

# *Peer-to-Peer- Netzwerke*



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

**Christian Schindelhauer**

Sommersemester 2006

7. Vorlesung

17.05.2006

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



# 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
  - Selbstorganisation
  - 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**



# Warum skaliert Gnutella nicht?

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

## ➤ Gnutella

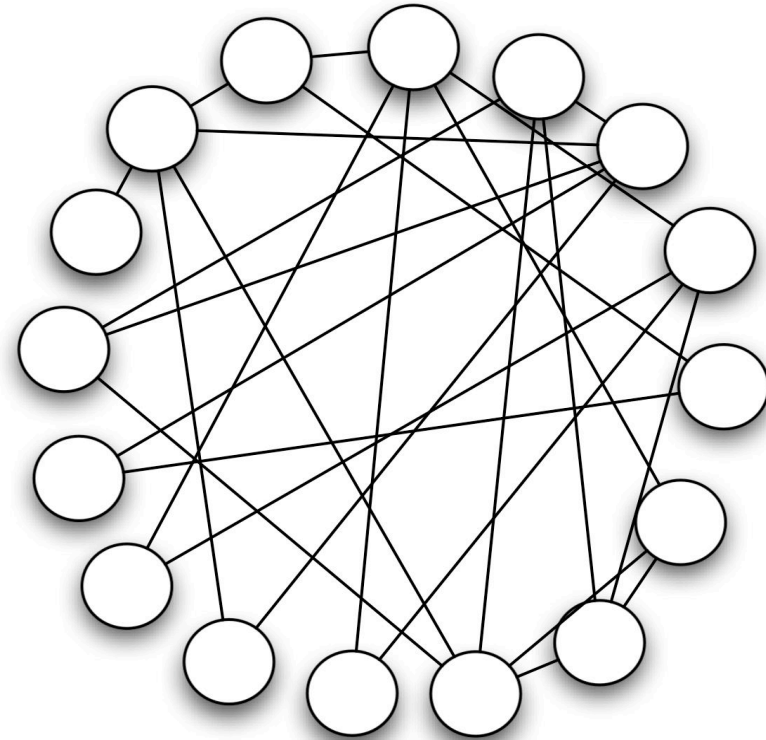
- Graph-Struktur ist zufälliger Verbindungsgraph
- Grad des Graphen klein
- Durchmesser gering
- Zusammenhang groß

## ➤ Suche aber aufwändig

- Um ein Datum sicher zu finden, muss das gesamte Netzwerk durchsucht werden

## ➤ Gnutella skaliert nicht, weil

- Keine Struktur in der Datenablage





# Zwei Fragen zur Informationsfindung

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

➤ **Wo ist es?**

➤ **Wie dorthin kommen?**

➤ **Napster:**

–Wo?

- Auf dem Server 😊

–Wie dorthin?

- Zum Serverstau 😞

➤ **Gnutella**

–Wo?

- Weiss nicht 😞

–Wie dorthin?

- Alle fragen 😞

➤ **Besser:**

➤ **Wo ist Datum  $x$ ?**

–An der Stelle  $f(x)$

–Was ist  $f(x)$ ?

- Eine allen Teilnehmern bekannte Abbildung von  $x$  auf einem Raum

➤ **Wie komme ich dorthin?**

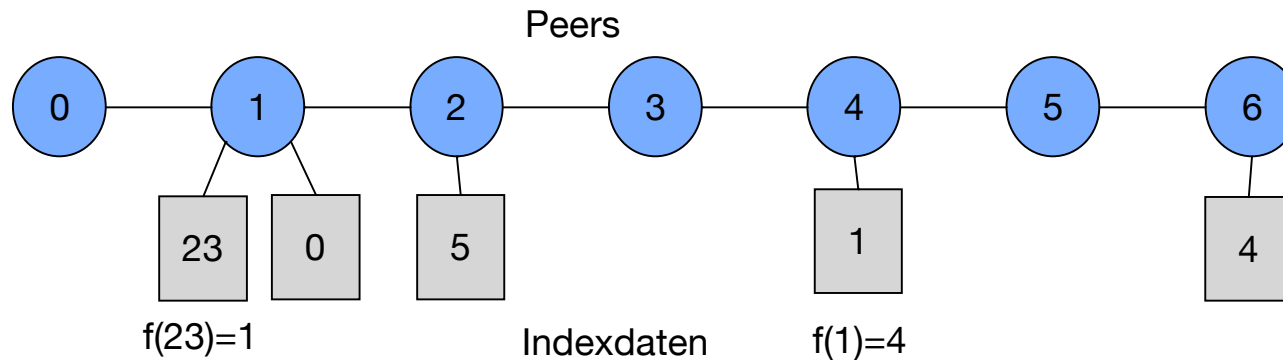
–Durch eine Route die mir den Weg von meinem Standort zu  $f(x)$  aufzeigt.



# Eine Hash-Tabelle als Peer-to-Peer-Netzwerk

## ➤ Jeder Peer steht für eine Speicherstelle $0,1,2,\dots,n-1$

- Eine allen Peers bekannte Hash-Funktion, z.B. für  $n = 7$ 
  - $f(x) = (3x+1 \bmod 23) \bmod 7$
- Peers sind als Kette verbunden



## ➤ Suche

- Berechne  $f(x)$
- Gehe zu Peer mit Adresse  $f(x)$  entlang der Linie



# Distributed Hash-Table (DHT)

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

## Hash-Tabellen

### ➤ Vorteile

- Suche einfach

### ➤ Nachteile

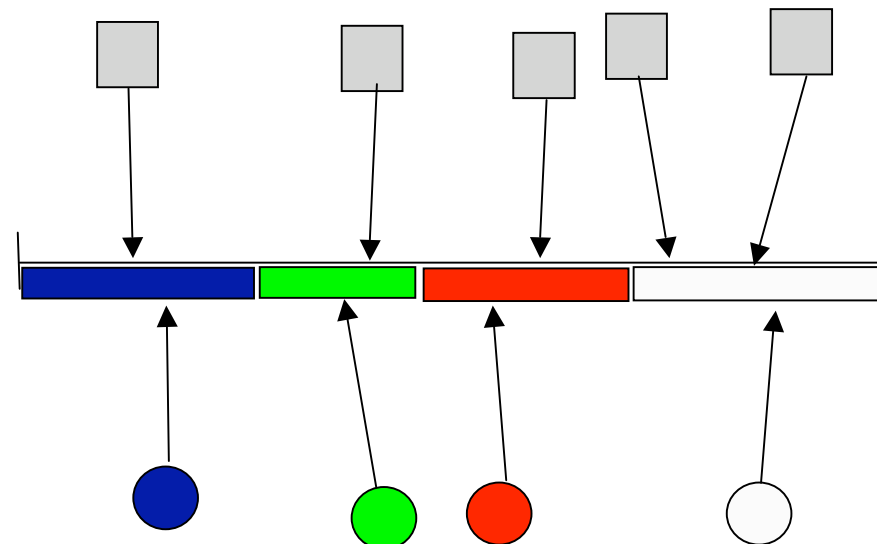
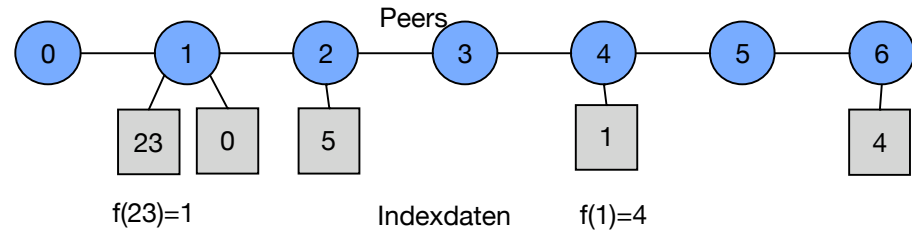
- Ein neuer Peer verursacht neue Wahl der Hash-Funktion
- Lange Wege

## 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



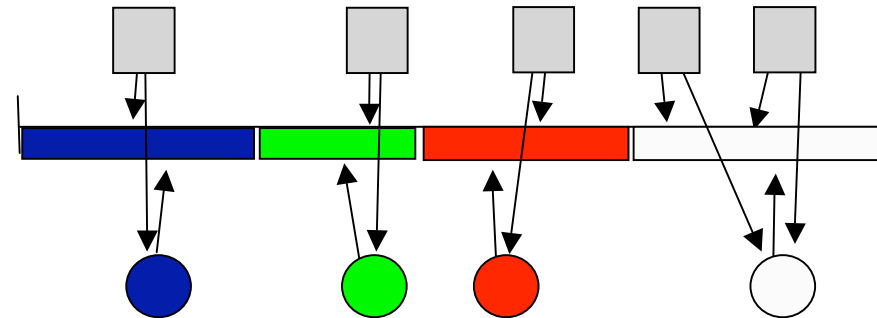


# Einfügen in die Distributed Hash-Table (DHT)

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

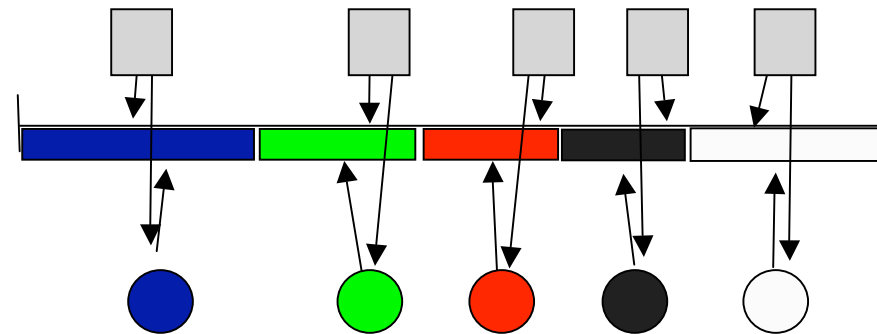
## ➤ Distributed Hash-Table

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



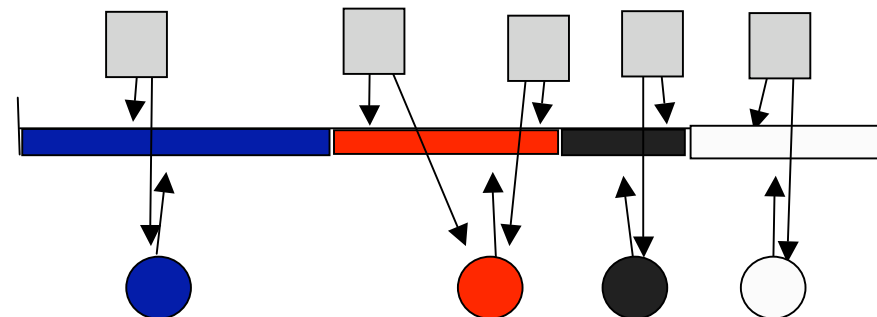
## ➤ Kommt ein neuer Knoten hinzu

- müssen die Nachbarn teilen



## ➤ Verlässt ein Knoten das Netzwerk

- übernehmen die Nachbarn sein Gebiet

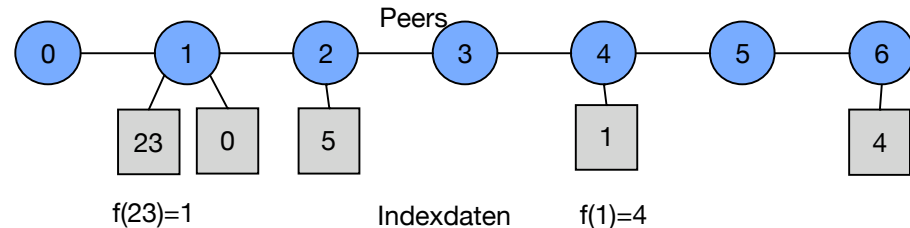




# Eigenschaften DHT

## ➤ Vorteile

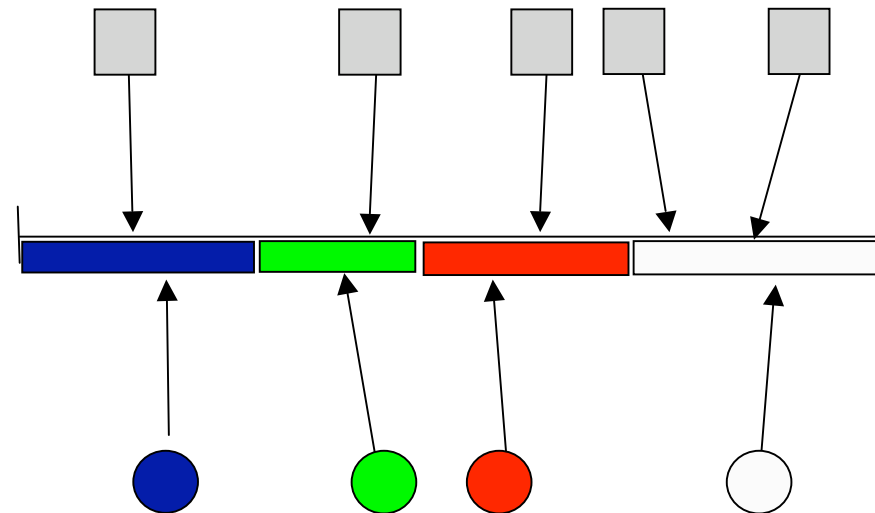
- Jedes Datum kann einem bestimmten Peer zugewiesen werden
- Einfügen und Entfernen von Peers erzeugt nur Veränderungen in den benachbarten Peers



## ➤ 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

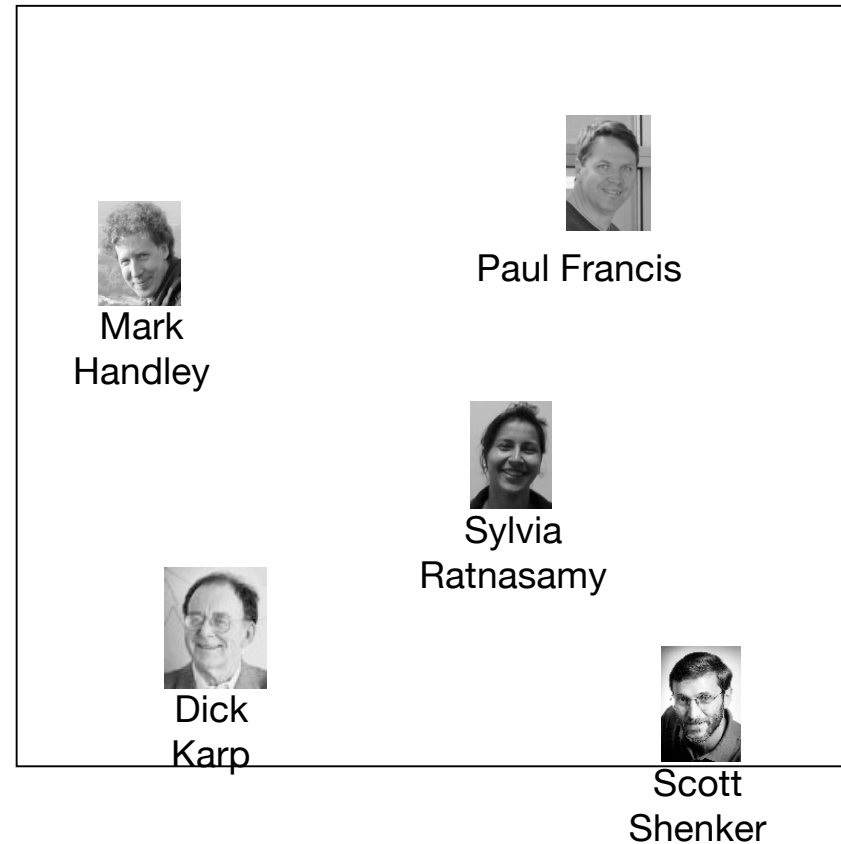




# A Scalable Content Addressable Network (CAN)

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

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

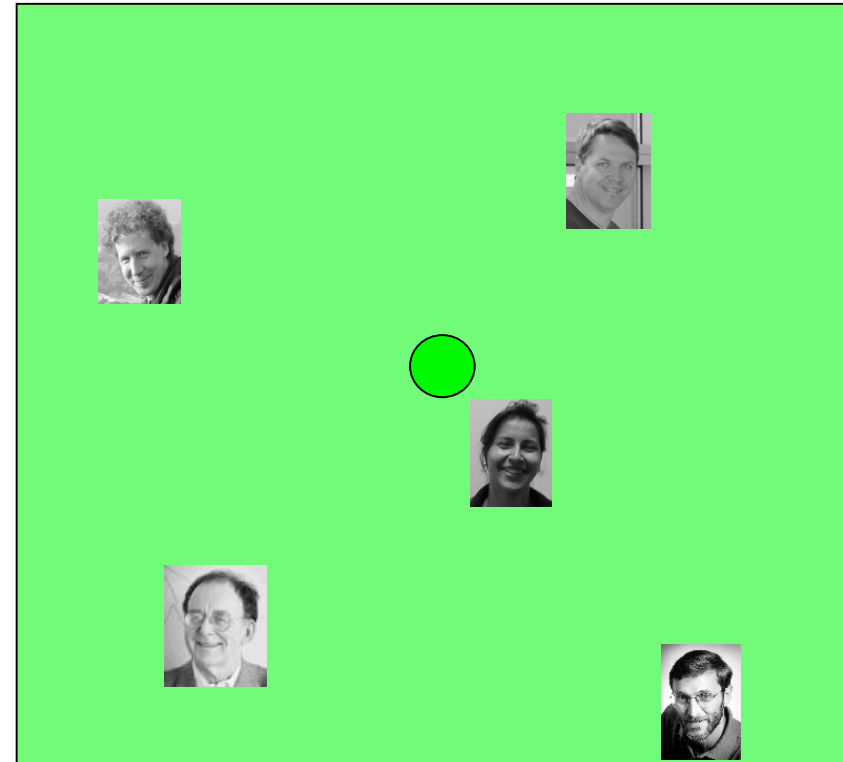




# Content Addressable Network (CAN)

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

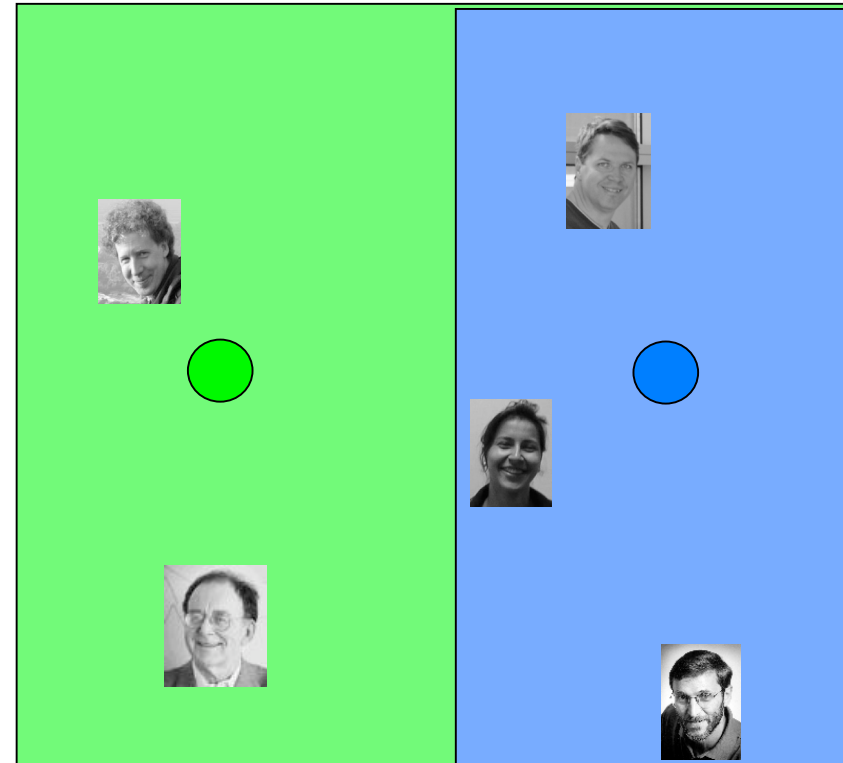




# Content Addressable Network (CAN)

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

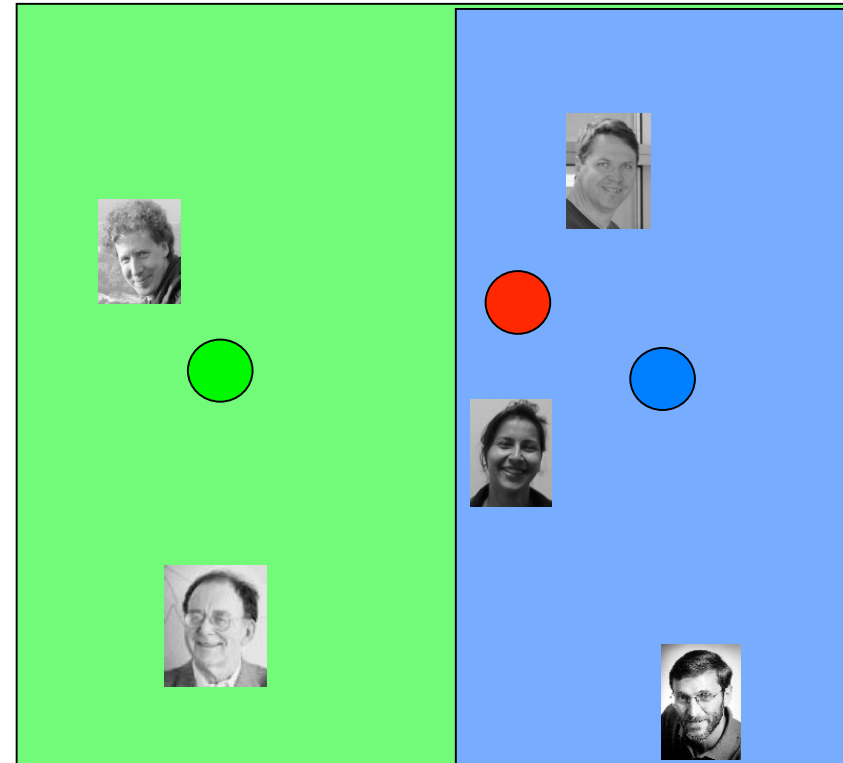




# Content Addressable Network (CAN)

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

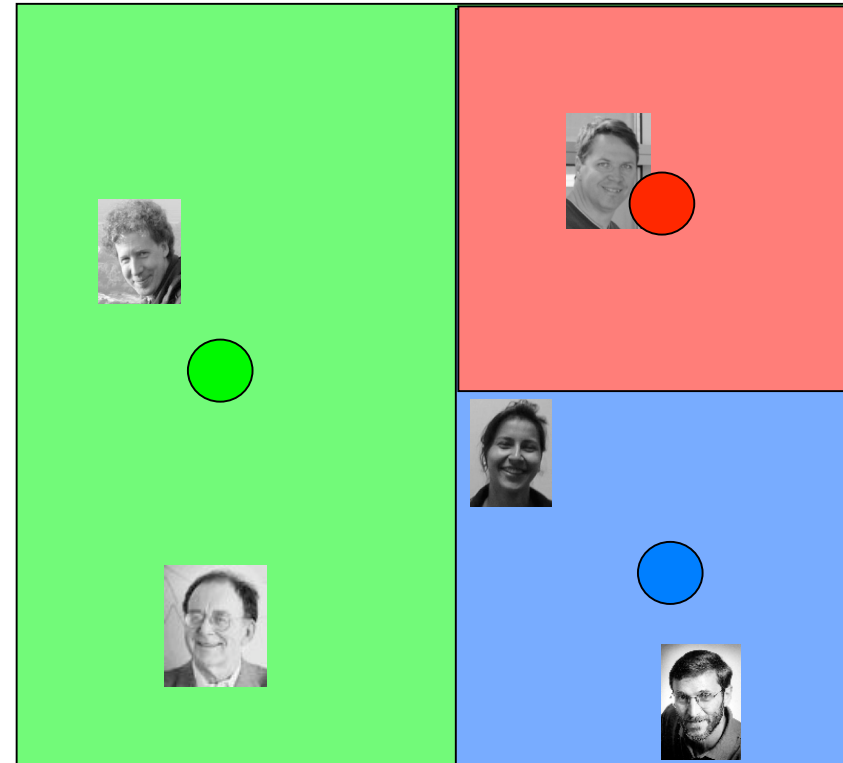




# Content Addressable Network (CAN)

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

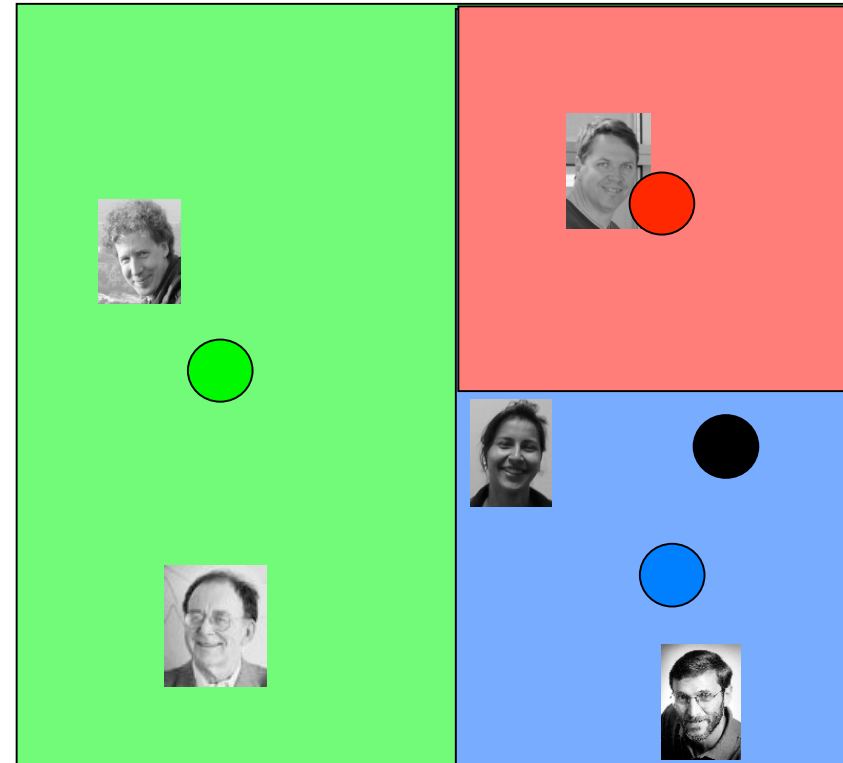




# Content Addressable Network (CAN)

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



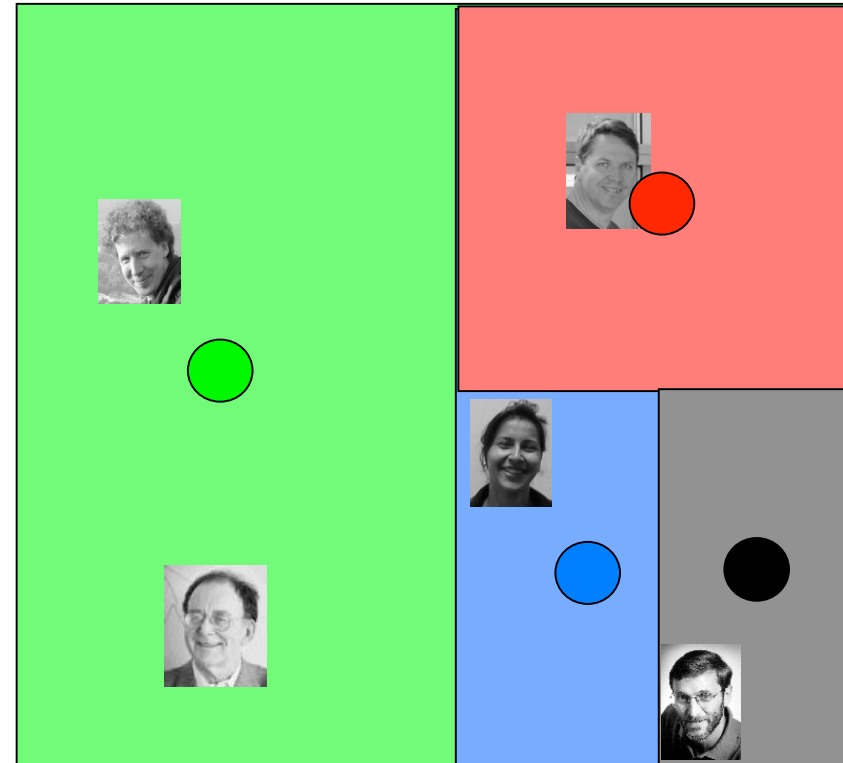




# Content Addressable Network (CAN)

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

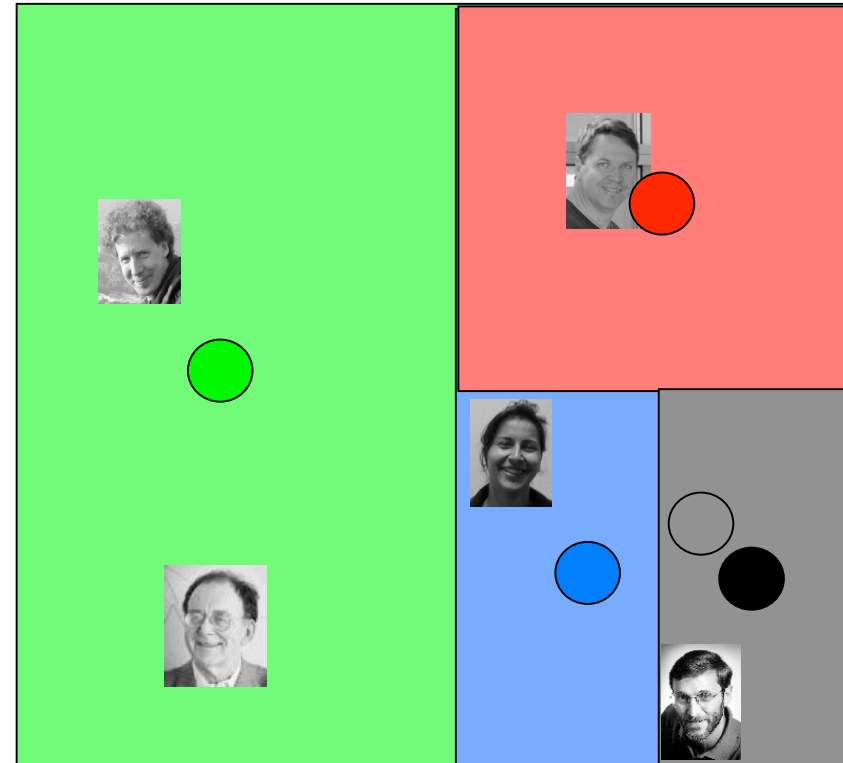




# Content Addressable Network (CAN)

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

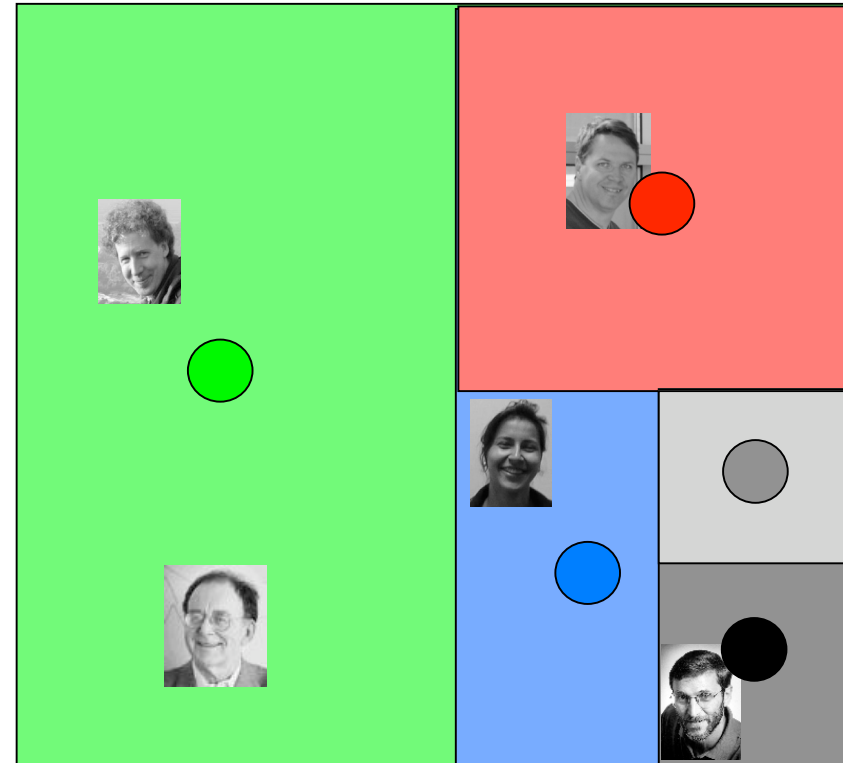




# Content Addressable Network (CAN)

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





# Wie groß/klein können solche Flächen werden

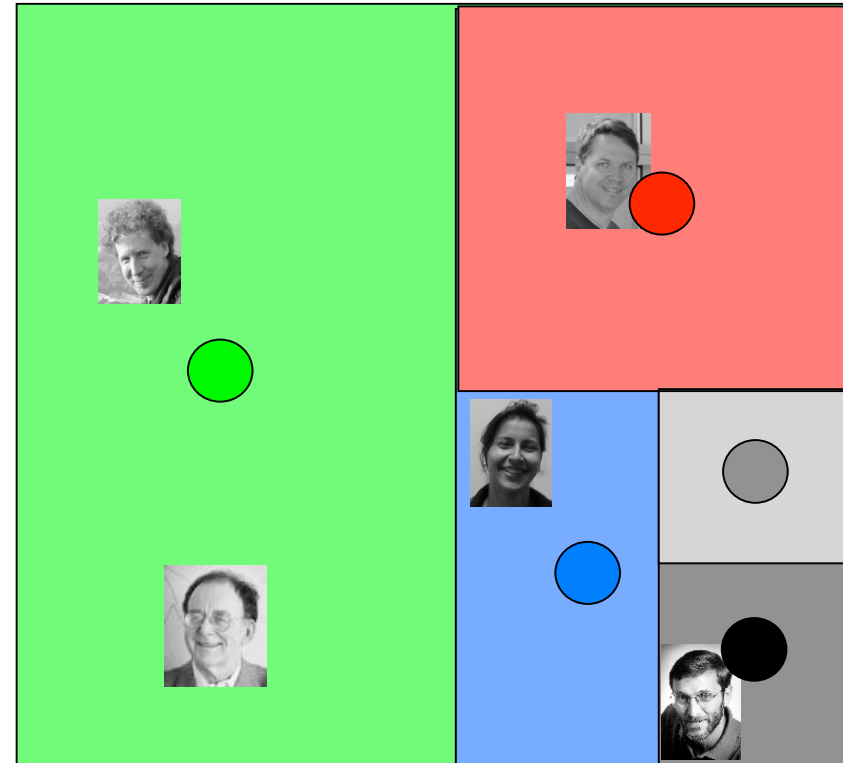
- $R(p)$  : Rechteck eines Peers  $p$   
 $A(p)$  : Fläche des Rechteck eines Peers  $p$   
 $n$  : Anzahl Peers  
Anfangsquadrat: Fläche 1

## Lemma

Für alle Peers  $p$  gilt

- $E[A(p)] = \frac{1}{n}$
- Sei  $P_{R,n}$  die Wahrscheinlichkeit, dass keines der  $n$  Peers in das Rechteck  $R$  hineinfällt. Dann gilt

$$P_{R,n} \leq e^{-n \text{Vol}(R)}$$





# Die erwartete Fläche eines Peers in CAN

Beweis von 1.  $E[A(p)] = \frac{1}{n}$

Seien  $\{1, \dots, n\}$  die Peers.

Dann gilt:

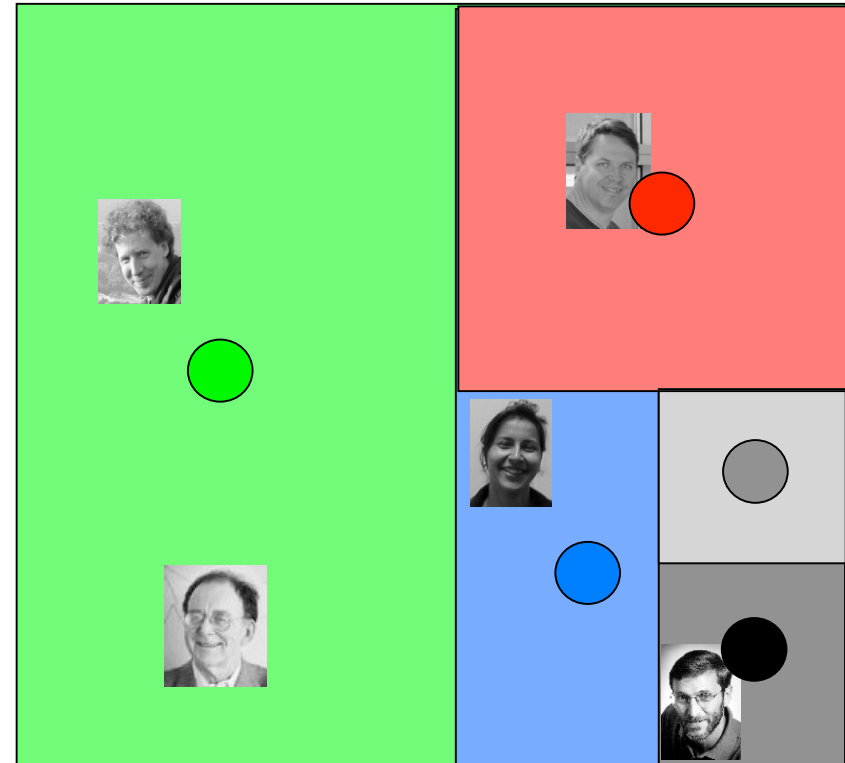
$$\sum_{i=1}^n A(i) = 1$$

Ferner gilt wegen Symmetrie

$$\forall i \in \{1, \dots, n\} : A(i) = A(1)$$

Damit gilt:

$$1 = \sum_{i=1}^n A(i) = E \left[ \sum_{i=1}^n A(i) \right] = \sum_{i=1}^n E[A(i)] = nE[A(1)]$$





# Ein nichtgetroffenes Rechteck

Beweis von 2.

$$P_{R,n} \leq e^{-n \text{Vol}(R)}$$

Betrachte ein Rechteck  $R$  der Fläche  $x = \text{Vol}(R)$

Die Wahrscheinlichkeit, dass ein Peer nicht in diese Fläche fällt, ist

$$1 - x$$

Die Wahrscheinlichkeit, dass  $n$  Peers nicht in  $R$  hineinfallen ist

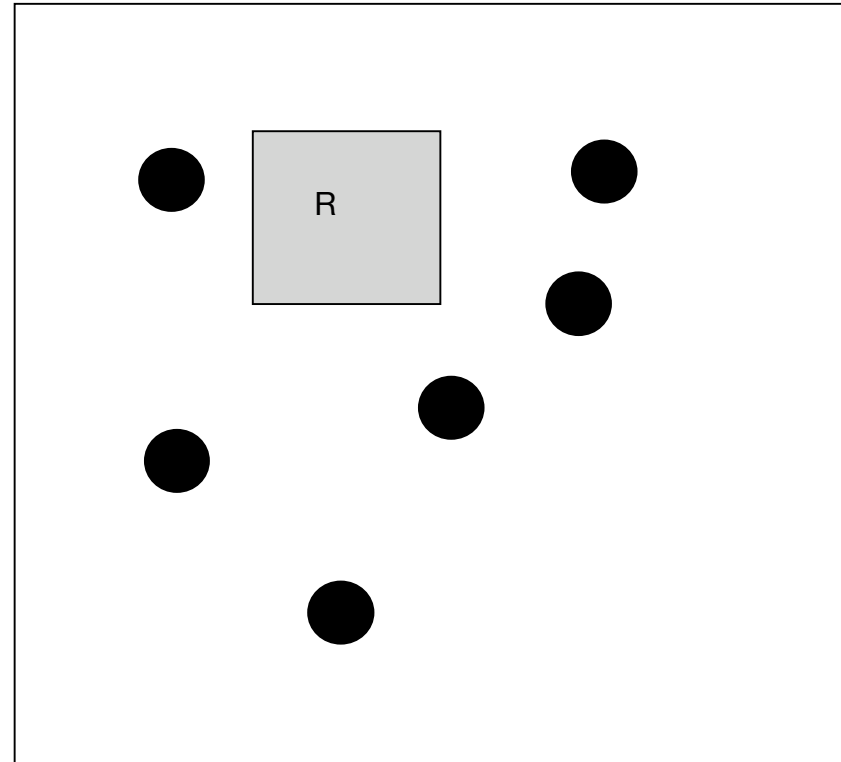
$$(1 - x)^n$$

Damit ist die Wahrscheinlichkeit dafür höchstens

$$(1 - x)^n = \left( (1 - x)^{\frac{1}{x}} \right)^{nx} \leq e^{-nx}$$

weil für alle

$$m > 1 : \left( 1 - \frac{1}{m} \right)^m \leq \frac{1}{e}$$





# Wie groß kann ein nicht getroffenes Rechteck sein?

Aus 2.

$$P_{R,n} \leq e^{-n \text{Vol}(R)}$$

folgt für ein Rechteck  $R_i$  Fläche  $2^{-i}$

$$P_{R_i, c 2^i \ln n} \leq e^{-c 2^i \ln n \text{Vol}(R_i)} = n^{-c}$$

Es genügen also  $c \cdot 2^i \cdot \ln n$  Peers um  $R_i$  mit Wahrscheinlichkeit  $1 - n^{-c}$

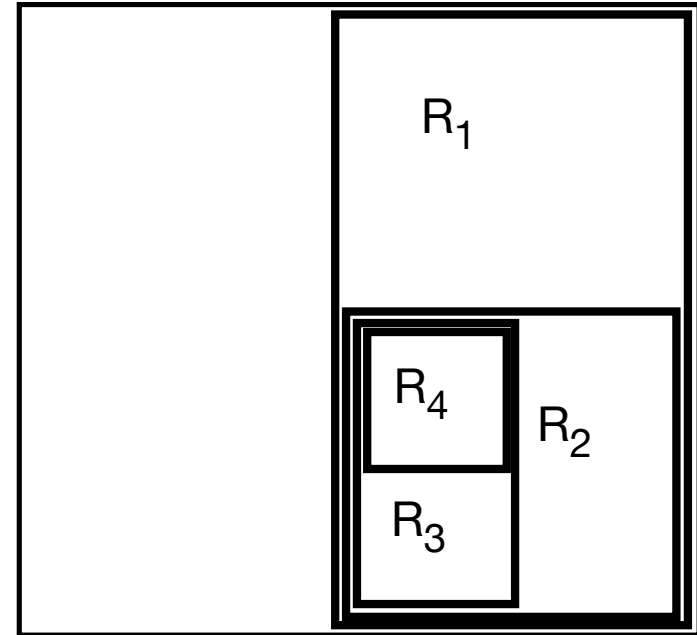
zu teilen. Diese kommen jetzt hintereinander.

Sei nun

$$i \in \left\{ 1, \dots, \log \frac{n}{2c \ln n} \right\}$$

$$\sum_{i=1}^{\log \frac{n}{2c \ln n}} c \cdot 2^i \ln n = c \cdot (\ln n) \cdot \sum_{i=1}^{\log \frac{n}{2c \ln n}} 2^i \leq c \cdot (\ln n) 2^{\log \frac{n}{c \ln n}} = n$$

Damit wird ein Rechteck der Fläche  $\frac{2c \ln n}{n}$  mit W'keit  $n^{-c} \log n$  nicht geteilt





# Wie gleichmäßig werden die Daten verteilt?

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

---

## ➤ Lemma

- Mit Wahrscheinlichkeit  $(\log n) n^{-c}$  wird ein Rechteck der Größe  $2c(\ln n)/n$  nicht geteilt.

## ➤ Wenn $m$ Elemente insgesamt gespeichert werden,

- so erhält jeder Peer also maximal  $2c(\ln n) m/n$  Elemente,
- während der Durchschnitt  $m/n$  Elemente speichert

## ➤ Also speichert jeder Peer höchstens $2c(\ln n)$ mal mehr als der Durchschnittspeer mit hoher Wahrscheinlichkeit.



# *Ende der 7. 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)