Informatik III

Wunschvorlesung Theorie der Peer-to-Peer-Netzwerke

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer

Wintersemester 2006/07

29. Vorlesung

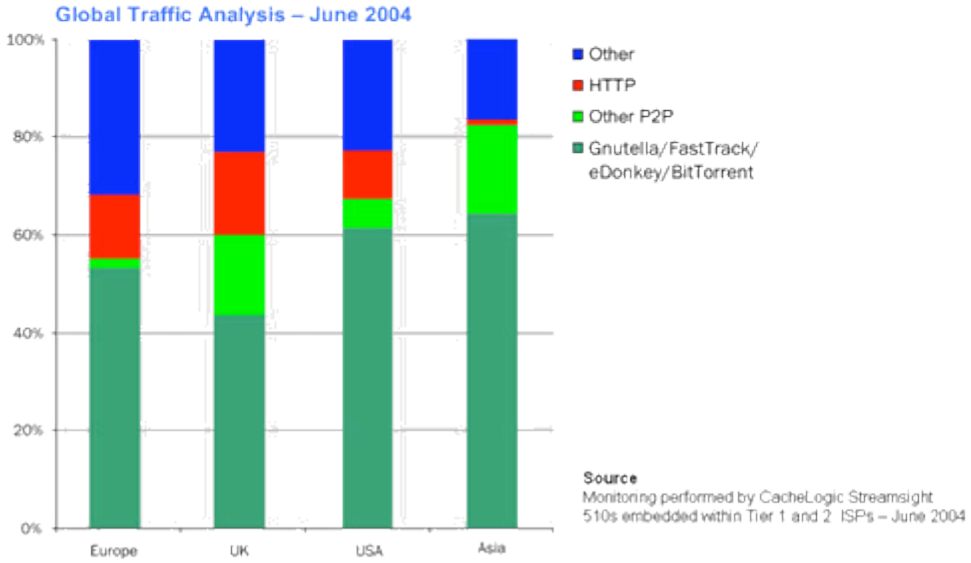
16.02.2007

P2P-Netzwerke 2005

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

➢ **Juni 2004**

– Quelle: CacheLogic



➢ **2005**

– Über 8 Mio. aktive Teilnehmer an Peer-to-Peer-Netzwerken zu jeder Zeit
– 10 PetaByte an Daten zu jeder Zeit
– Mehr als die Hälfte des gesamten Internet-Traffic ist Peer-to-Peer
– Mehrere Peer-to-Peer-Netzwerke durch Gerichtsprozesse stillgelegt
– Tausende von Einzelklagen gegen Peer-to-Peer-Nutzer wegen Verletzung des Urheberschutzes

Meilensteine Praxis

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Napster (1999)**
  - seit 1999, bis 2000 (Gerichtsurteil)
- **Gnutella (2000)**
  - Neue Version (Gnutella 2) in 2002
- **Edonkey (2000)**
  - Später: **Overnet** unter Verwendung von Kademlia
- **FreeNet (2000)**
  - Anonymisierung der Teilnehmer
- **JXTA (2001)**
  - Open Source Peer-to-Peer-Netzwerk-Plattform
- **FastTrack (2001)**
  - bekannt durch KaZaa, Morpheus, Grokster
- **Bittorrent (2001)**
  - Nur Download-System, keine Suche
- **...**

Meilensteine Theorie

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Distributed Hash-Tables (DHT) (1997)**
  - Urspr. für Lastverteilung zwischen Web-Servern
- **CAN (2001)**
  - Effiziente verteilte DHT-Datenstruktur für P2P-Netzwerke
- **Chord (2001)**
  - Effiziente verteilte P2P-Datenstruktur mit logarithmischer Suchzeit
- **Pastry/Tapestry (2001)**
  - Effiziente verteilte P2P-Datenstruktur aufbauend auf Routing von Plaxton
- **Kademlia (2002)**
  - P2P-Lookup basierend auf XOr-Metrik
- **Viele weitere interessante Netzwerke**
  - Viceroy, Distance-Halving, Koorde, Skip-Net, P-Grid, ...

Was ist ein P2P-Netzwerk?

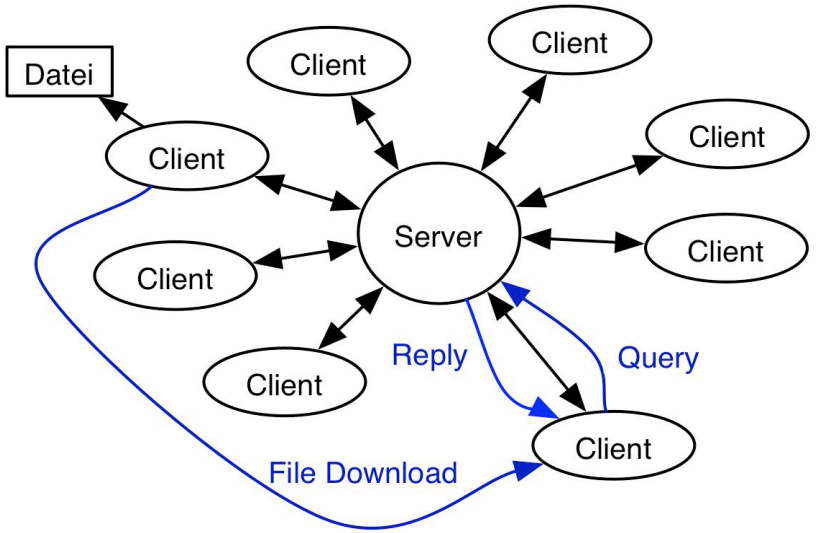
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Was ist ein Peer-to-Peer-Netzwerk nicht?**
  - Ein Peer-to-Peer-Netzwerk ist kein Client-Server-Netzwerk!
- **Ethymologie: Peer**
  - heißt Gleicher, Ebenbürtiger, von lat. par
- **Definition**
  - *Peer-to-Peer*
    - bezeichnet eine Beziehung zwischen gleichwertigen Partnern
  - *P2P*
    - Internet-Slang für Peer-to-Peer
  - Ein *Peer-to-Peer-Netzwerk* ist ein
    - Kommunikationsnetzwerk zwischen Rechnern im Internet
    - in dem es keine zentrale Steuerung gibt
    - und keine zuverlässigen Partner.
- **Beobachtung**
  - Das Internet ist (eigentlich auch) ein Peer-to-Peer-Netzwerk
  - Definitionen zu ungenau

Wie funktioniert Napster?

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

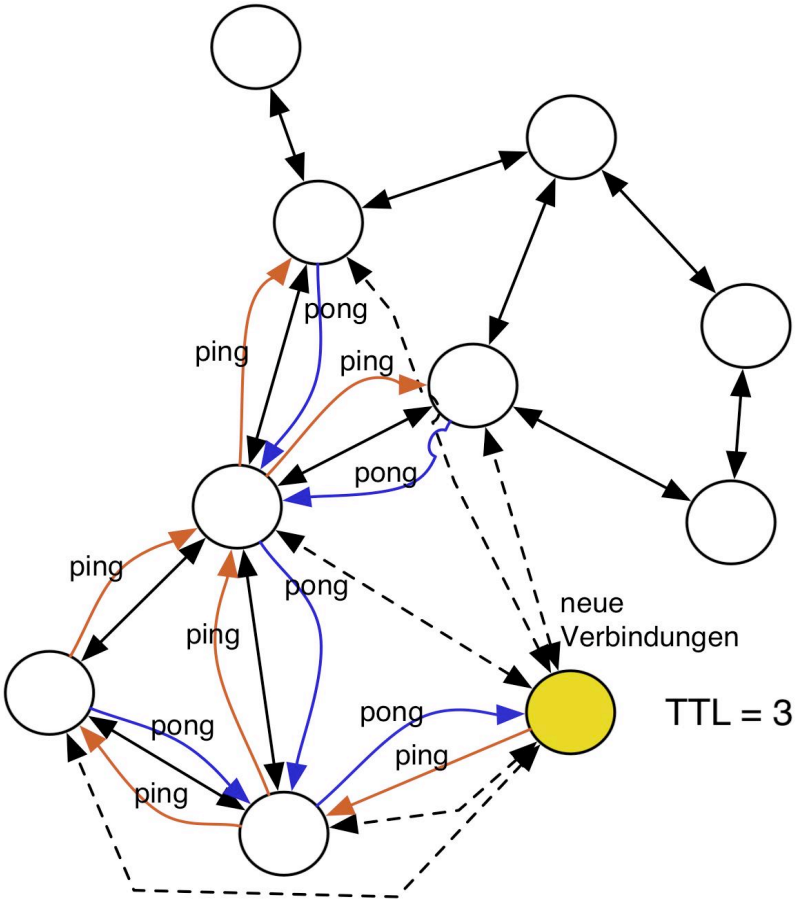
- **Client-Server-Struktur**
- **Server unterhält**
  - Index mit Meta-Daten
    - Dateiname, Datum, etc
  - Tabelle der Verbindungen der teilnehmenden Clients
  - Tabelle aller Dateien der teilnehmenden Clients
- **Query**
  - Client fragt nach Dateinamen
  - Server sucht nach passenden Teilnehmern
  - Server antwortet, wer die Datei besitzt
  - Anfrage-Client lädt Datei von datei-besitzenden Client herunter



Gnutella - Originalversion - Anbindung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Nachbarschaftslisten**
  - Gnutella verbindet direkt mit anderen Clients
  - Beim Download wird eine Liste von Clients mitgeliefert
  - Diese werden ausprobiert bis ein Aktiver sich meldet
  - Ein aktiver Client gibt dann seine Nachbarschaftsliste weiter
  - Nachbarschaftslisten werden imme weiter verlängert und gespeichert
  - Die Anzahl aktiver Nachbarn ist beschränkt (typisch auf fünf)

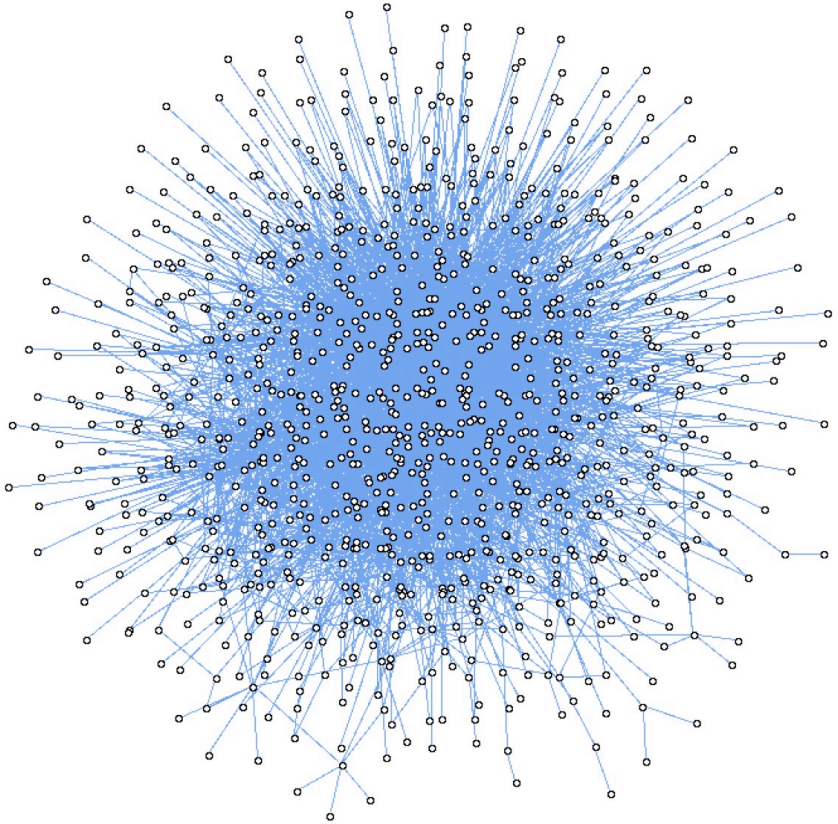


Gnutella - Originalversion - Anbindung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Protokoll**
  - **`Ping`**
    - Teilnehmeranfrage
    - werden weiter gereicht gemäß TTL-Feld (time to live)
  - **`Pong`**
    - Reaktion auf Ping
    - Werden auf dem Anfragepfad zurückgereicht
    - IP und Port des angefragten Teilnehmers
    - Anzahl und Größe zur Verfügung gestellter Dateien

- **Graphstruktur**
  - entsteht durch zufälligen Prozess
  - unterliegt Pareto-Verteilung
  - entsteht unkontrolliert

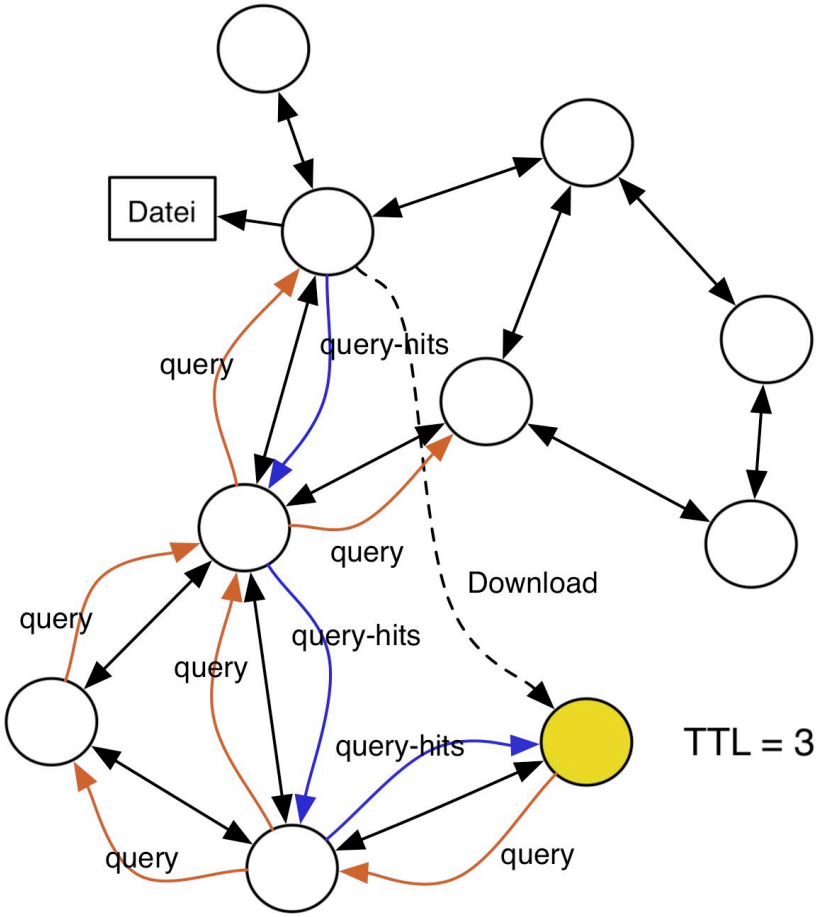


Gnutella Schnappschuss im Jahr 2000

Gnutella - Originalversion - Anfrage

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Dateianfrage**
  - wird an alle Nachbarn geschickt
  - diese senden sie an ihre Nachbarn
  - bis zu einer vorgegebenen Anzahl von Hops
    - TTL-Feld (time to live)
- **Protokoll**
  - **`Query`**
    - Anfrage nach Datei wird bis zu TTL-hops weitergereicht
  - **`Query-hits`**
    - Antwort auf umgekehrten Pfad
- **Wenn Datei gefunden wurde, direkter Download**

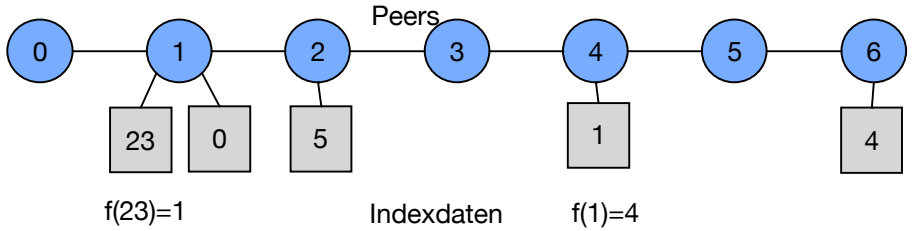


Distributed Hash-Table (DHT)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

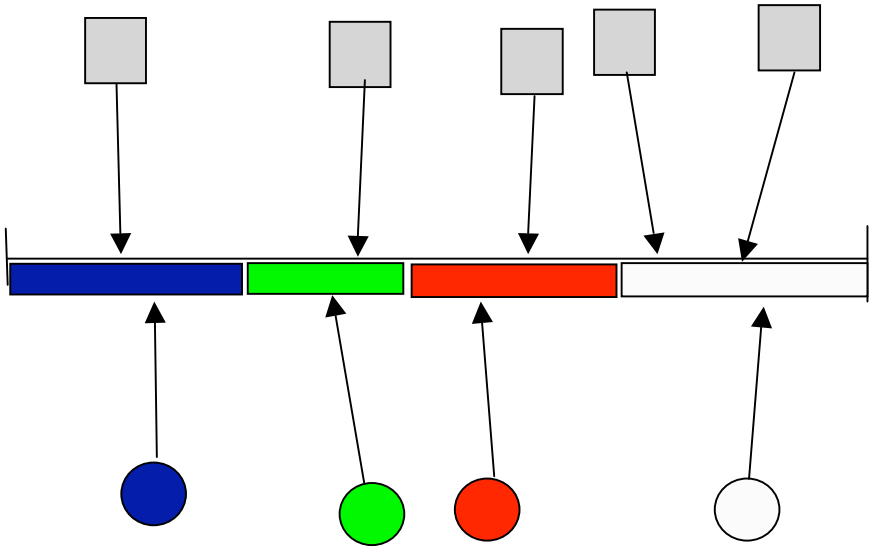
Hash-Tabellen

- **Vorteile**
  - Suche einfach
- **Nachteile**
  - Ein neuer Peer verursacht neue Wahl der Hash-Funktion
  - Lange Wege



Distributed Hash-Table

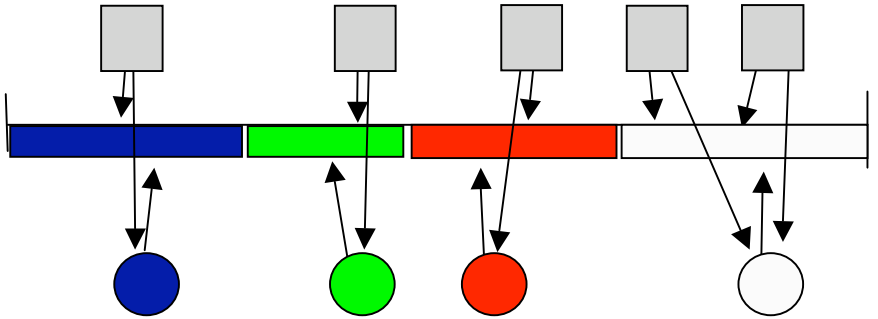
- **Peers werden an eine Stelle ge“hash“t und erhalten Bereiche des Wertebereichs der Hashfunktion zugeteilt**
- **Daten werden auch ge“hash“t**
  - Je nach Bereich den Peers zugeordnet



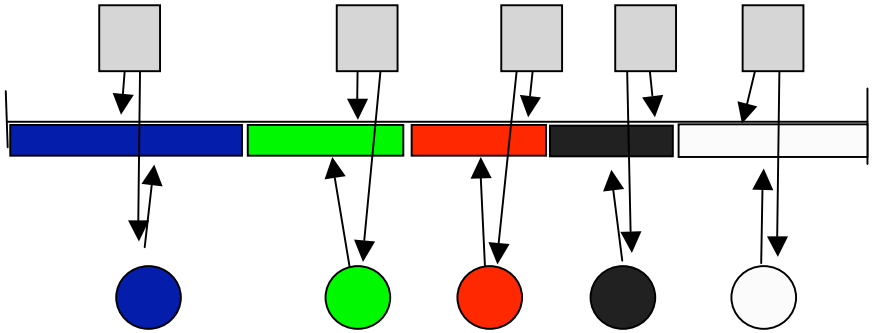
Einfügen in die Distributed Hash-Table (DHT)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

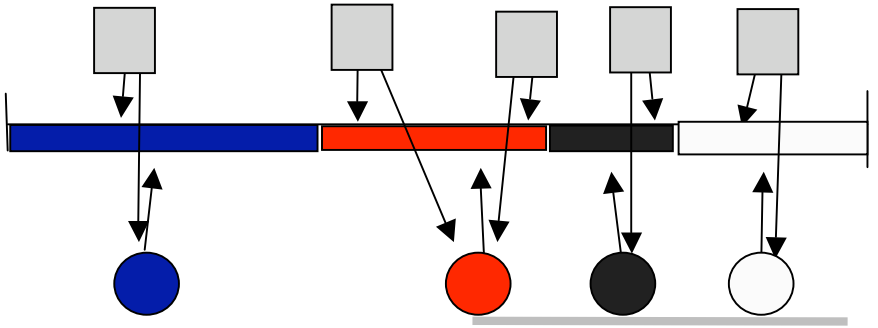
- **Distributed Hash-Table**
  - Peers werden an eine Stelle ge“hash“t
  - Dokumente ebenso
  - Jeder ist für einen Bereich verantwortlich



- **Kommt ein neuer Knoten hinzu**
  - müssen die Nachbarn teilen



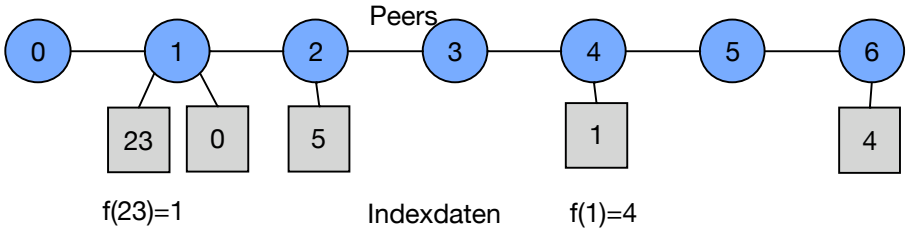
- **Verlässt ein Knoten das Netzwerk**
  - übernehmen die Nachbarn sein Gebiet



Eigenschaften DHT

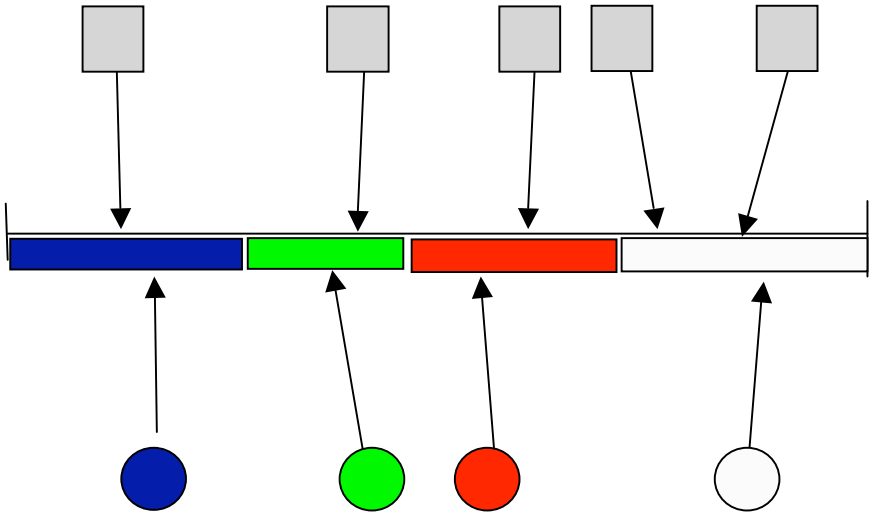
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Vorteile**
  - Jedes Datum kann einem bestimmten Peer zugewiesen werden
  - Einfügen und Entfernen von Peers erzeugt nur Veränderungen in den benachbarten Peers



- **DHTs werden von vielen P2P-Netzwerken benutzt**

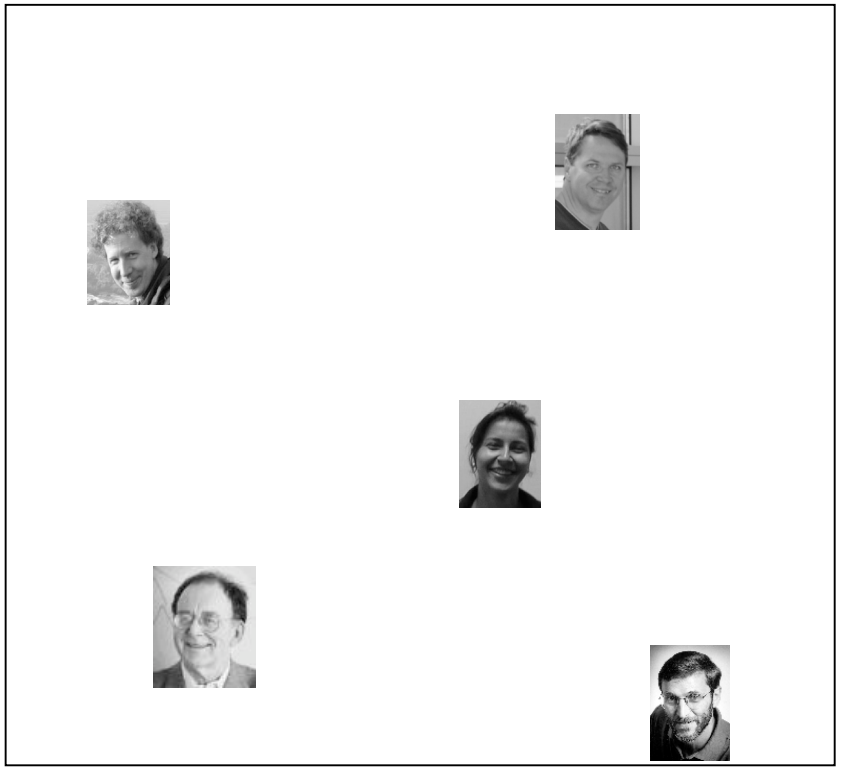
- **Noch zu klären:**
  - Die Verbindungsstruktur



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

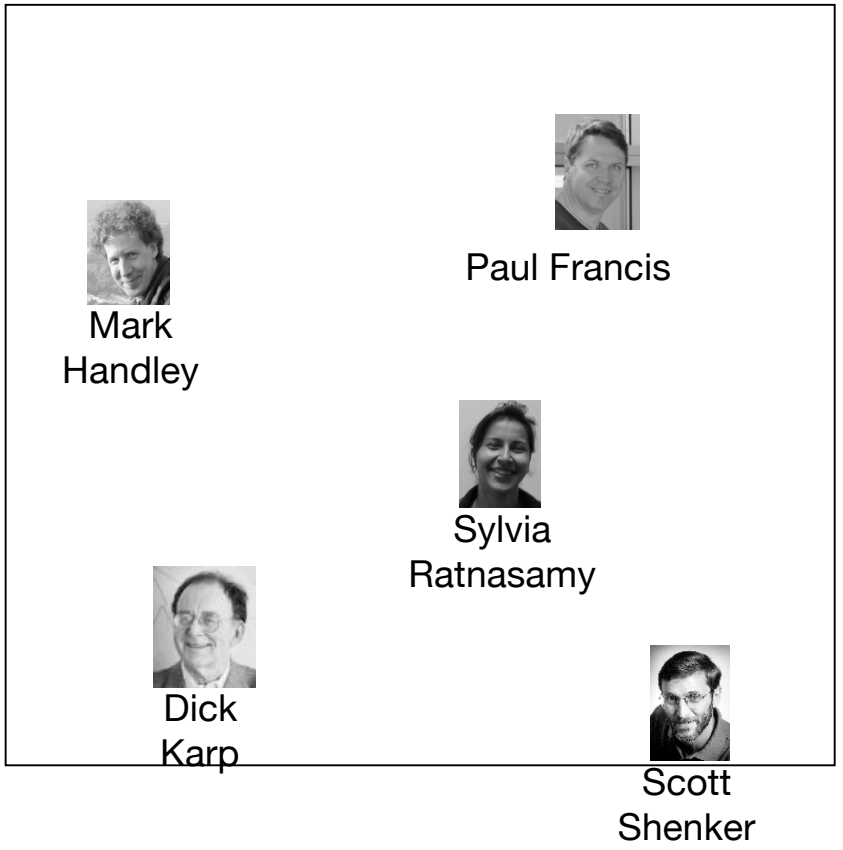
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**



A Scalable Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

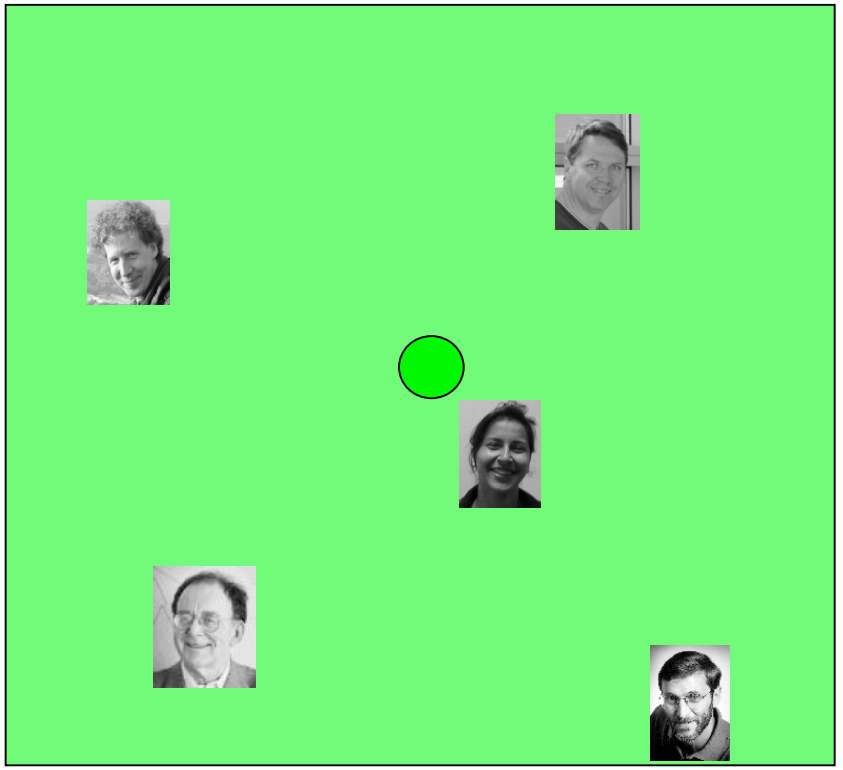
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

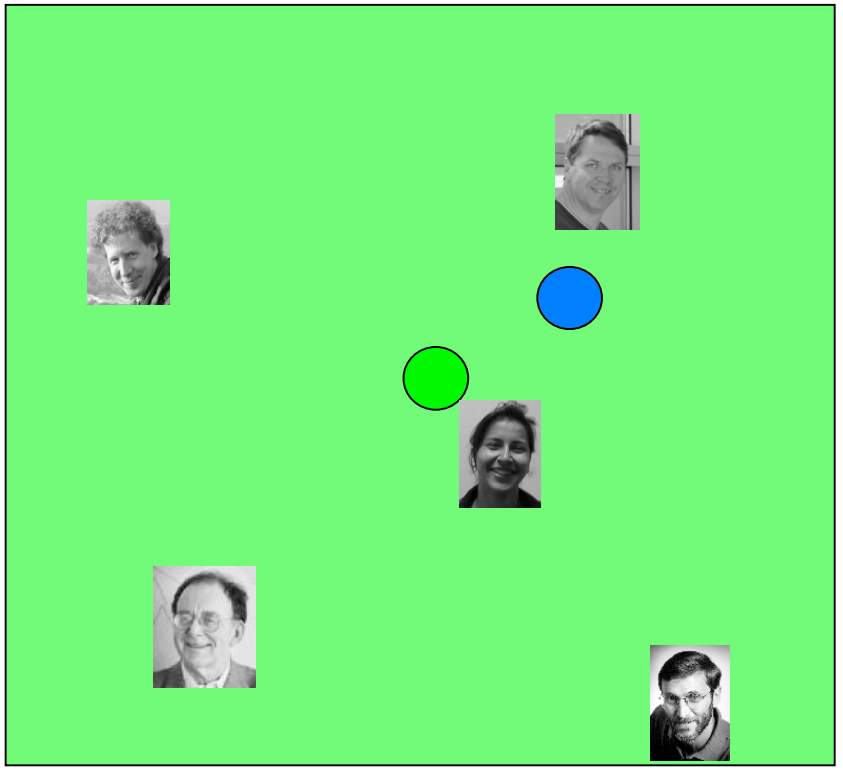
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

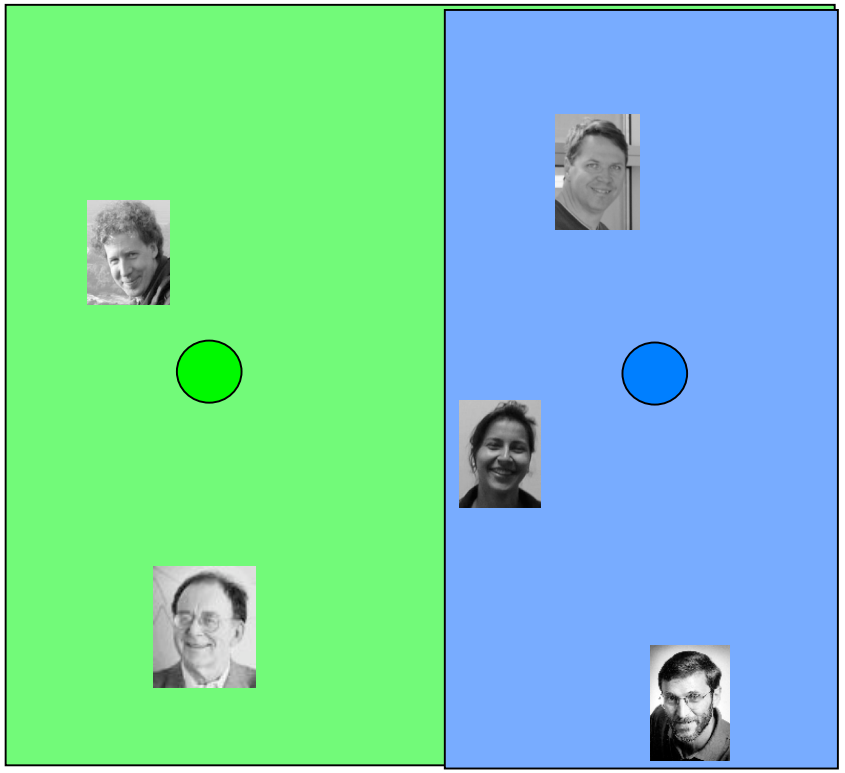
- Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet
- Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer
- Der Besitzer einer Fläche speichert alle Einträge in der Fläche
- Ein Peer wählt einen zufälligen Punkt in der Ebene



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

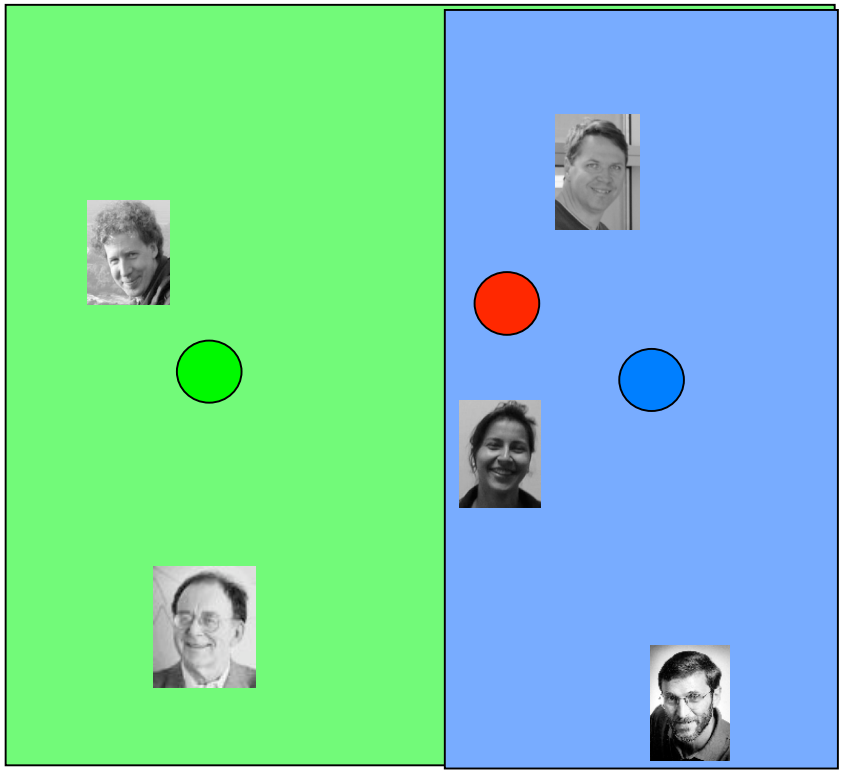
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

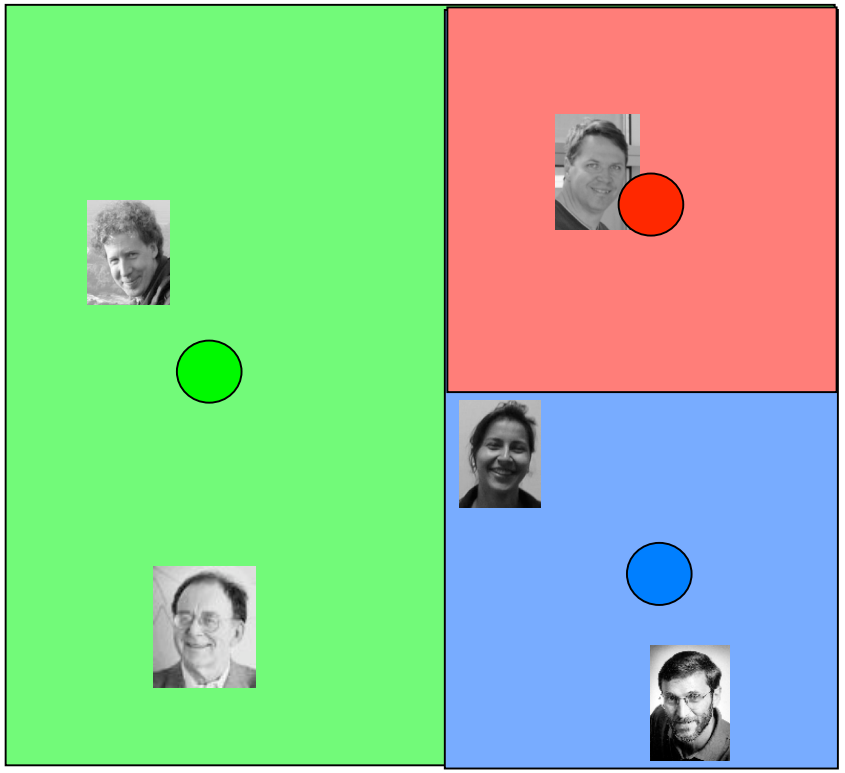
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

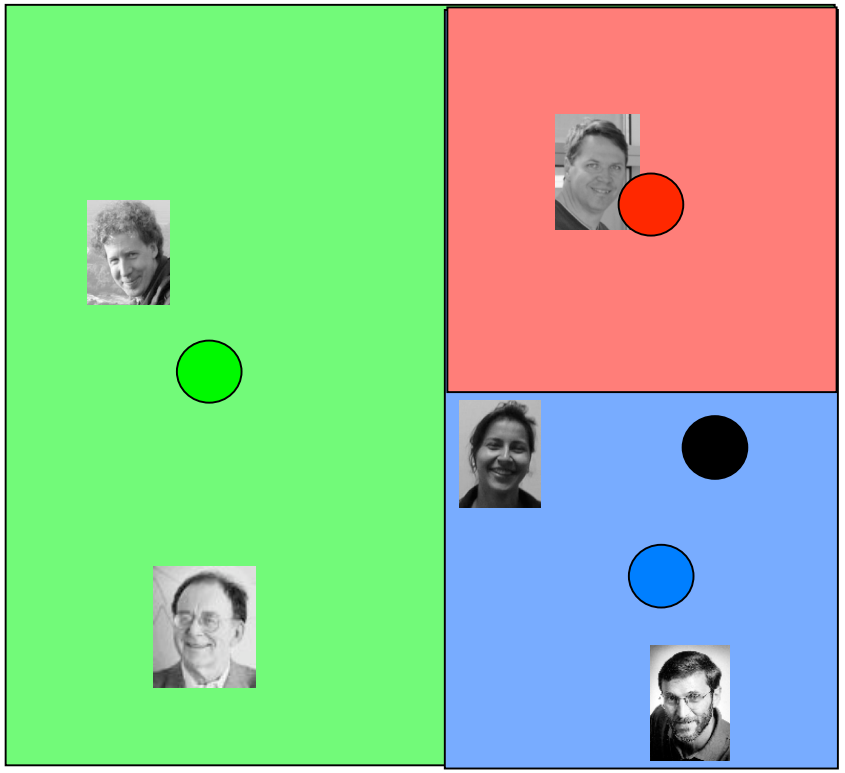
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

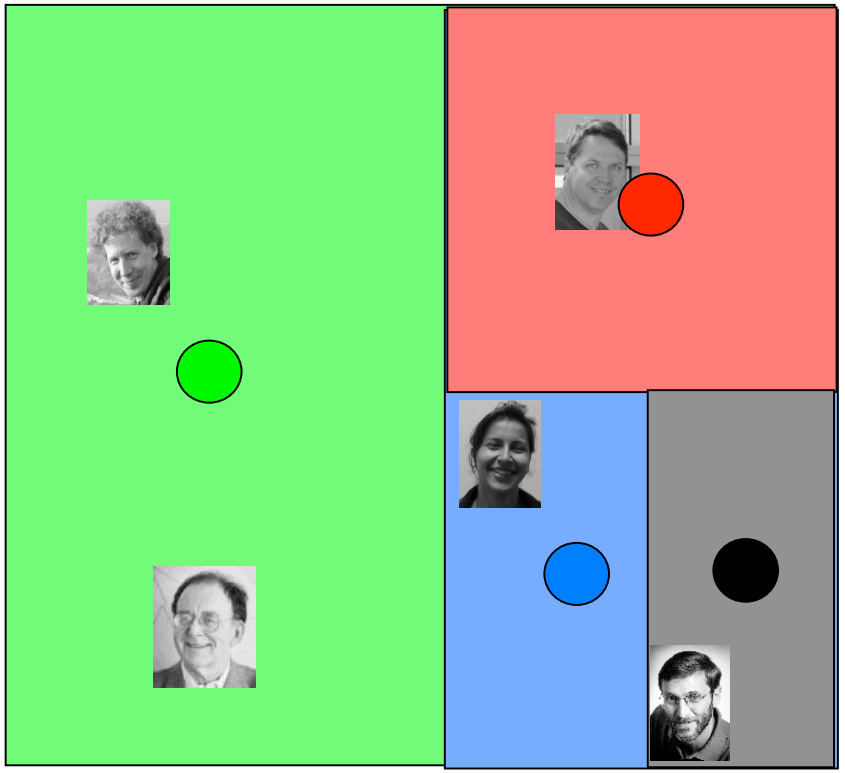
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

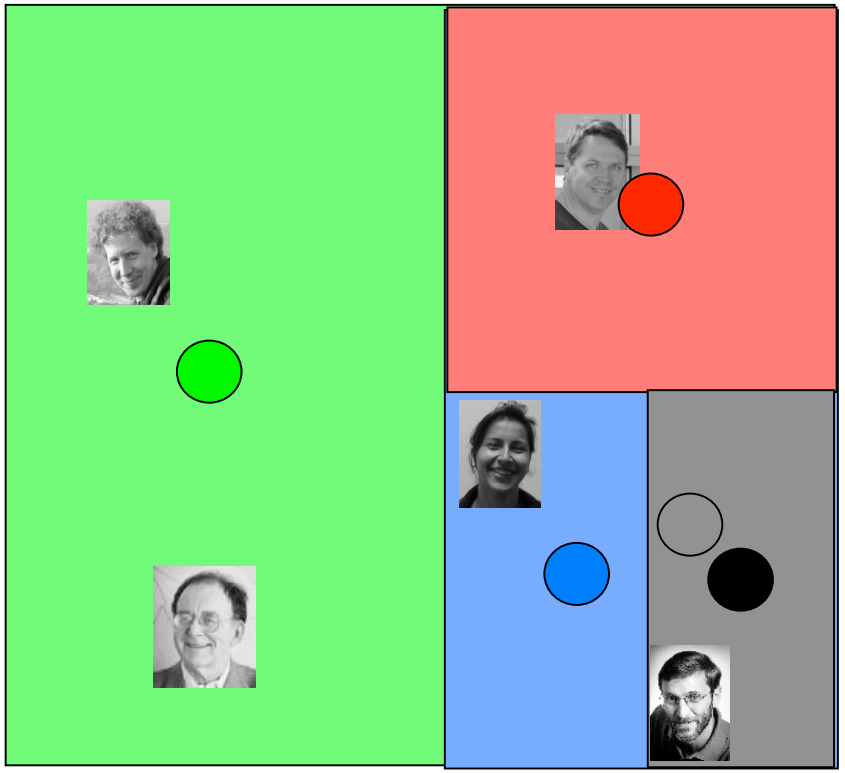
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

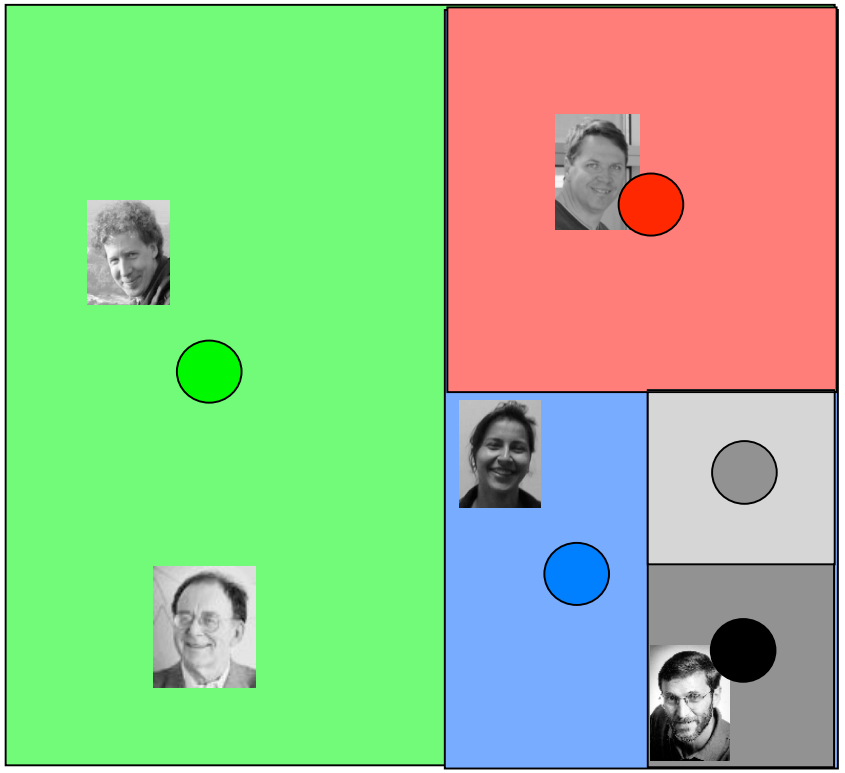
- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



Content Addressable Network (CAN)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Dateien werden in durch (zweiwertige)-Hash-Funktion in das Quadrat abgebildet**
- **Am Anfang ist ein leeres Quadrat mit nur einem Peer als Besitzer**
- **Der Besitzer einer Fläche speichert alle Einträge in der Fläche**
- **Ein Peer wählt einen zufälligen Punkt in der Ebene**
  - Der Besitzer des entsprechenden Quadrats teilt seine Fläche und
  - übergibt die Hälfte dem neuen Peer



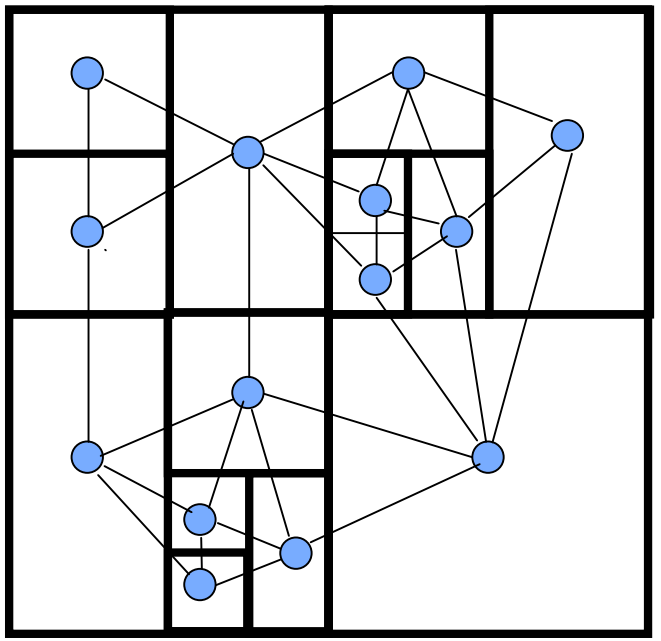
Lookup in CAN

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Zuerst wird Ort des Indexes durch Berechnung der Hash-Funktion bestimmt**
- **Zwischen den Besitzer benachbarter Rechtecke bestehen Kanten**
- **Anfrage wird in Richtung des Index weitergeleitet**

- **d Dimension des Quadrats**
  - 1: Linie
  - 2: Quadrat
  - 3: Würfel
  - 4: ...

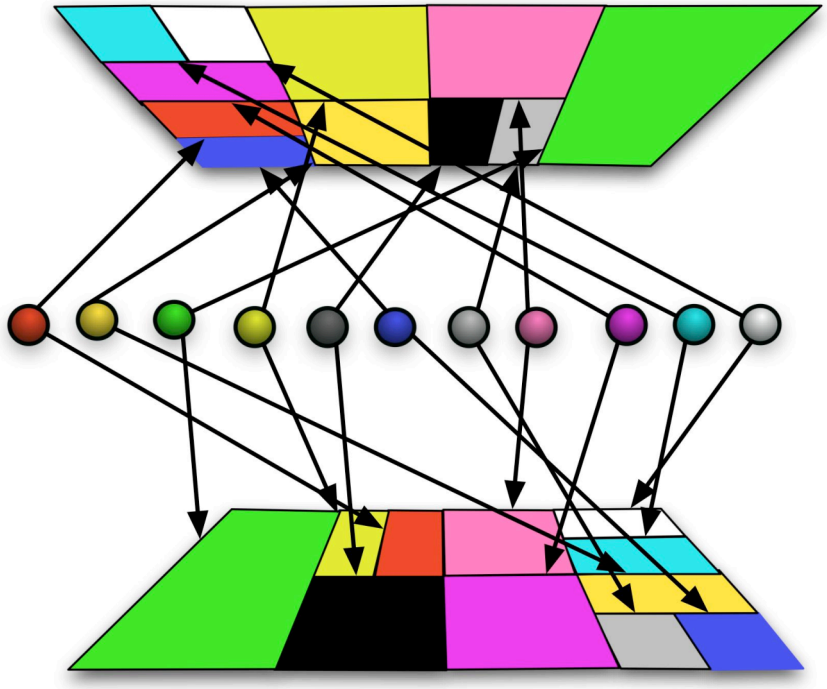
- **Erwartete Anzahl Hops in d Dimensionen: $n^{1/d}$**
- **Durchschnittlicher Grad eines Knotens: O(d)**



Mehrere Realitäten

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

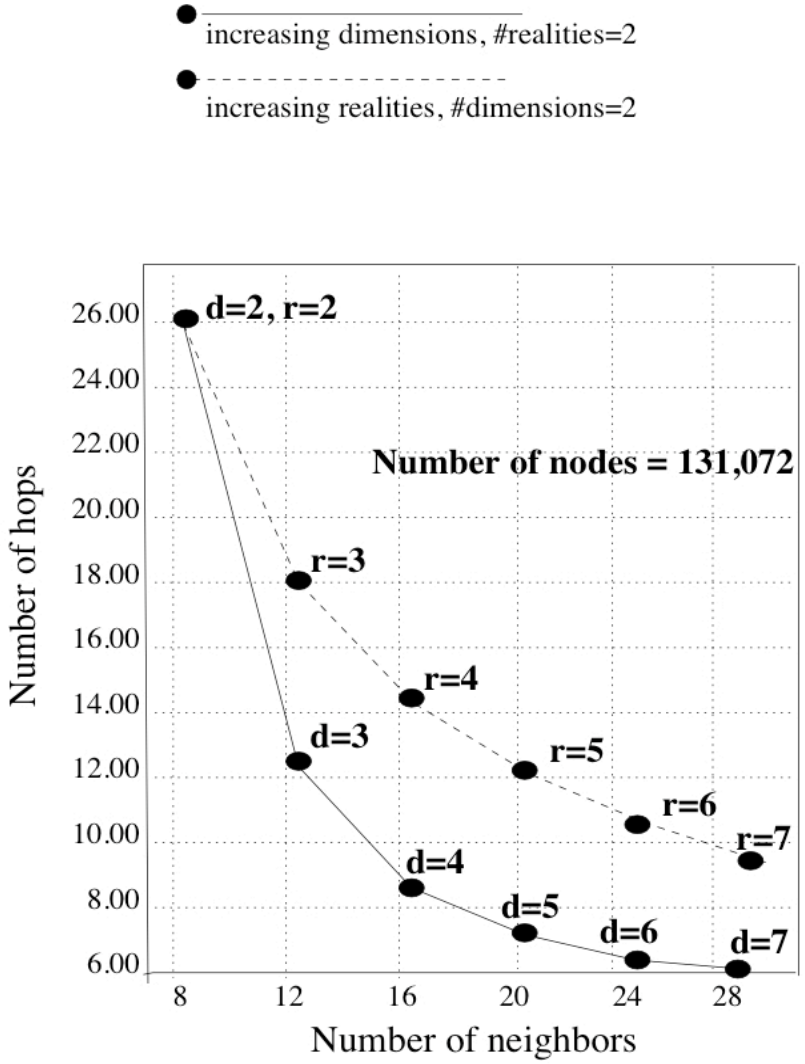
- **Simultan werden r CAN-Netzwerke aufgebaut**
- **Jedes CAN-Netzwerk wird Realität genannt**
- **Auf der Suche nach einem Feld**
  - springt man zwischen den Realitäten
  - wählt man die Realität, in welcher der Abstand zum Ziel am geringsten ist
- **Vorteile**
  - Hohe Robustheit



Realitäten versus Dimensionen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

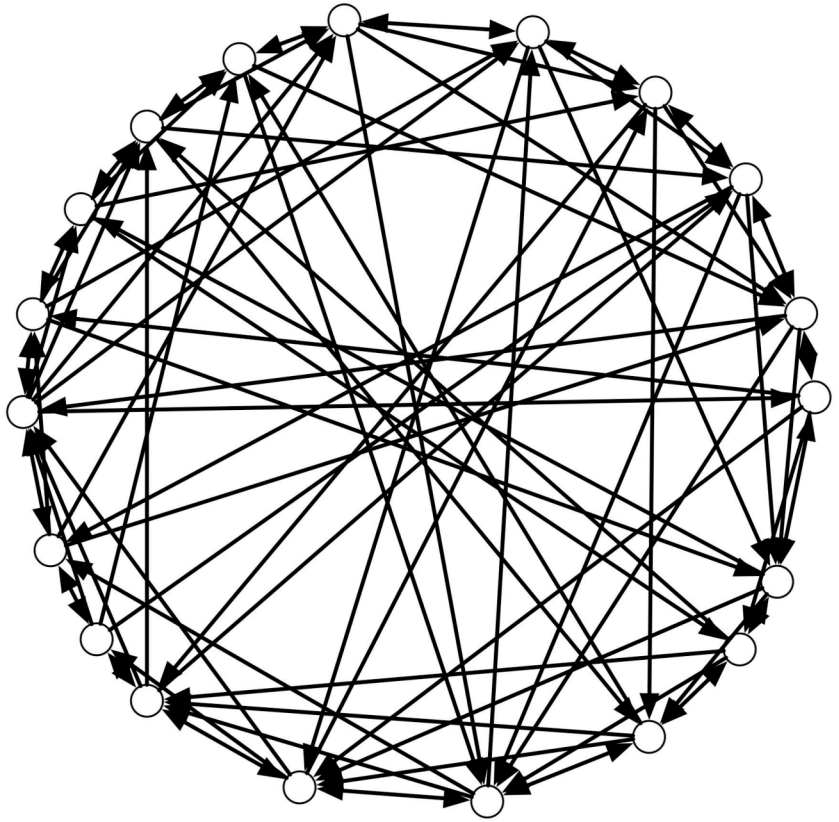
- **Dimensionen verkürzen die Wege besser**
- **Realitäten erzeugen robustere Netzwerke**



Chord

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **von Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek und Hari Balakrishnan (2001)**
- **DHT mit Hash-Bildbereich $\{0,..,2^m-1\}$**
  - für genügend großes m
- **Ring-Verknüpfung der Peers**
- **Abkürzungen im Ring durch exponentiell gestaffelte Zeiger auf Nachfolger**



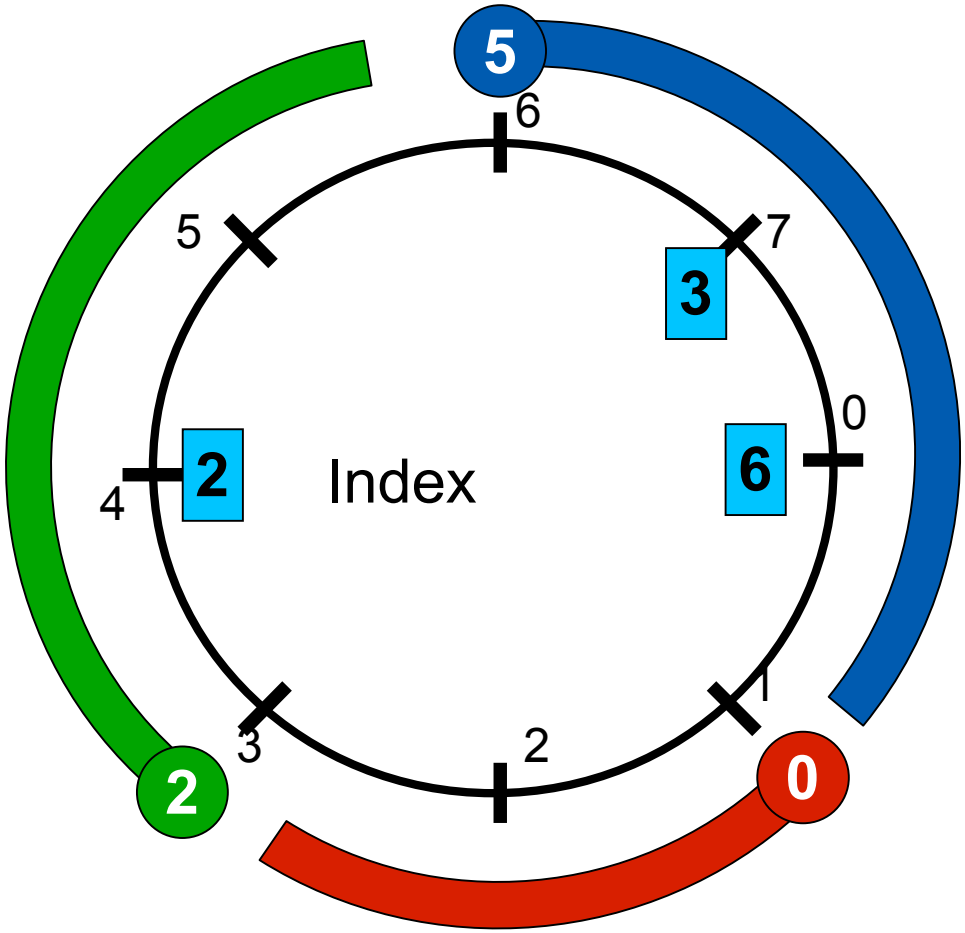
Chord als DHT

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **n: Knotenanzahl, Knotenmenge V**
- **k: Anzahl Schlüssel, Schlüsselmenge K**
- **m: Hashwertlänge: m >> log max{K,N}**

- **Zwei Hash-Funktionen bilden auf {0,..,2^m-1} ab**
  - $r_V(b)$: bildet Peer b zufällig auf {0,..,2^m-1} ab
  - $r_K(i)$: bildet Index i zufällig auf {0,..,2^m-1} ab

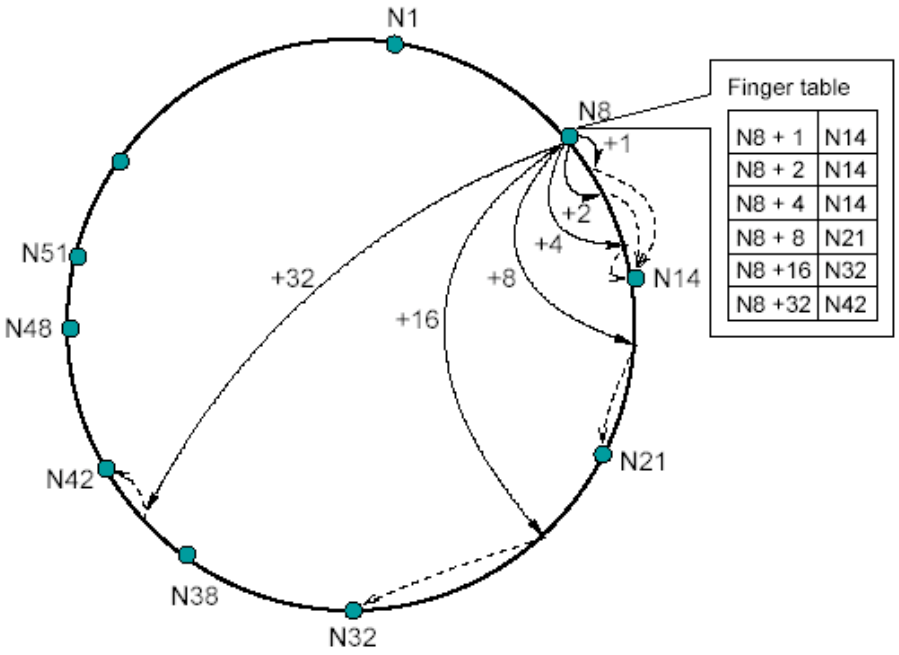
- **Abbildung von i auf einen Peer b = $f_V(i)$**
  - $f_V(i) := \arg\min_{b \in V} (r_B(b)-r_K(i))$



Die Datenstruktur von Chord

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

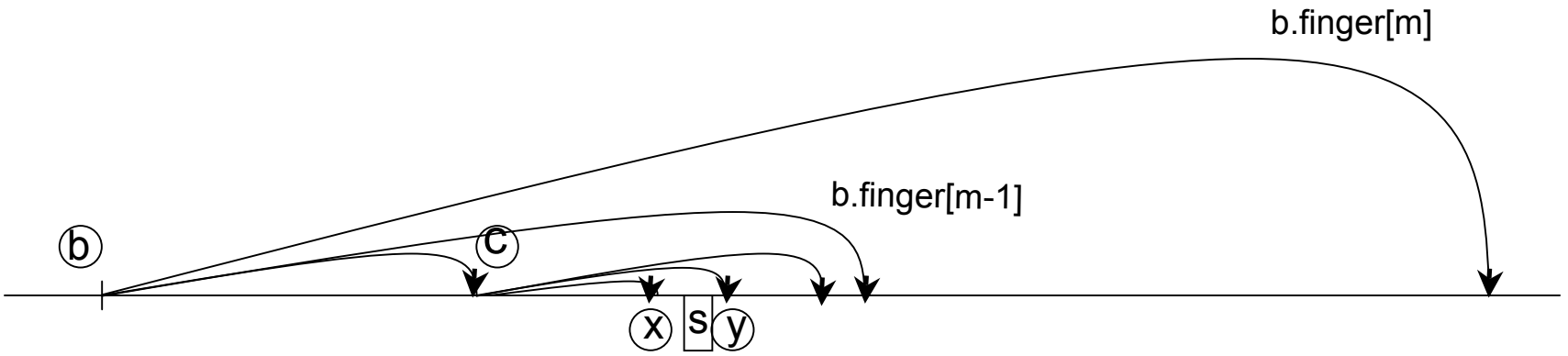
- **Für jeden Knoten b:**
  - successor: Nachfolger
  - predecessor: Vorgänger
  - Für i ∈ {0,..m-1}
    - Finger[i] := Der Knoten der dem Wert $r_V(b+2^i)$ folgt
- **Für kleine i werden die Finger-Einträge immer gleich**
  - Nur unterschiedliche Fingereinträge werden gespeichert
- **Lemma**
  - Die Anzahl unterschiedlicher Finger-Einträge für Knoten b ist mit hoher Wahrscheinlichkeit O(log n)
- **Hohe Wahrscheinlichkeit = $1-n^{-c}$**



Suchen in Chord

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Theorem**
  - Die Suche braucht mit hoher W'keit O(log n) Sprünge
- **Beweis:**
  - Mit jedem Sprung wird die Entfernung zum Ziel mindestens halbiert
  - Zu Beginn ist der Abstand höchstens 2^m
  - Der Mindestabstand zweier benachbarter Peers ist $2^m/n^c$ mit hoher W'keit
  - Damit ist die Laufzeit beschränkt durch c log n



Problembeschreibung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Motivation**
  - nicht nur die Verhinderung des berechtigten Zugriffs staatlicher Verfolgungsbehörden gegen die gesetzeswidrige Verletzung von Urheberschutzgesetzen
  - Zensur und Verfolgung in Diktaturen
- **Grade der Anonymität**
  - Autor
    - Wer hat das erzeugt?
  - Server
    - Wo wird das gespeichert?
  - Leser
    - Wer hat sich das geholt?
  - Dokument
    - Welche Dokumente werden auf einen bestimmten Peer gespeichtert?

Methoden der Anonymisierung

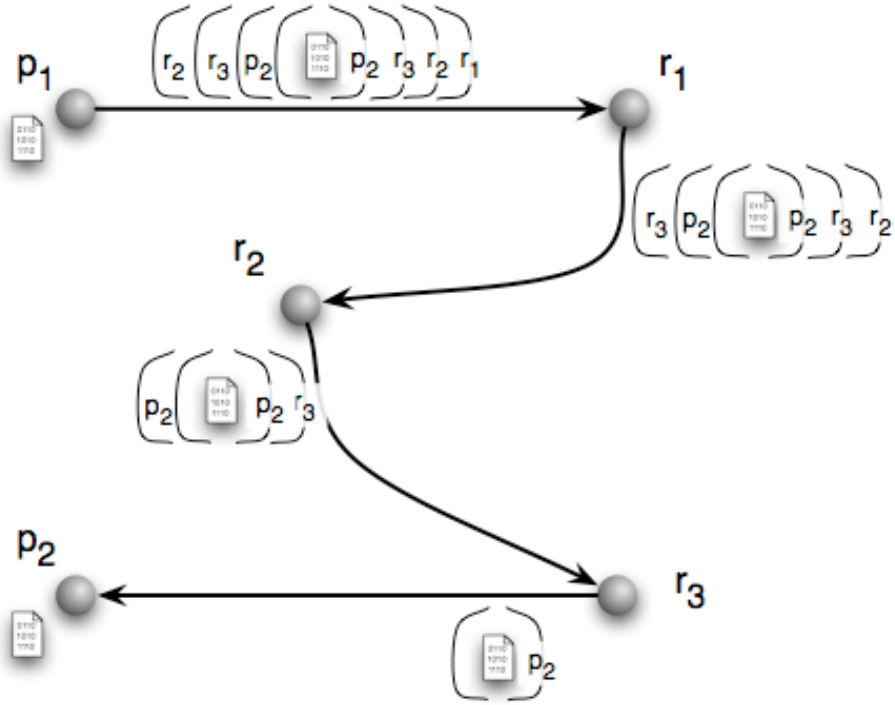
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Dining Cryptographers**
  - Wer hat's geschickt?
- **Onion Routing**
  - Verwickelte Umwege...
- **F2F-P2P**
  - Friend-to-Friend
- **Dark-Net**
  - War das was?
- **Steganographie**
  - nichts zu sehen...
- **Verschlüsselte Inhalte**
  - Denn sie wissen nicht, was sie speichern...
- **Verschlüsselte, unterschriebene Index-Einträge**
  - gezeichnet: Zorro

Onion Routing

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Von David Goldschlag, Michael Reed, and Paul Syverson**
- **Ziel**
  - Schutz der Privatsphäre von Sender und Empfänger einer Nachricht
  - Schutz der übermittelten Nachricht
- **Annahme**
  - Spezielle Infrastruktur (Onion Routers), die bis auf wenige Ausnahmen kooperieren
- **Methode:**
  - Basierend auf Mix Cascades (D. Chaum)
  - Nachricht wird von der Quelle zum Ziel über Zwischenstationen geleitet (Proxies - Onion Routers)
  - Onion Routers wählen unvorhersehbar andere Onion Routers als Zwischenstationen
  - Zwischen Sender, Onion Routers und Empfängern ist die Nachricht jeweils symmetrisch verschlüsselt
  - Jeder Onion Router kennt nur die nächste Zwischenstation
  - Die Nachricht ist wie eine Zwiebel mehrfach für die Zwischenstationen verschlüsselt
- **Onion Routers sind eine frewillige Infrastrukturerweiterung des Internets**
  - Verstehen sich nicht als Peer-to-Peer-Netzwerk



Friend-to-Friend

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

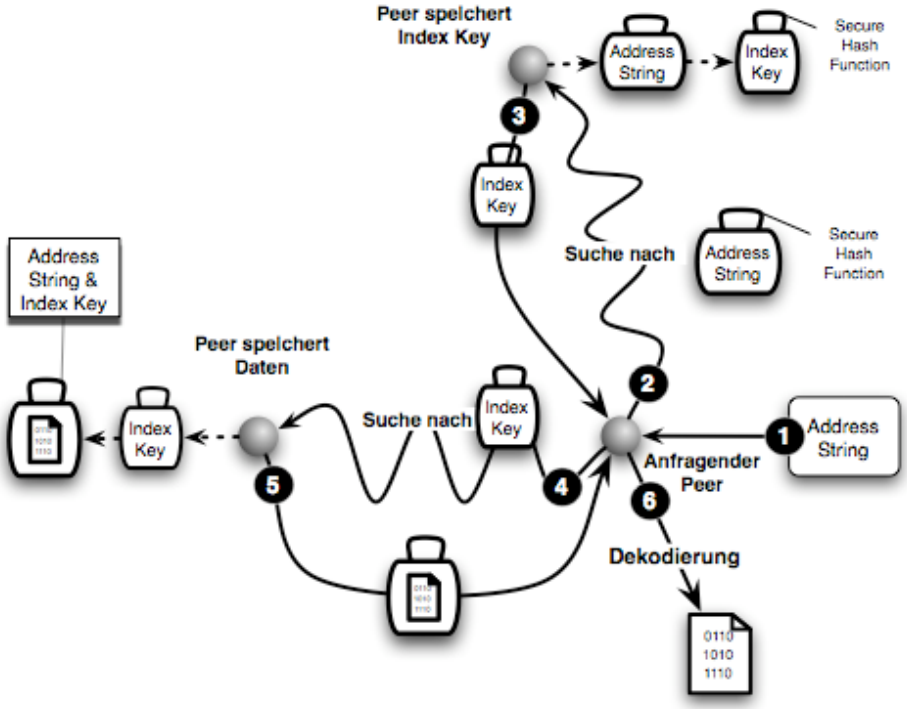
- **von Dan Bricklin (2000)**
- **Peer-to-Peer-Netzwerk mit Verbindungen nur zwischen Personen, die sich gegenseitig vertrauen**
  - weitere Verbindungen werden nicht aufgebaut
- **Kommunikation läuft über lange Pfade im Netzwerk**
  - ist jeweils verschlüsselt
  - Nachricht für den Router nicht erkennbar
  - Statt IP-Adresse wird eine Identifikation weitergeleitet
- **Vorteil**
  - IP-Adresse wird niemals veröffentlicht
  - Absolute Sicherheit

Free-Net

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **von Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore Hong, 2000**
- **Ziel**
  - Peer-to-Peer-Netzwerk
  - Erlaubt Veröffentlichung, Replikation, Beschaffung von Daten
  - Anonymität von Autoren und Lesern
- **Dateien**
  - sind orts-unabhängig referenziert
    - durch verschlüsselte und unterzeichnete Index-Dateien
    - Autor ist nicht rekonstruierbar
  - sind gegen unbefugtes Überschreiben oder Löschen geschützt
  - sind verschlüsselt
    - Inhalt ist nur durch Kenntnis der andernorts abgelegten Index-Datei in Kombination mit dem Suchbegriff lesbar
  - werden repliziert
    - auf dem Anfragepfad der Suchanfrage
  - und nach dem “Least Recently Used” (LRU) Prinzip gelöscht

- **Netzwerkstruktur**
  - stark verwandt mit Gnutella
  - Netzwerkaufbau durch Nachbarkanten
    - aber kein F2F-Netzwerk, da bei der Suche Abkürzungen eingebaut werden können
  - Ähnlich wie Gnutella ist das Netzwerk Pareto-verteilt
- **Speichern von Dateien**
  - Jede Datei kann durch den kodierten Adress-String und dem signierten Index-Schlüssel (signed subspace key) gefunden, entschüsselt und gelesen werden
  - Jede Datei wird mit der Information des Index-Schlüssels gespeichert, aber ohne kodierten Adress-String
  - Dadurch kann kein Server diese Datei lesen
    - es sei denn er führt eine Wörterbuch-Attacke durch
- **Speichern von Index-Daten**
  - Der Adress-String, kodiert durch eine kryptographische Hash-Funktion führt zu den passenden Peer, der die Index-Daten bestehend aus dem Adress-String und dem signierten Index-Schlüssel besteht
  - Mit diesen Index-Daten kann die Datei gefunden werden



Gnu-Net

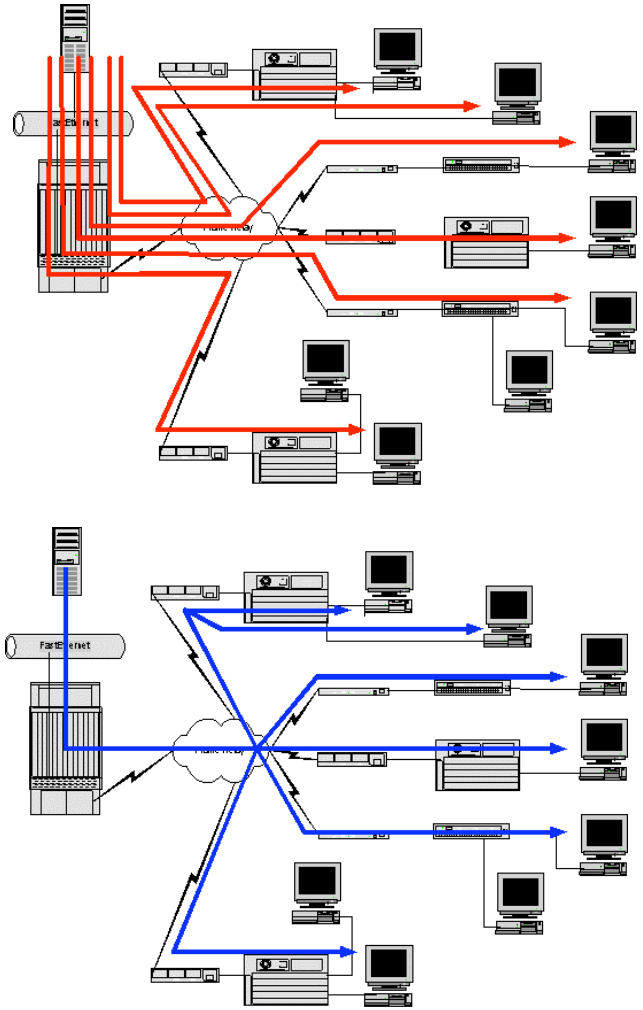
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Krista Bennett, Christian Grothoff, Tzvetan Horozov, Ioana Patrasca, Tiberiu Stef, 2006**
- **Ziele**
  - Vertrauenswürdiges, anonymes, verteiltes File-Sharing
  - wenig Nachrichten-Verkehr, geringer CPU-Overhead
  - Abwehrmaßnahmen gegen bösartige Hosts
- **Methoden**
  - GNUnets teilt große Dateien in Blöcke, die durch einen baumförmigen Code zusammengehalten werden
    - Kodierte Knoten beschreiben die Hash-Werte der Kinder im Baum
  - Trust-Management
    - Knoten können jeder Zeit ohne zentrale Kontrolle dem Netzwerk beitreten
    - Knoten starten mit geringen Vertrauen (untrusted)
    - Erst durch positive Mitwirkung wird das Vertrauen in diese Peers erhöht
    - Je größer das Vertrauen, desto mehr Anfragen dürfen sie in das Netzwerk stellen

IP Multicast

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Motivation**
  - Übertragung eines Stroms an viele Empfänger
- **Unicast**
  - Strom muss mehrfach einzeln übertragen werden
  - Bottleneck am Sender
- **Multicast**
  - Strom wird über die Router vervielfältigt
  - Kein Bottleneck mehr

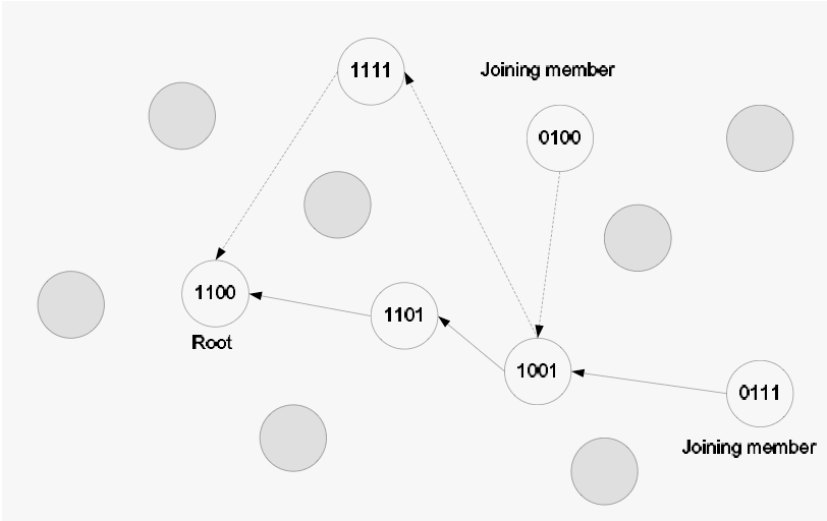


Bilder von Peter J. Welcher
www.netcraftsmen.net/.../ papers/multicast01.html

Scribe

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

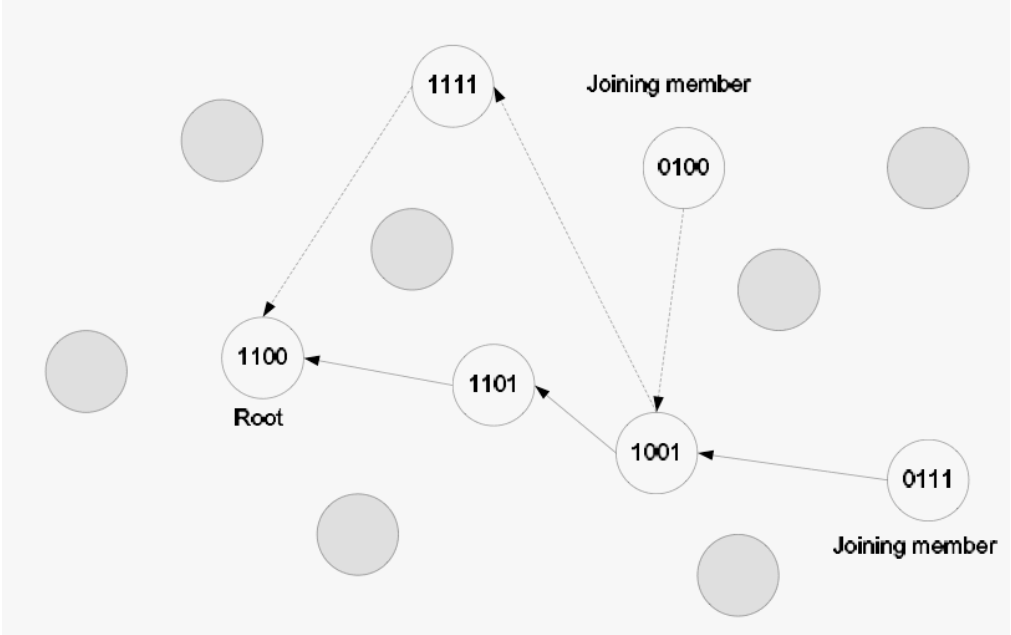
- **Multicast-Baum im Overlay-Netzwerk**
- **Scribe [2001] basiert auf Pastry**
  - Castro, Druschel, Kermarrec, Rowstron
- **Vergleichbare Ansätze**
  - CAN Multicast [2001] basiert auf CAN
  - Bayeux [2001] basiert auf Tapestry
- **Andere Ansätze**
  - Overcast [´00] und Narada [´00]
  - bauen auch Multicast-Tree auf Unicast-Verbindungen
  - skalieren nicht



Funktionsweise Scribe

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Create**
  - GroupID wird einem Peer gemäß nächsten Pastry-Index zugewiesen
- **Join**
  - Interessierter Peer macht Look-up zur Group-ID
  - Sobald ein Peer im Multicast-Baum gefunden worden ist, wird neuer Teilpfad eingefügt
- **Download**
  - Nachrichten werden baumförmig verteilt
  - Knoten duplizieren Teile

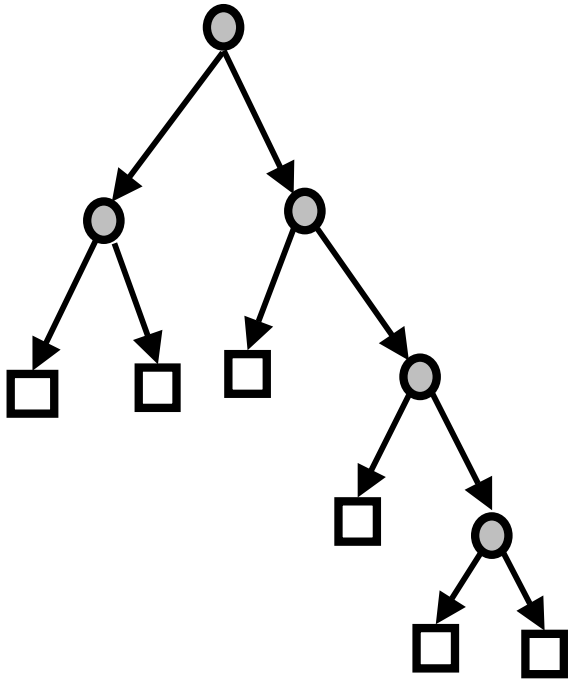


Split-Stream Motivation

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Multicast-Bäume benachteiligen gewisse Knoten**
- **Lemma**
  - In jedem Binärbaum ist die Anzahl der Blätter = Anzahl interner Knoten +1
- **Schlussfolgerung**
  - Fast die Hälfte aller Knoten verteilen Daten
  - Während die andere Hälfte profitiert
  - Als interner Knoten hat man den doppelten Upload (verglichen mit dem Durchschnitt)
- **Lösung: größerer Grad?**
- **Lemma**
  - In jedem Baum mit Grad d gilt für die Anzahl interner Knoten k und die Blätter b:

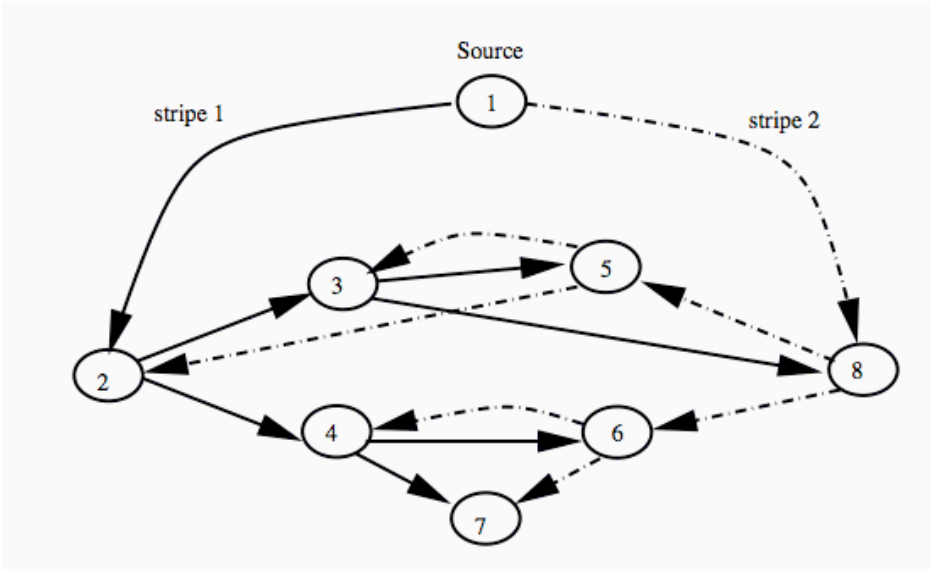
    $(d-1)\,k = b - 1$
- **Schlussfolgerung**
  - Damit müssen noch weniger Peers mehr Upload verrichten als im Binärbaum



Split-Stream Lösung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Castro, Druschel, Kermarrec, Nandi, Rowstron, Singh 2001**
- **Idee**
  - Teile die Datei der Größe B in k kleinere Teile
  - Verwende anderen Multicast-Baum für jeden der Teile
  - Dadurch wird jeder Peer mal als Blatt oder als Verteil-Knoten fungieren
    - außer der Quelle
- **Der Upload jedes Knotens ist dann (im Idealfall) höchstens der Download**



Bittorrent

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Bram Cohen**
- **Bittorrent ist ein reales (sehr erfolgreiches) Peer-to-Peer-Netzwerke**
  - konzentriert sich auf Download
  - verwendet (implizit) Multicast-Trees für die Verteilung der Daten
- **Beschreibung ist Peer-orientiert und nicht Daten-orientiert**

- **Ziele:**
  - Möglichst effizienter Download einer Datei unter Zuhilfenahme der Upload-Fähigkeit der Peers
  - Möglichst effiziente Ausnutzung des Upload von Peers
    - In der Praxis ist der Upload der Bottleneck
    - z.B. wegen der asymmetrischen Protokoll-Gestaldung von ISDN, Bitübertragungsschicht von DSL
  - Fairness zwischen den Peers
    - Seeders versus Leechers
  - Verwendung verschiedener Quellen

Bittorrent
Koordination und Datei

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

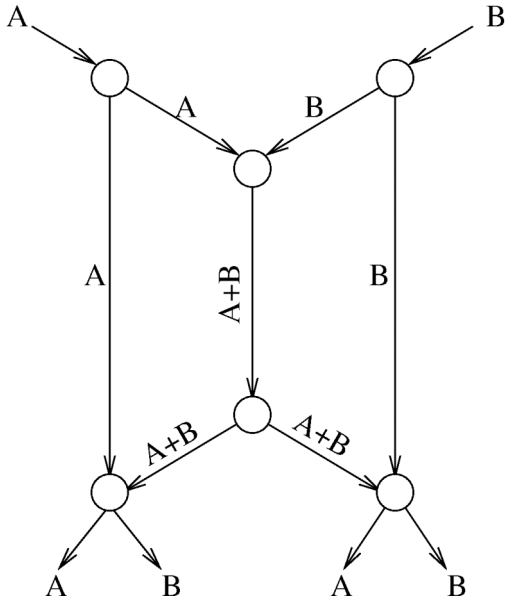
- **Zentrale Koordination**
  - durch sogenannten Tracker-Host
  - Für jede Datei gibt der Tracker eine Menge von Zufallspeers aus der Menge der downloadenden Peers
  - Zusätzlich: Ausgabe des Hash-Codes und anderer Kontroll-Information
  - Tracker-Hosts haben keine Dateien
    - Trotzdem kann das Anlegen einer Tracker-Datei auf einem Tracker-Host rechtliche Probleme ergeben (Urheberschutzsgesetz)
- **Datei**
  - ist in kleinere Dateien zerlegt (in Tracker-Datei festgelegt)
  - Jeder teilnehmende Host kann heruntergeladenen Teil weiterverbreiten, sobald er vollständig erhalten wurde
  - Damit ist Bittorrent die Umsetzung eines Split-Stream-ähnlichen Protokolls
- **Interaktion zwischen Peers**
  - Zwei Peers tauschen die Information über ihre vorhandenen Teile aus
  - Gemäß der Politik von Bittorrent werden dann noch nicht vorhandene Teile von dem einen Peer zum anderen übertragen

Netzwerk-Kodierung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *(IEEE Transactions on Information Theory*, IT-46, pp. 1204-1216, 2000)**

- **Beispiel:**
  - Die Bits A und B sollen übertragen werden
  - Über jede Leitung darf nur ein Bit übertragen werden
  - Werden nur die Bits unverändert übertragen, so
    - kann entweder nur links oder nur rechts A und B erhalten werden
  - Durch Verwendung des Xor A+B kann in beiden das Ergebnis bekommen werden

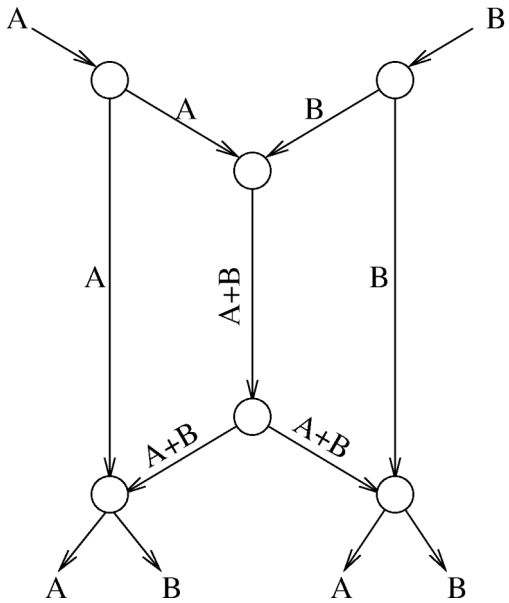


Netzwerk-Kodierung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *(IEEE Transactions on Information Theory*, IT-46, pp. 1204-1216, 2000)**

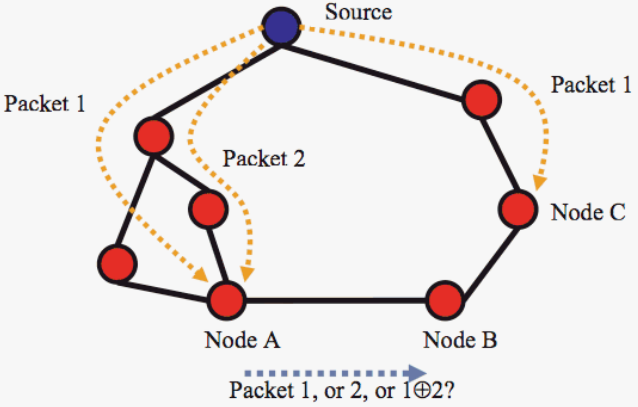
- **Theorem [Ahlswede et al.]**
  - Es gibt einen Netzwerk-Code für jeden Graphen, so dass man so viel Information von den Quellen erhalten kann,
  - wie das zugehörige Fluss-Problem erlaubt.

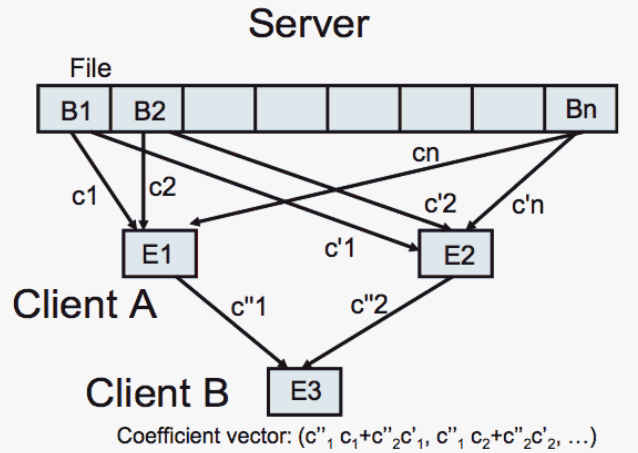


Praktische Netzwerk-Kodierung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Christos Gkantsidis, Pablo Rodriguez Rodriguez, 2005**
- **Ziel**
  - Überwendung des Coupon-Collector-Problems bei der Partitionierung von Dateien
    - Eine Datei aus m Teilen kann von den Vorgängern erhalten werden, wenn die Summe ihrer Teile mindestens m ist
  - Optimale Übertragung von Dateien hinsichtlich der verfügbaren Bandbreite
- **Methode**
  - Verwende als Codes Linear-Kombinationen der Teile einer Datei
    - Entstehender Code beinhaltet den Vektor
  - Bei der Verteilung werden Linear-Kombination rekombiniert zu neuen Teilen
  - Beim Empfänger werden die Linear-Kombinationen gesammelt
  - und mittels Matrix-Invertierung die Original-Datei rekonstruiert





Kodierung und Dekodierung

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- Datei: $x_1, x_2, \ldots, x_m$
- Codes: $y_1, y_2, \ldots, y_m$
- Mit zufälligen Variablen r_{ij}

$$(r_{i1} r_{i2} \ldots r_{im}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = y_i$$

- Also

$$\begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

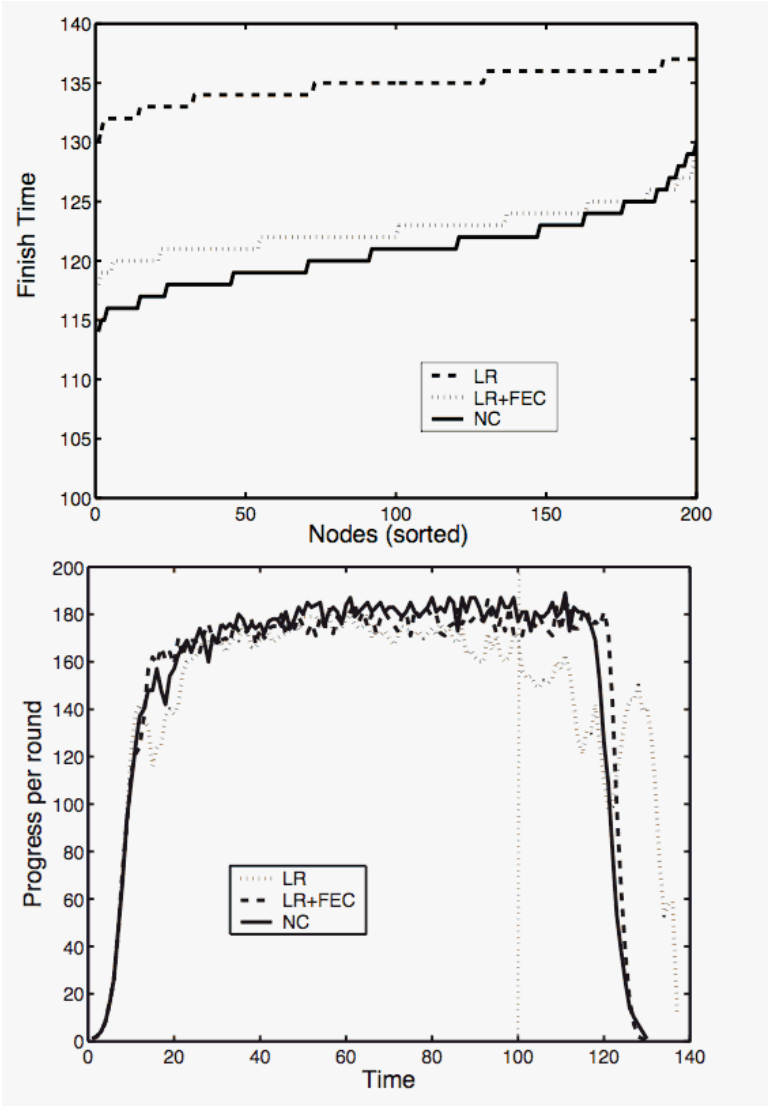
- Falls die Matrix (r_{ij}) invertierbar ist, erhält man

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

Geschwindigkeit von Network-Coding

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Vergleich**
  - Network-Coding (NC) versus
  - Local-Rarest (LR) und
  - Local-Rarest+Forward-Error-Correction (LR+FEC)



Probleme mit Network-Coding

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Zusatzaufwand: Speichern der Variablen**
  - Pro Block 1 Variablen-Vektor
  - Z.B. 4 GByte-Datei hat mit 100 kByte-Block hat Variablenvektor von
    - 4 GByte/100 kByte = 40 kByte
    - Damit entsteht ein Overhead von rund 40% pro Datei
  - Besser: 4 GByte und 1 MByte-Block
    - ergibt 4kByte Overhead von 0,4%
- **Dekodierungsaufwand**
  - Invertierung einer m x m- Matrix benötigt Zeit $O(m^3)$ und Speicher $O(m^2)$
  - Damit ist der Speicherverbrauch bei
    - 4 kByte-Variablen-Vektor = 16 MByte
    - 40 kByte-Variablen-Vektor = 1,6 GByte
- **Schreib-/Lese-Zugriffe**
  - Um m Blöcke zu kodieren muss jeder Teil der Datei m-mal gelesen werden
  - Zur Dekodierung muss jeder Code-Teil ebenfalls m-mal gelesen werden
  - Abnutzung der Speichermedien (Festplatten)
  - Zeitverbrauch für Lese/Schreib-Operationen (Disk-Cache wird nicht ausgenutzt)

Ende der 27. Vorlesung

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer

Wintersemester 2006/07

27. Vorlesung

08.02.2007