CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Peer-to-Peer Networks

03: DHT

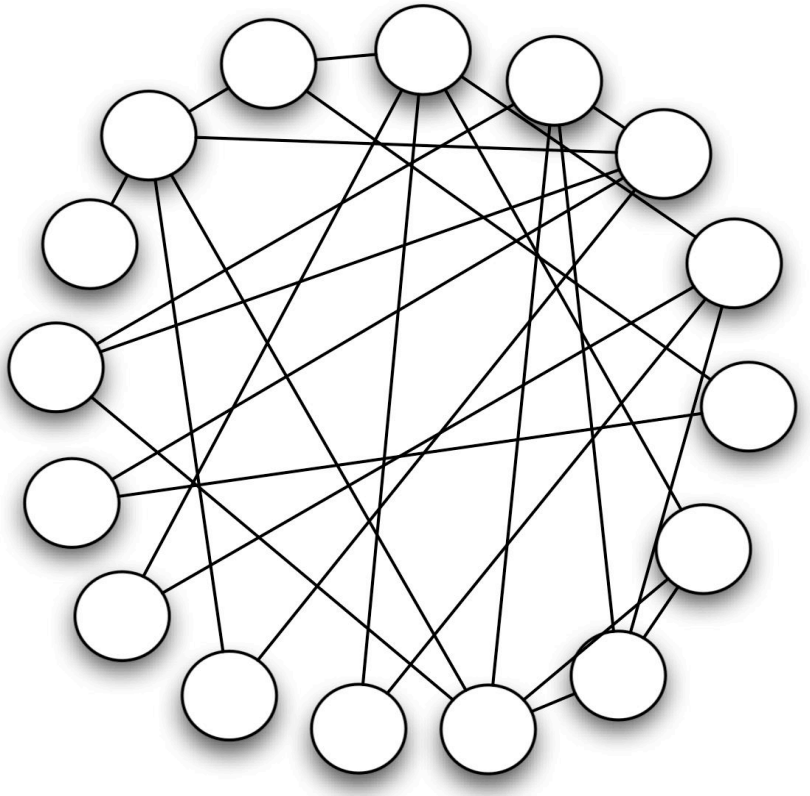
Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

UNI FREIBURG

CoNe Freiburg

Why Gnutella Does Not Really Scale

- Gnutella
  - graph structure is random
  - degree of nodes is small
  - small diameter
  - strong connectivity
- Lookup is expensive
  - for finding an item the whole network must be searched
- Gnutella‘s lookup does not scale
  - reason: no structure within the index storage



UNI FREIBURG

CoNe Freiburg

Two Key Issues for Lookup

- Where is it?
- How to get there?
- Napster:
  - Where? on the server
  - How to get there? directly
- Gnutella
  - Where? don‘t know
  - How to get there? don‘t know
- Better:
- Where is x?
  - at f(x)
- How to get there?
  - all peers know the route

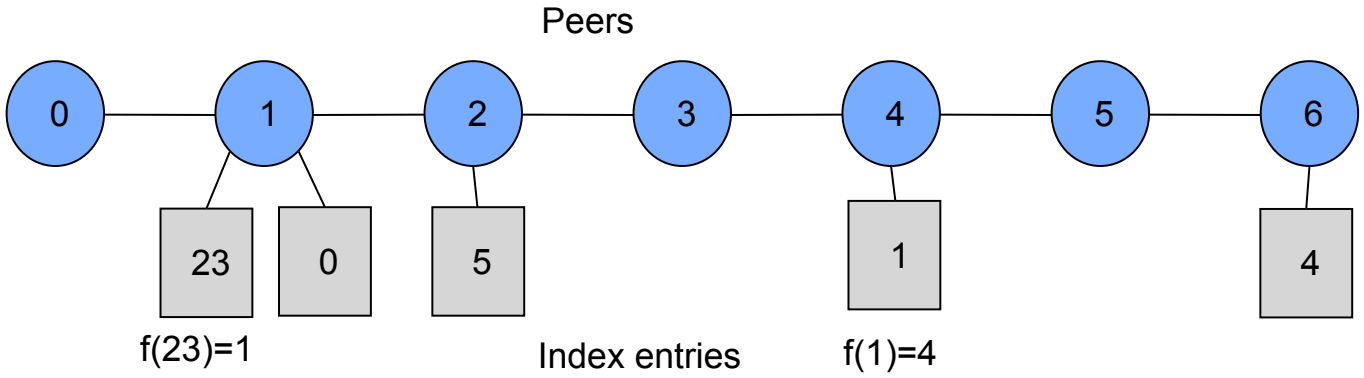
UNI FREIBURG

CoNe Freiburg

(Bad) Idea: Use Hashing

- Give each of n peers a number 0,1,..,n-1
  - use hash function
    - e.g. f(x) = (3x+1 mod 23) mod 7
  - peers are connected on a chain

- Lookup
  - compute f(x)
  - forward message to f(x) along the chain

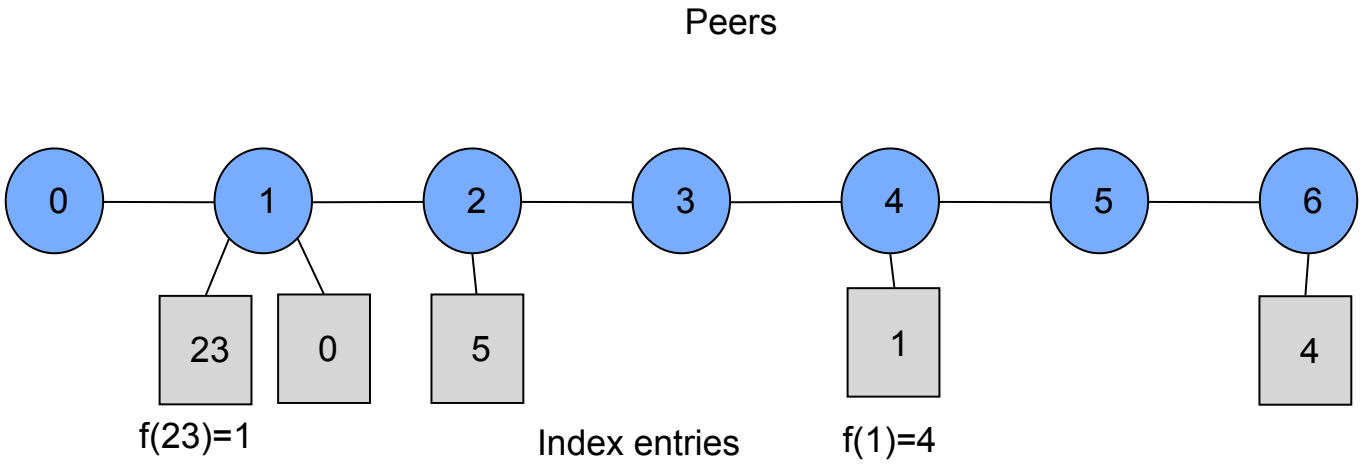


UNI FREIBURG

CoNe Freiburg

Problems with Pure Hashing

- Insert and deletion of peers critical
  - if a peer leaves without warning then network breaks up
  - inserting a peer implies readjusting the whole entries
    - hash function must be changed to new version
  - how to assign the numbers to peers?
- Lookup is not efficient
  - takes linear time on the average
  - the peers in the middle see 50% of all lookups

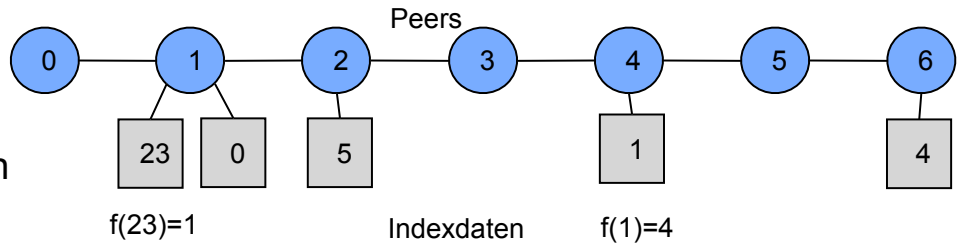


UNI FREIBURG

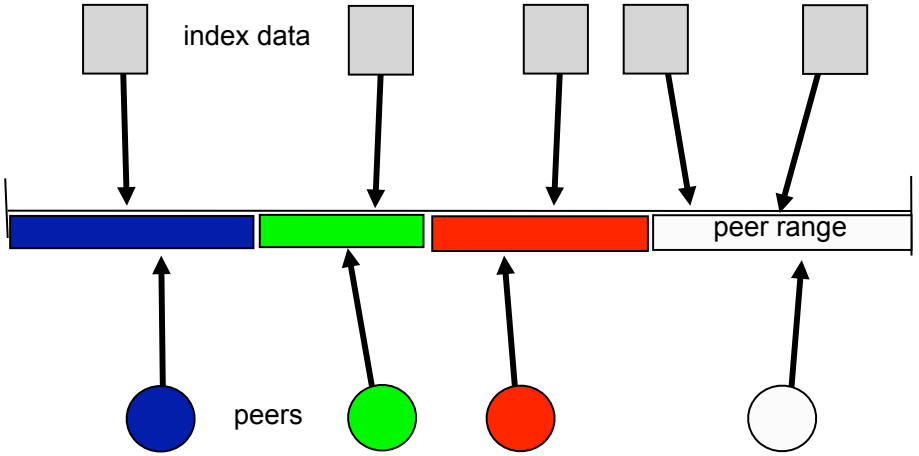
Distributed Hash-Table (DHT)

- Hash table
  - does not work efficiently for inserting and deleting
- Distributed Hash-Table
  - peers are „hashed“ to a position in an continuos set (e.g. line)
  - index data is also „hashed“ to this set
- Mapping of index data to peers
  - peers are given their own areas depending on the position of the direct neighbors
  - all index data in this area is mapped to the corresponding peer
- Literature
  - “Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web”, David Karger, Eric Lehman, Tom Leighton, Mathhew Levine, Daniel Lewin, Rina Panigrahy, STOC 1997

Pure (Poor) Hashing



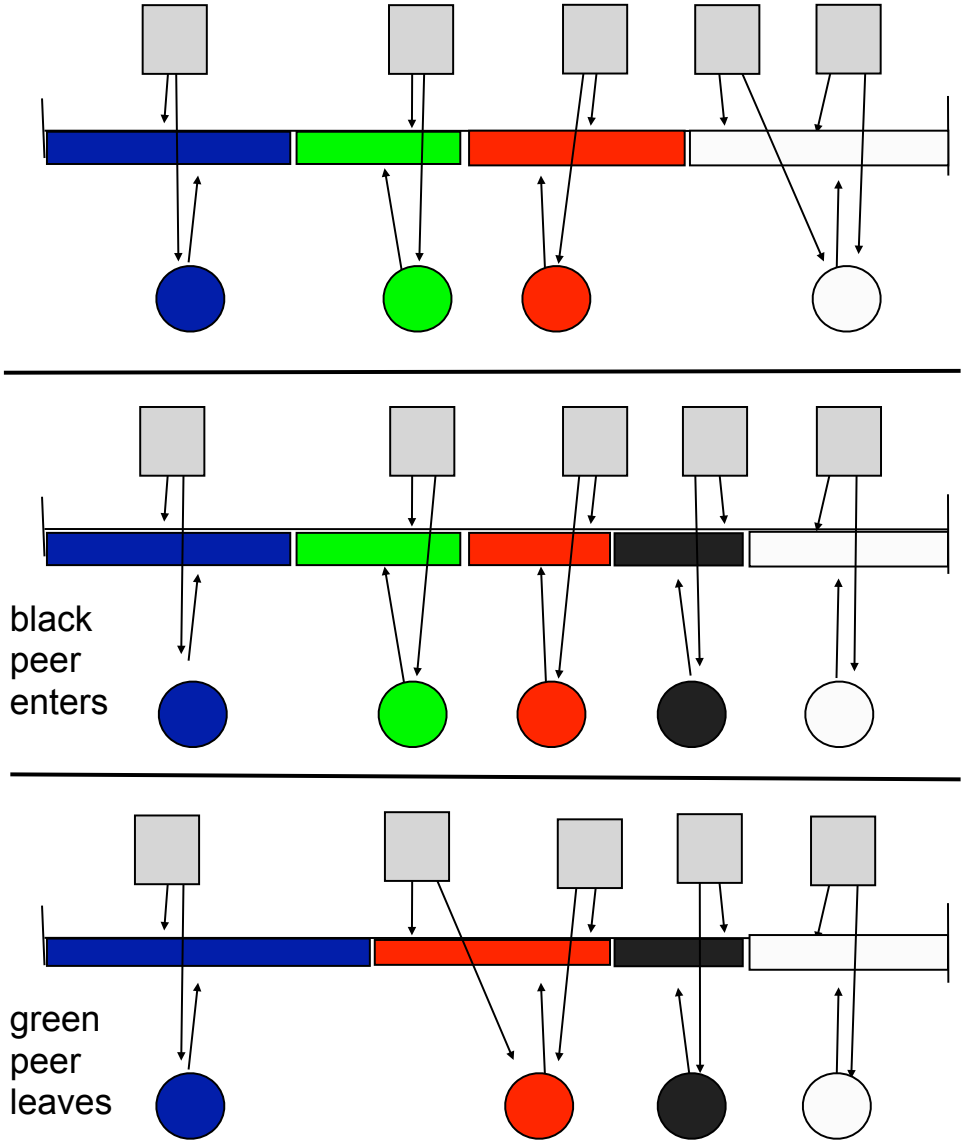
DHT



CoNe Freiburg

Entering and Leaving a DHT

- Distributed Hash Table
  - peers are hashed to to position
  - index files are hashed according to the search key
  - peers store index data in their areas
- When a peer enters
  - neighbored peers share their areas with the new peer
- When a peer leaves
  - the neighbors inherit the responsibilities for the index data

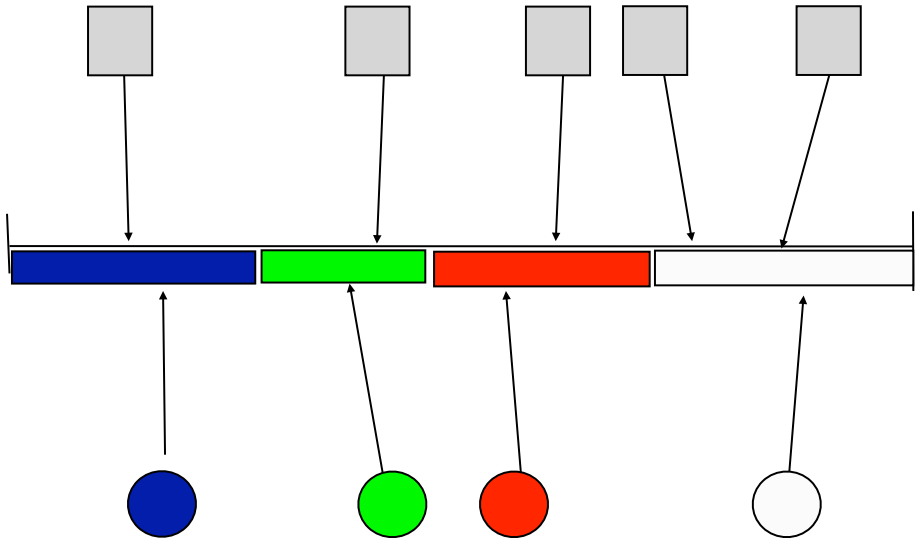


UNI FREIBURG

CoNe Freiburg

Features of DHT

- Advantages
  - Each index entries is assigned to a specific peer
  - Entering and leaving peers cause only local changes
- DHT is the dominant data struction in efficient P2P networks
- To do:
  - network structure



UNI FREIBURG

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Peer-to-Peer Networks

03: DHT

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

UNI FREIBURG