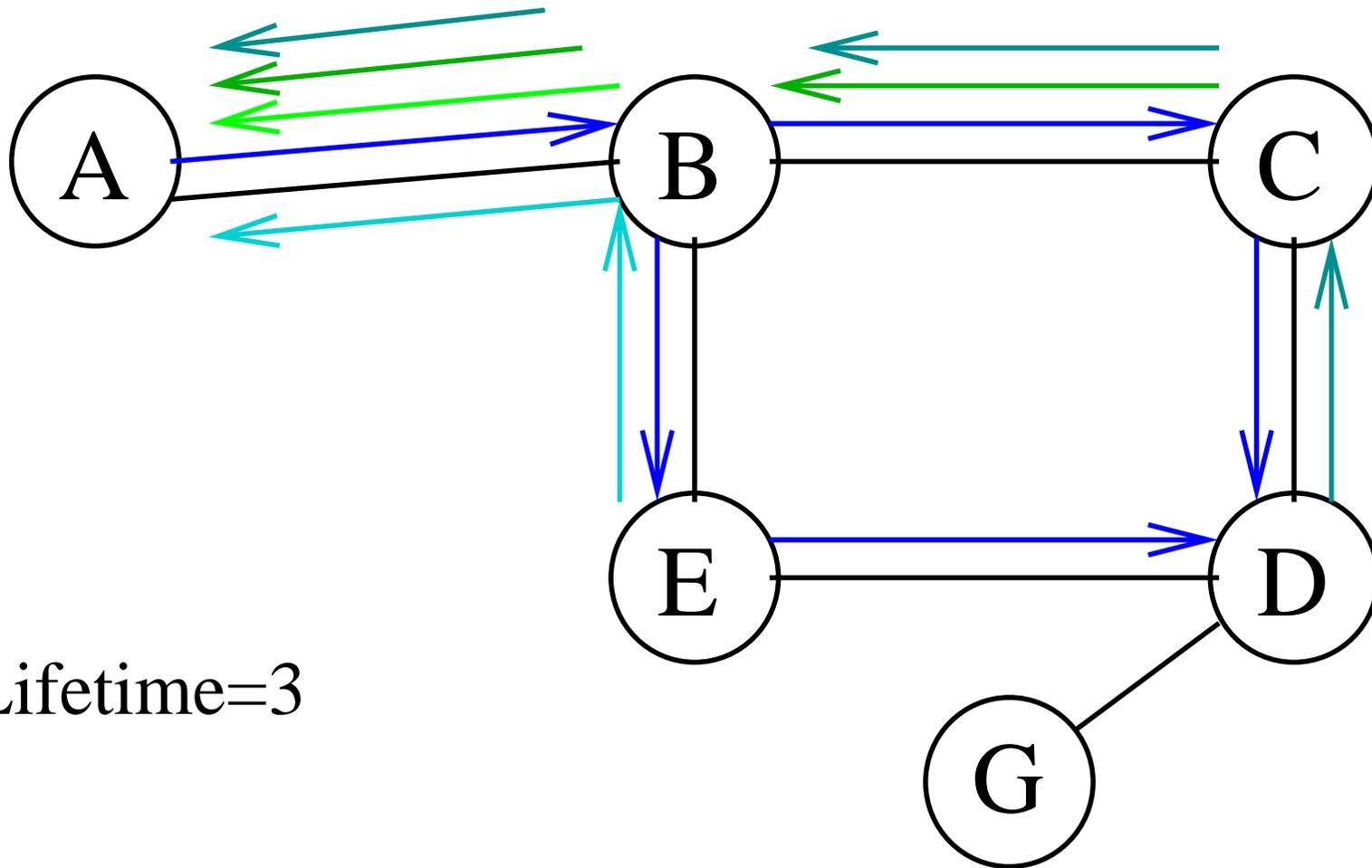
Beispiel 2: Gnutella (2)

- Gnutella-Protokoll - Merkmale der P2P-Protokolle
Nachrichtentypen:

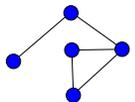
Type	Funktion	Inhalt
ping	Kontaktaufnahme	–
pong	Antwort auf ping	IP-Adresse und Portnr. Zahl der Dateien und deren Volumen in KB
query	Suchanfrage	Suchkriterium; minimale Bandbreite
query-hit	positive Antwort auf Suchanfrage	IP-Adresse, Portnr., vorhandene Bandbreite Anzahl der Treffer, Treffermenge
push	Download-Anforderung eines Peers hinter Firewall	Peer-Id; Index des Datei; IP-Adr, u. Portnr., an den die Daten zu senden sind



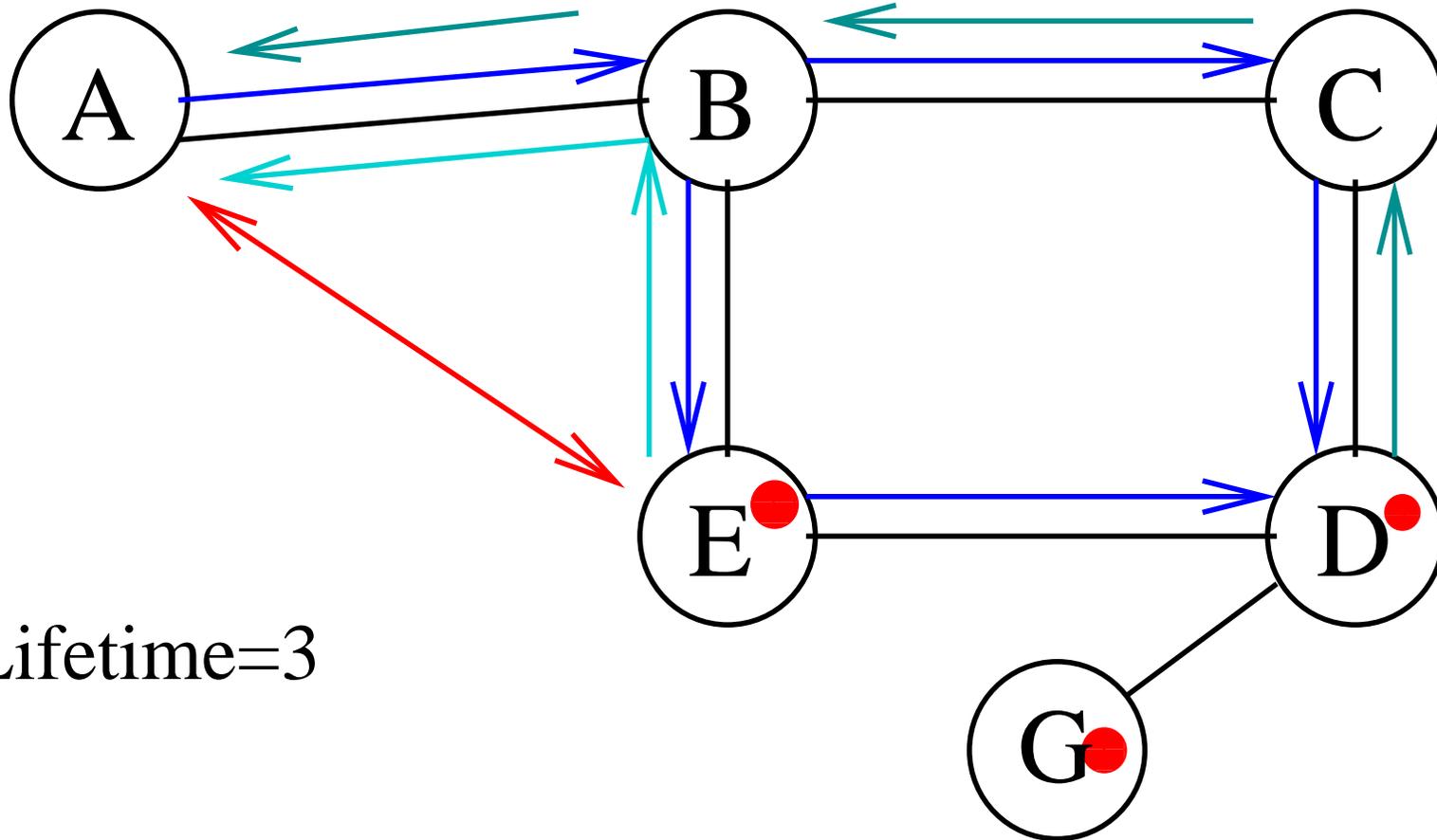
Gnutella - Anmelden



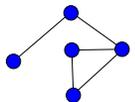
Lifetime=3



Gnutella - Suche

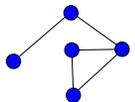


Lifetime=3



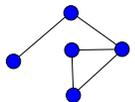
Gnutella - Bandbreite

- Messungen im Gnutella-Netz:
 - Anfrage ca. 560 bit lang (Brutto) - Last: 25% Anfragen, 50% ping, 25% Rest
 - Verbindungen: durchschnittlich 3 Peers kontaktiert (Verkehrsanalyse?)
- Schlußfolgerung
 - $10 \text{ Anfragen/Sekunde} * 560 \text{ bit/Anfrage} * \frac{100}{25} * 3 \text{ (Verbindungen)} = 67.200 \text{ bit/Sekunde}$
 - ISDN: 64 Kbit/Sekunde \mapsto Bandbreitenlimit
- Caching verringert Netzlast, kostet bei Zwischenstationen Speicher



Gnutella - Zusammenfassung

- + : reines P2P-System, hohe Trefferrate, hohe Fehlertoleranz, gute Anpassung bei Fluktuation
- - hohe Netzlast, z.B. eine Anfrage bei C Nachbarn mit TTL=7 - bis $2 \times \sum_{i=0}^{TTL} C \times (C - 1)^i = 26240$ möglich
 - keine Abschätzung von Ausführungszeit, i.a. nicht das ganze Netz erreicht. Netztopologie unbekannt, kann nicht genutzt werden.
 - Keine Bewertung, Free-Riding.

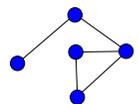


Beispiel 3: KaZaA

KaZaA-Homepage: www.kazaa.com

Hybrides P2P-Netzwerk - hierarchisch

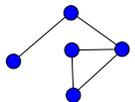
- Reaktion auf Bandbreiten-Engpaß bei Gnutella, Rechner mit breitbandiger Anbindung (und großer Rechenleistung) \mapsto Superpeers (verwalten Dateiindizes, bearbeiten Suchanfragen, Weiterleitung an andere Superpeers)
- Superpeers kommunizieren miteinander, Peers melden sich bei einem Superpeer an und stellen bei ihm Suchanfragen. Keine Anonymität vorgesehen.
- Direkter Download Peer \leftrightarrow Peer.
- Sehr gute Skalierbarkeit.
- Version 2: Integrity Rating - Qualitätsbewertung der Dateien durch Nutzer, participation level - up-/download-Verhalten der Nutzer bewertet, Nutzer mit hohem Level bei Download bevorzugt.



Beispiel 4: Freenet

(<http://freenet.sourceforge.net/>)

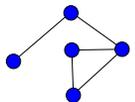
- Ziel: anonymer Tausch von Informationen
 - Quelle und Ziel von Daten kann nicht festgestellt werden.
 - Daten werden verschlüsselt gespeichert und „durchgeroutet“, es ist für jeden Peer aufwändig festzustellen, welche Daten er hat.
 - Rechtliche Würdigung schwierig
- Unterstützt dynamische Replikation, kein Löschen möglich
- Adressierung der Daten unabhängig von der Lokalisierung, Sucher muß ein Kennwort kennen, welches der „Autor“ einer Datei vergibt und veröffentlicht. Daraus wird eine ID berechnet.



Freenet (2)

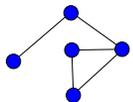
Ziele des Projekts:

- Anonymität von Autoren und Lesern
- Abstreitbarkeit für Serverbetreiber
- Verteilte Datenhaltung mit Datenreplikation
Jeder Peer speichert nicht seine Daten, sondern die, die er auf Anfragen empfangen hat. Dabei spezialisiert sich ein Knoten auf einen Teil des Adressraumes
- Adaptives, kooperatives Routing
Service: Jeder Peer wird bei einer Anfrage seinerseits versuchen, die Daten zu erhalten und dem Anfrager zu senden. Jeder Peer kennt einige wenige andere Peers und deren Spezialisierungen. Kann er eine Anfrage nicht selbst beantworten, leitet er sie den Peer unter seinen „Bekanntem“ weiter, dessen Spezialisierung der gesuchten ID am ähnlichsten ist.



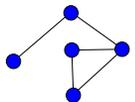
Freenet (3) - Technische Fragen

- open source Projekt (JAVA)
- Suchmechanismen werden verbessert - adaptives Routen
- Dateiteilung und Fehlerkorrektur
- Details zum Routing und Verschlüsselung später
- Installation (unter UNIX): Entpacken, Shellskript starten, Fragen zur Installation beantworten, ggf. default-Werte übernehmen.



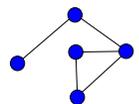
Freenet (4) Bewertung

- Unterstützt Anonymität von Autoren und Lesern
- + Reine P2P-Architektur (fehlertolerant, robust)
 - + Replication verbessert Verfügbarkeit der Daten
 - + Adaptives Routing verringert Netzlast
 - + Gute Anpassung bei Fluktuation
 - + Routing verhindert Free-Riding
- - Netztopologie unbekannt und ungenutzt
 - Keine Ausführungszeit von Anfragen garantiert
 - Keine Erfolgsgarantie auch bei Existierenden Daten.
 - Keine Bewertung der Peers.



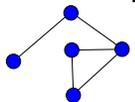
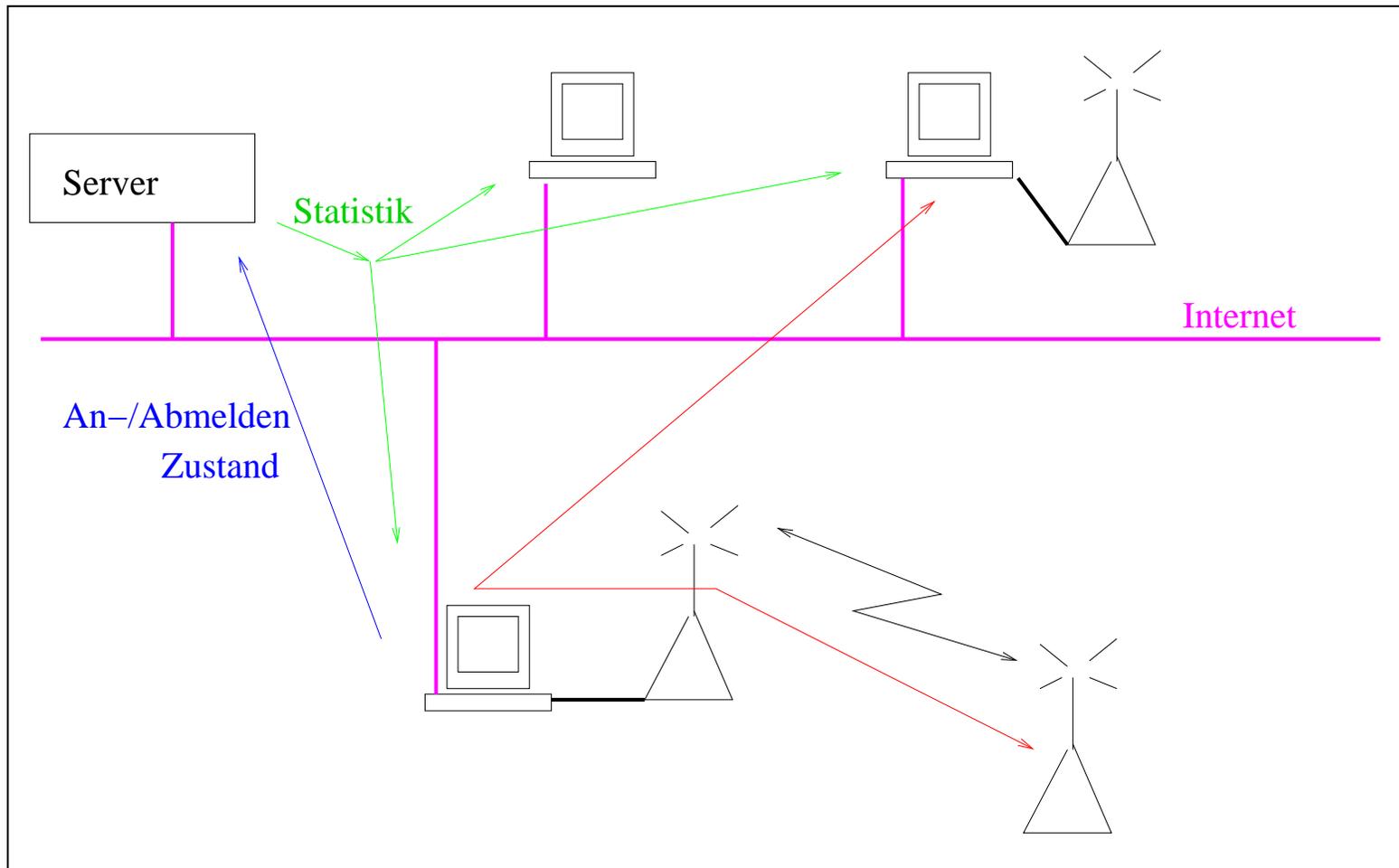
Beispiel 5: Echolink

- Ziel: Steuerung/Benutzung einer Funkstation über das Internet,
Copyright: Synergetics, LLC
URL: www.echolink.org
- Strenger rechtlicher Rahmen: Authentifizierung notwendig
- Hybride Architektur
 - Server: Registrierung der Peers, Anmeldung/Authentifikation, aktuelle Teilnehmerstatistik
 - Peer: direkte Verbindung untereinander
- Echtzeitanwendung: Sprachübertragung



Echolink II

Systemübersicht



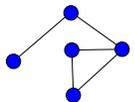
Vergleich P2P - andere netzbasierte Kommunikationsformen

Ereignis-basierte Systeme

symmetrisch, dynamisch, keine feste Bindung: jeder Teilnehmer erzeugt und erhält *events*, er erklärt Interesse an best. Events und wird bei Auftreten benachrichtigt (passiv - P2P aktive Anfragen.)

Nachrichtenverteilung

(z.B. NEWS), typ. Client-Server-Architektur, Nachrichten nach Kanälen geordnet und verteilt, Client meldet sich für Kanal an, auch Bezahlssysteme bekannt



Vergleich: P2P - andere netzbasierte Kommunikationsformen (2)

Mobile Agenten

erfüllen eine Aufgabe.

bewegen sich autonom in den Knoten eines Netzwerk nach Regeln, die für jeden MA andere sein können, Bewegung des A. hat Ähnlichkeiten zur Weiterleitung von Anfragen, diese aber systemweit (ähnlich - gleiche Routingstrategie).

MA deutlich flexibler, können mit anderen MA interagieren.

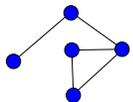
Pflege von mobilem Code deutlich schwieriger,

Schutz vor mobilen Viren - ?

Schutz des MA vor Viren - ?.

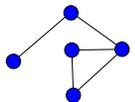
Föderierte Datenbanken

Von allen Formen der MRDB am ehesten mit P2P-Anwendungen vergleichbar, aber i.A. typ. Client-Server-Architektur, die beteiligten DBVS sind fest und i.A. verfügbar. Datenbanken u. Datenverteilung i.A. bekannt, 2PC-Protokoll, Autonomieeinschränkung.



Free-riding

- Untersuchung des Problems bei Gnutella
Quelle: Eytan Adar and Bernardo A. Huberman. *Free Riding on Gnutella*. Tech. report. Xerox PARC. Sept.9,2000
24 Stunden Verkehr beobachtet:
 - 70% der Nutzer stellen nichts zum Download bereit
 - 50% der Antworten kommen von 1% der Hosts
- Free-riding ist ein soziales, kein technisches Problem
- Mathematisch beweisbar: Free-riding als optimale Lösung eines entsprechenden Spiels.
- 3 Hypothesen bestätigt
 - H1: Ein signifikanter Anteil der Nutzer sind Free-Rider.
 - H2: Free-Rider sind gleichverteilt
 - H3: Häufig werden Daten bereitgestellt, die niemand abrufen.

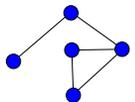


Free-riding (2): Statistik 1

Hypothese 1: Ein signifikanter Anteil der Nutzer sind Free-Rider.
(Quelle: ebenda)

Beobachtet wurden 33335 Hosts

- 22.094 (66%) stellen nichts bereit
- 24.347 (73%) stellen 10 oder weniger Dateien bereit
- 1% (333) der Hosts halten 37% (1.142.645) aller angebotenen D. bereit
- 5% (1667) der Hosts halten 70% (2.166.043) bereit
- 10% (3334) der Hosts halten 87% (2.692.082) bereit



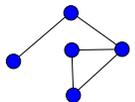
Free-riding (3): Statistik 2

Hypothese 3: Viele stellen Daten bereit, die niemand abrufen. (Quelle ebenda)

Untersucht wurden 11585 Hosts:

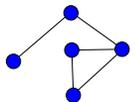
- 1% liefert fast 47% aller Antwortnachrichten - 25% liefern 98% aller Antworten - 63%

(7.349) gaben niemals eine Antwort auf eine Anfrage



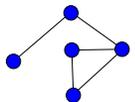
Free-riding (4)

- Free-Rider verursachen Netzlast, Einsinken der Performance
- Fragen:
 - Kann mit einem großen Anteil von Free-Ridern ein Netz lahmgelegt werden?
 - Kann ein (kleiner) Anteil von Free-Ridern die Netztopologie verbessern?
- Praxis und Spieltheorie: Durch Einführen einer Bewertung des Verhaltens der Peers ändert sich deren Zielfunktion und damit das für sie optimale Verhalten



P2P - Datenbanken

- P2P derzeit meist auf Dateiniveau, keine strukturierten Ressourcen. C und D von ACID in P2P-Systemen nicht gewährleistet, keine Transaktionsunterstützung.
- Literatur: Risse, T. und Knesevic, P. : *A Peer-to-Peer XML Database*. Fraunhofer IPSI.2003
P2P XML Datenbank, XML-Dokumente verteilt gespeichert, jeder Knoten nur Teildokument, verteilte Hashtabellen zur Repräsentation der Baumstruktur des XML-Dokuments.



Zusammenfassung:Hauptanforderungen

Bei der Entwicklung eines P2P-Systems sind folgende Gesichtspunkte wichtig:

- Welche Anzahl von Nachrichten tritt auf? Welche Anforderungen werden an den Internetzugang gestellt (z.B. Bandbreite)? Wie skaliert das Netz?
- Wie werden Ressourcen gefunden?
- Wie werden Updates (durch das Netz) propagiert?
- Wie gut toleriert das P2P-System Fehler und dynamische Veränderungen im eigenen Netz und im darunterliegenden Internet?
- Kann das P2P-Netz durch Angriffe außer Funktion gesetzt werden?
- Wahrung der Anonymität
- Welche Bewertung des Verhaltens der Peers existiert? (Objektivität, Manipulierbarkeit).

