Lecture 13

Internet Protokolle II

Distributed Hash Tables

Thomas Fuhrmann

Network Architectures
Computer Science Department
Technical University Munich

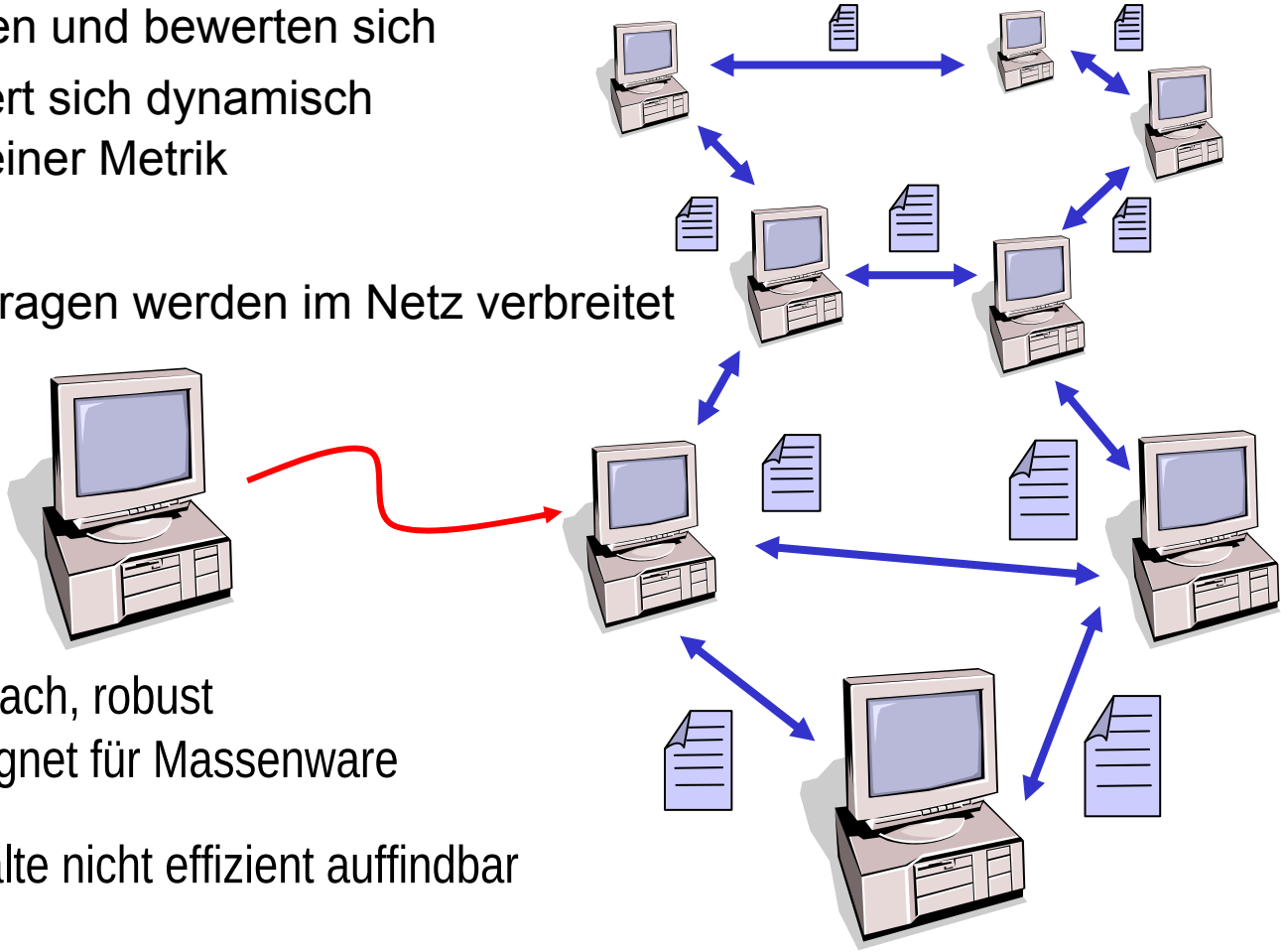
Unstrukturierte Peer-to-Peer-Netze

Netzwerkaufbau

- Neuer Peer kontaktiert einen existierenden Peer
- Peers empfehlen und bewerten sich
- Topologie ändert sich dynamisch entsprechend einer Metrik

Inhaltssuche

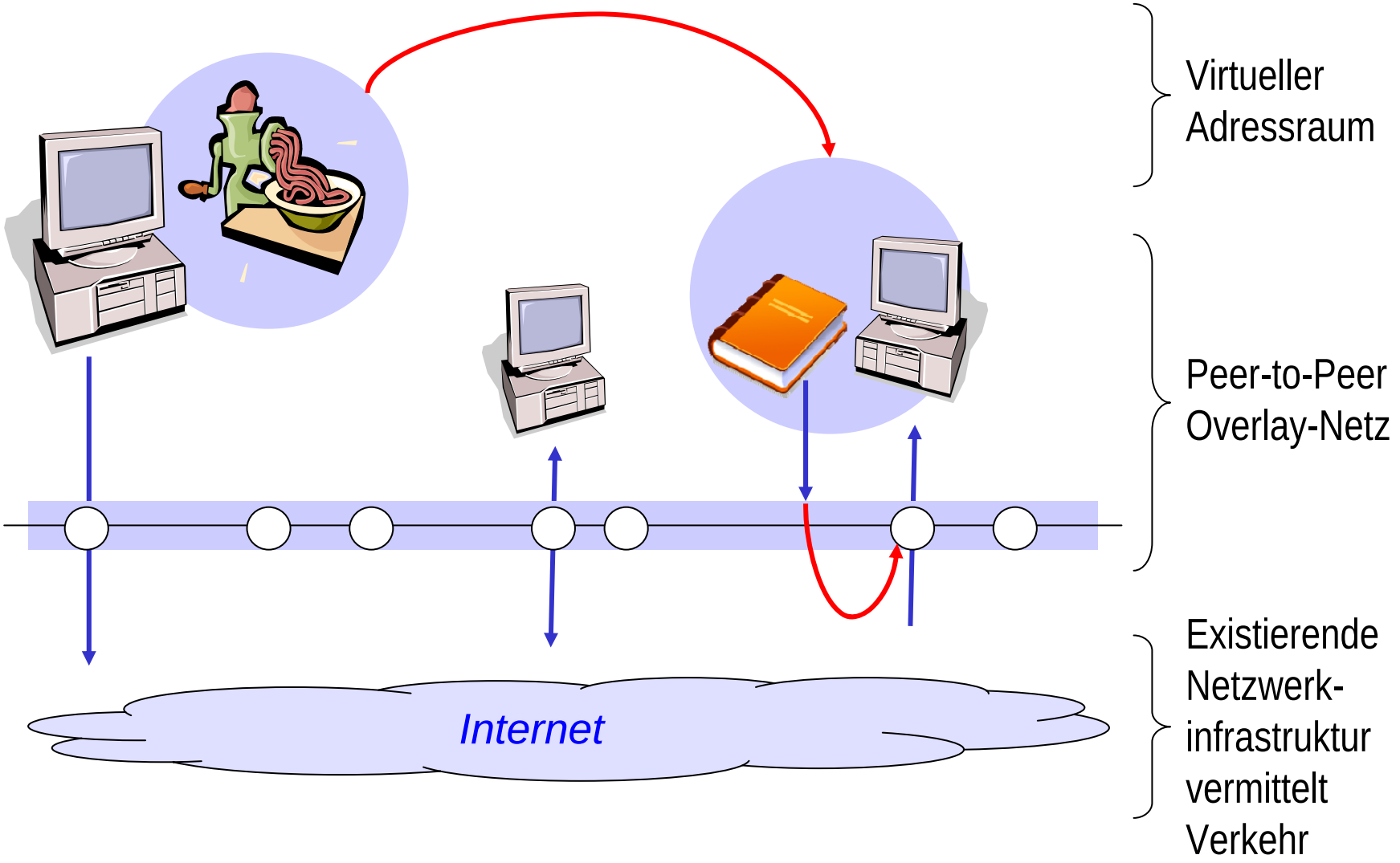
- Inhalte und Anfragen werden im Netz verbreitet



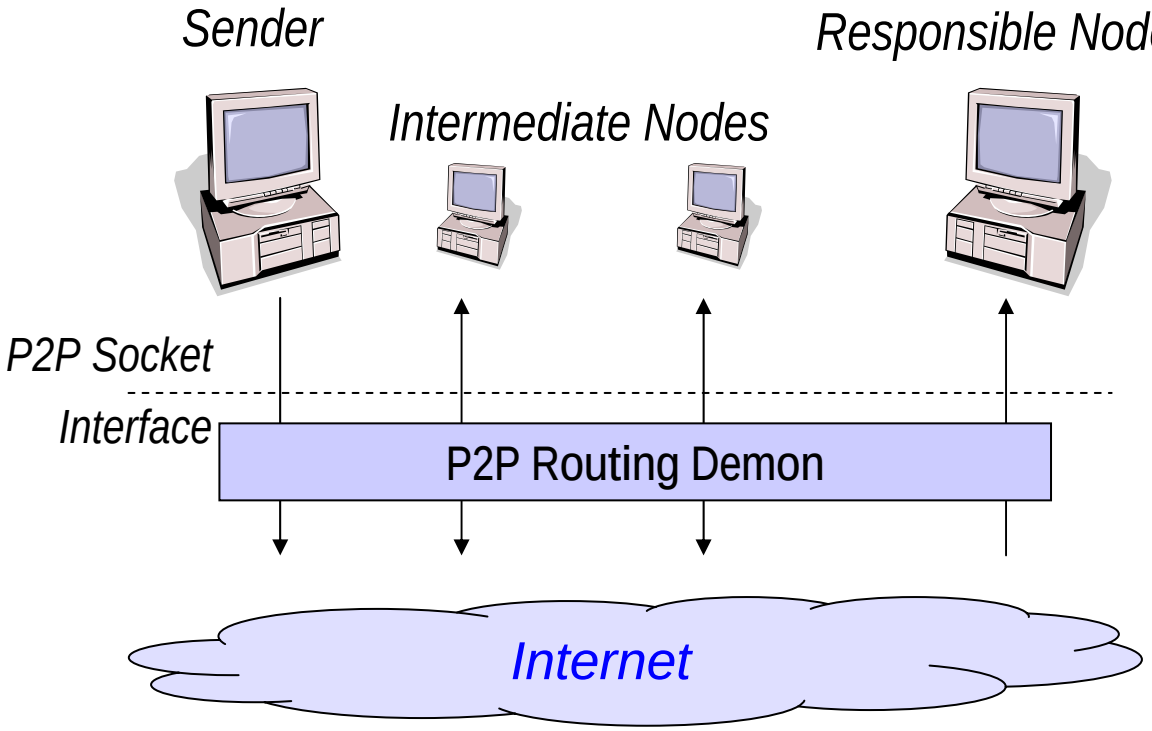
✓ Spontan, einfach, robust
sehr gut geeignet für Massenware

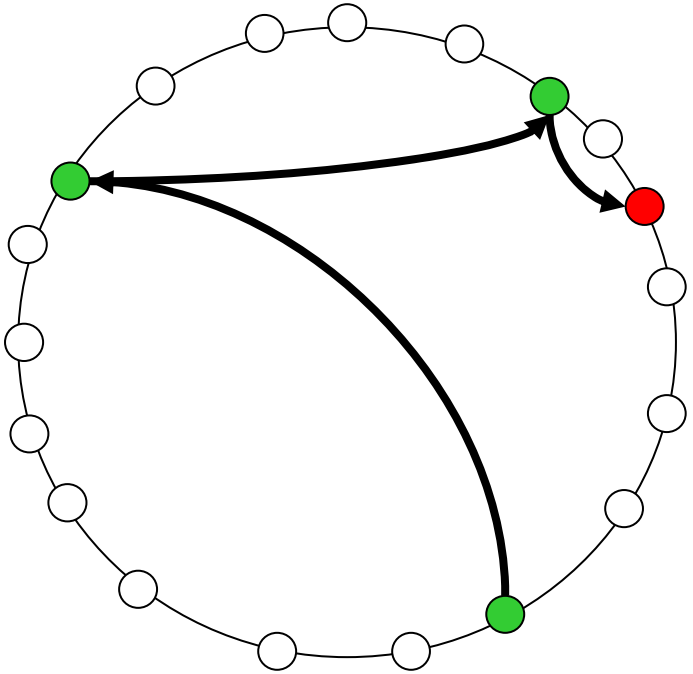
👎 Einzelne Inhalte nicht effizient auffindbar

Strukturierte Peer-to-Peer-Netze

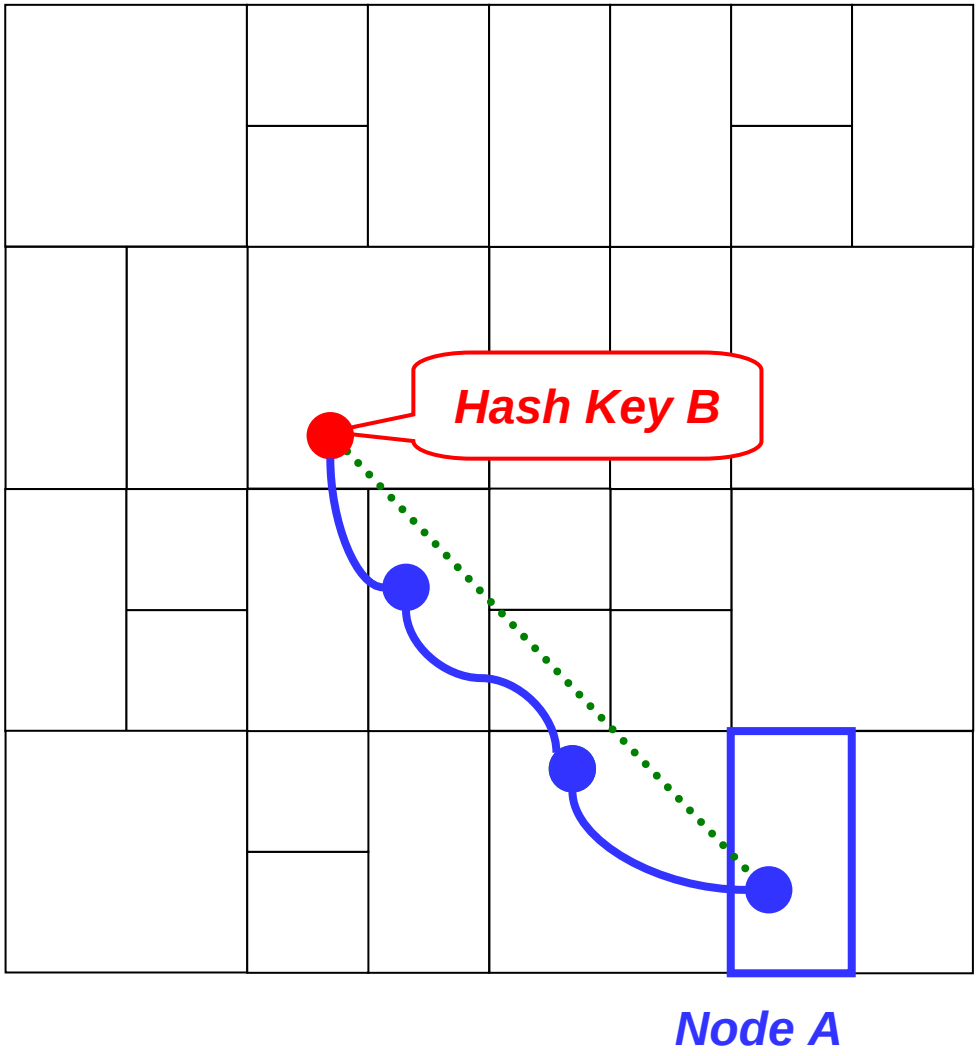


Key Based Routing

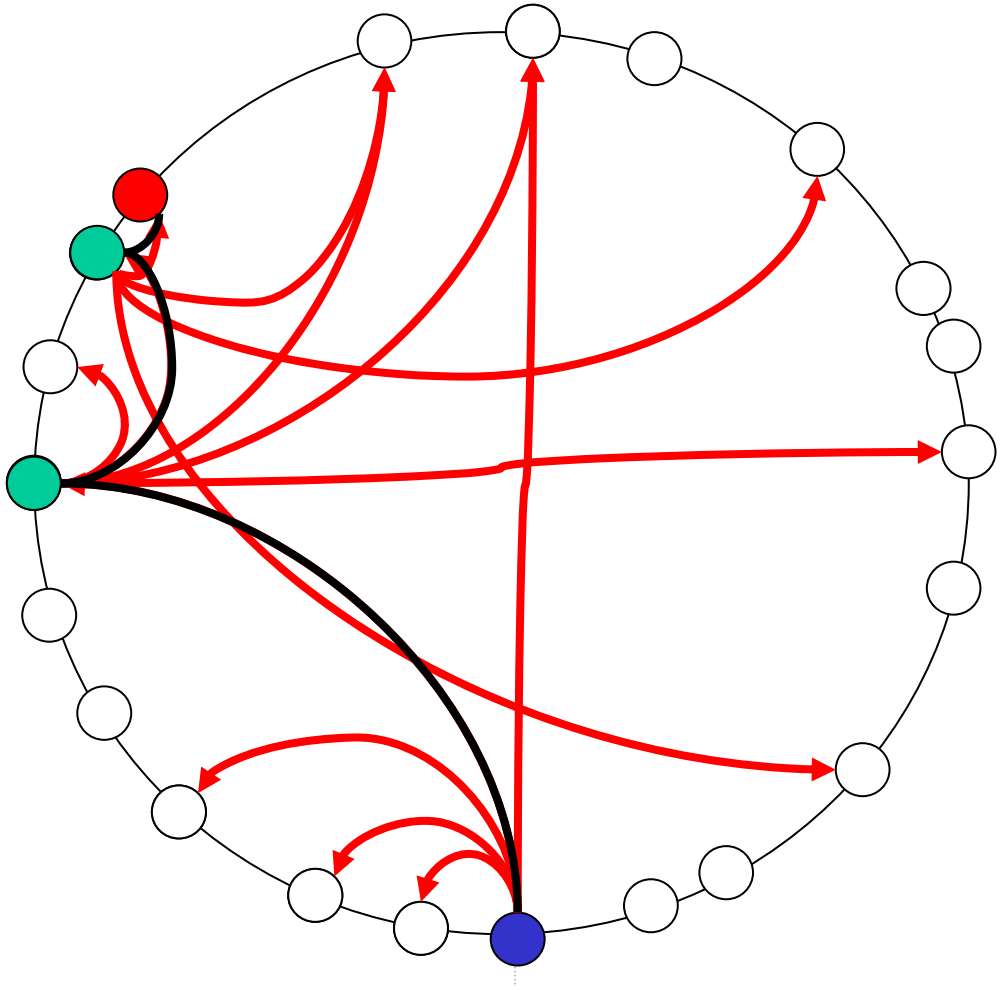




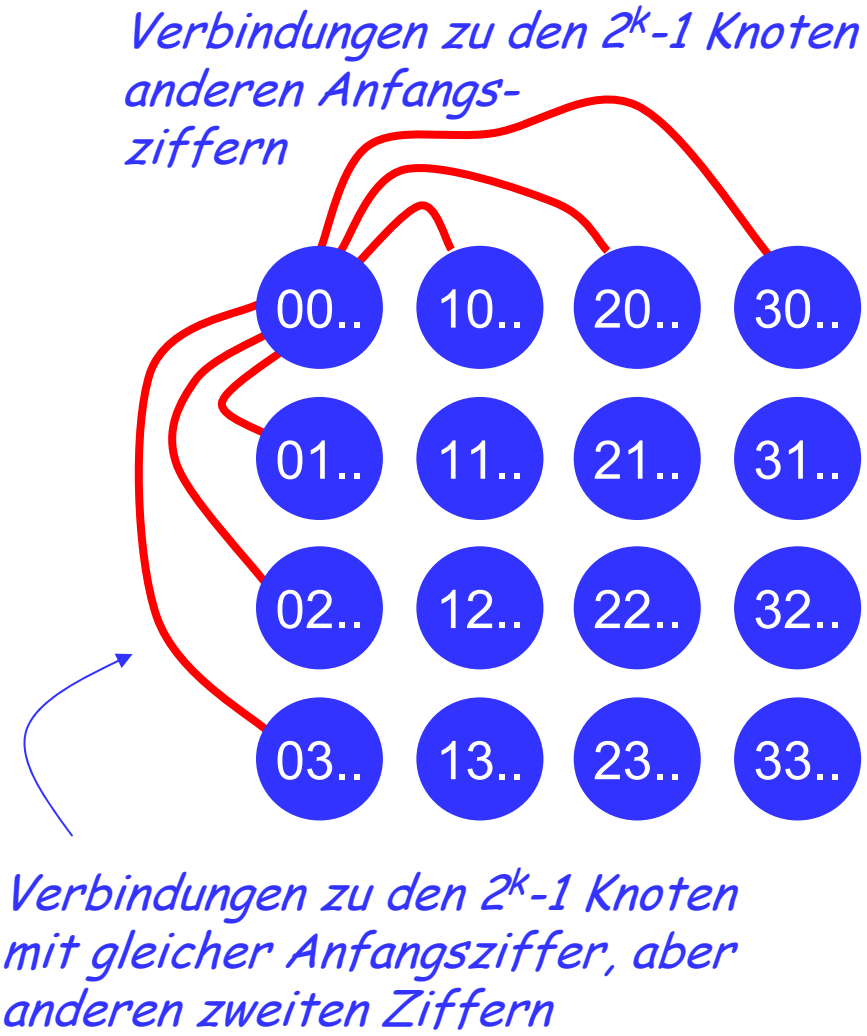
Content Addressable Network



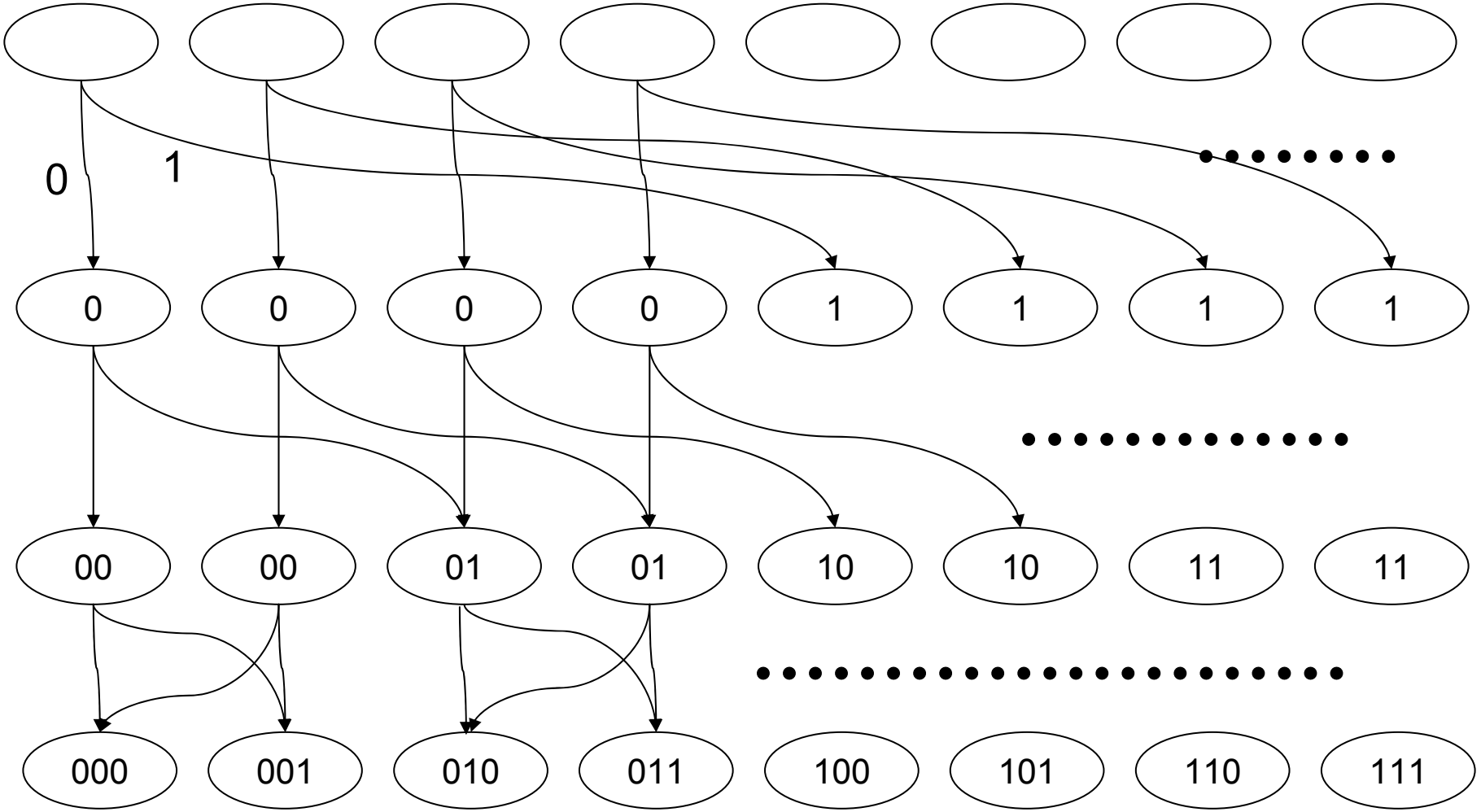
Chord



Pastry

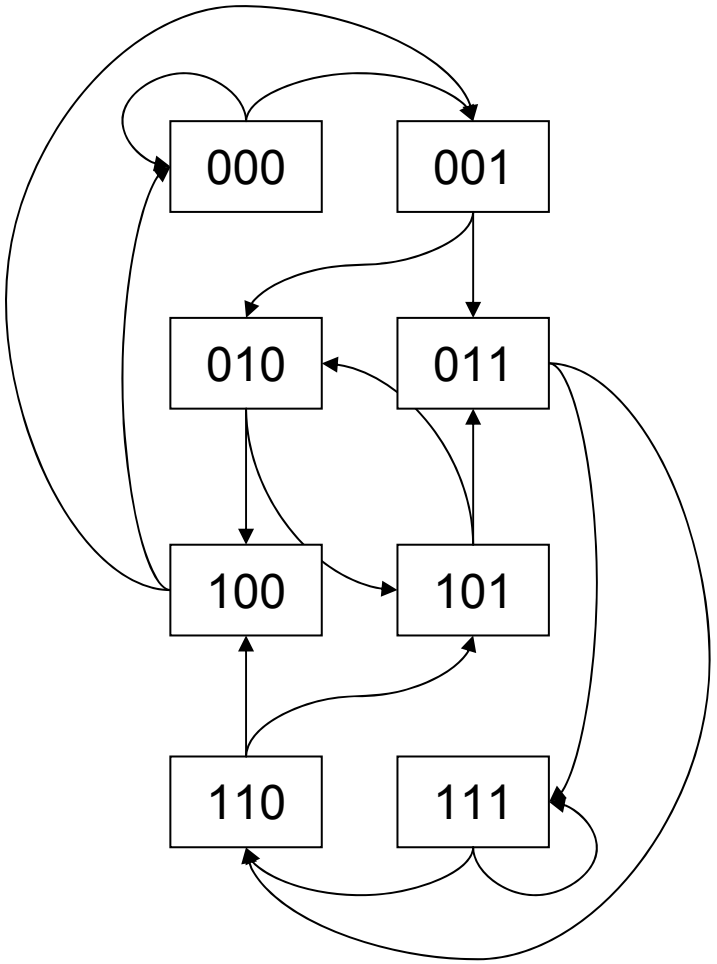

Verbindungen zu den 2^k-1 Knoten anderen Anfangs-ziffern
00..
10..
20..
30..
01..
11..
21..
31..
02..
12..
22..
32..
03..
13..
23..
33..
Verbindungen zu den 2^k-1 Knoten mit gleicher Anfangsziffer, aber anderen zweiten Ziffern

Butterfly Netze



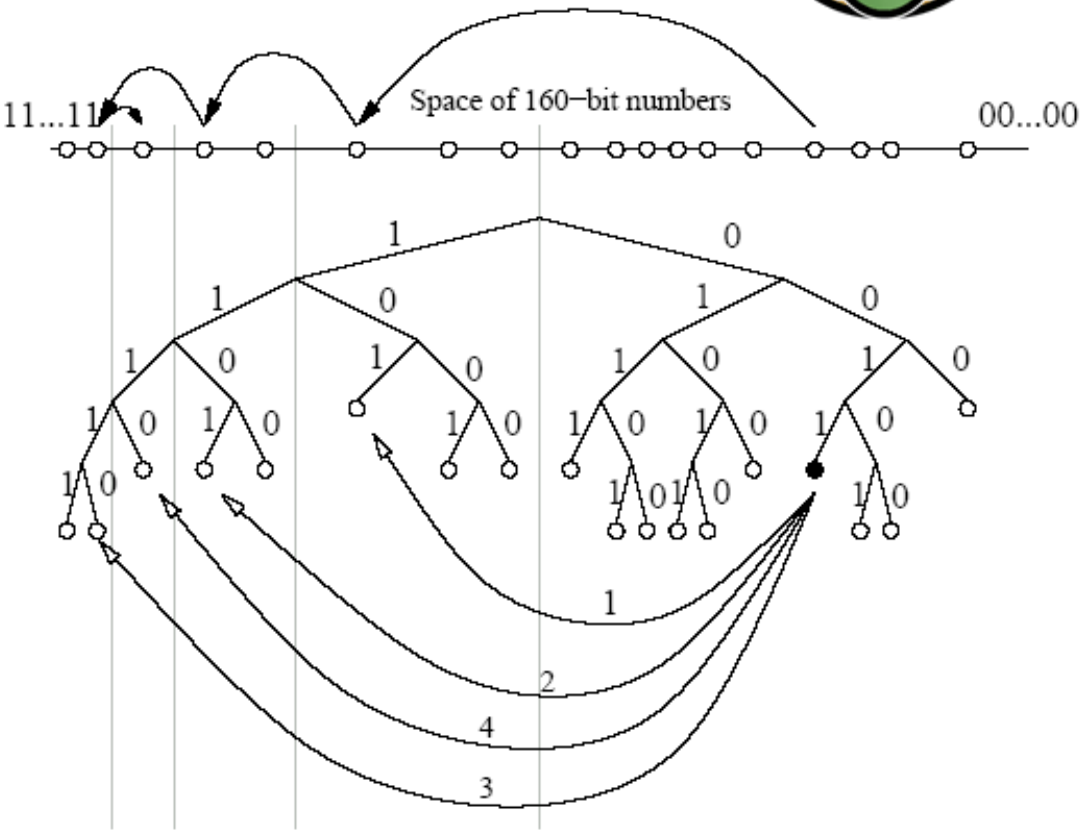
De Bruijn Netze

- Geringe praktische Bedeutung, da anfällig gegen Störungen
- Vorteil: O(1) Zustand, O(log N) Schritte
- Nachteil: Keine Freiheit bei der Topologiewahl
- Vorteil oder Nachteil: Pfade von verschiedenen Quellen konvergieren oft erst beim Ziel.



Kademlia

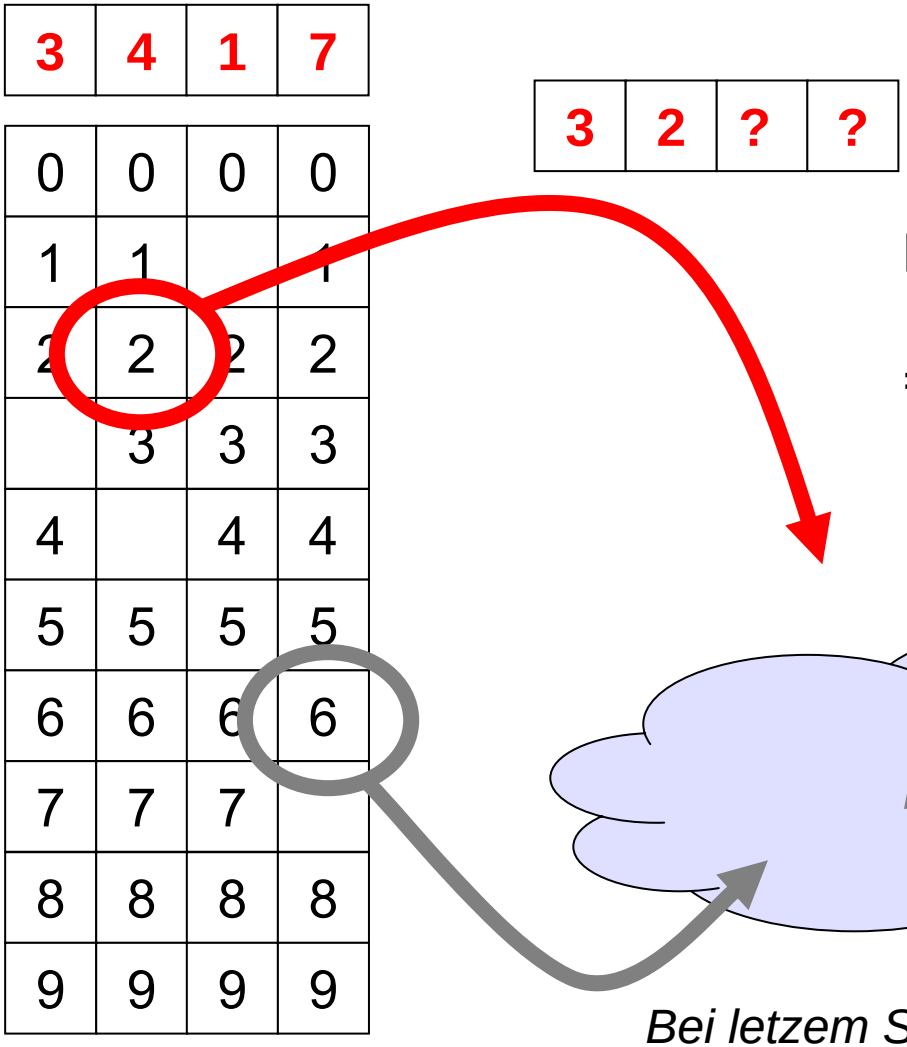
- Ein-dimensionaler Schlüsselraum
- Abstand über XOR-Metrik definiert, d.h. d(a,b) := a XOR b
- D.h. es werden z.B. die Bits schrittweise vom höchst- zum niedrig-wertigsten auf den Zielwert gebracht.
- Dazu gibt es pro Stelle eine Liste („Bucket“) mit bis zu k=20 Peers.
- Kademlia ist Grundlage z.B. für das eDonkey Filesharing System.



Nähengewahrheit (1)

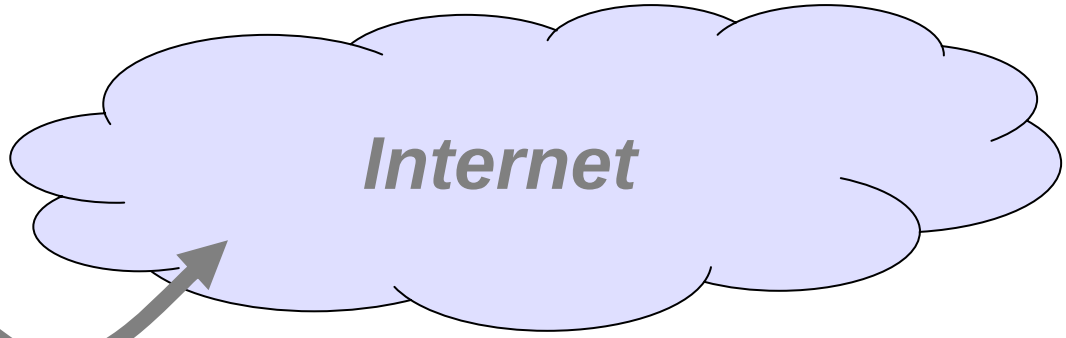
- Pastry erlaubt explizit die Freiheit an jeder Stelle einen beliebigen Peer auszuwählen, der dem grundsätzlichen Kriterium („Ziffer kann geändert werden“) genügt.
- Chord funktioniert auch wenn man nicht genau die Peers in exponentiell steigendem virtuellen Abstand wählt.
  - Extremfall: Jeder Peer kennt nur seinen virtuellen Nachfolger. Dann kann jedes Ziel in O(N) Schritten erreicht werden.
  - Zusätzliche Overlay-Verbindungen verringern die erforderliche Schrittzahl.
- Die Wahl physisch guter Peers erhöht die Leistungsfähigkeit des Systems:
  - Wahl guter Peers zum Aufbau von Overlay-Verbindungen (=Proximity Neighbor Selection)
  - Wahl guter Peers während des Routings (=Proximity Route Selection)

Nähengewahrheit (2) - PNS bei Pastry



Proximity Neighbor Selection

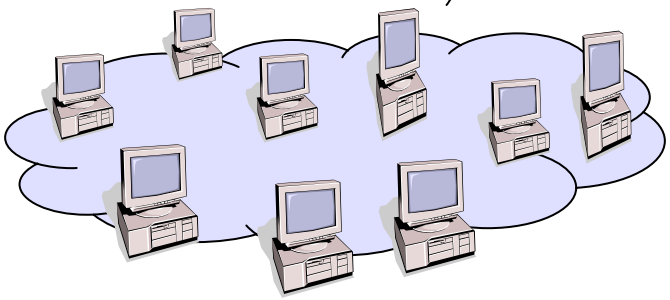
= Wähle einen beliebigen Peer der die entsprechende Stelle korrigieren kann!

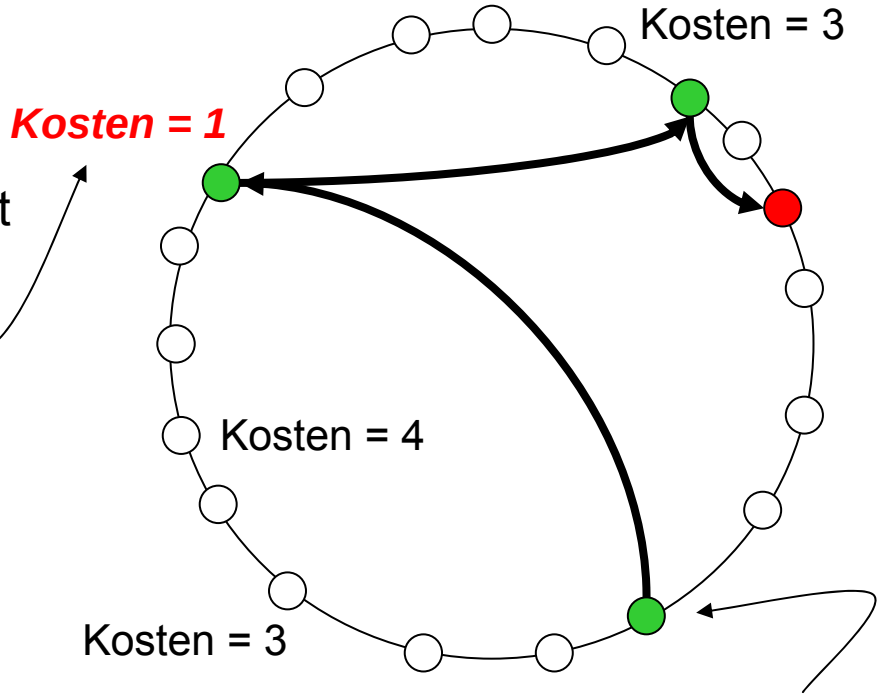


Bei letzem Schritt ist die Wahl jedoch festgelegt.

PNS und PRS bei Chord

PNS: Wähle aus allen Peers diejenigen aus, die geringe Kosten haben.
Jedoch: Jeder exponentiell größer werdende Ringabschnitt muss mit ausgewählten Peers besetzt sein.

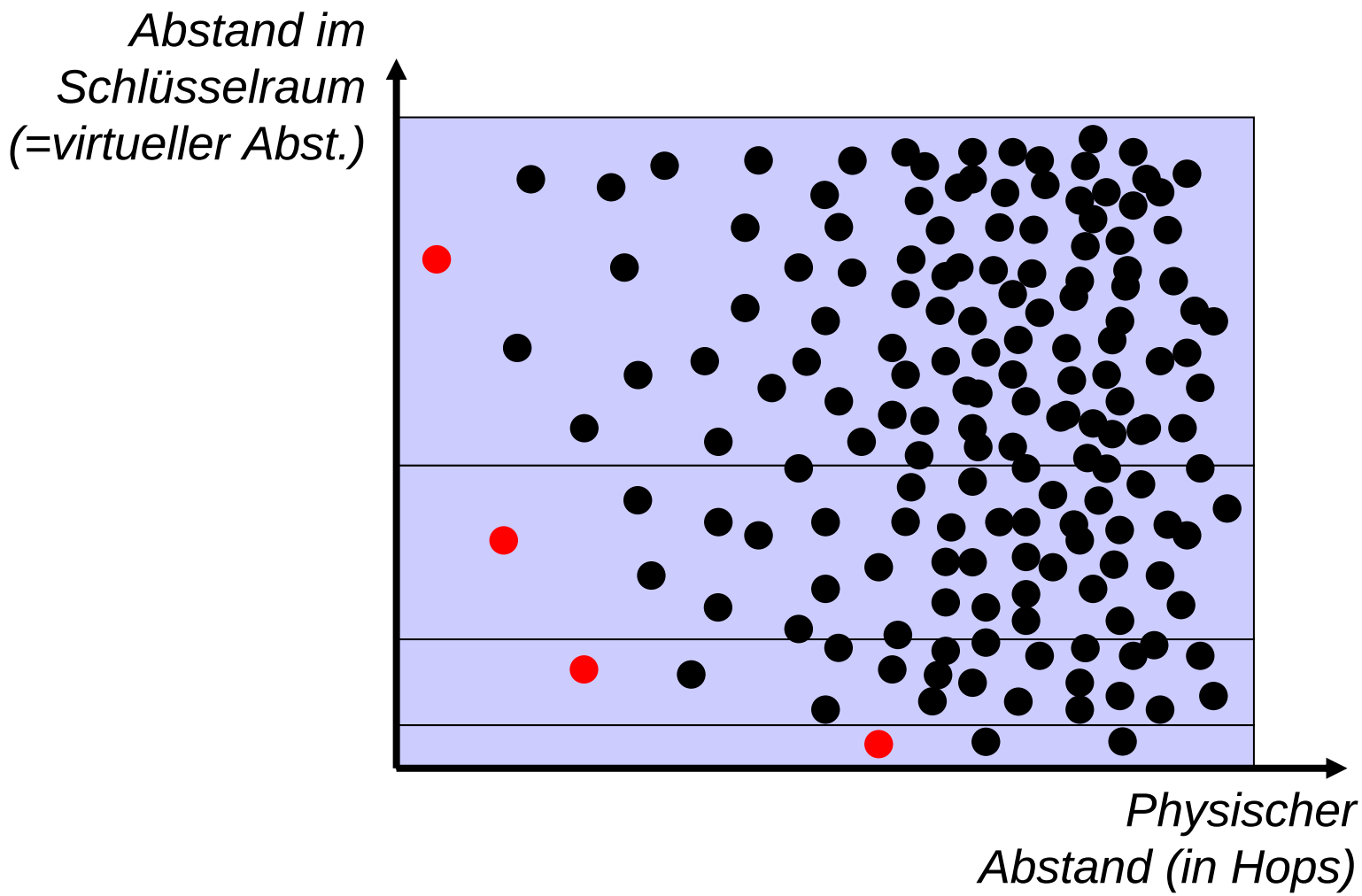




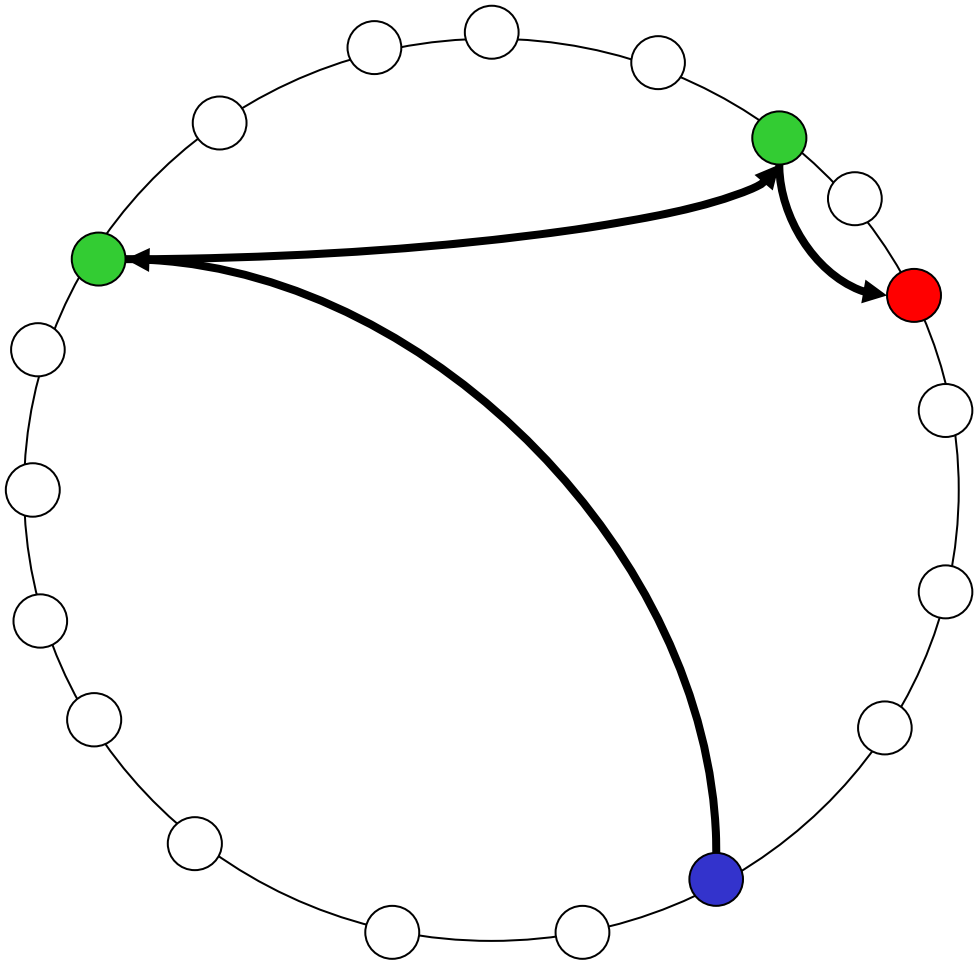
PRS: Wähle beim Routen im die Peers, die näher am Ziel sind und möglichst geringe Kosten haben.

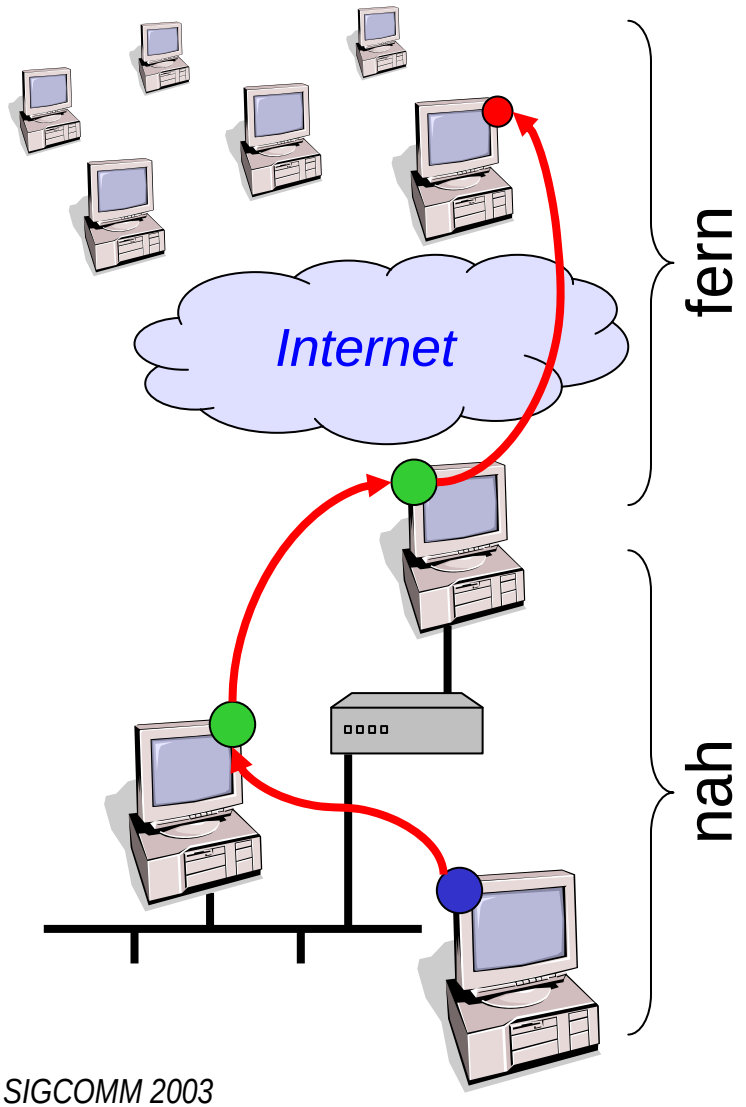
PRS ist in so fern einfacher, als dass man ja bereits alle möglichen Peers unter den man wählt in der lokalen Routing-Tabelle hat.

Virtuelle versus physische Distanz



Nähengewahrheit bei Chord

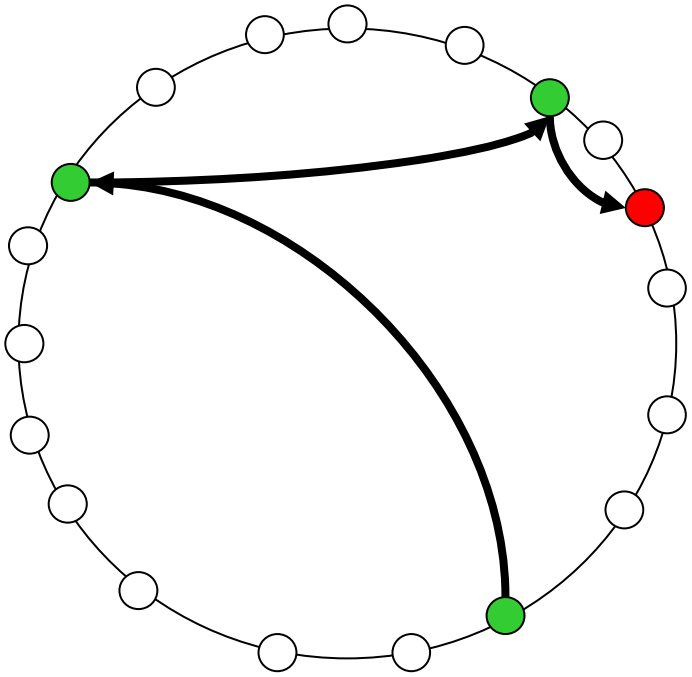




Gummadi, et al. The Impact of DHT Routing Geometry on Resilience and Proximity, SIGCOMM 2003

Chord – Zusammenfassung

- Jeder Peer wählt sich eine zufällige Position im Schlüsselraum (=virtuelle Adresse).
- Jeder Peer unterhält O(log N) Verbindungen (=„Finger“) ...
  - ... und zwar zu Peers in exponentiell wachsendem Abstand im Schlüsselraum.
  - Dabei werden physisch nahe Peers bevorzugt (→Proximity Neighbor Selection, PNS)
- Eine Anfrage wird in Ringrichtung weitergereicht bis der Ziel-Peer erreicht ist …
  - ... und zwar über die vorhandenen Finger.
  - Dabei werden physisch nahe Peers bevorzugt (→Proximity Route Selection, PRS)



Rules for Filling the Finger Table

Peer-to-Peer zur Inhaltssuche

- Strukturierte P2P Systeme benötigen eindeutigen Schlüssel zur Suche
  - Beispiel: Dateiname vollständig angegeben sein.
- Unstrukturiertes P2P kann beliebige Suchalgorithmen umsetzen
  - Beispiel: Dateinamen mit Wildcards

- Häufig will man auch nach Texten suchen, die bestimmte Schlüsselworte enthalten
  - Beispiel: Web-Recherche mit Google
- Oder man benötigt eine unscharfe Suche nach inhaltlichen Zusammenhängen

- Lösungsansatz bei strukturiertem P2P mittels direktem Ausnutzen der inhaltlichen Beziehung beim Speichern der Objekte:
  - Im Overlay benachbarte Peers speichern auch inhaltlich benachbarte Objekte.

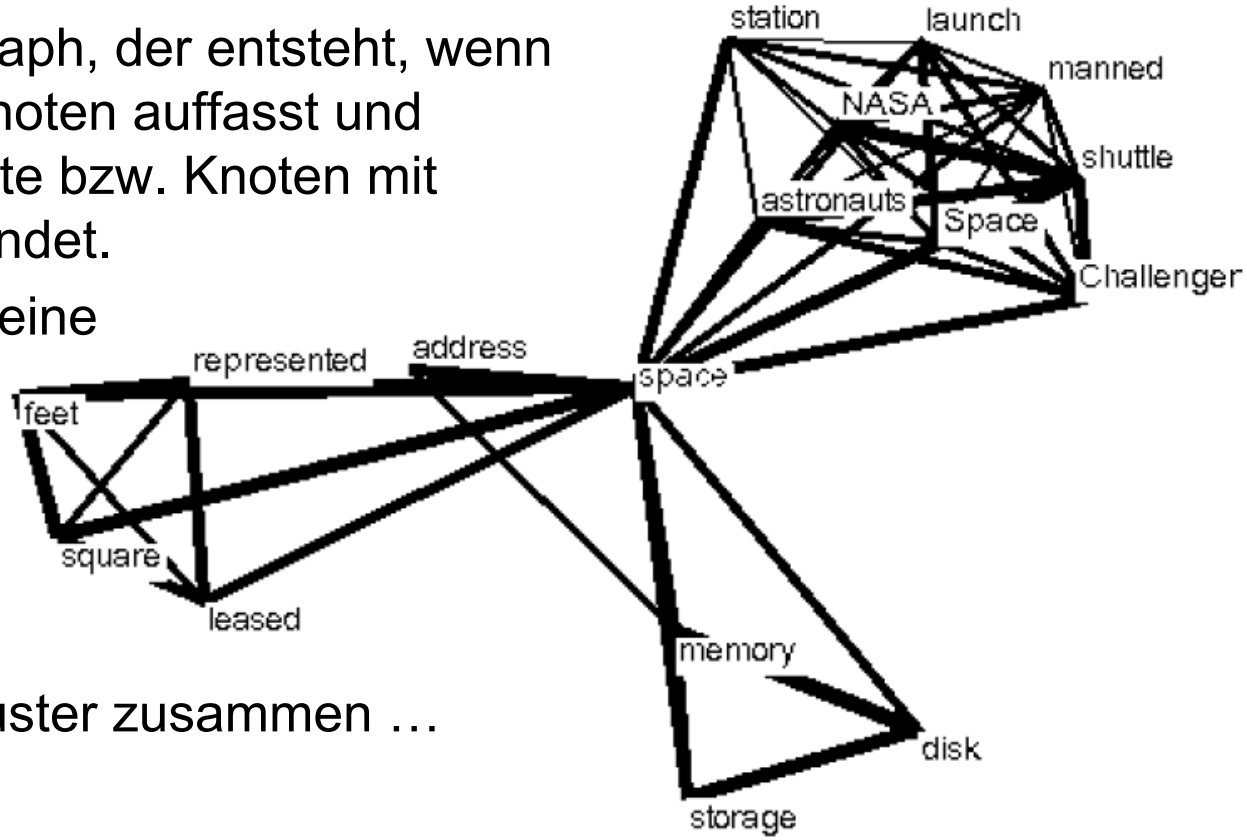
Semantische Overlays mit CAN

- Grundidee:
  - Verorte Texte nach ihren Inhalten in einem hochdimensionalen Indexraum.
  - Jeder Text ist ein Vektor. Dimensionen des Vektorraums sind die verschiedenen Worte. Koordinate ist die Häufigkeit des jeweiligen Wortes im Text. (=Vector Space Model)
  - Da einige Worte meist gemeinsam auftreten, genügt es, Unterräume des Vektorraums zu betrachten. (=Latent Semantic Indexing)
  - Die so erhaltenen Unterräume projiziert man auf ein CAN.
- Dieses Verfahren bildet ähnliche Texte (gemessen am gemeinsamen Vokabular) auf den gleichen oder benachbare Peers ab.

Chunqiang Tang, at al. „Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks“, SIGCOMM 2003

Abbildung der Inhaltsstruktur

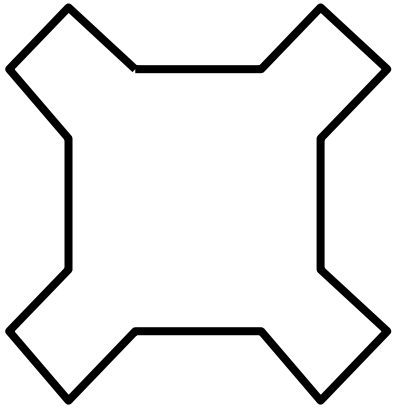
- Grundidee:
  - Betrachte den Graph, der entsteht, wenn man Worte als Knoten auffasst und benachbarte Worte bzw. Knoten mit einer Kante verbindet.
  - Synonyme sind keine Nachbarn, haben aber viele gemeinsame Nachbarn.
  - Fasse die so entstehenden Cluster zusammen …



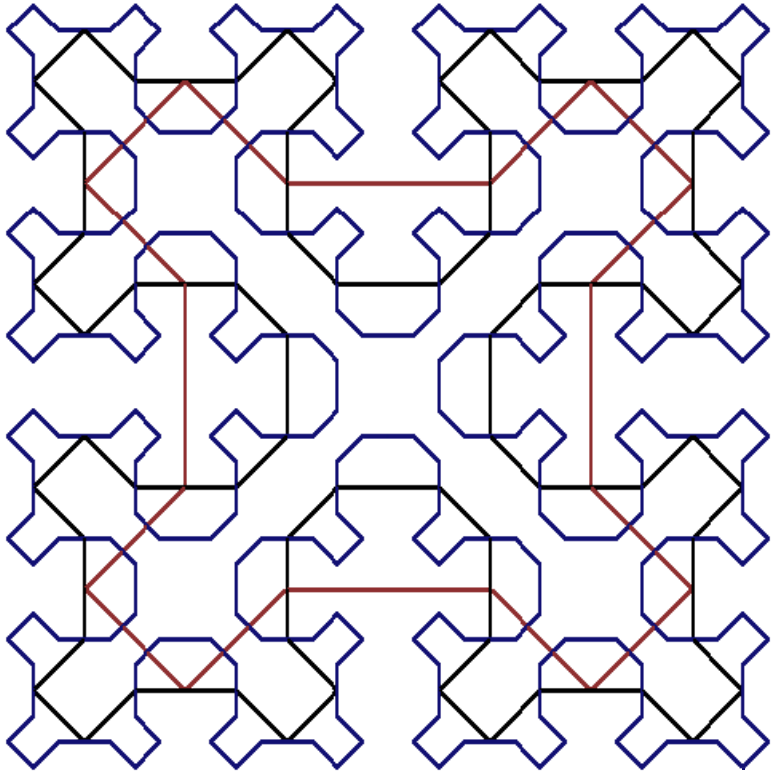
- Offene Fragen:
  - Wie lässt sich diese Idee skalierbar und effizient auf Overlay-Netze übertragen?

Quelle: Stefan Bordag, Gerhard Heyer and Uwe Quasthofff, *Small Worlds of Concepts and Other Principles of Semantic Search*, IICS 2003

Space-Filling Curves and P2P Systems



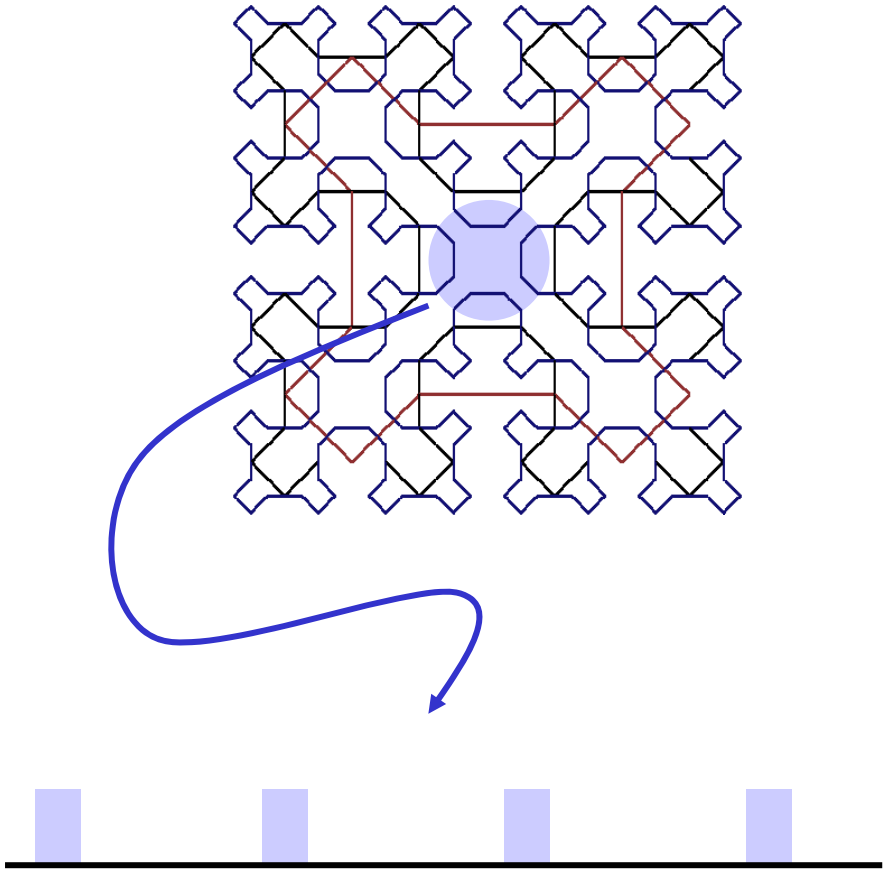
Sierpiński curves are a recursively defined sequence of continuous closed plane fractal curves discovered by Wacław Sierpiński, which in the limit n $\rightarrow\infty$ completely fill the unit square: thus their limit curve, also called the Sierpiński curve, is an example of a space-filling curve.



Source: Wikipedia

Space-Filling Curves and P2P Systems

- Space-filling curves can map higher-dimensional structures to one-dimensional DHTs
  - Geographic information such as points of interest on a map
  - Astronomical catalogues of stars
- Depending on the type of curve a source region is more or less scattered across the key space.
- Input data that is widely scattered requires more search effort.

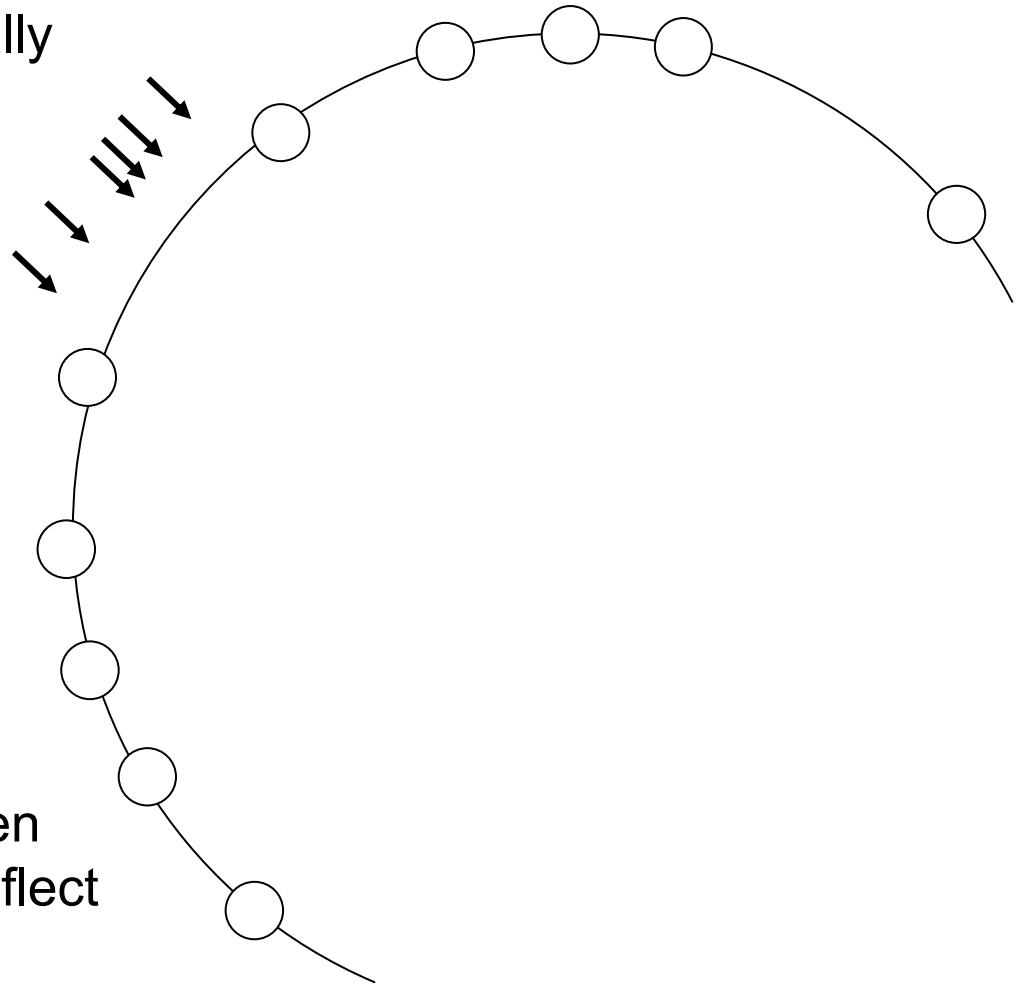


Hash Key Distribution

Hash keys are not equally spaced, but follow a random distribution.

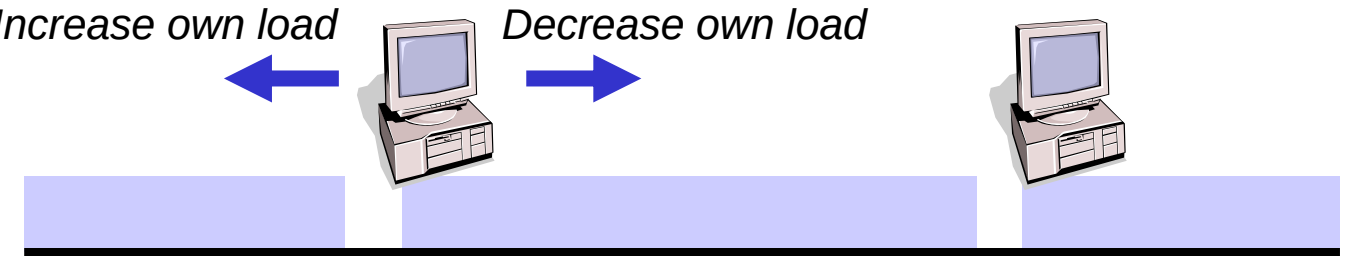
Nodes are not equally spaced either.

Thus, the load is not fairly split up between the peers. It does not reflect the peers‘ capabilities.



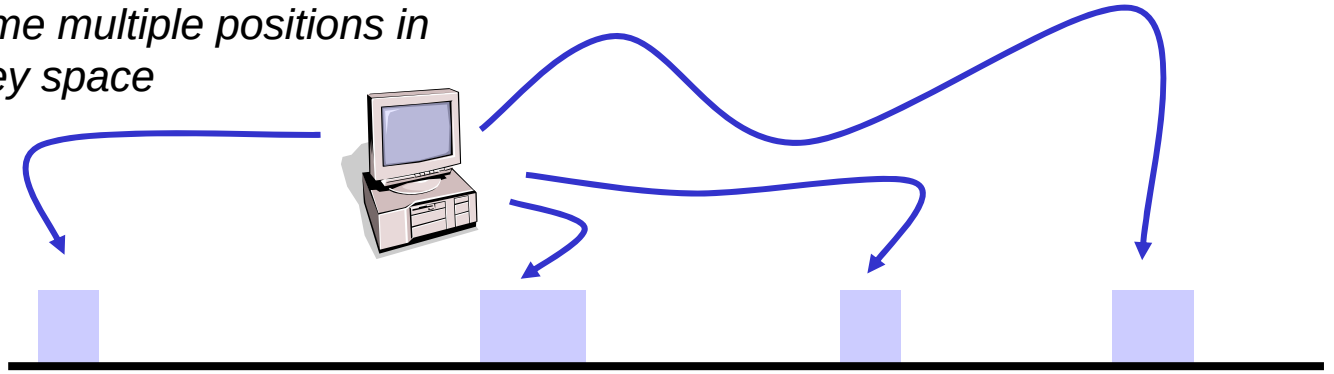
Load Balancing

1.



2.

Assume multiple positions in the key space



Range Queries

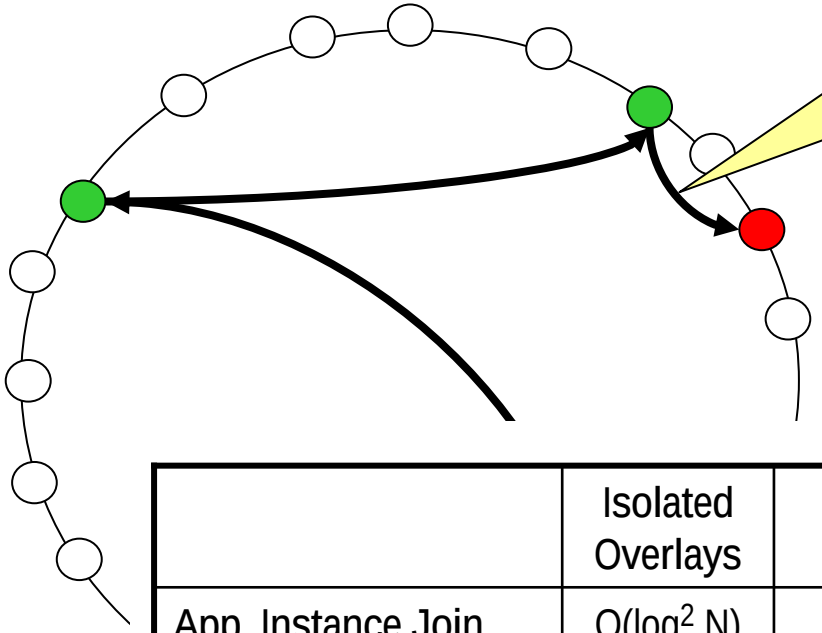
- Load Balancing is extremely important for range queries
- E.g., Content that is mapped with space-filling curves is not hashed.
- Thus hot spots in the input data are directly visible in the load distribution.
- Proposed solution (besides normal load balancing):
  Reflect key density in peer density.

Motivation for Multiple Application KBR

Dim Chord

Open DHT

Probleme bei Strukturierten Overlays



Overlay-Kanten müssten regelmäßig auf Verfügbarkeit geprüft werden!

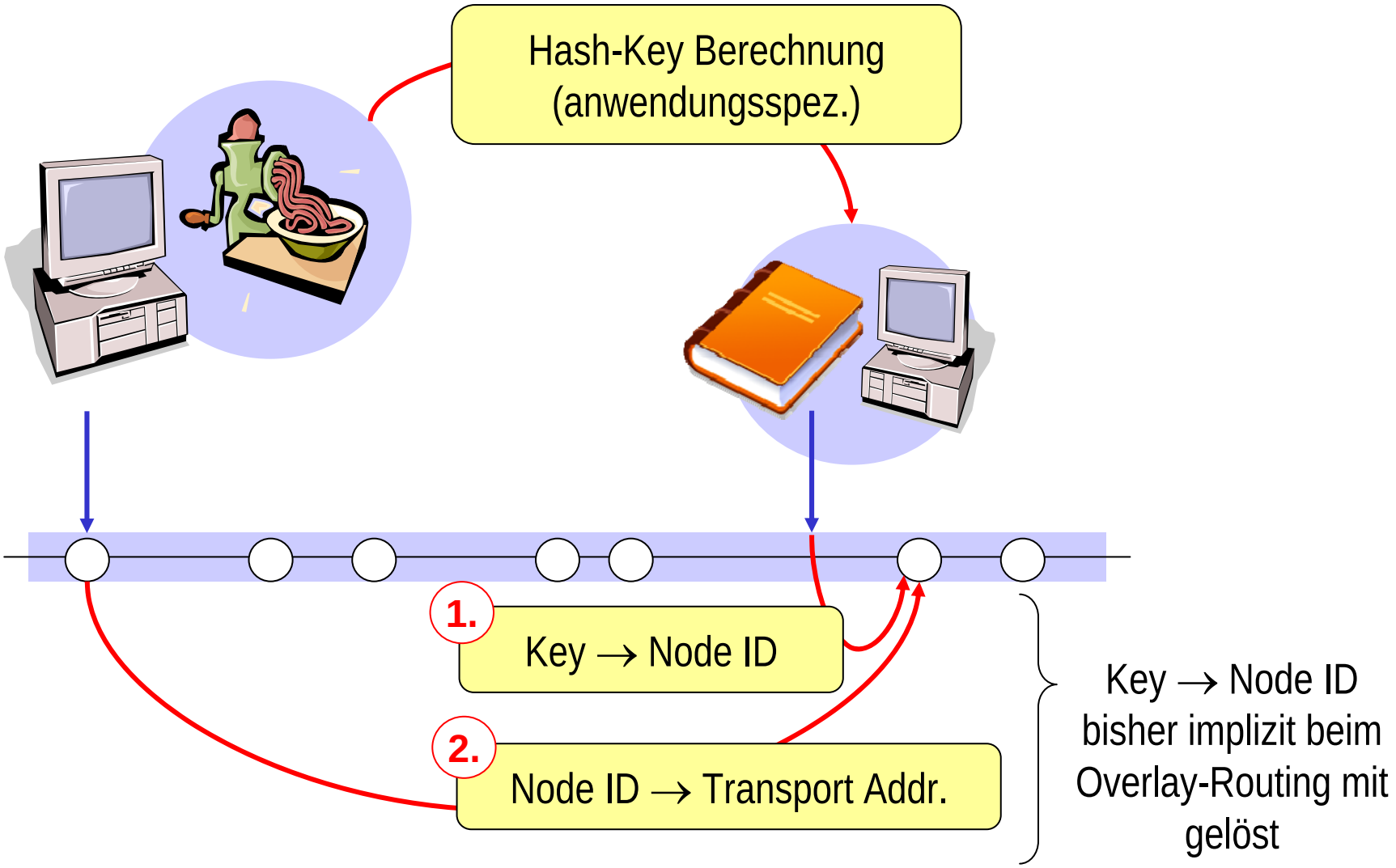
Prüfung bei geringer Anwendungslast oder schwachen Peers unnötig aufwändig.

	Isolated Overlays	DimChord	OpenDHT w/ ReDiR	Our Proposal
App. Instance Join	$O(\log^2 N)$	$O(\log M)$	$O(\log N) \bullet O(\log M^*)$	$O(\log M^*)$
State per App. Inst.	$O(\log N)$	$O(\log M)$	$O(1)$	$O(1)$
Load Ratio	$O(1)$	$O(\log M)$	$O(1)$	$O(1)$
Routing (worst case)	$O(\log N)$	$O(\log M) + O(\log M^*)$	$O(\log N) \bullet O(\log M^*)$	$O(\log M^*)$

N nodes per application (avg.), M nodes overall, M nodes in DHT*

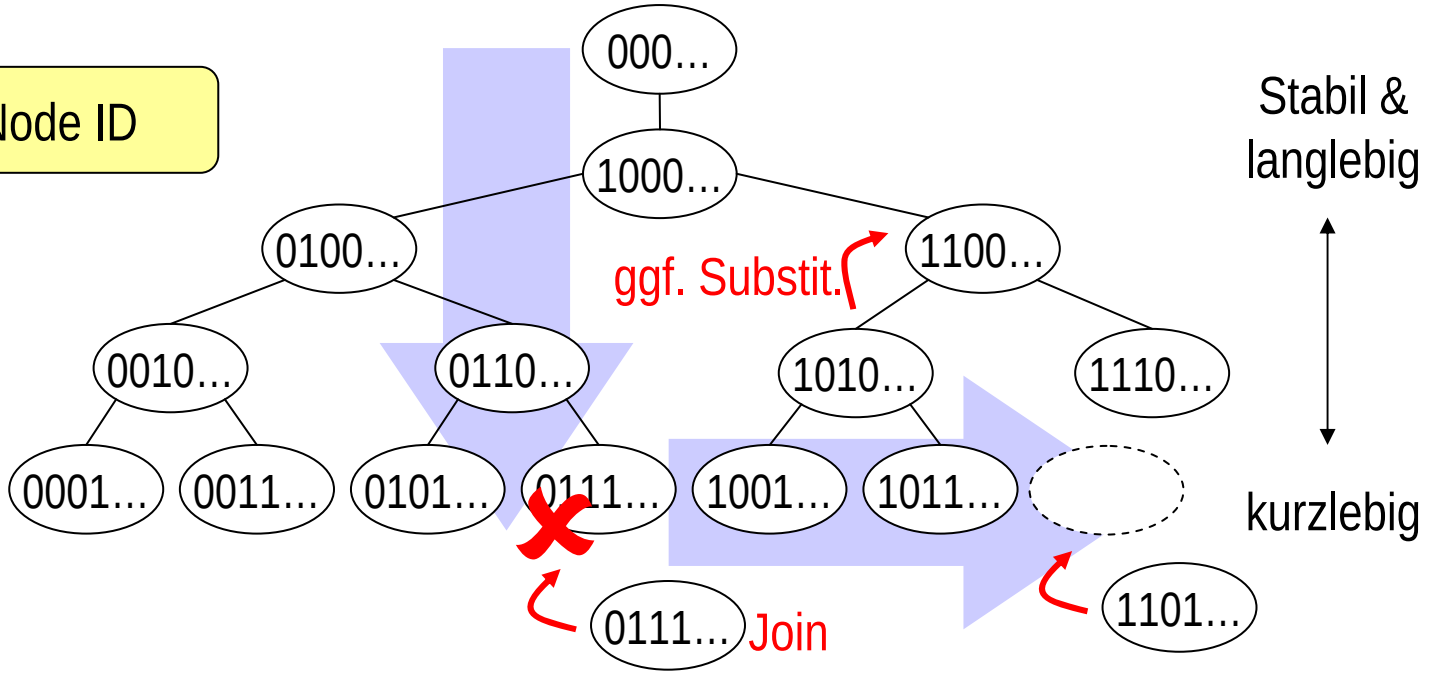
Di et al, Providing KBR Services for Multiple Applications, IPTPS 2008

KBR Dienst für mehrere Anwendungen (1)



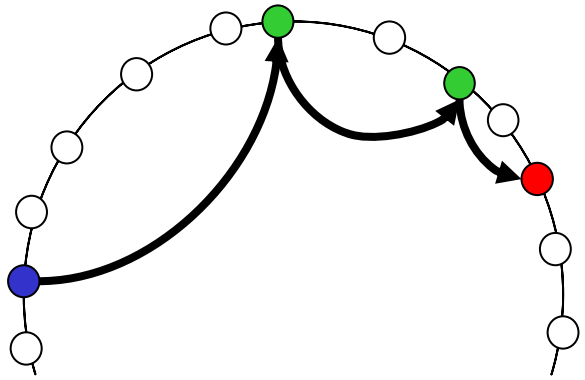
KBR Dienst für mehrere Anwendungen (2)

1. Key → Node ID



2. Node ID → Transport Addr.

Verteilte Hash-Tabelle zum Nachschlagen (nur stabile Peers machen mit)



Zusammenfassung

- Strukturierte P2P Systeme geben eine Overlay-Topologie vor.
  - PNS nutzt die Freiheit bei der Wahl der Overlay-Verbindungen zwischen verschiedenen Peers wählen zu können.
  - PRS nutzt die Freiheit während der Weiterleitung von Nachrichten zwischen verschiedenen Overlay-Verbindungen wählen zu können.
- Anwendungsbeispiele meist Filesharing
  - Gnutella & Co. basieren auf unstrukturiertem P2P mit Super Peers
  - FreeNet & Co. bereiten den Weg für strukturierte P2P Netze
  - eDonkey & Co. basieren auf Kademlia
  - BitTorrent (vgl. nächste Vorl.) mischen unstrukturierte mit strukturierten Elementen
- Vereinzelte Konzepte Overlay-Toplogie an Inhaltsstruktur anzupassen

Bootstrapping P2P

Questions?

Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de