Peer-to-Peer-Netzwerke

CoNe Freiburg

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

Christian Schindelhauer
Sommersemester 2006
13. Vorlesung
22.06.2006
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**

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

Viceroy

A Scalable and Dynamic Emulation of the Butterfly

Dahlia Malkhi Moni Naor David Ratajczak

2001

Erreichbarer Durchmesser bei Grad log n

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

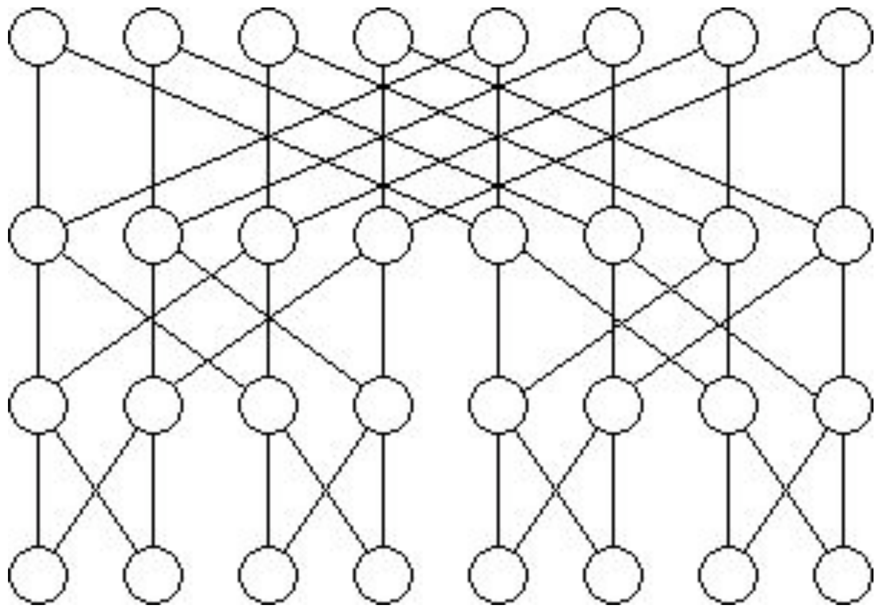
- **CHORD:**
  - Durchmesser O(log n)
  - Grad O(log n)
- **Gesucht:**
  - Netzwerk mit kleinem Ausgrad
  - D.h. Eingrad, Ausgrad konstant
  - Durchmesser O(log n)
- **Lösung:**
  - Viceroy
  - Distance-Halving-Netzwerk

Definition Butterfly-Graph

- **Knoten: (i, S)**
  - –i ist aus {1,..,k}
  - –S ist k-stelliger Binärstring
- **Interpretation**
  - –m = 2^k Knoten pro Ebene
  - –k Ebenen
  - –In der Regel werden die Knoten der k-ten Ebene zweimal gezeichnet
- **Kanten: Von (i, (b_1,..,b_i, ..., b_k)) nach**

  ((i+1) mod k, (b_1,..,b_i, ..., b_k)) und

  ((i+1) mod k, (b_1,..,$\neg b_i$, ..., b_k))



Eigenschaften Butterfly-Graph

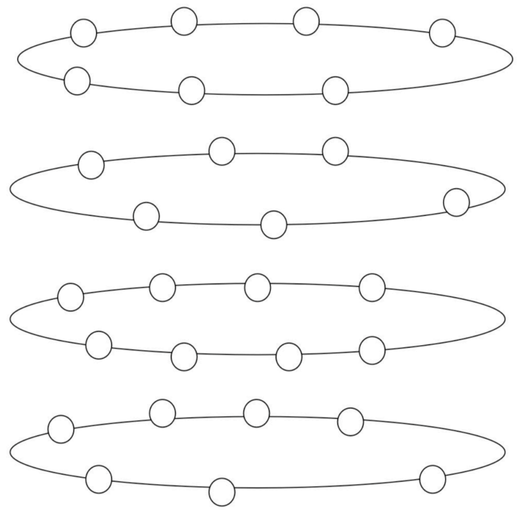
- **Kleiner Grad**
  - Ein+Ausgrad = 4
- **Kleiner Durchmesser**
  - mit log m = log n + log log n optimal
- **Gute Simluationseigenschaften**
  - Andere Netzwerke können sehr effizient in einen Butterfly-Graphen eingebettet werden
  - D.h. Eine Kante eines anderen Netzwerks wird durch sehr kurze Pfade im Butterfly-Graph ersetzt
- **Einfache Routing-Algorithmen**
  - Routing-Entscheidung in konstanter Zeit möglich
- **Keine Flaschhälse**
  - Kein Knoten wird beim Routing mit Nachrichten überlastet
- **Hohe Fehlertoleranz**
  - Ausfall von Knoten kann gut toleriert werden

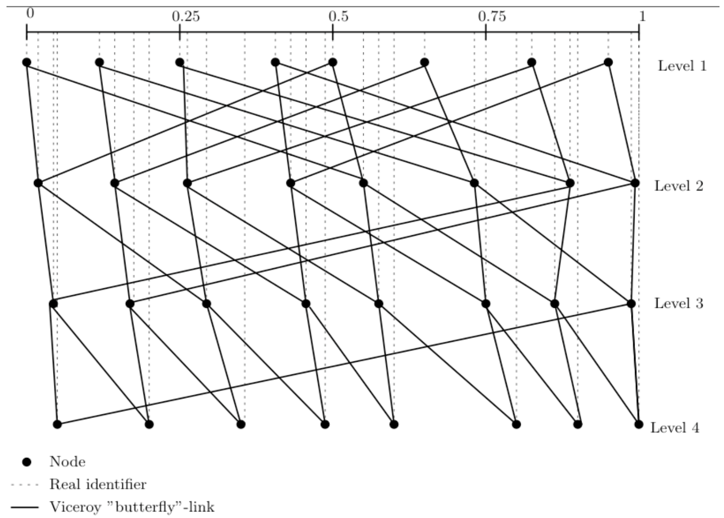
Übersicht Viceroy

- **Zielsetzung**
  - Skalierbarkeit
  - Bewältigung von Dynamik
  - Verteilung Traffic
- **Congestion**
  - Max. Anzahl Nachrichten die ein Peer zu bewältigen hat
- **Kosten für Peer-Aufnahme/Entfernen**
  - Minimierung Anzahl Nachrichten/Zeit
- **Länge des Suchpfads**
  - Entspricht dem Begriff der Dilation
- **Viceroy war das erste P2P-Netzwerk mit optimaler Grad/Durchmesser-Beziehung**
  - Techniken lassen sich auf andere Netzwerke übertragen

Aufbau Viceroy

- **Knoten in Viceroy**
  - Wählen anfangs zufällig eine Ebene des Butterfly-Netzwerks
  - (Hierzu ist es notwendig, log n zu kennen)
- **Kombination aus drei Netzwerk-Strukturen**
  1. Ein Ring für alle Knoten
     - Verkettet alle Knoten
  2. Für jede Ebene ein Ring
     - mit dem Intervall [0,1)
  3. Das Butterfly-Netzwerk zwischen den Ebenen,
     - d.h.in Ebene i gibt es Links von (i,x) zu
       1. Nachfolger von (i+1,x)
       2. Nachfolger von (i+1,x+2^i)
       3. Vorgänger von (i-1, x)
       4. Vorgänger von (i-1, x-2^i)





Berechnung von log n

- **Betrachte Ringnachbar auf dem Ring für alle**
- **Sei d der Abstand auf dem Ring [0,1)**
- **Dann gilt**
  - $E[d] = 1/n$
  - $P[d > c (\log n)/n] < n^{-O(1)}$
  - $P[d < 1/n^{1+c}] < n^{-c}$
- **Damit ist -log d mit hoher Wahrscheinlichkeit eine (1+ε)-Approximation von log n**

Einfügen eines Peers

1. **Füge Knoten an zufällige Stelle des Rings für alle ein**
2. **Bestimme log n durch Messen des Abstands zweier Peers**
3. **Wähle zufällige Ebene i uniform aus [1,..,log n]**
4. **Suche Stelle im ViceRoy-Netzwerk durch LookUp ausgehend vom Ringnachbarn**
5. **Füge Peer in die Ebene des ViceRoy-Netzwerks, indem**
   - der Peer in den Ring i des Netzwerks eingefügt wird
   - die Ausgangszeiger von (i,x)
     - Nachfolger von (i+1,x)
     - Nachfolger von (i+1,x+2^i)
     - Vorgänger von (i-1, x)
     - Vorgänger von (i-1, x-2^i)
   - ausgehend von den Zeigern des Nachbarn im Ring i gesucht werden

➢ **Laufzeit/Anzahl Nachrichten**
   - Zeit von Lookup (O(log n)) +
   - Finden der der Nachfolger/Vorgänger (O(log n))

Suche

- **Peer (i,x) erhält Suchanfrage nach (j,y)**

```
IF i=j und |x-y| ≤ (log n)²/n THEN
    Leite Suchanfrage auf Ringnachbar der Ebene i weiter
ELSE
    IF y weiter rechts als x+2^i THEN
        Leite Suchanfrage an Nachfolger von (i+1,x+2^i) weiter
    ELSE
        Leite Suchanfrage an Z = Nachfolger von (i+1,x) weiter
        IF Nachfolger Z weiter rechts als x THEN
            Suche auf dem Ring (i+1) von Z aus einen Knoten (i+1,p) mit p<x
        FI
    FI
FI
```

Lemma

Mit hoher Wahrscheinlichkeit benötigt Suche O(log n) Zeit und Nachrichten

Eigenschaften von ViceRoy

- **Ausgrad konstant**
- **Erwarteter Eingrad konstant**
- **Durchmesser O(log n)**
- **Durch den Butterfly-Graph kann die Kommunikationslast balanciert werden**
  - Butterfly-Graphen betten beliebige andere Graphen effizient ein

Die Sache mit dem Grad

- **Ausgrad: 2+2+2+2 = 8**
- **Bei perfekter Verteilung, d.h. Abstand zum Nachbarn $\Theta(1/n)$**
  - Eingrad konstant
- **Tatsächlich**
  - Eingrad: erwartet konstant
  - aber auch $\Omega(\log n)$ kommt mit Wahrscheinlichkeit $1-n^{-1}$ vor
- **Problem:**
  - Große Abstände auf den Viceroy-Ring ziehen viele Links an
  - Kleine Abstände bombardieren die Ringnachbarn
- **Lösung:**
  - Ordne die Peers **nicht** durch Hashing zu
  - Verwende das Prinzip der mehrfachen Auswahl (multiple choice).

Das Prinzip der mehrfachen Auswahl

- **Beim Einfügen würfelt jeder Peer c log n Positionen**
- **Für jede Position p(j) wird die Abstand a(j) zwischen potentiellen linken und rechten Nachbar gemessen**
- **Peer wird an einer Position p(j) in der Mitte zwischen linken und rechten Nachbarn eingefügt, wo a(j) maximal in der Auswahl ist**

Lemma

Mit hoher Wahrscheinlichkeit ist der Abstand zweier Peers im den log n Viceroy-Ringen nur einen konstanten Faktor größer oder kleiner als der durchschnittliche Abstand (log n)/n.

Beweis der Korrektheit des Prinzips der mehrfachen Auswahl

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

Lemma

Nach Einfügen von n Peers auf einem Ring [0,1), wobei das Prinzip der mehrfachen Auswahl angewendet wird, bleiben nur Intervalle der Größe 1/(2n), 1/n und 2/n übrig.

1. Teil: Mit hoher W'keit gibt es kein Intervall länger als 2/n:

Beweis folgt aus folgenden Lemma:

Lemma*

Sei das längste Intervall c/n (c kann von n abhängig sein).
Dann sind nach Einfügen von 2n/c Peers alle Intervalle kürzer als c/(2n) mit hoher W'keit.

Wendet man dieses Lemma für c=n/2,n/4, ...,4 an, folgt der erste Teil des Lemmas

Beweis der Korrektheit des Prinzips der mehrfachen Auswahl

Lemma

Nach Einfügen von n Peers auf einem Ring [0,1), wobei das Prinzip der mehrfachen Auswahl angewendet wird, bleiben nur Intervalle der Größe 1/(2n), 1/n und 2/n übrig.

Beweis

2. **Teil: Keine Intervalle kleiner als 1/(2n) entstehen**
   Die Gesamtlänge der Intervalle der Größe 1/(2n) ist vor jedem Einfügen höchstens n/2

   Ein solches Intervall wird mit Wahrscheinlichkeit höchstens 1/2 gewählt

   Die Wahrscheinlichkeit, dass unter c log n Intervallen nur solche gewählt werden, ist

$$2^{-c\log n} = n^{-c}$$

   Damit wird für c>1 mit polynomiell kleiner Wahrscheinlichkeit ein Intervall der Länge 1/(4m) weiter unterteilt.

Chernov-Schranke

- **Bernoulli-Experiment**
  - Entweder 0 mit Wahrscheinlichkeit 1-p
  - Oder 1 mit Wahrscheindlichkeit p

Theorem 3 Chernoff-Schranke
Seien $x_1, \ldots, x_n$ unabhängige Bernoulli-Experimente mit $\mathrm{P}[x_i = 1] = p$ *und* $\mathrm{P}[x_i = 0] = 1 - p$*. Sei* $S_n = \sum_{i=1}^{n} x_i$.

1. *Dann gilt für jedes* $c \in [0, 1]$:

$$\mathbf{P}[S_n \ \geq \ (1 + c)\mathbf{E}[S_n]] \ \leq \ e^{-\frac{c^2}{3}pn} \ .$$

2. *Für* $c \in [0, 1]$:

$$\mathbf{P}[S_n \leq (1 - c)\mathbf{E}[S_n]] \ \leq \ e^{-\frac{c^2}{2}pn} \ .$$

Beweis von Lemma*

Sei das längste Intervall c/n (c kann von n abhängig sein). Dann sind nach Einfügen von 2n/c Peers alle Intervalle kürzer als c/(2n) mit hoher W'keit.

- Betrachte ein Intervall der Länge c/n
- Mit W'keit c/n wird ein solches Intervall von einem Peer getroffen
- Angenommen es werden für jeden Peer t log n Intervalle untersucht
- Die erwartete Anzahl von Treffern in

$$E[X] = \frac{c}{n} \cdot \frac{2n}{c} \cdot t \log n = 2t \log n$$

- Die Chernov-Schranke liefert dann

$$P[X \leq (1-\delta)E[X]] \leq n^{-\delta^2 t}$$

- Ist $\delta^2 t \geq 2$ werden all diese Intervall mindestens $2(1-\delta)t \log n$-mal getroffen
- Jedesmal wenn ein Intervall geteilt wird, gab es t log n (zusätzliche) Treffer
- Wir wählen $2(1-\delta) \geq 1$
- Dann wird jedes Intervall (mit hoher W'keit) der Länge c/n geteilt.

Resumee ViceRoy

- **Butterfly-Graph**
  - für Routing sehr gut geeignet
  - viele erprobte gute Algorithmen bekannt
- **Erstes Peer-to-Peer-Netzwerk mit konstanten Ein- und Ausgrad**
- **Aber:**
  - Mehrfache Ringstruktur relativ kompliziert
  - Durch das Prinzip der mehrfachen Auswahl ist der Aufwand zum Einfügen $O(\log^2 n)$
  - Das Prinzip der mehrfachen Auswahl kann auch bei DeBrujin-Netzwerken angewendet werden
    - Diese sind wesentlich einfacher
    - Haben die gleichen Netzwerkeigenschaften

Distance Halving

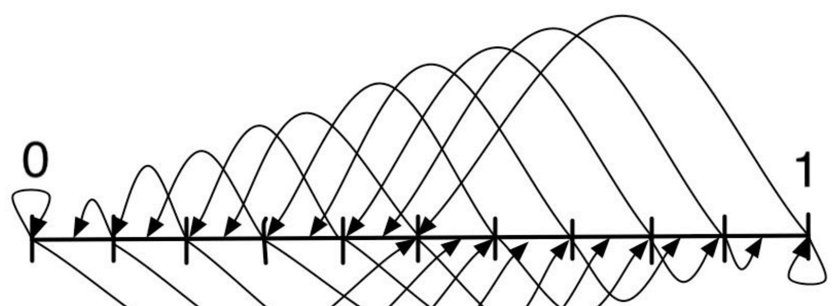
Moni Naor Udi Wieder

2003

Kontinuierliche Graphen

- **Sind unendliche Graphen mit kontinuierlicher Knotenmenge und Kantenmenge**

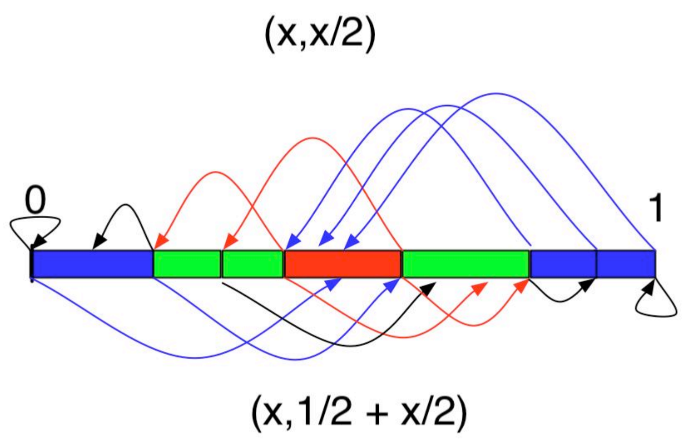
- **Der verwendete Graph**
  - $x \in [0,1)$
  - Kanten:
    - (x,x/2), genannt Links-Kanten
    - (x,1+x/2), genannt Rechts-Kanten
  - Und entsprechende Kanten in Gegenrichtung
    - (x/2,x)
    - (1+x/2,x)

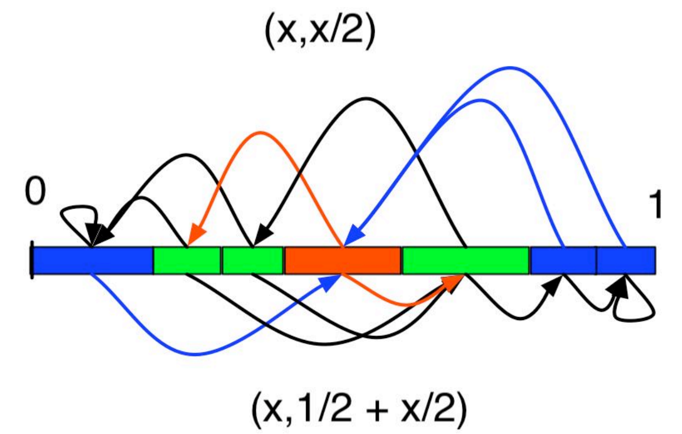


Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Der Ubergang vom kontinuierlichen zum diskreten Fall

- **Betrachte diskrete Teilintervalle des kontinuierlichen Raums**
- **Füge Kante zwischen Teilintervall A und B ein,**
  - falls es $x \in A$ und $y \in B$ gibt, so dass (x,y) Kante des kontinuierlichen Graphs ist
- **Teilintervalle entstehen durch sukzessive Halbierung**
- **Damit ist der Knotengrad von Distance-Halving konstant,**
  - falls das Verhältnis aus größten und kleinsten Intervall konstant ist
- **Durch Mehrfache Auswahl (multiple choice principle) kann dies erreicht werden**





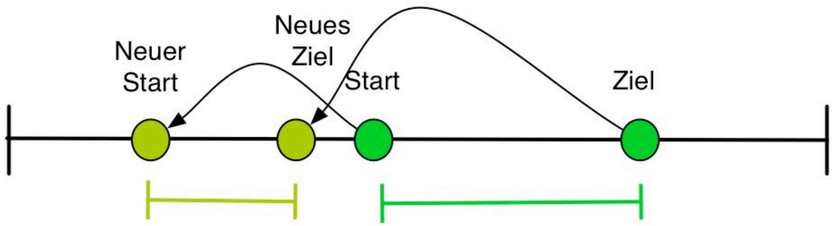
Aufbau von Distance-Halving

- Peers werden den Teilintervallen zugewiesen
- Zusätzlich werden benachbarte Teilintervalle durch Doppelverkettung verbunden
- Das größte Intervall hat (mit hoher Wahrscheinlichkeit) Länge 2/n
- Das kleinste Intervall (mit hoher W'keit) Länge 1/(2n)
- Damit ist der Ein- und Aus-grad konstant
- Der Durchmesser ist logarithmisch

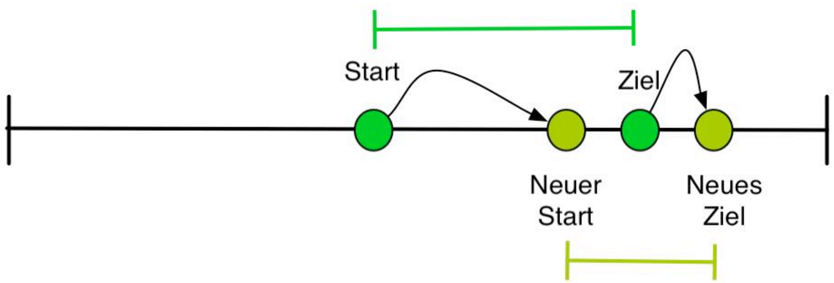
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Lookup in Distance-Halving

- Durch Abbildung der Links-Kanten wird der Start-Ziel-Anstand halbiert
- Folge den Links-Kanten 2+log n Schritte
- Dann kann durch einen Schritt auf dem Ring der aktuell Ziel-Peer erreicht werden
- Zum Schluss wird auf den Links-Kanten rückwärts das Ziel angesteuert



- Genau die selbe Beobachtung gilt für die Rechtskanten



Lemma

Lookup funktioniert in O(log n) Hops und Nachrichten

Congestion beim Lookup

- **Auch können Links- und Rechts-Kanten beliebig abwechselnd benutzt werden (wenn die umgedrehte Reihenfolge rückwärts beachtet wird).**
- **Analog zu Valiants Routing-Ergebnis auf dem Hyper-Würfel kann man hier zeigen**
- **Die Congestion ist höchstens O(log n),**
  - d.h. Jeder Peer befördert höchstens O(log n) Pakete, wenn jeder Peer eine Anfrage stellt
- **Das selbe Resultat kann man für Viceroy zeigen**

Einfügen von Peers in Distance-Halving

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

1. **Führe mehrfache Auswahl durch,**
   - d.h. c log n Anfragen an zufällige Intervalle
   - Wähle größtes Intervall aus und
   - halbiere dieses Intervall
2. **Aktualisierung Ring-kanten**
3. **Aktualisierung Links- und Rechts-Kanten**
   - Durch Verwendung der Links- und Rechts-Kanten des Nachbarn

Lemma

Einfügen von Peers in Distance-Halving benötigt Zeit und Nachrichten $O(\log^2 n)$

Resumee Distance-Halving

- **Einfaches effizientes P2P-Netzwerk**
  - Grad O(1)
  - Durchmesser O(log n)
  - Lastbalancierung
  - Lookup O(log n)
  - Einfügen $O(\log^2 n)$

- **Das Prinzip des Übergangs von kontinuierlichen zu diskreten Graphen schon bekannt**
  - Chord
  - Koorde
  - ViceRoy
- **Hier aber erstmals formalisiert**

Ende der
13. Vorlesung

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

Peer-to-Peer-Netzwerke

Christian Schindelhauer
schindel@informatik.uni-freiburg.de