

# *Peer-to-Peer- Netzwerke*



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

**Christian Schindelhauer**

Sommersemester 2006

15. Vorlesung

28.06.2006

**[schindel@informatik.uni-freiburg.de](mailto:schindel@informatik.uni-freiburg.de)**



# Evaluation der Lehre im SS2006

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

- 
- **Umfrage zur Qualitätssicherung und -verbesserung der Lehre**
    - unter den Studierenden
    - in anonymer Form
    - Online-Fragebogen oder zum Ausdrucken
  
  - <http://www.informatik.uni-freiburg.de/~welte/lehrevaluation/ss2006/lehrevaluation.html>
  
  - **Frist bis zum 30. Juni (Freitag)**
  
  - **Sprechstunde:**
    - Dienstag 14-15 Uhr



# 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 geordneter Speicherung**
    - P-Grid
    - Skip-Net und Skip-Graphs
  - **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**



# P-Grid

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

---

➤ **Karl Aberer, P-Grid: A Self-Organizing Access Structure for P2P Information Systems, LNCS 2171, 2001, 179-194**

➤ **Problem:**

- DHT (Distributed Hash-Tables) zerstören Datenzusammenhang
- Hashing zerhackt jegliche Lokalität

➤ **Ziel:**

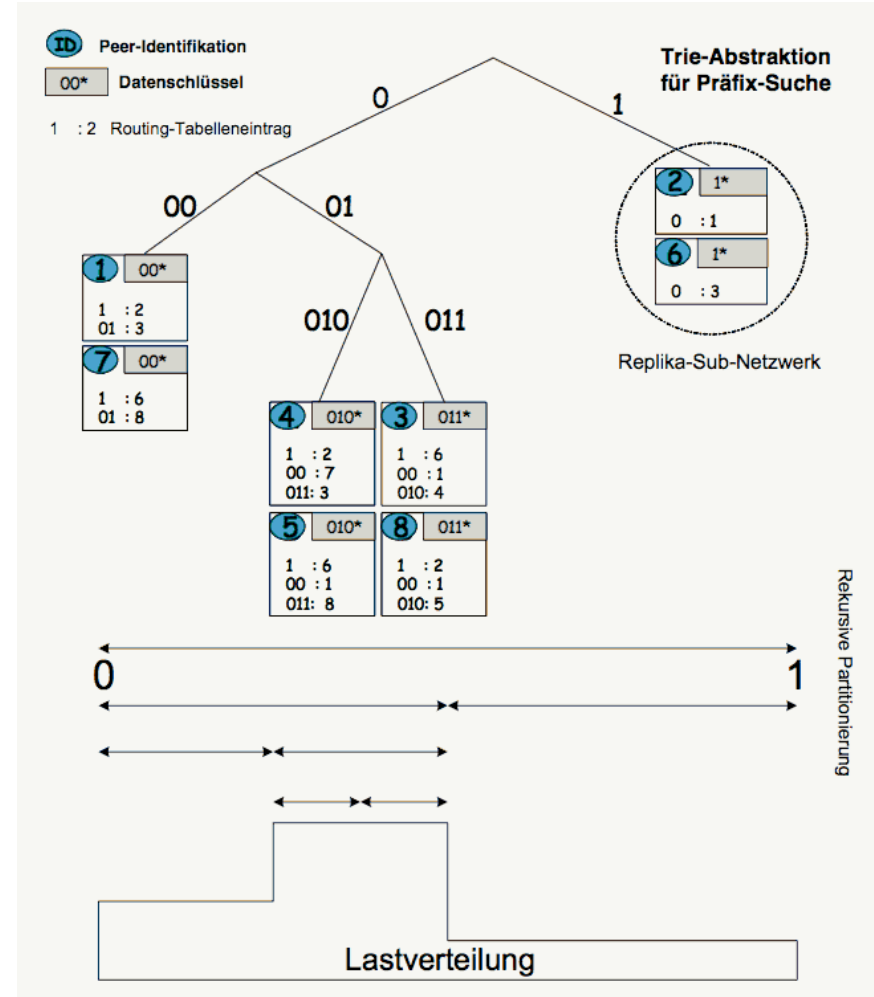
- Strukturiertes Overlay-Netzwerk
- Wie in einem Baum



# P-Grid

## ➤ Grundprinzipien

- Unterteilung des Schlüsselraums rekursiv in Partitionen mit
  - annähernd gleicher Anzahl der Schlüssel (Datenverweise)
  - annähernd gleicher Anzahl von Peers
- Rekursives Unterteilen erzeugt eine **Baumstruktur**
  - Verwendung eines TRIE

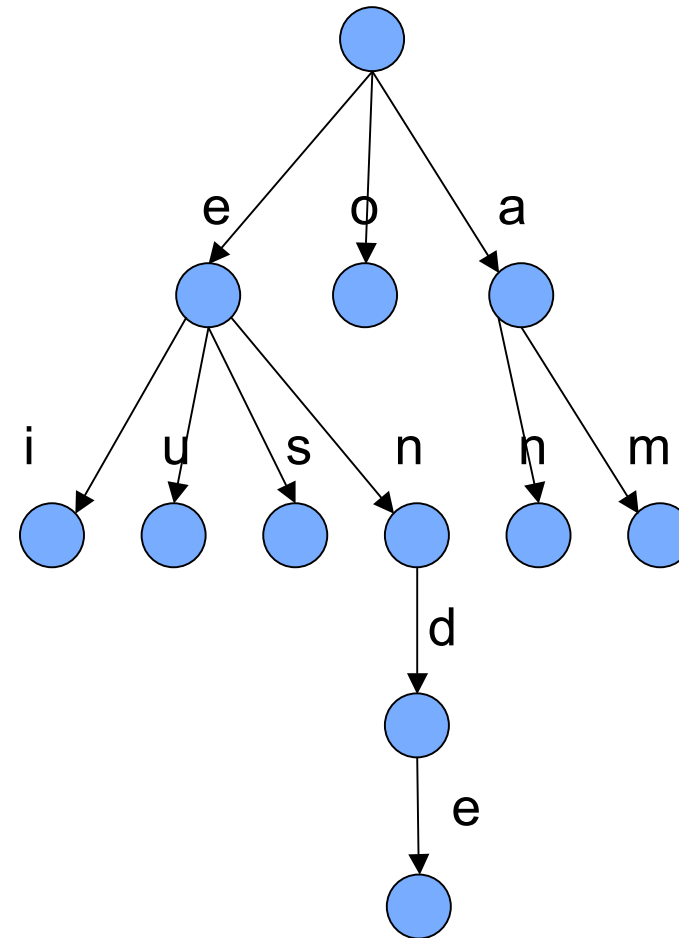


Aus: „Das P-Grid-Overlay-Netzwerk: Von einem einfachen Prinzip zu einem komplexen System“ Karl Aberer, Anwitaman Datta, Manfred Hauswirth, Roman Schmidt



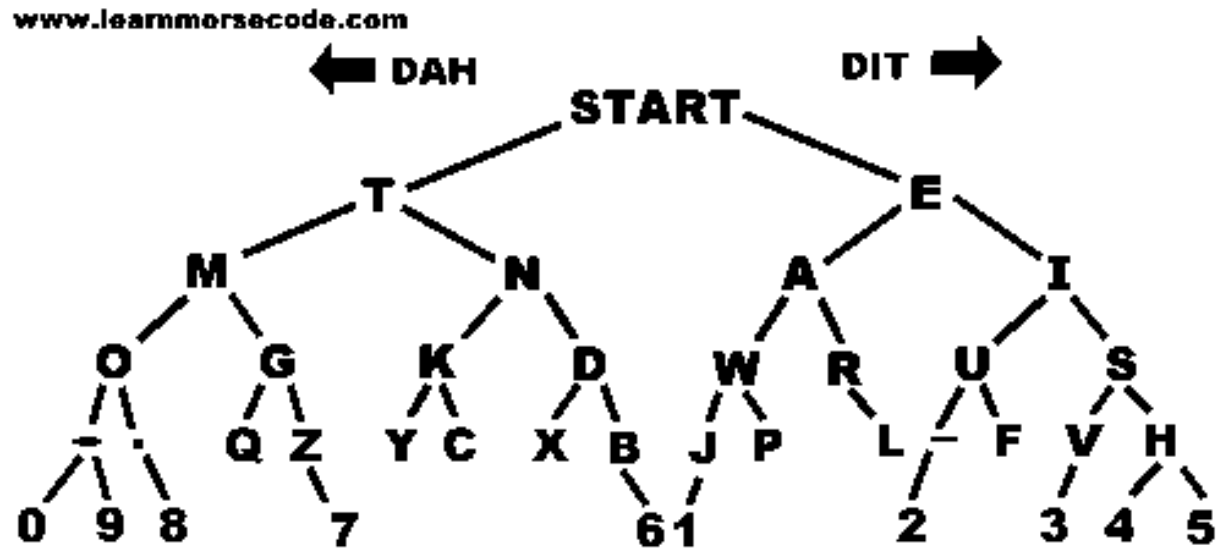
# TRIE = „reTRIEval TREE“

- **Ein Trie (gesprochen wie „Tree“)**
  - ist ein Baum
  - zum Speichern oder Kodieren von Text
  - Effiziente Suche für gleiche Präfixe
- **Struktur**
  - Kanten werden mit Buchstaben/Symbolen beschriftet
  - Knoten tragen die Wörter/Daten
- **Abbildung**
  - Man folgt gemäß der Buchstaben des Texts der Baumstruktur
  - Endet das Wort, hat man den zugehörigen Knoten erreicht
  - Jeder Knoten bezeichnet ein Wort
    - und umgekehrt





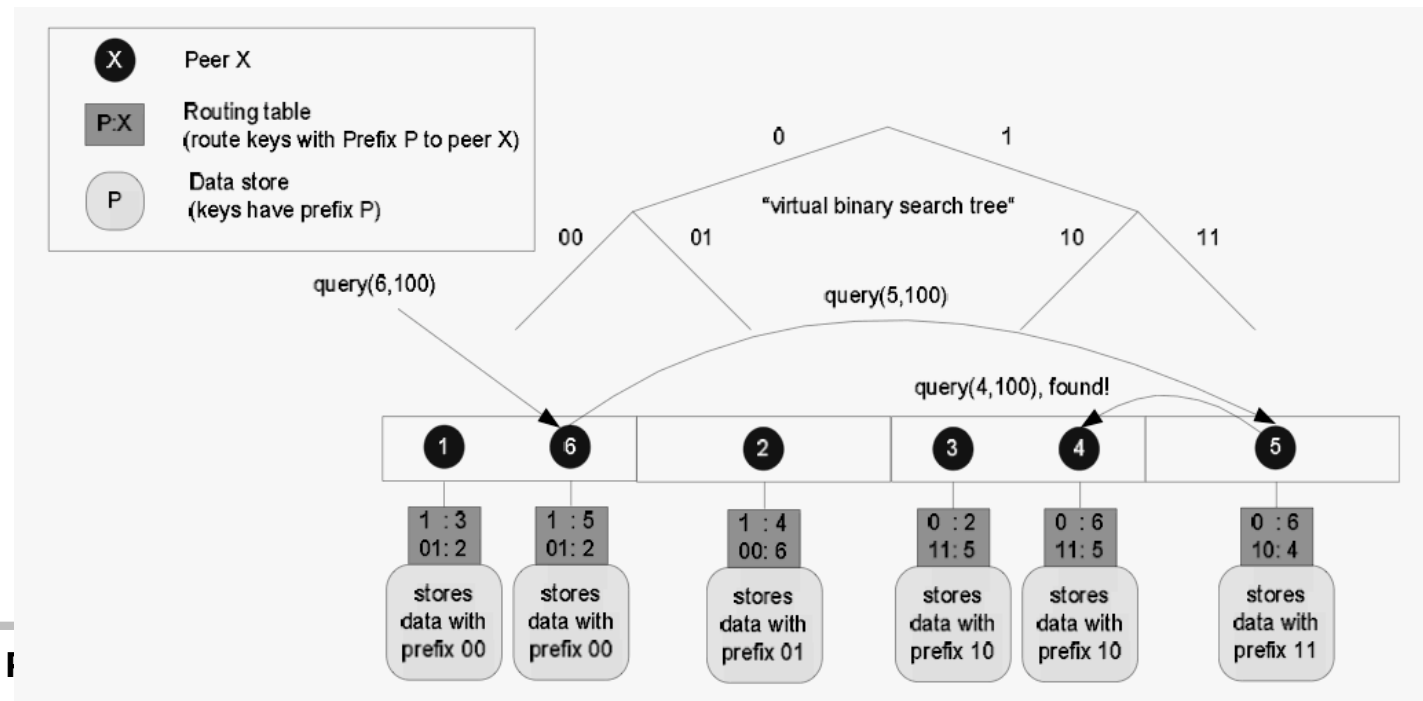
# Beispiel eines TRIE





# Struktur von P-Grid

- **Peers werden in die Blätter eines Binärbaums abgelegt**
- **Jeder Peer hat Zeiger auf mindestens einen Peer eines benachbarten Teilbaums**
  - ähnlich wie in Pastry/Tapestry
- **Datenschlüssel aus Intervall [0,1)**
  - Daten werden in die Blätter des Baums abgebildet
- **Die Anzahl der Peers in benachbarten Teilbäumen werden gemäß der Datenschlüssel ausbalanciert**
  - Dies geschieht durch paarweise Interaktion von Peers







# Balancierungs- eigenschaften

➤ **Die Pfadlänge ist dynamisch:**

- Sobald nur noch ein Peer in einem Teilbaum ist, wird dieser nicht weiter aufgeteilt

➤ **Die Anzahl der Peers in einem Teilbaum wird proportional zu der Anzahl der Dateneinträge gewählt**

➤ **Beim Einfügen:**

- Peers initiieren Interaktion mit zufällig ausgewählten Peers
- Mit geeignet gewählten Wahrscheinlichkeit
  - aufgrund des Zustände der Peers
  - und deren Datenmenge
- wählt der Peer einen Teilbaum aus
- und verfolgt den gewählten Teilbaum in die Tiefe



# Suche in P-Grid

## ➤ Theorem

- Die erwartete Anzahl von Hops bei der Suche in P-Grid ist  $O(\log n)$

## ➤ Beweis:

- Zwar können die Bäume beliebig tief sein
- Aber: Jeder Teilbaum der mehr als die Hälfte der Peers besitzt wird in erwarteter Anzahl von 2 Sprüngen erreicht,
  - weil mit Wahrscheinlichkeit  $1/2$  in einem Schritt
  - mit W'keit  $1/4$  in höchstens zwei Schritten
  - mit W'keit  $1/8$  in höchstens drei Schritten
  - ... ergibt: erwartet 2 Schritte
- Dann halbiert sich die Anzahl der erreichbaren Peers nach erwartungsgemäß nach drei Schritten
  - = 2 Schritten bis zum Teilbaum der Größe  $n/2$
  - + 1 Schritt zu dessen Verringerung unter  $n/2$
- Damit ist die erwartete Suchtiefe  $3 \log n$ .



# Bootstrapping

➤ **Hauptproblem: Partitionierung der Bäume zu Beginn**

– Peers sind noch unentschieden

➤ **Lösung: folgende Heuristik**

➤ **AEP (Adaptive Eager Partitioning)**

1. Jeder Peer, der sich noch nicht entschieden hat, initiiert Interaktionen mit zufällig ausgewählten Peers, bis er eine Entscheidung getroffen hat.
2. Wenn der kontaktierte Peer sich noch nicht entschieden hat, dann führen die Peers mit einer Wahrscheinlichkeit von  $\alpha(p)$  eine balancierte Aufteilung durch und nehmen wechselseitige Referenzen in ihre Routing-Tabellen auf.
3. Hat sich der kontaktierte Peer schon für 1 entschieden, dann entscheidet sich der Initiator mit einer Wahrscheinlichkeit von  $\beta(p)$  für 0 und mit einer Wahrscheinlichkeit von  $1 - \beta(p)$  für 1.
  - Im ersten Fall, referenziert der initiiierende Peer den kontaktierten Peer,
  - im zweiten Fall fordert er eine Referenz zur anderen Partition von diesem Peer an.

➤  **$\beta(p)$  genügt (z.B.) der Gleichung:** 
$$p = 1 - \frac{1}{\beta}(1 - 2^\beta)$$

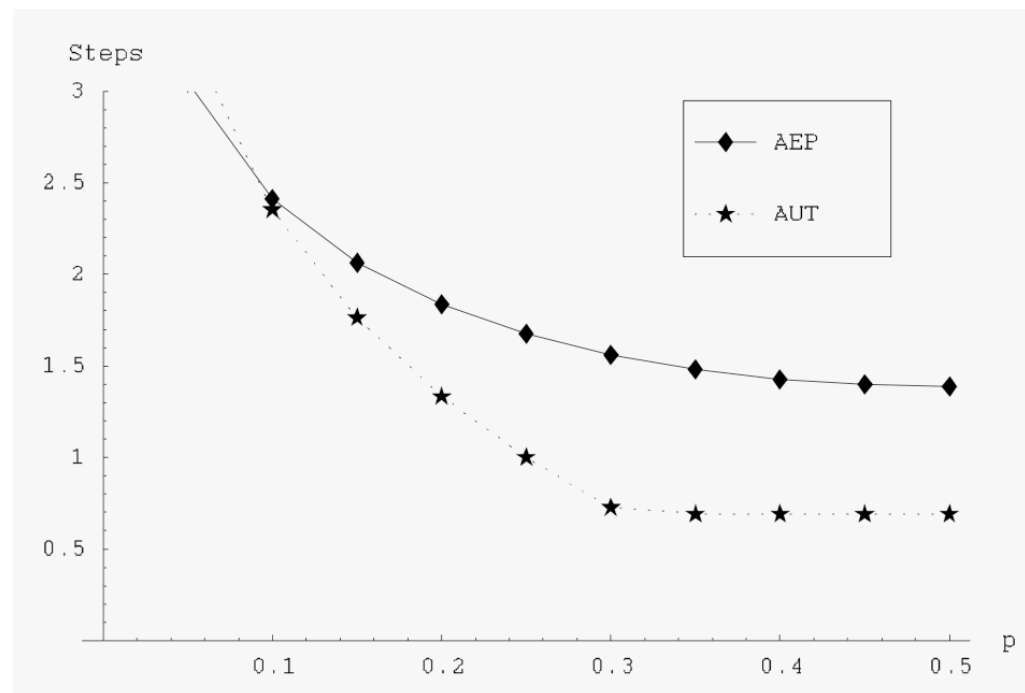


# Bootstrapping

## ➤ Autonomous Partitioning (AUT)

- Wähle zufällig Partition mit W'keit  $p$
- Kontaktiere zufällige Peers bis ein Partner für den Nachbar-Baum gefunden wurde

## ➤ Vergleich der Laufzeit





# Weitere Eigenschaften von P-Grid

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

## ➤ Lokale Lastbalancierung

- Im Netzwerk werden die Peers abhängig von der erfahrenen Last umbalanciert
- Die Baumstruktur passt sich dadurch der Datenlast an

## ➤ Dynamische Adressen

- Werden durch Baumadressierung abgebildet

## ➤ Dezentrales Trust-Management

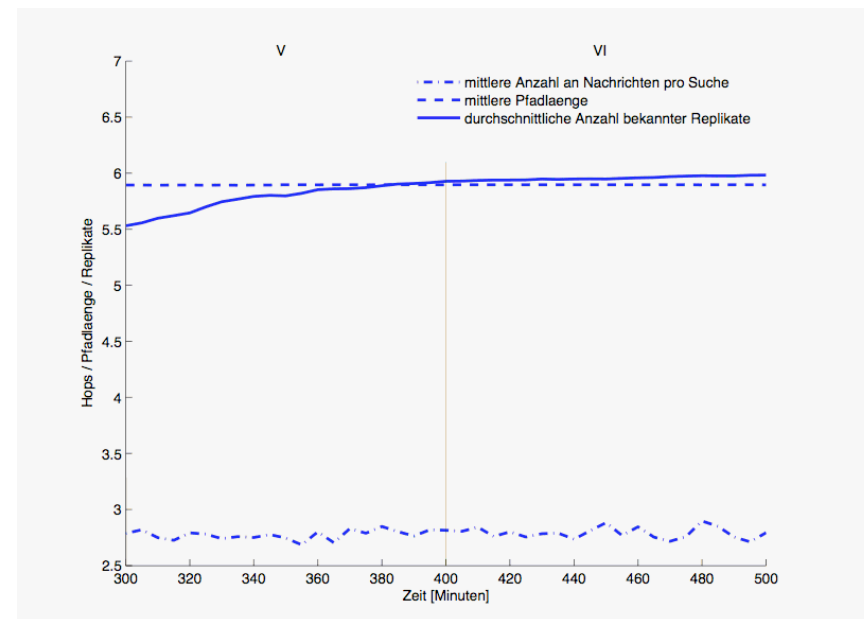
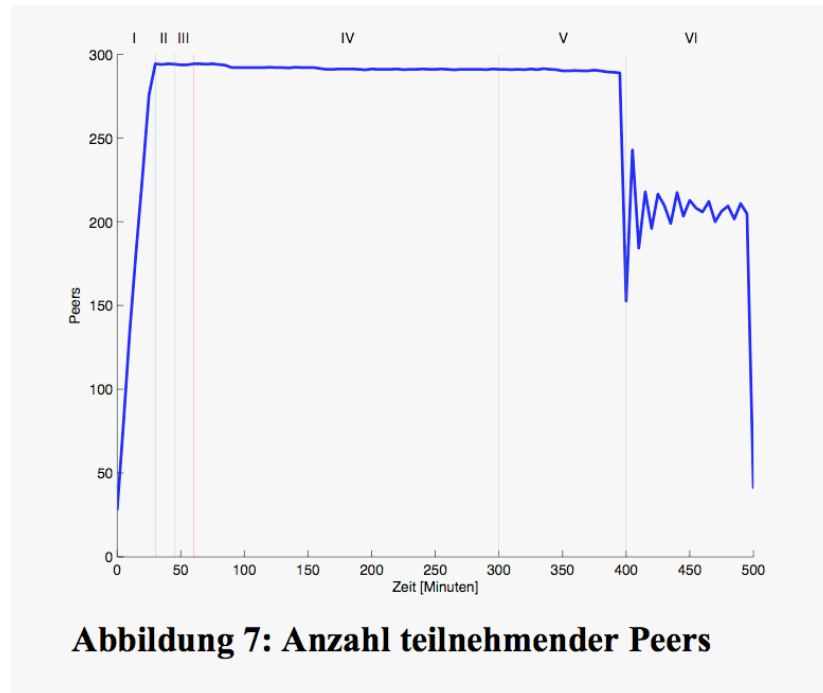
- Als selbstorganisierender Algorithmus

## ➤ Updates

- Vertraut auf epidemischer Informationsverbreitung (Rumor Spreading)
- Erzeugt konsistente Sicht auf dem Baum

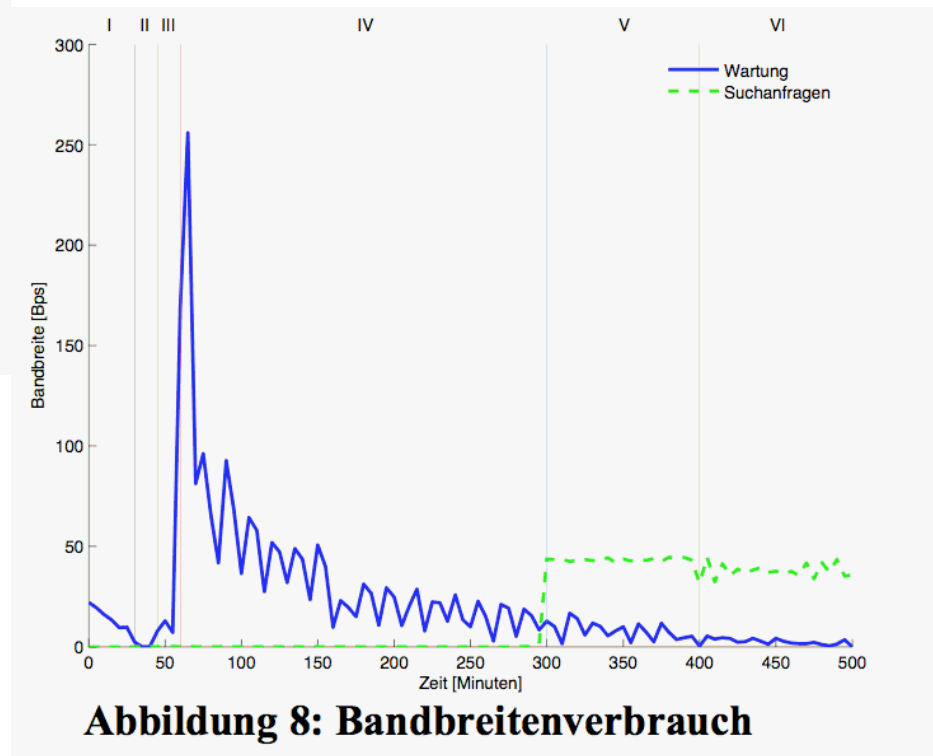
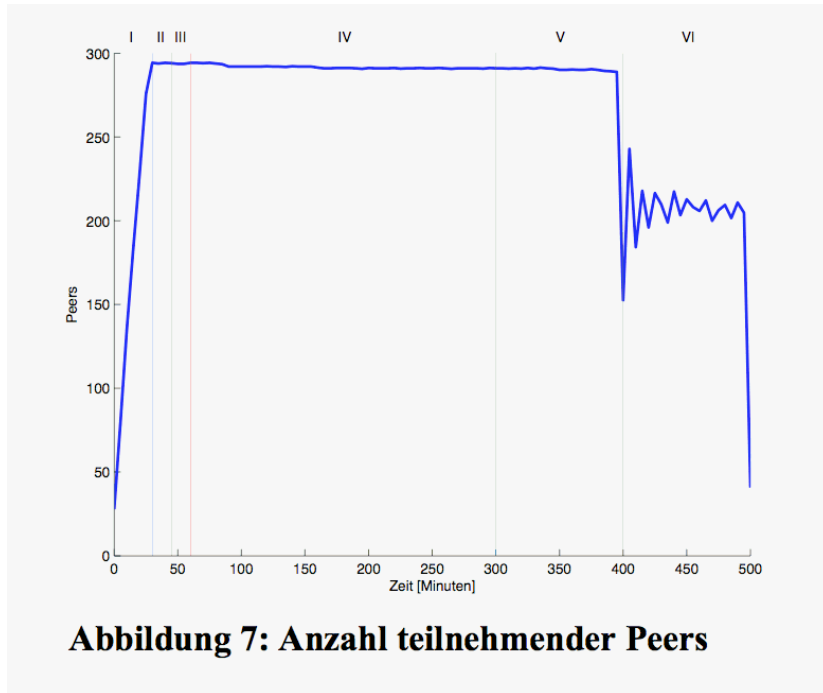


# Ergebnisse





# Bandbreite





# Skip-Net

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

- 
- **SkipNet: A Scalable Overlay Network with Practical Locality Properties**  
**Nicholas J.A. Harvey, Michael B. Jones, Stefan Saroiu, Marvin Theimer, Alec Wolman, 2003**
    - Microsoft Research, Microsoft Corporation, Redmond, WA
    - University of Washington, Seattle, WA
  - **Problem:**
    - Geordnete Speicherung der Daten auf Peers
    - Ohne komplizierte Selbstbalancierung
  - **Lösung**
    - Verwendung des Skip-Graphen
    - Weiterentwicklung der Skip-Liste

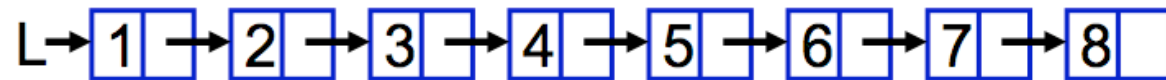




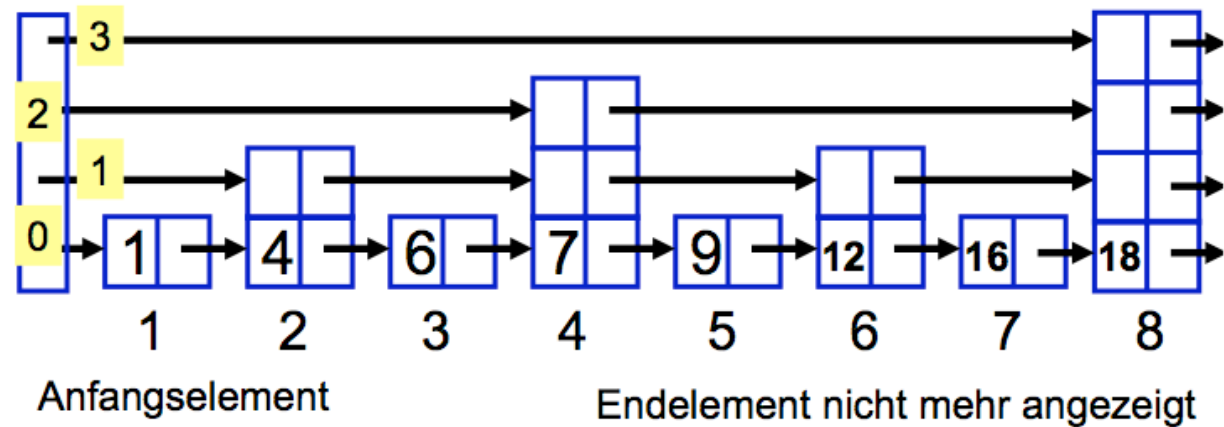
# Skip-Listen

- **Durch Einfügen von zusätzlichen Zeigern in eine einfach verkettete Liste**
- **kann man auch dort effizient suchen**
- **Höhe eines Elements erhält man durch wiederholten Münzwurf:**
  - x-mal Kopf hintereinander entspricht Höhe
- **Erwartete Suchzeit:**
  - $O(\log n)$

Einfach verkettete Liste:



Niveau i des Zeigers





# Skip-Graphs

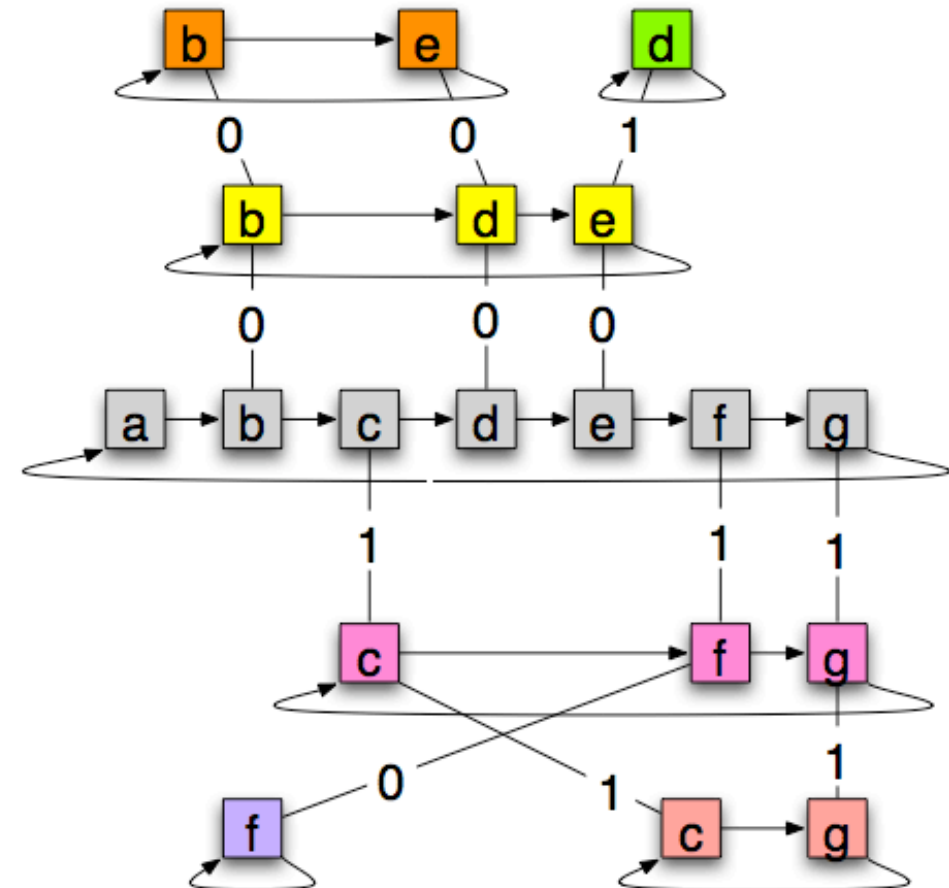
➤ **J. Aspnes and G. Shah. Skip graphs, 2003**

➤ **Idee:**

- Die “Verlierer” des Münzwurfs nehmen an einem eigenen Spiel teil

➤ **Eigenschaften:**

- “unverwüstlich” (highly resilient)
  - hoher Anteil der Knoten kann gelöscht werden bis Netzwerk auseinander fällt
- Durchmesser, Grad:  $O(\log n)$  mit hoher Wahrscheinlichkeit
- Sortierung der Daten bleibt gleich





# Skip-Net

- 
- **Harvey, Jones, Saroiu, Theimer, Wolman, SkipNet: A Scalable Overlay Network with Practical Locality Properties 2003**
  - **Prinzip:**
    - Daten werden sortiert abgelegt auf die Peers auf dem Ring
    - Knoten-ID wird als Skip-Graph-Zufallsbits verwendet
  - **Suche nach Daten:**
    - Wähle den weitest springenden (höchsten) Zeiger, der nicht hinter das Datum fällt
  - **Suche nach (numerischer) Knoten-ID:**
    - Gehe zum Nachbarn bis die erste Ziffer stimmt
    - Dann folge der Struktur eins höher
    - Wiederhole dies mit der nächsten Ziffer
  - **Anzahl Hops: jeweils  $O(\log n)$  mit hoher Wahrscheinlichkeit**
    - bei gleichmäßiger Verteilung der Datenindizes
    - bei zufälliger Wahl der Knoten-IDs

# *Ende der 15. Vorlesung*



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

Peer-to-Peer-Netzwerke  
Christian Schindelhauer  
[schindel@informatik.uni-freiburg.de](mailto:schindel@informatik.uni-freiburg.de)