Peer-to-Peer Networks

07 Degree Optimal Networks

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

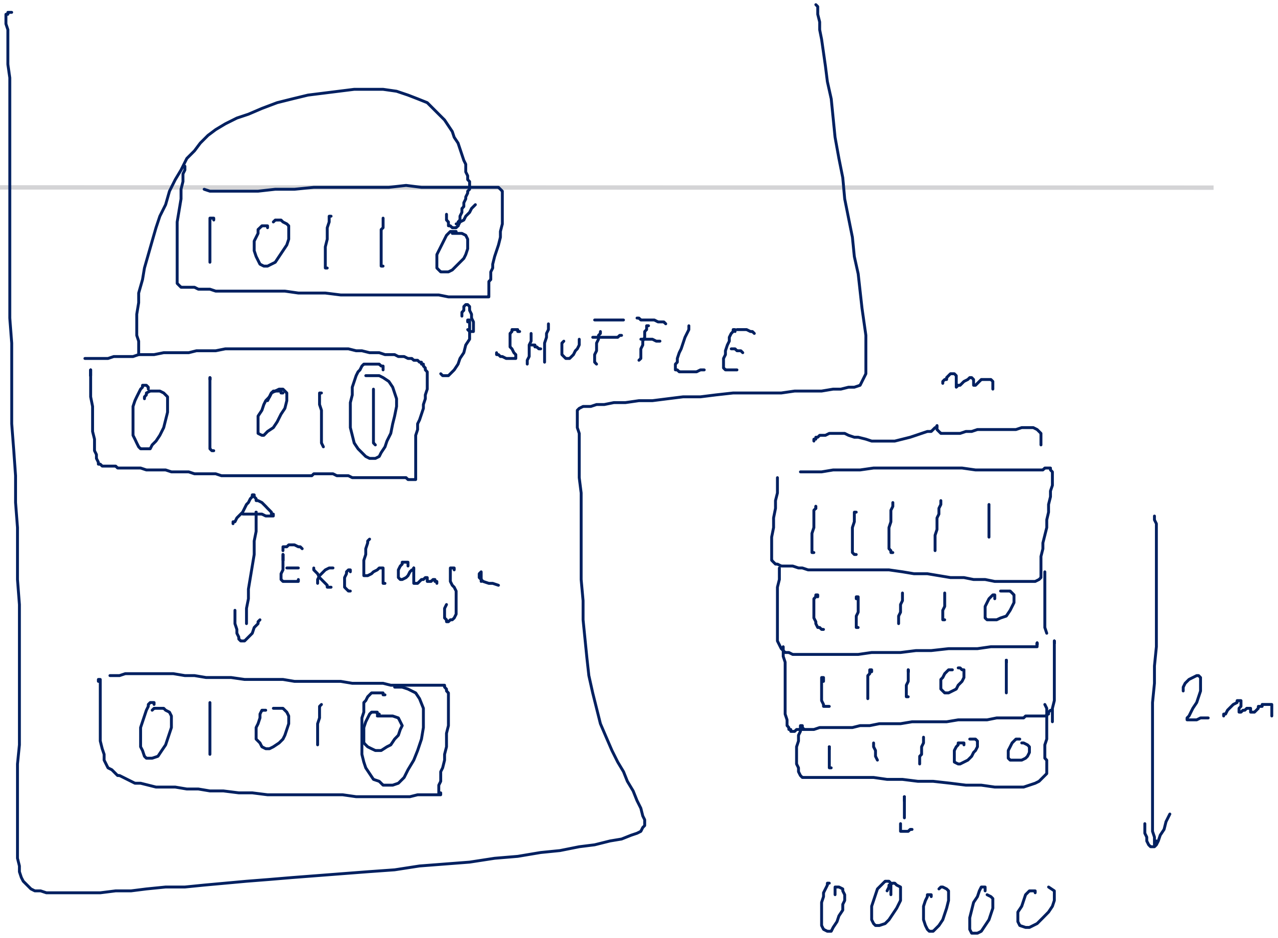
Degree Optimal Networks

Koorde

M. Frans Kaashoek and David R. Karger 2003

Dutch

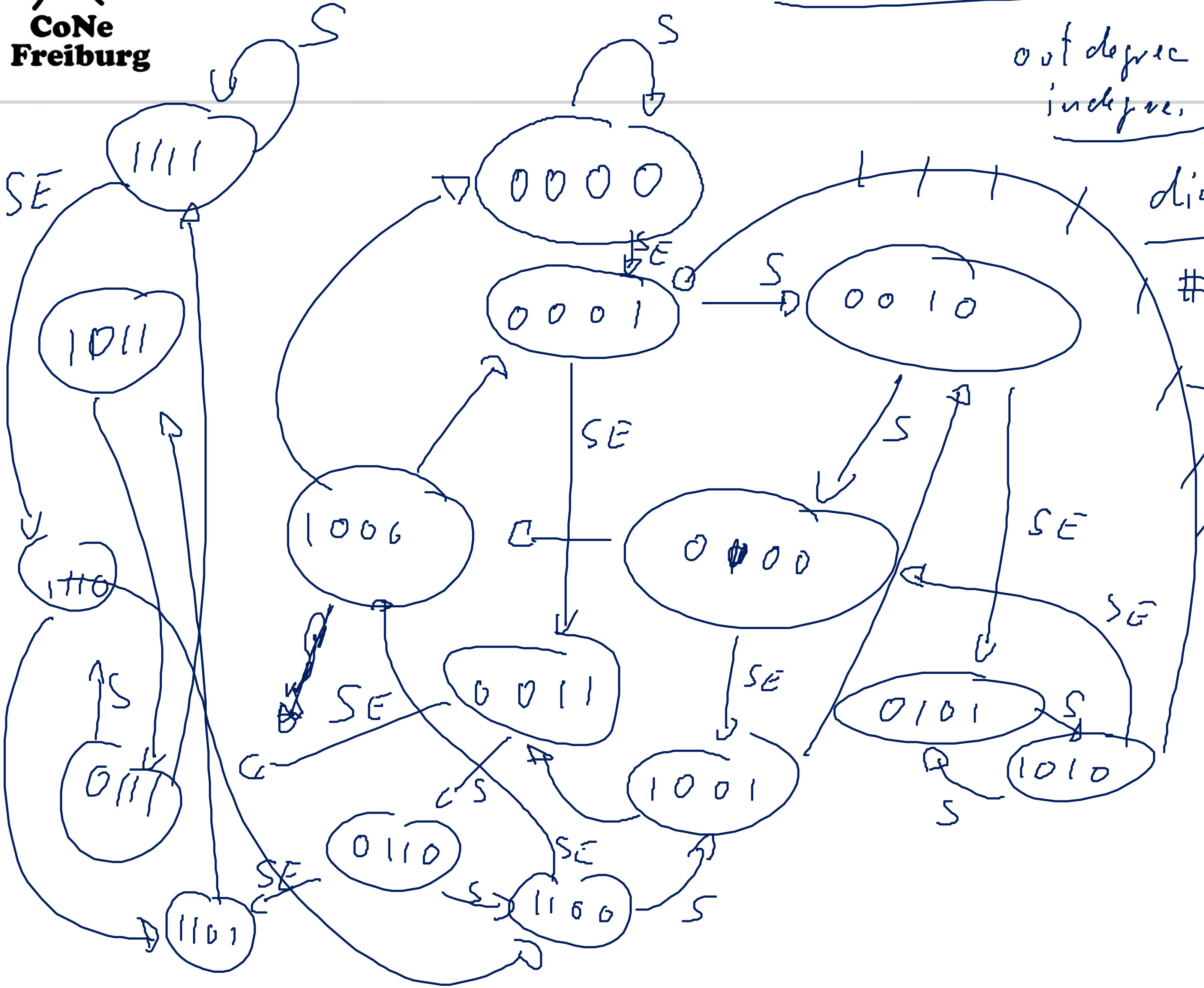
Chord


1 0 1 1 0
SHUFFLE
0 1 0 1 0
Exchange
0 1 0 1 0
m
1 1 1 1 1
1 1 1 1 0
1 1 1 0 1
1 1 1 0 0
0 0 0 0 0
2m

De Brujn Graph

outdegree : 2

indegree : 2



diameter?

#nodes

$n = 2^m$

$\to m$

$= \log_2 n$

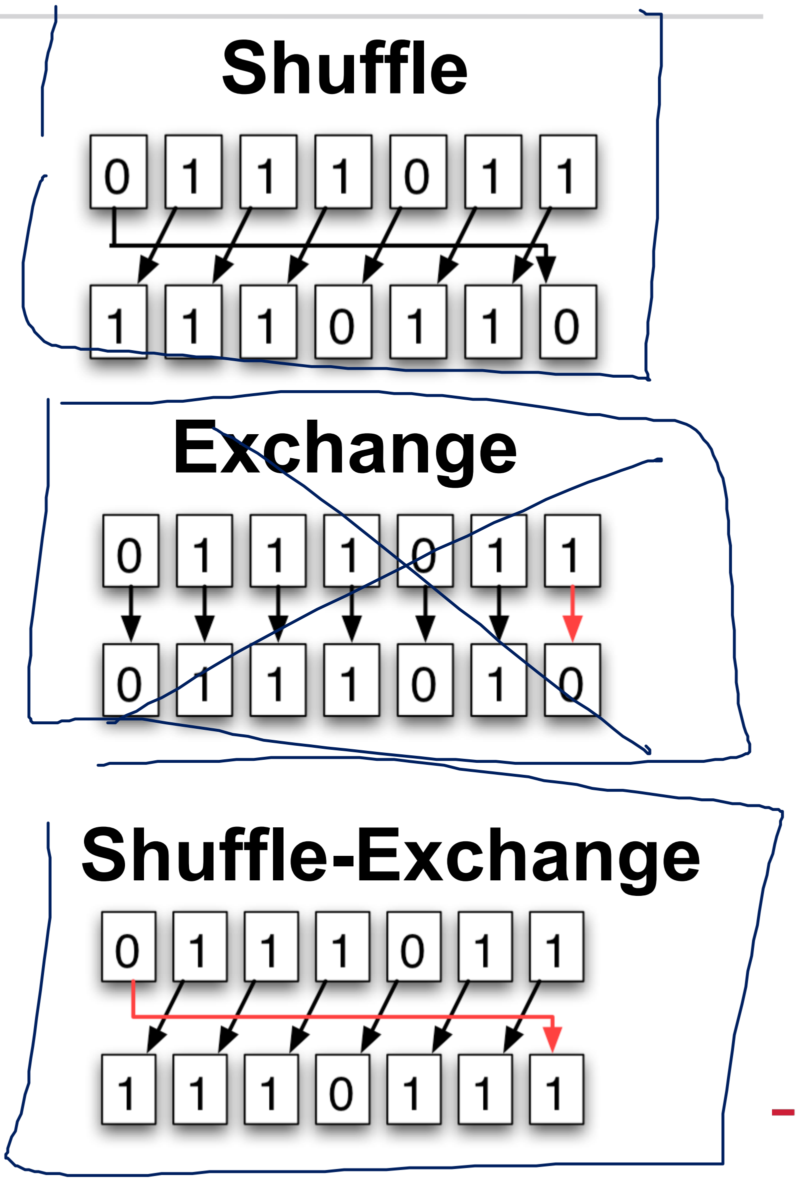
CoNe Freiburg

Shuffle, Exchange, Shuffle-Exchange

$s_i \in \{0,1\}$

- Consider binary string s of length m
  - shuffle operation:
    - $\text{shuffle}(s_1, s_2, s_3, \ldots, s_m) = (s_2, s_3, \ldots, s_m, s_1)$
  - exchange:
    - $\text{exchange}(s_1, s_2, s_3, \ldots, s_m) = (s_1, s_2, s_3, \ldots, \neg s_m)$
  - shuffle exchange:
    - $SE(S) = \text{exchange}(\text{shuffle}(S)) = (s_2, s_3, \ldots, s_m, \neg s_1)$
- Observation:

Every string a can be transformed into a string b by at most m shuffle and shuffle-exchange operations



UNI FREIBURG

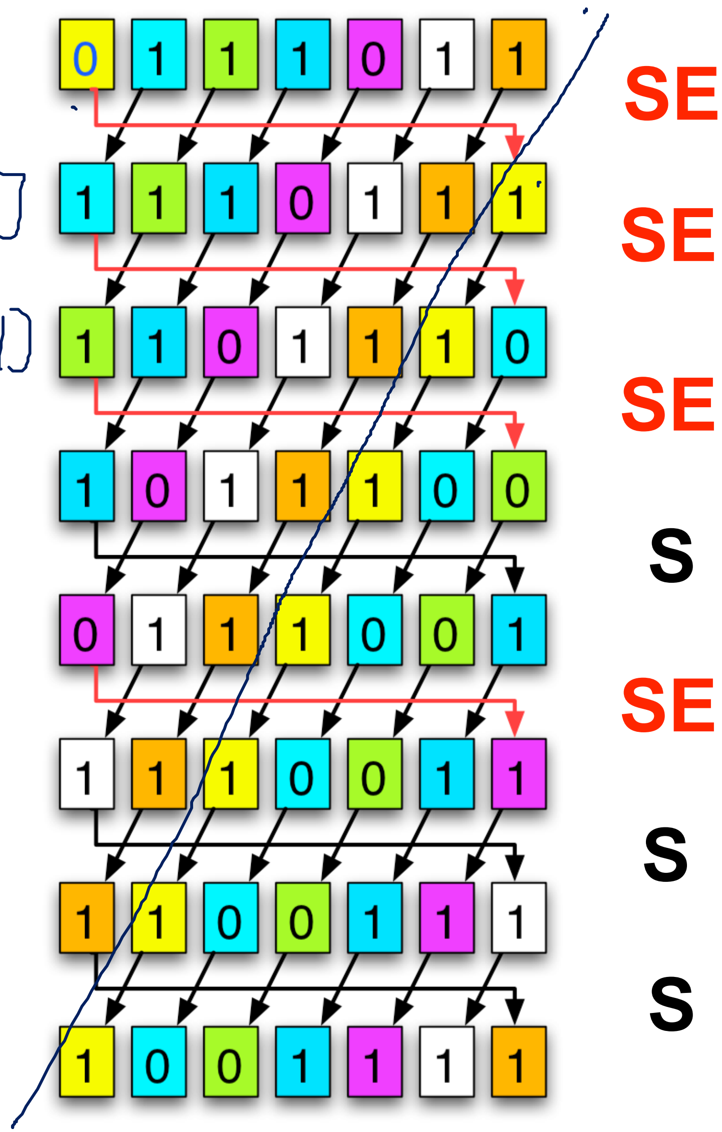
Magic Trick

- Observation

Every string a can be transformed into a string b by at most m shuffle and shuffle exchange operations Beispiel:

From	0	1	1	1	0	1	1
to	1	0	0	1	1	1	1
via	SE	SE	SE	S	SE	S	S

operations



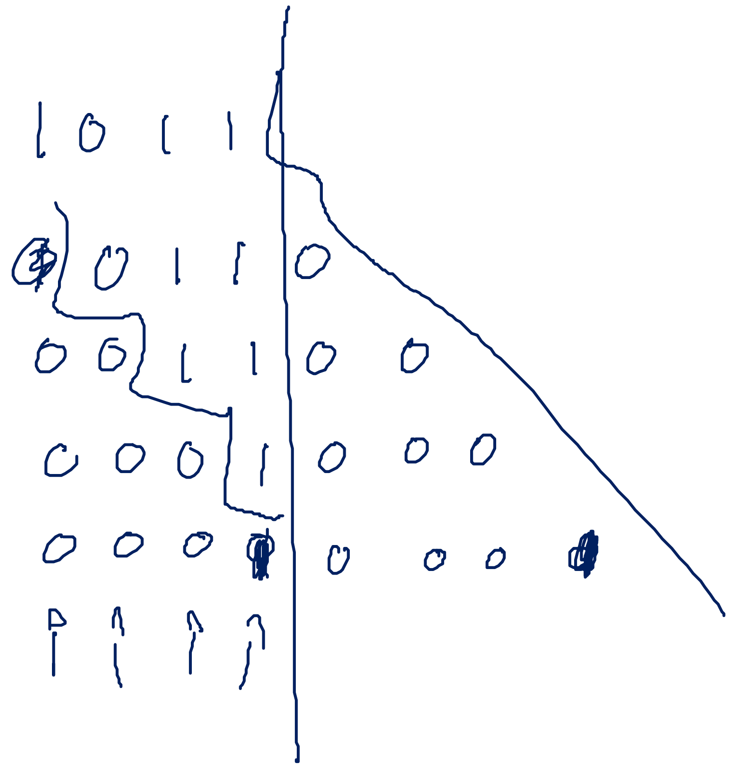
CoNe Freiburg

1 0 1 1
0 0 0 1

(1) 1 0 1 0
SE S SE S

st. 0 0 0 1 1 0 0 0 0 0
ta. 0 0 1 1 0 0 0 0 0 0

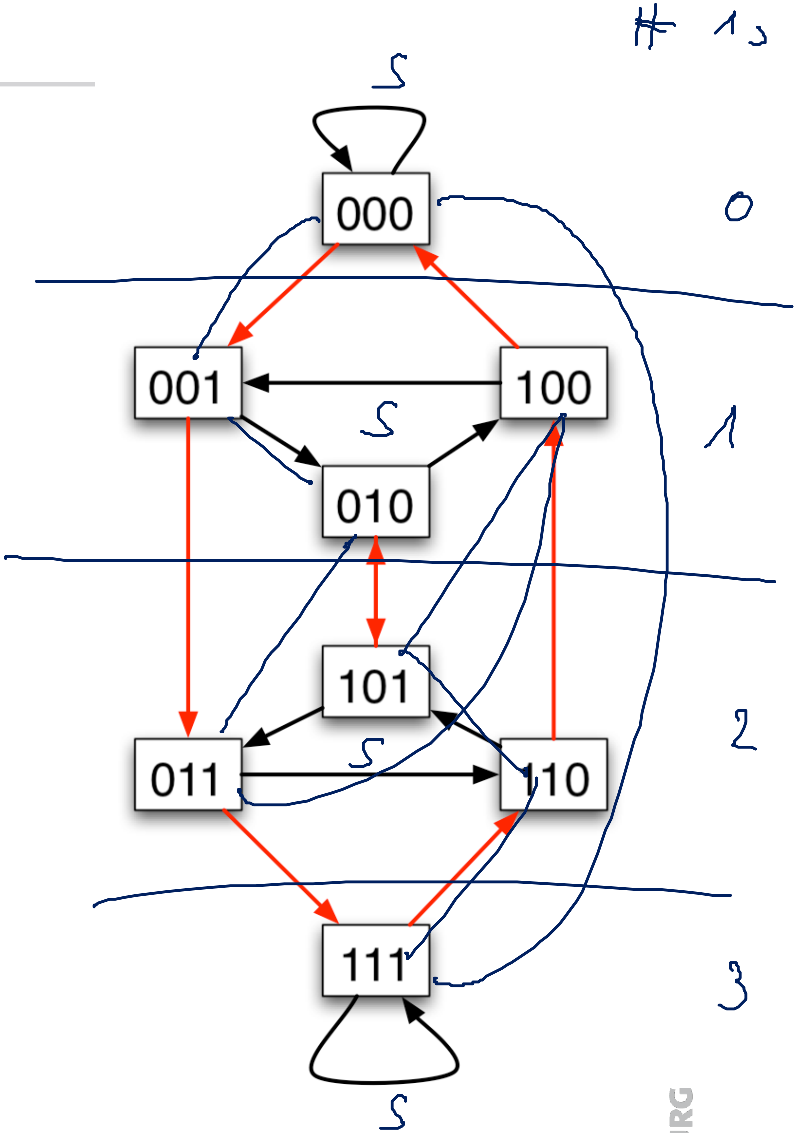
S S SE S SE S S S S



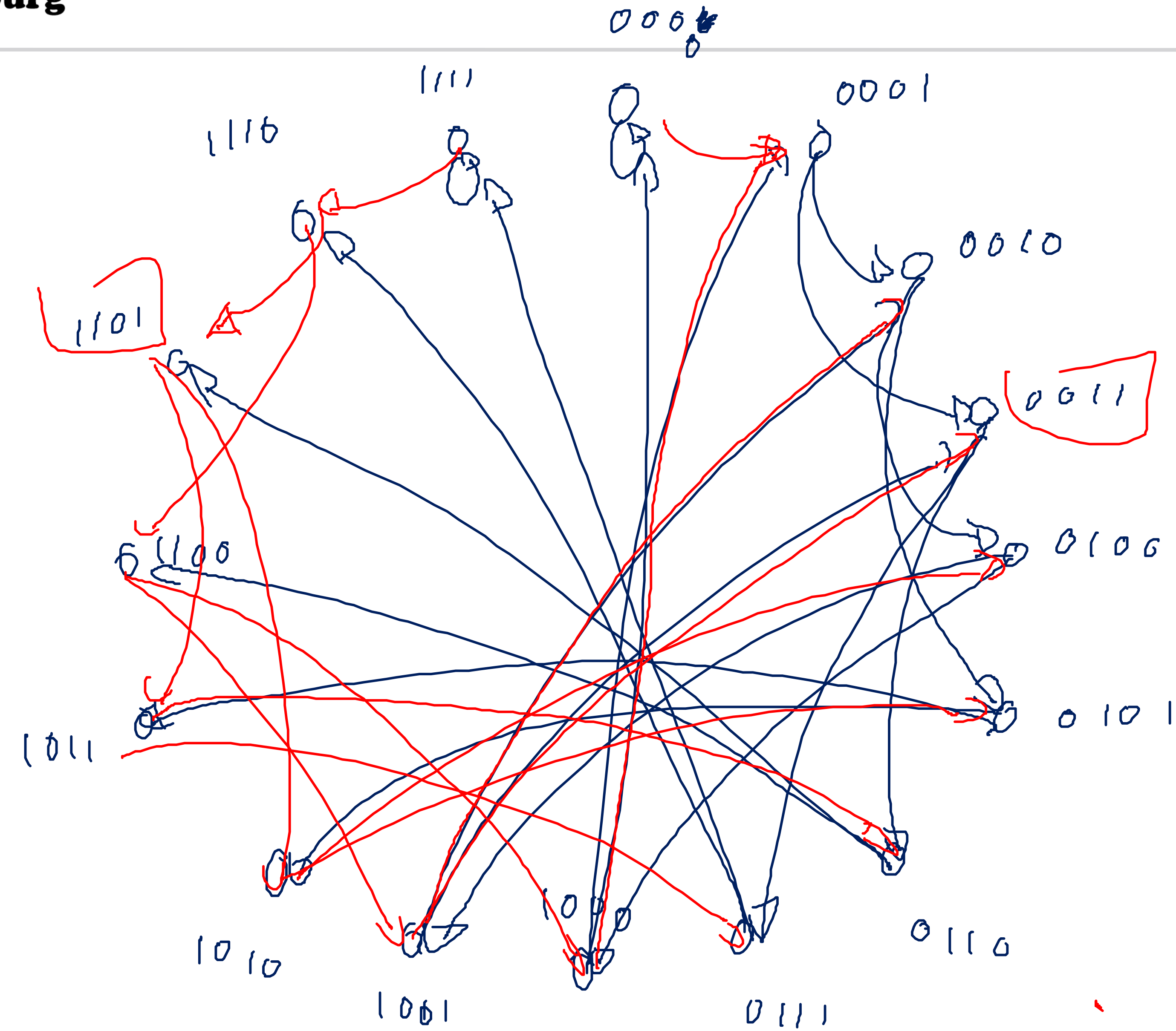
UNI FREIBURG

The De Bruijn Graph

- A De Bruijn graph consists of n=2m nodes,
  - each representing an m digit binary strings
- Every node has two outgoing edges
  - (u,shuffle(u))
  - (u, SE(u))
- Lemma
  - The De Bruijn graph has degree 2 and diameter log n
- Koorde = Ring + DeBruijn-Graph

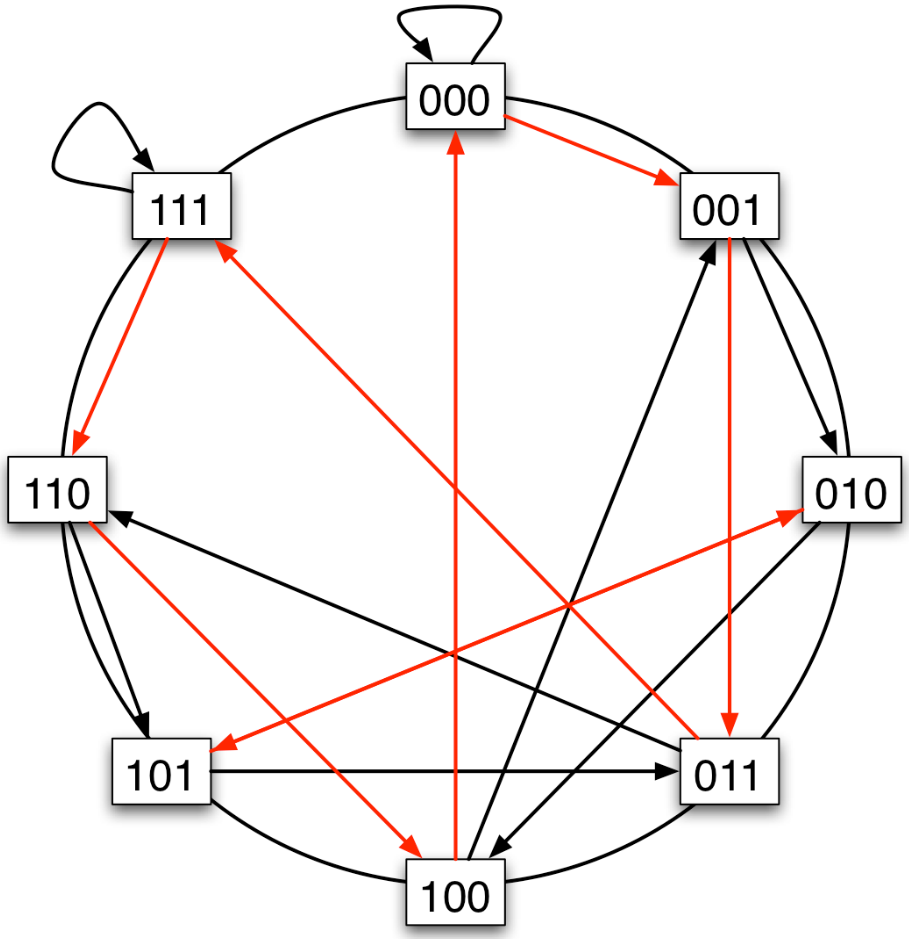


Koorde


0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

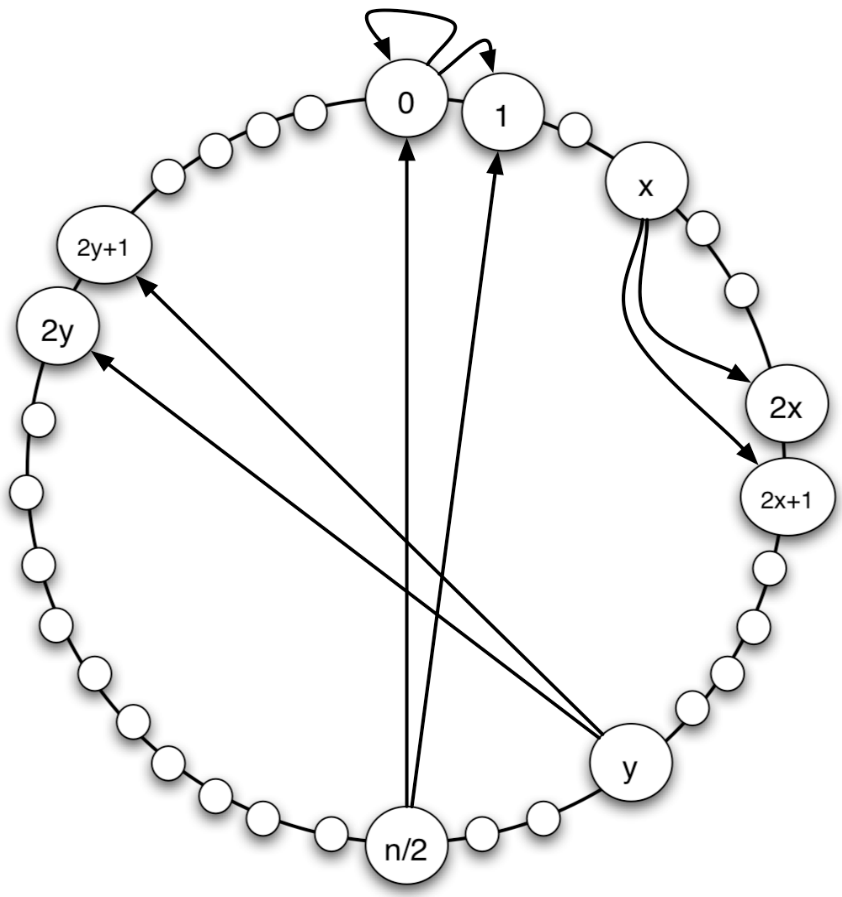
Koorde = Ring + DeBruijn-Graph

- Consider ring with 2^m nodes and De Bruijn edges



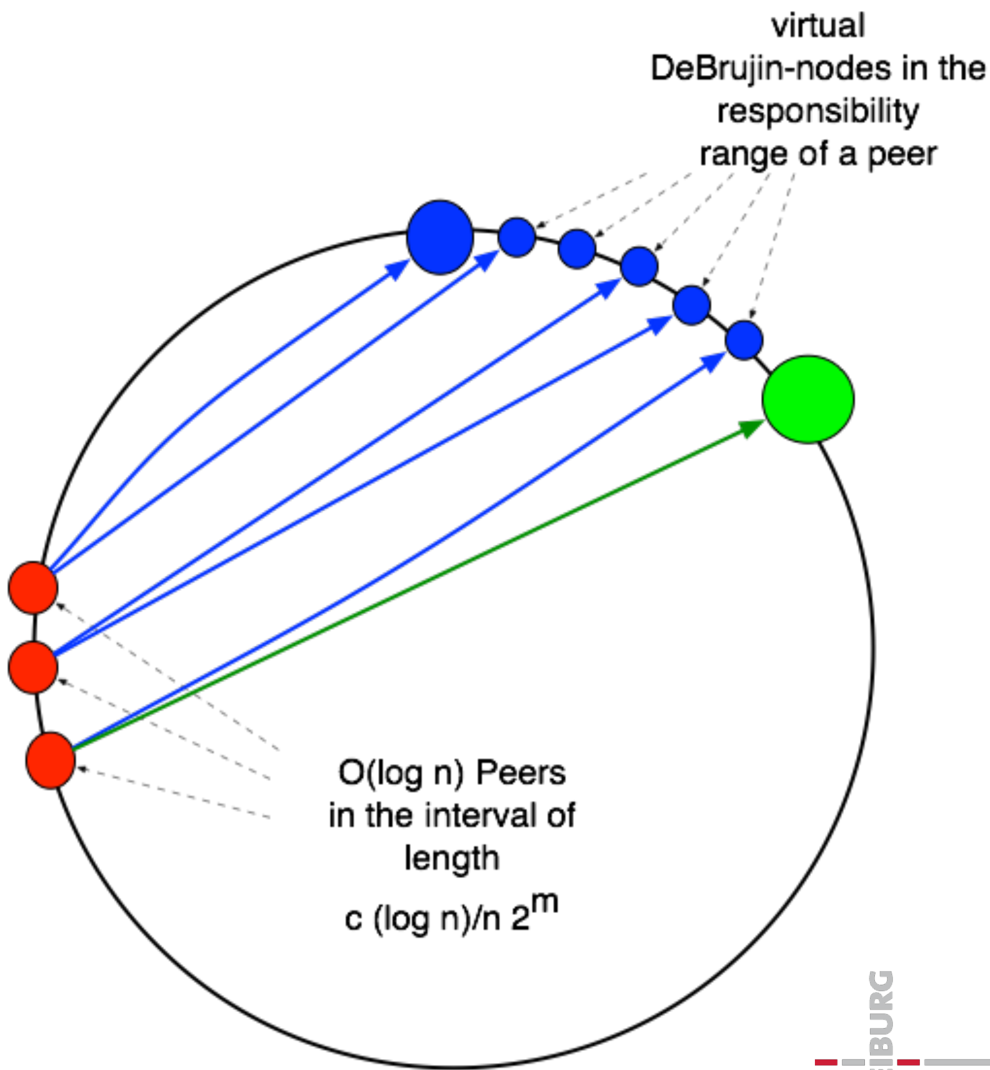
Koorde = Ring + DeBruijn-Graph

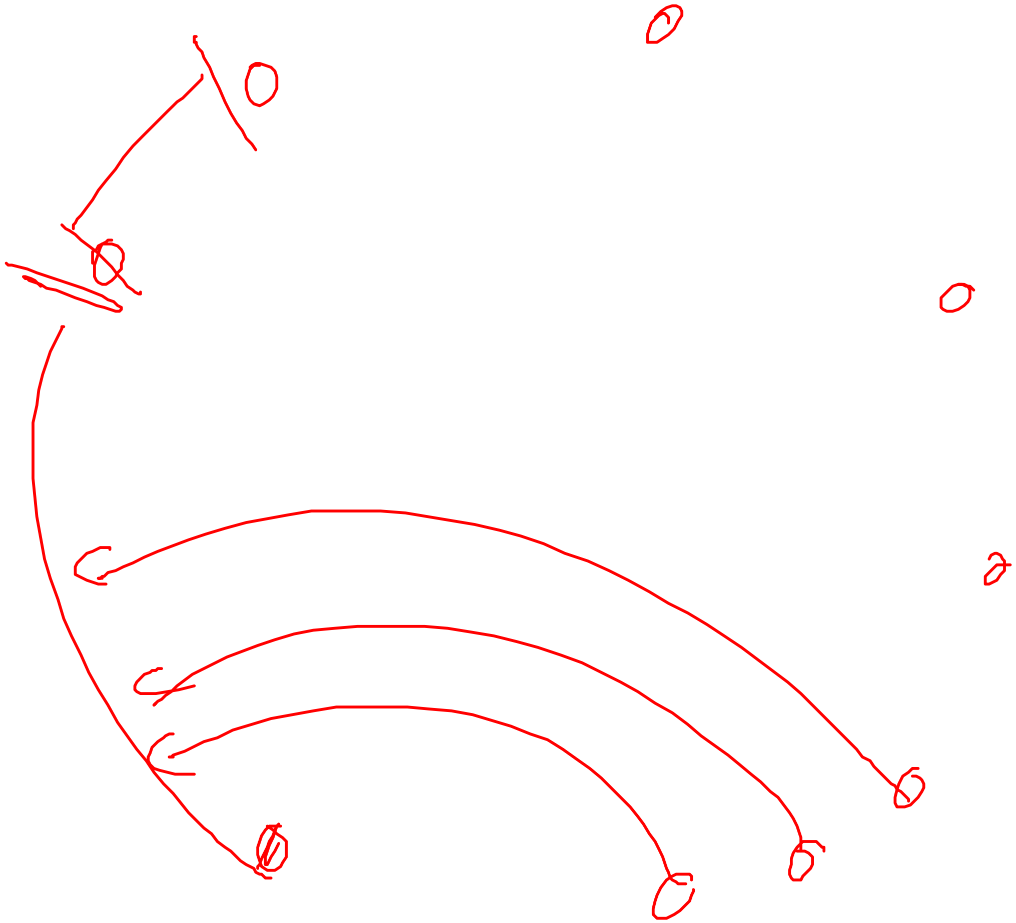
- Note
  - shuffle($s_1, s_2, \ldots, s_m$) = ($s_2, \ldots, s_m, s_1$)
    - shuffle (x) = (x div 2^{m-1})+(2x) mod 2^m
  - SE(S) = ($s_2, s_3, \ldots, s_m, \neg s_1$)
    - SE(x) = 1-(x div 2^{m-1})+(2x) mod 2^m
  - Hence: Then neighbors of x are
    - 2x mod 2^m and
    - 2x+1 mod 2^m



Virtual DeBruijn Nodes

- To avoid collisions we choose
  - m > (2+c) log (n)
- Then the probability of two peers colliding is at most n^{-c}
- But then we have much mor nodes in the graph than peers in the network
- Solution
  - Every peer manages all DeBruijn nodes between his position and his successor on the ring
  - only for incoming edges
  - outgoing edges are considered only from the peer‘s poisition on the ring





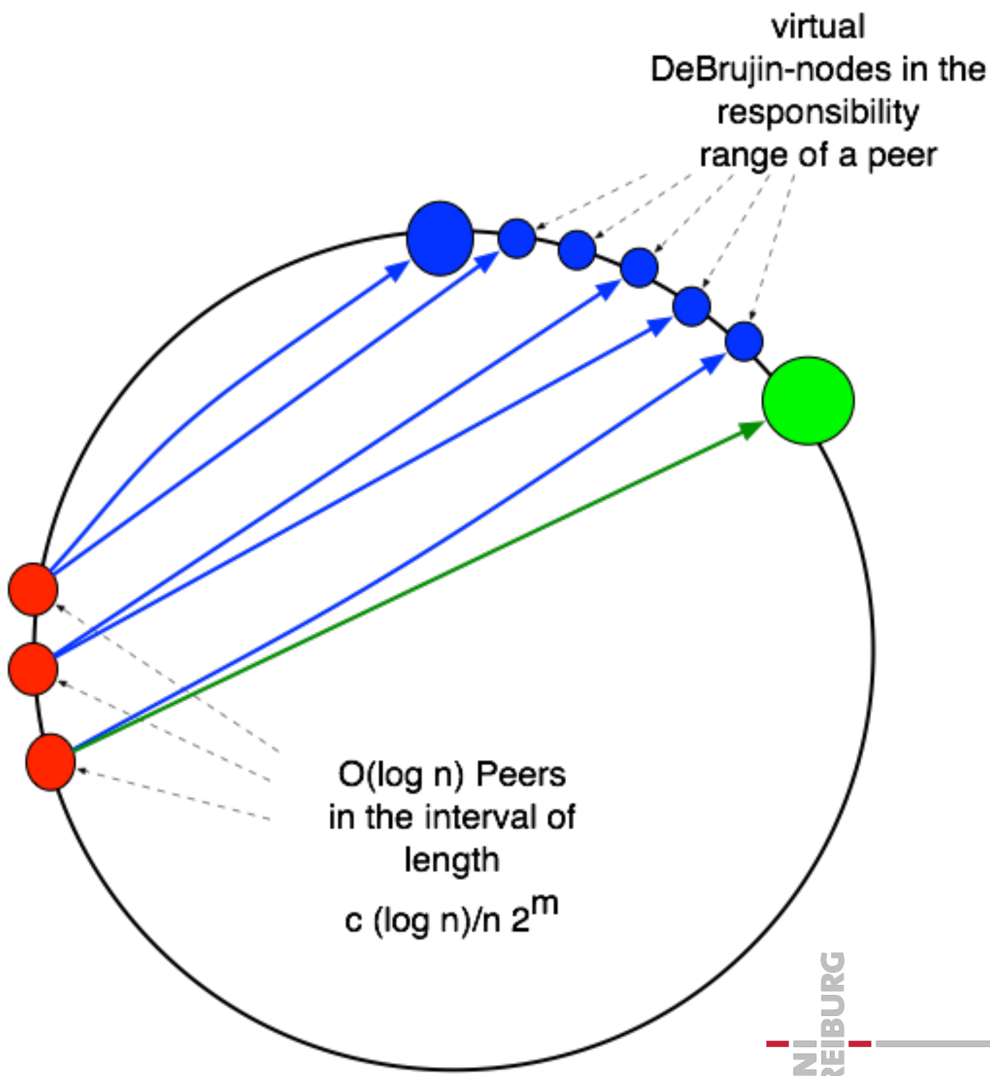
Properties of Koorde

- Theorem
  - Every node has four pointers
  - Every node has at most O(log n) incoming pointers w.h.p.
  - The diameter is O(log n) w.h.p.
  - Lookup can be performed in time O(log n) w.h.p.

- But:
  - Connectivity of the network is very low.

Properties of Koorde

- Theorem
  - 1. Every node has four pointers
  - 2. Every node has at most O(log n) incoming pointers w.h.p.
- Proof:
  - 1. follows from the definition of the De Bruijn graph and the observation that only non-virtual nodes have outgoing edges
  - 2. The distance between two peers is at most c (log n)/n 2^m with high probability
  - The number of nodes pointing to this distance is therefore at most c (log n) with high probability



Properties of Koorde

- Theorem
  - The diameter is $O(\log n)$ w.h.p.
  - Lookup can be performed in time $O(\log n)$ w.h.p.
- Proof sketch:
  - The minimal distance of two peers is at least $n^{-c} 2^m$ w.h.p.
  - Therefore use only the $c \log n$ most significant bits in the routing
    - since the prefix guarantees that one end in the responsibility area of a peer
  - Follow the routing algorithm on the De Bruijn-graph until one ends in the responsibility area of a peer

0 0 0 1 0

~~2~~ 0 0 1 0 0

0 0 1 0 1

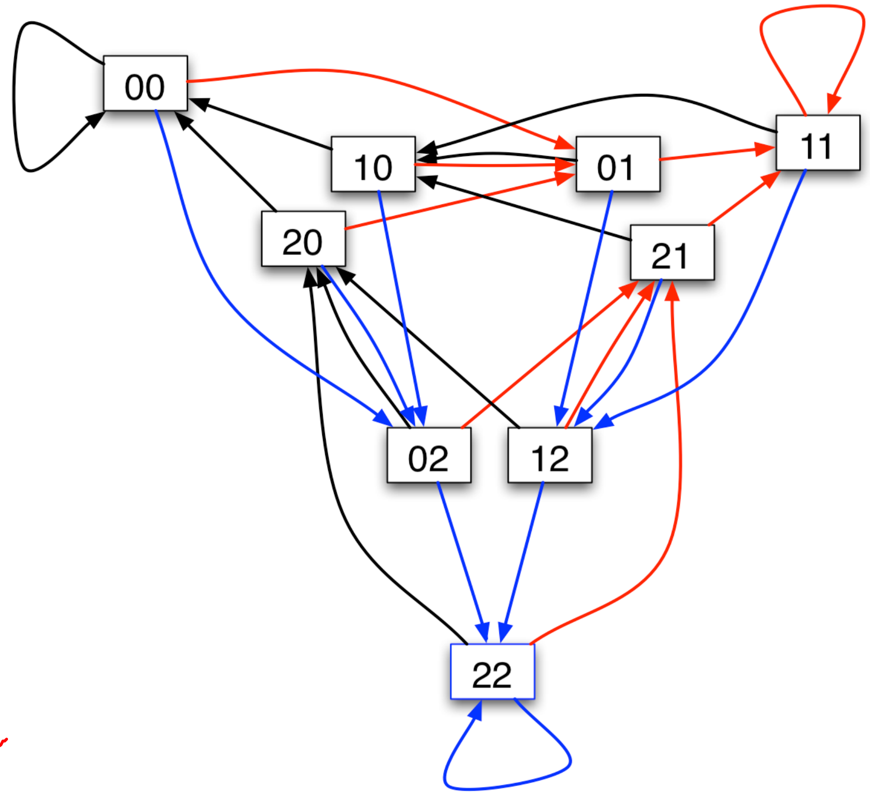
0 0 1 0 2

0 0 1 0 3

0 0 1 0 4

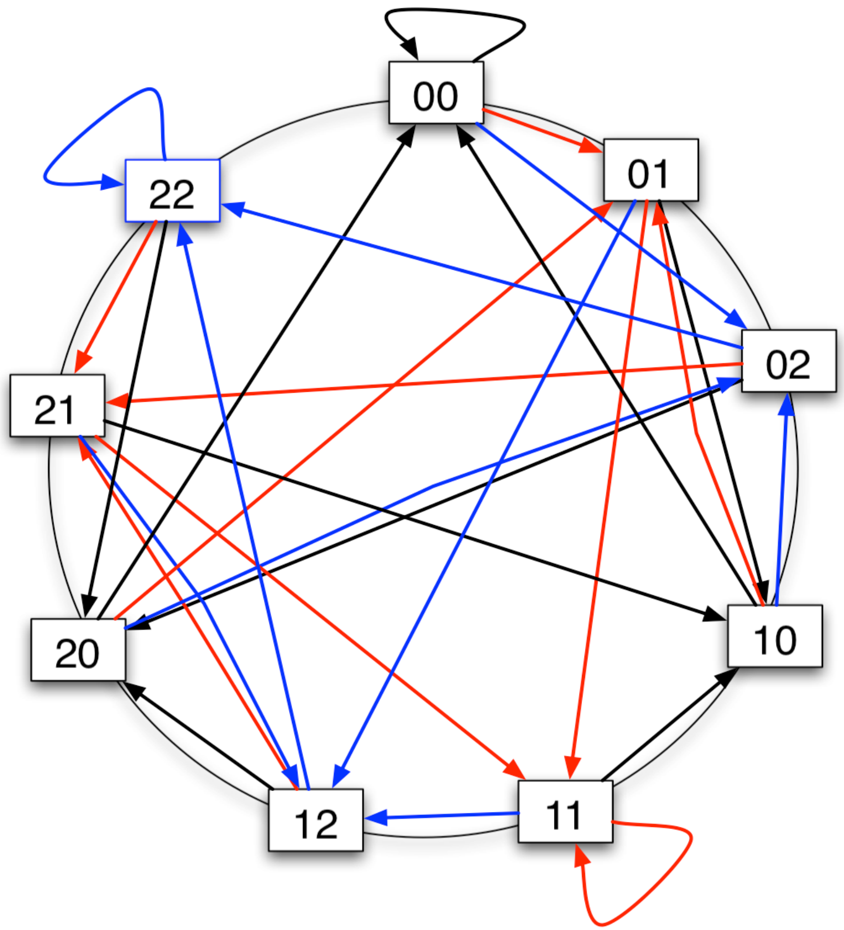
Degree k-DeBruijn-Graph

- Consider alphabet using k letters, e.g. k = 3
- Now, every k-De Bruijn-node has successors
  - (kx mod km)
  - (kx +1 mod km)
  - (kx+2 mod km)
  - ... (kx+k-1 mod km)
- Diameter is reduced to
  - (log m)/(log k) ⟷ Pastry
- Graph connectivity is increased to k



k-Koorde

- Straight-forward generalization of Koorde
  - by using k-De Bruijn graphs
- Improves lookup time and messages to O((log n)/(log k)) steps



Peer-to-Peer Networks

07 Degree Optimal Networks

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