Peer-to-Peer-Netzwerke

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

Christian Schindelhauer

Sommersemester 2006

9. Vorlesung

24.05.2006

schindel@informatik.uni-freiburg.de

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

Inhalte

- Kurze Geschichte der Peer-to-Peer-Netzwerke
- Das Internet: Unter dem Overlay
- Die ersten Peer-to-Peer-Netzwerke
  - Napster
  - Gnutella
- CAN
- Chord
- Pastry und Tapestry
- Gradoptimierte Netzwerke
  - Viceroy
  - Distance-Halving
  - Koorde
- Netzwerke mit Suchbäumen
  - Skipnet und Skip-Graphs
  - P-Grid
- Selbstorganisation
  - Pareto-Netzwerke
  - Zufallsnetzwerke
  - Metrikbasierte Netzwerke
- Sicherheit in Peer-to-Peer-Netzwerken
- Anonymität
- Datenzugriff: Der schnellere Download
- Peer-to-Peer-Netzwerke in der Praxis
  - eDonkey
  - FastTrack
  - Bittorrent
- Peer-to-Peer-Verkehr
- Juristische Situation

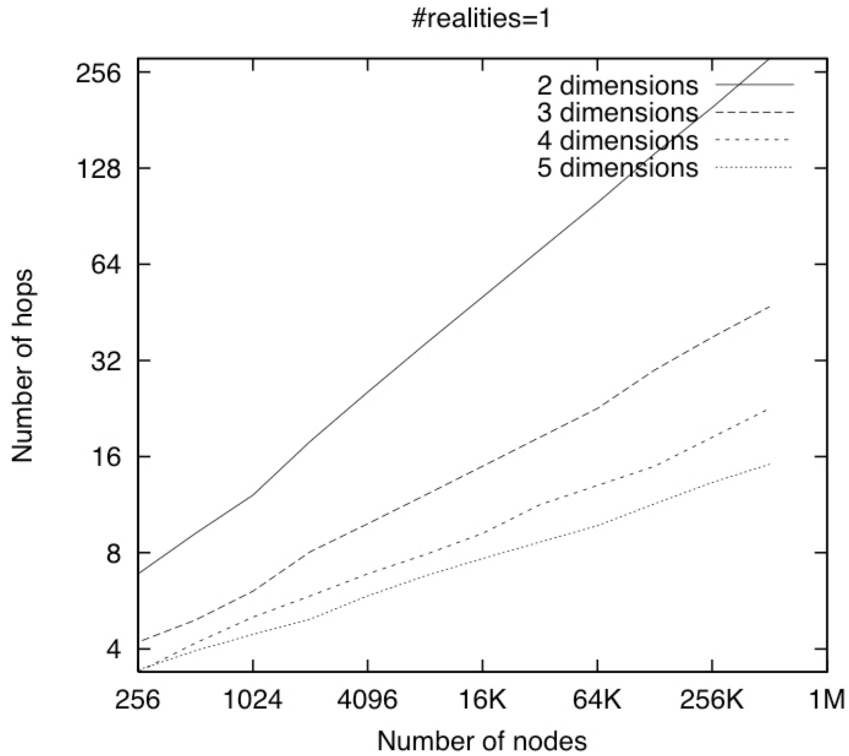
Systemverbesserungen für CAN

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

1. Mehrdimensionale Räume
2. Verschiedene Realitäten
3. Abstandsmetrik für Routing
4. Überladen der Zonen
5. Mehrfaches Hashing
6. Topologie-angepasste Netzwerkkonstruktion
7. Gleichmäßigere Partitionierung
8. Caching, Replikation und Hot-Spot-Management

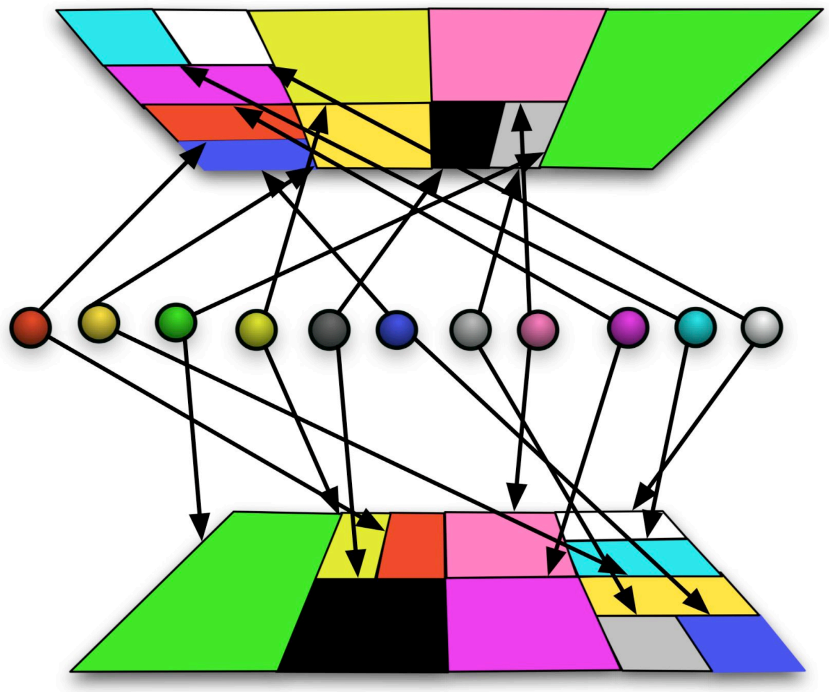
Mehrdimensionale Räume

- **D-dimensionaler Raum (statt 2-D)**
  - 1: Linie
  - 2: Quadrat
  - 3: Würfel
  - ...
- **Die erwartete Pfadlänge bei d Dimensionen ist O($n^{1/d}$)**
- **Erwartete Anzahl von Nachbarn O(d)**



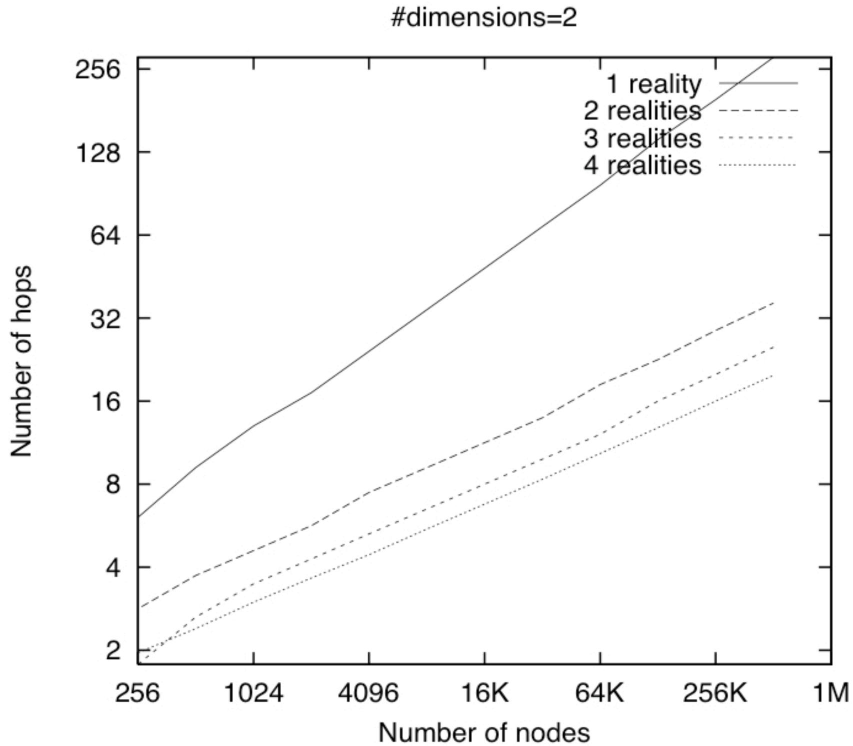
Mehrere Realitäten

- **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



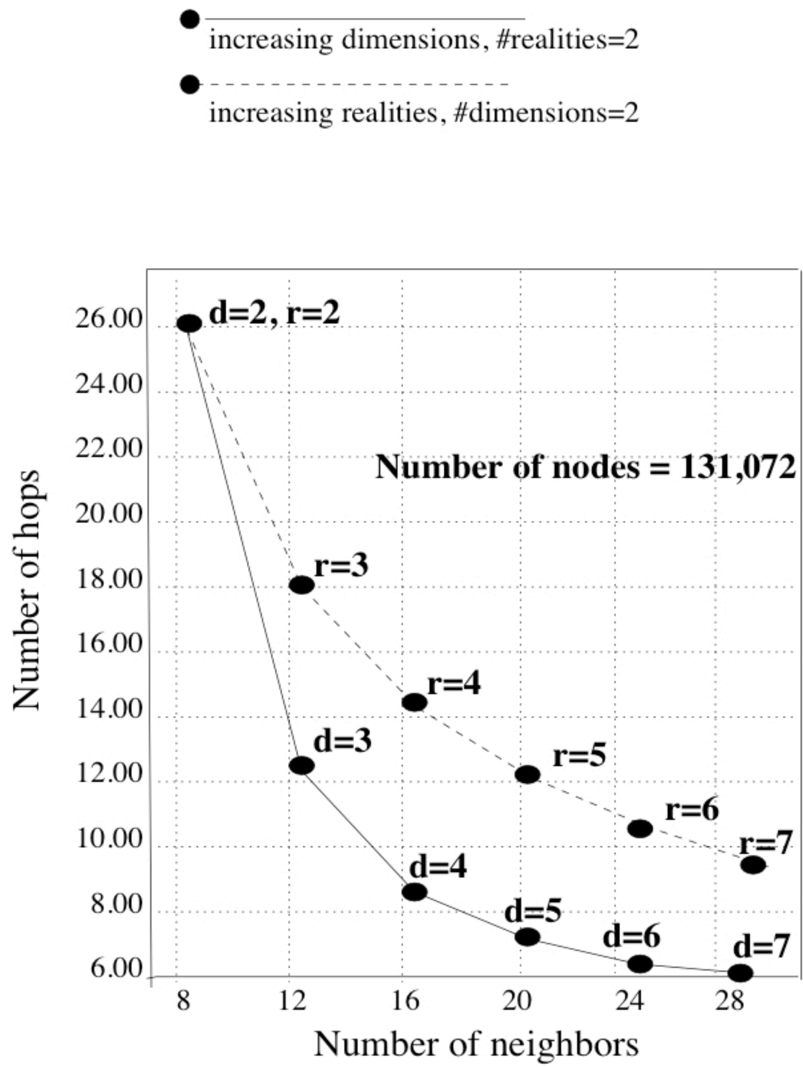
Mehrere Realitäten

- **Vorteile**
  - Hohe Robustheit
  - Kürzere Wege



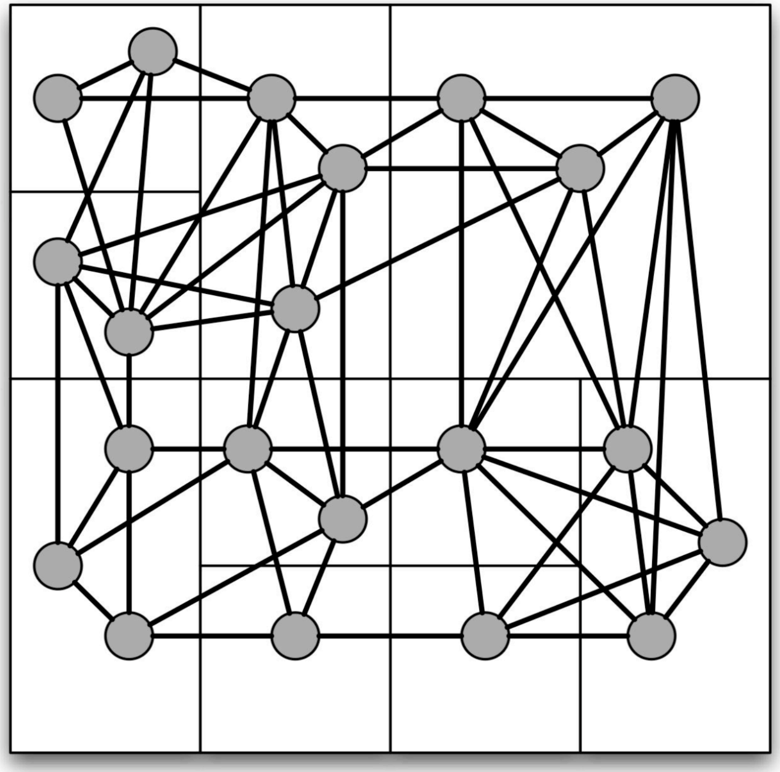
Realitäten versus Dimensionen

- Dimensionen verkürzen die Wege besser
- Realitäten erzeugen robustere Netzwerke



Überladen von Zonen

- **In jede Zone werden bis zu MAXPEERS (z.B. 10) Peers abgelegt**
  - Jeder Peer kennt alle Peers seiner Zone
  - und jeweils einen der Nachbarzone
  - Dadurch werden Routen nicht verlängert
- **Wege verkürzen sich um O(MAXPEERS)**
- **Latenzzeit kann verkürzt werden**
  - indem jeder Peer den nächsten Peer der Nachbarzone wählt
- **Verbesserte Fehlertoleranz**



Abstandsmetriken für Routing

- **Durch Messung der RTT (round trip time) wird Abstandsmessung vorgenommen**
- **Bevorzuge kürzesten Nachbarn gemäß dieser Metrik**
- **Vorteil:**
  - Verringerung der Latenzzeit um konstanten Faktor
- **Bessere Zeitersparnis Topologie-angepasste Netzwerkkonstruktion**
- **Siehe auch Übungsaufgabe**

Mehrfaches Hashing

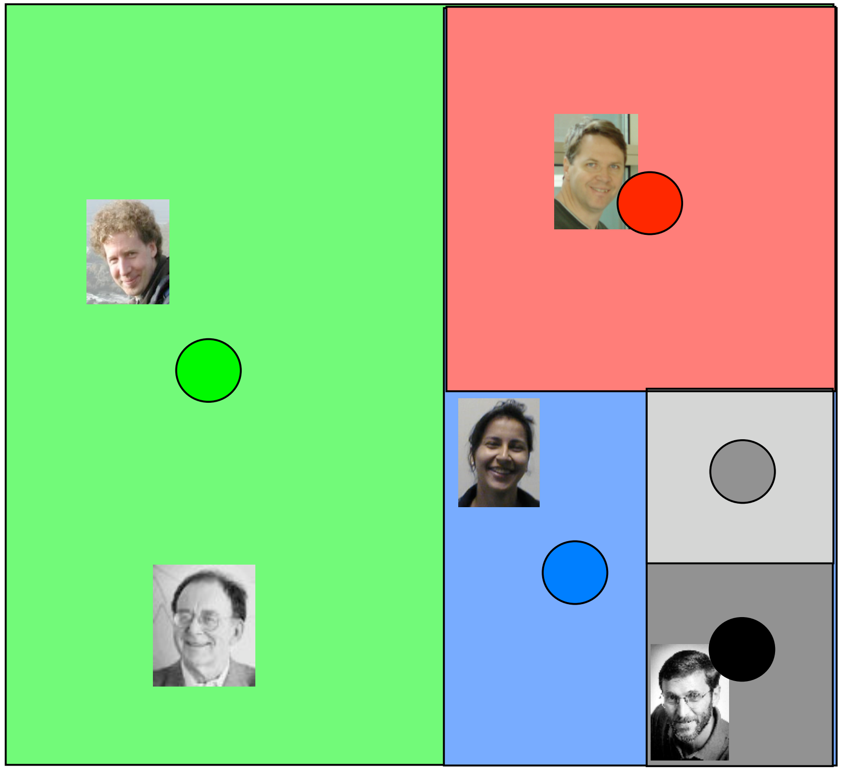
- **Daten werden nicht nur an einmal, sondern mehrfach abgespeichert,**
  - indem man den Schlüssen mit Zahl k aus {1,2,..,COPIES} kombiniert
- **Dadurch erhöhte Robustheit**
- **Geringere Entfernungen**
  - Lookup nur zu nächster Kopie
  - Anzahl Hops indirekt proportional zu Anzahl Kopien

Topologie-angepasste Netzwerkkonstruktion

- **Die gemessenen Latenzzeiten zu m ausgezeichneten Peers, genannt Landmarken, dienen als Positionsinformation**
- **Die Zeiten werden sortiert**
- **Die sortierte Liste der Landmarken dient als Schlüssel**
- **Dieser Schlüssel wird jetzt als Basis für die Abbildung auf Bildbereich gewählt**
  - Dabei wird keine „echte“ Hashfunktion gewählt
  - Sondern eine die ähnliche Permutation in nahe Bereiche abbildet
- **Dadurch**
  - Nahe Knoten kommen in den gleichen Bereich
  - Enorme Verkürzung der Latenzzeiten
- **Aber**
  - Wahl der Landmarken schwierig
  - Gefahr der ungleichen Aufgabenverteilung

Bewertung CAN

- **Vorteile**
  - Einfaches robustes Verfahren
  - Balanciert die Datenmenge
  - Kleiner Grad
  - Netzwerk ist stark zusammenhängend, dadurch robust
  - Kennt verschiedene Wege zum Ziel und kann dadurch Routen optimieren
- **Nachteile**
  - Lange Wege (polynomiell lang)
  - Stabilität durch geringe Nachbarzahl gefährdet



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

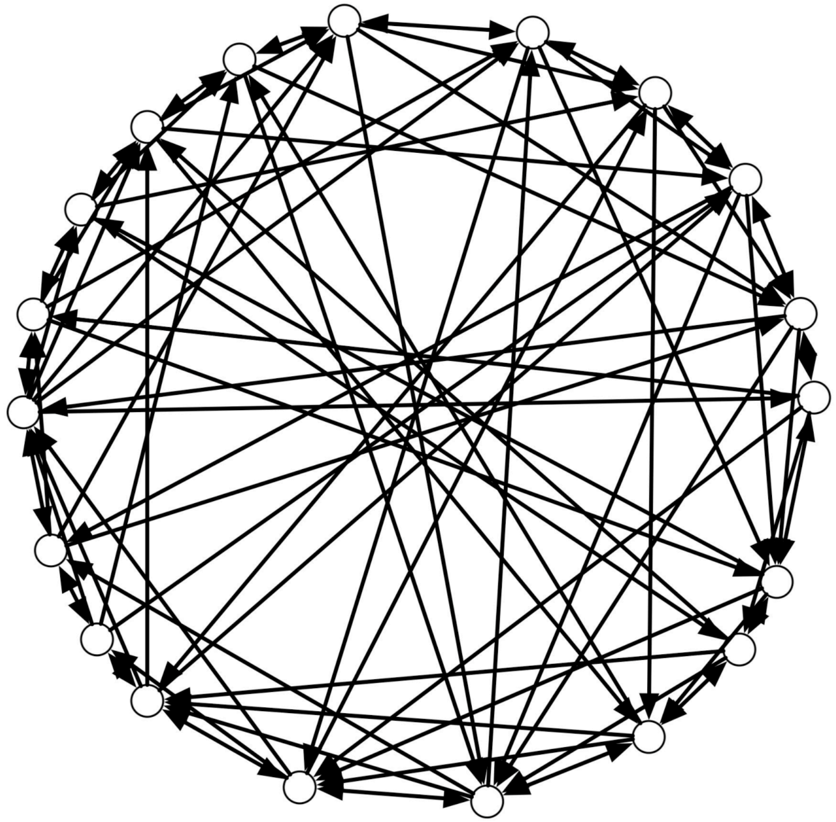
Inhalte

- **Kurze Geschichte der Peer-to-Peer-Netzwerke**
- **Das Internet: Unter dem Overlay**
- **Die ersten Peer-to-Peer-Netzwerke**
  - Napster
  - Gnutella
- **CAN**
- **Chord**
- **Pastry und Tapestry**
- **Gradoptimierte Netzwerke**
  - Viceroy
  - Distance-Halving
  - Koorde
- **Netzwerke mit Suchbäumen**
  - Skipnet und Skip-Graphs
  - P-Grid
- **Selbstorganisation**
  - Pareto-Netzwerke
  - Zufallsnetzwerke
  - Metrikbasierte Netzwerke
- **Sicherheit in Peer-to-Peer-Netzwerken**
- **Anonymität**
- **Datenzugriff: Der schnellere Download**
- **Peer-to-Peer-Netzwerke in der Praxis**
  - eDonkey
  - FastTrack
  - Bittorrent
- **Peer-to-Peer-Verkehr**
- **Juristische Situation**

Chord

- **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**

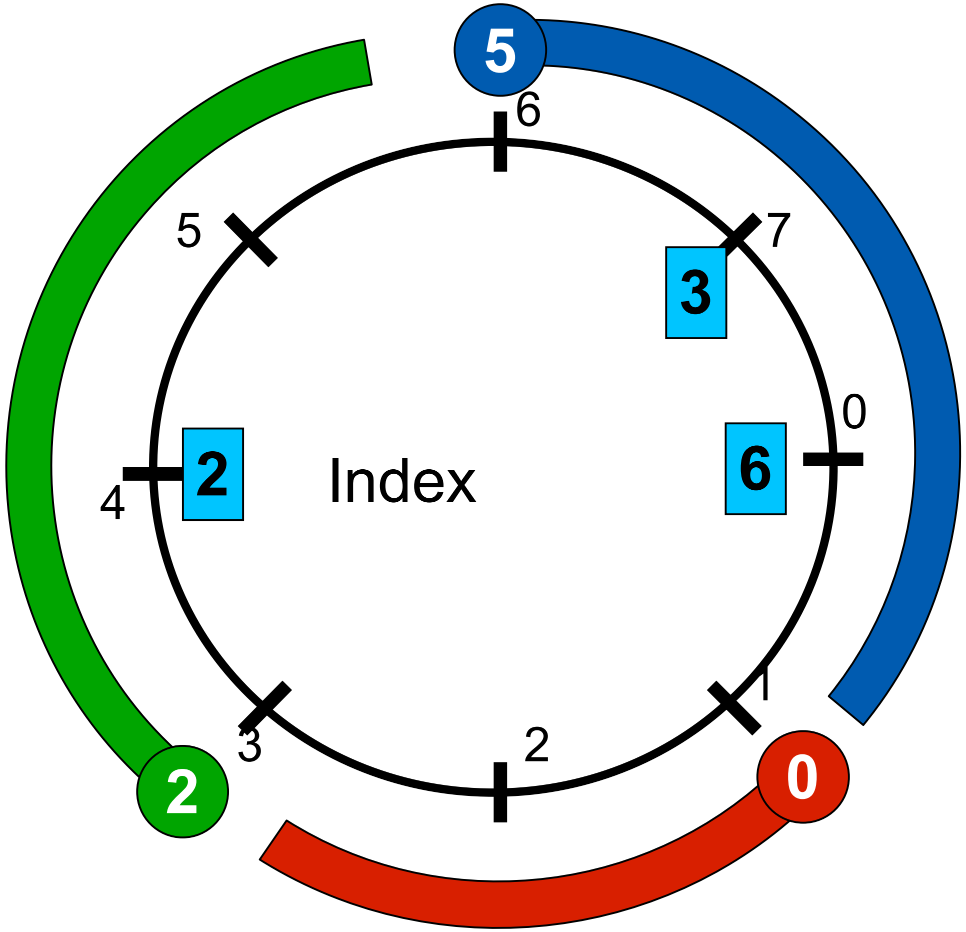


Chord als DHT

- **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

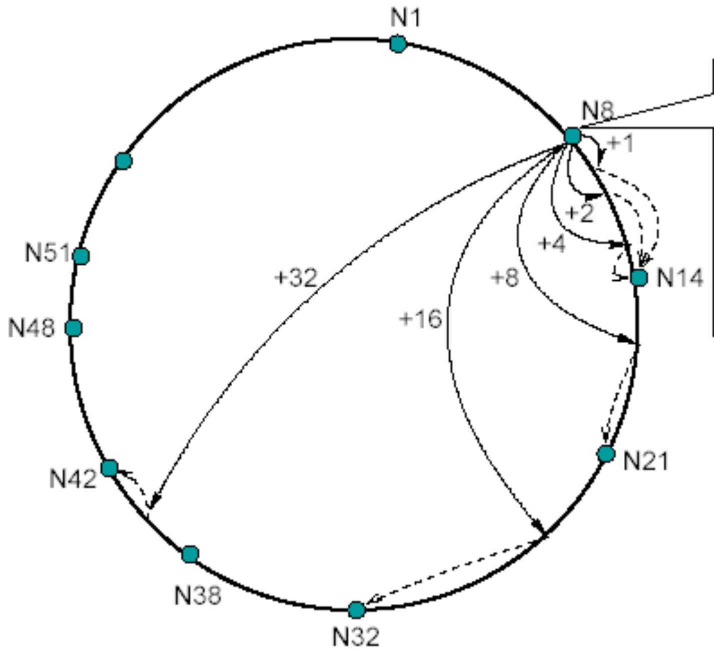
- **Abbildung von i auf einen Peer $b = f_V(i)$**
  - $f_V(i) := \arg\min_{b \in V} (r_B(b)-r_K(i))$



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

Die Datenstruktur von Chord

- **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)



N8 + 1	N14
N8 + 2	N14
N8 + 4	N14
N8 + 8	N21
N8 +16	N32
N8 +32	N42

- **Hohe Wahrscheinlichkeit = $1-n^{-c}$**

Balance in Chord

- **n: Anzahl der Knoten im P2P-Netzwerk**
- **k: Anzahl der Schlüssel $\geq$ 1**

- **Theorem**
  - Die Datenstruktur von Chord hat folgende Eigenschaften
    - Balance&Load: Mit pol. W‘keit ($1-n^{-c}$) werden in jedem Knoten höchstens O(k/n log n) Schlüssel gespeichert
    - Dynamik: Tritt ein neuer Knoten hinzu oder verlässt ein Knoten das Netzwerk müssen mit pol. W‘keit höchstens O(k/n log n) Schlüssel bewegt werden.
- **Beweis**
  - …

Eigenschaften der Datenstruktur

- **Lemma**
  - Der Abstand $|r_V(b.succ) - r_V(b)|$ ist
    - im Erwartungswert $2^m/n$,
    - mit hoher Wahrscheinlichkeit höchstens $O((2^m/n) \log n)$ und
    - mit hoher Wahrscheinlichkeit mindestens $(2^m/n)/ n^c$ für eine Konstante $c>0$
    - In einem Intervall der Länge $w\, 2^m/n$ sind mit hoher Wahrscheinlichkeit
      - $\Theta(w)$ Knoten, falls $w=\Omega(\log n)$
      - höchstens $O(w \log n)$ Knoten, falls $w=O(\log n)$
- **Lemma**
  - Die Anzahl der Knoten, die einen Fingerzeiger auf Knoten b besitzen ist
    - im Erwartungswert $O(\log n)$
    - mit pol. Wahrscheinlichkeit höchstens $O(\log n)$

Suchen in Chord

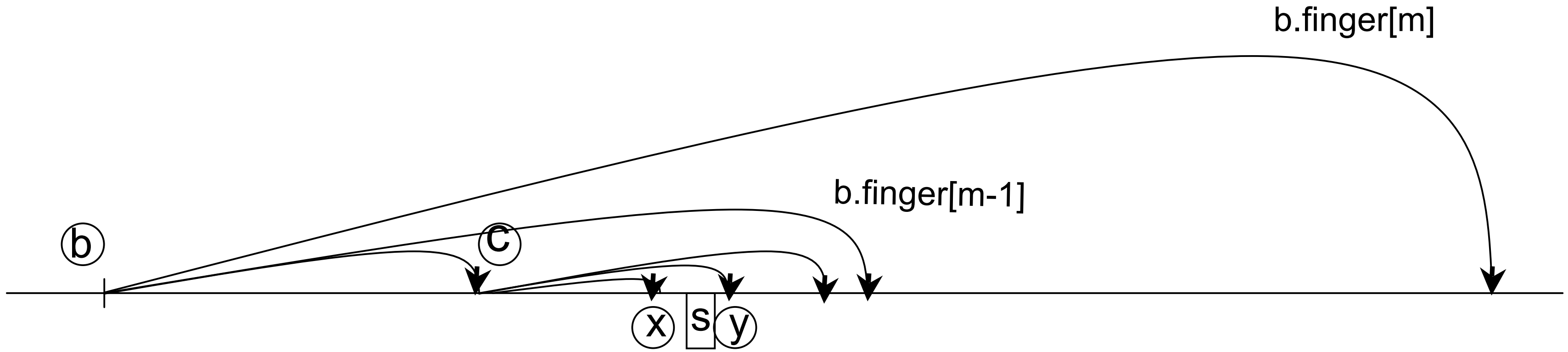
- **Theorem**
  - Die Suche braucht mit hoher W'keit O(log n) Sprünge

- **Suchalgorithmus für Element s:**
  - Abbruch(b,s):
    - Knoten b,b'=b.succ gefunden, mit rK(s) $\in [r_V(b), r_V(b')|$
  - Hauptroutine: Starte mit irgendeinem Knoten b

```
while not Abbruch(b,s) do
    for i=m downto 0 do
        if r_K(s) ∈ [r_V(b.finger[i]),r_V(finger[i+1])] then
            b ← b.finger[i]
        fi
    od
```

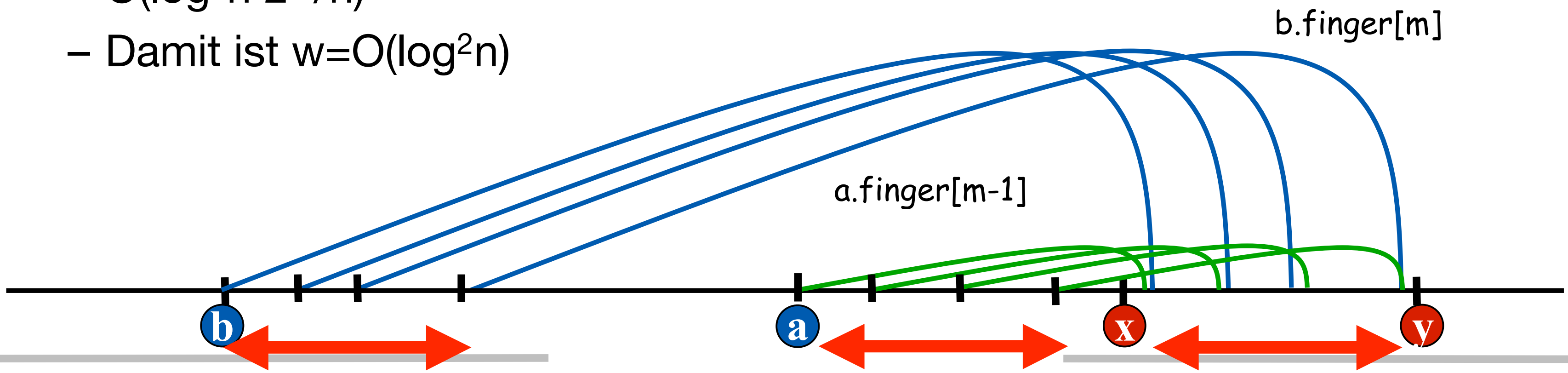
Suchen in Chord

- **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



Fingeranzahl

- **Lemma**
  - Der Ausgrad im CHORD-Netzwerk ist O(log n) mit hoher W'keit
  - Der Eingrad im CHORD-Netzwerk ist $O(\log^2 n)$ mit hoher W'keit
- **Beweis**
- **Der minimale Abstand zweier Peers ist $2^m/n^c$ (mit hoher W'keit)**
  - Damit ist der Ausgrad beschränkt durch c log n (mit hoher W'keit)
- **Der maximale Abstand zweier Peers ist $O(\log n\ 2^m/n)$**
  - Jeder Peer, der mit einem seiner Finger auf diese Linie zeigt, erhöht den Eingrad des nachstehenden Peers.
  - Die Gesamtlänge der Streckenabschnitte, wo solche Peers liegen ist $O(\log^2 n\ 2^m/n)$
  - Damit ist $w=O(\log^2 n)$



Ende der 9. Vorlesung

CoNe
Freiburg

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

Peer-to-Peer-Netzwerke

Christian Schindelhauer
schindel@informatik.uni-freiburg.de