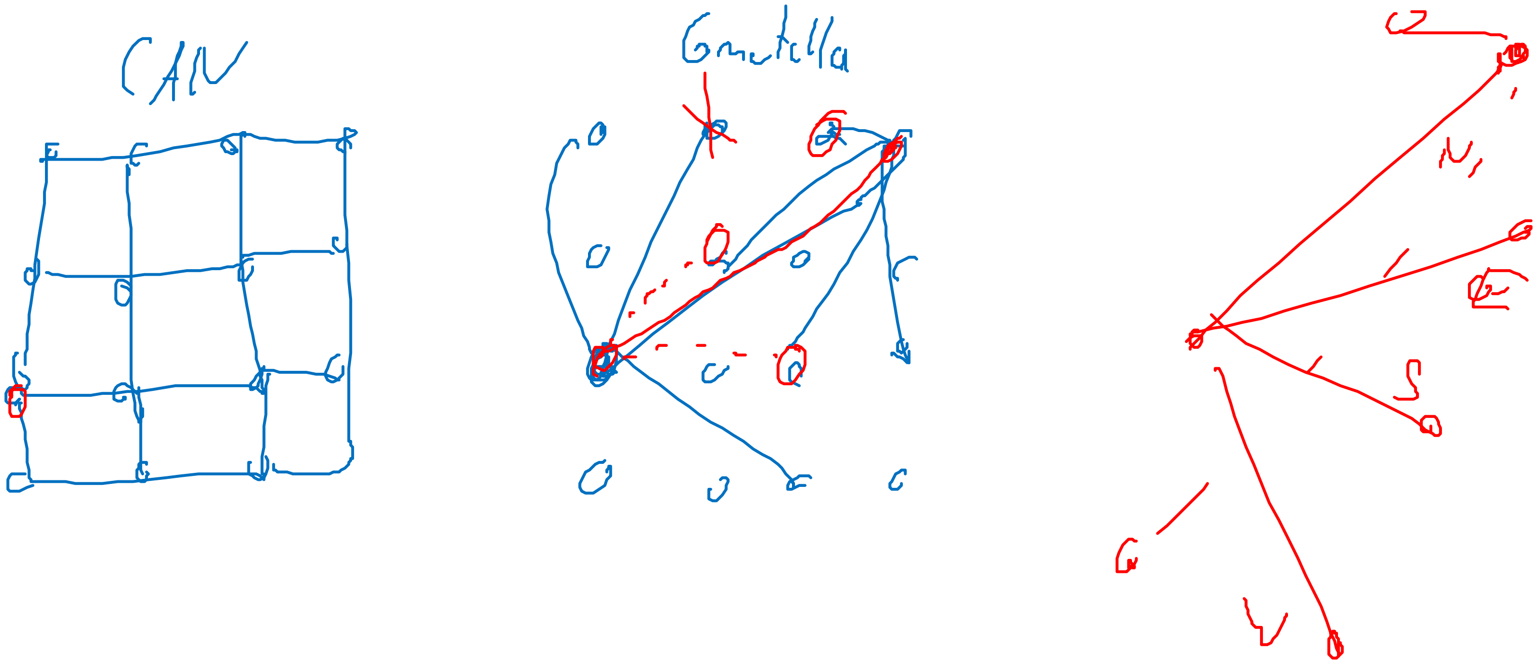
Peer-to-Peer Networks

15 Self-Organization

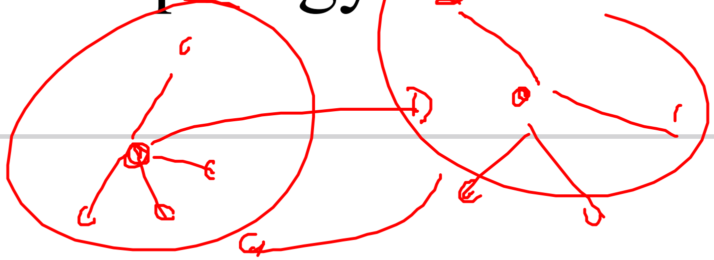
Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

Topology-Management

- T-Man: Fast Gossip-based Construction of Large-Scale Overlay Topologies Mark Jelasity Ozalp Babaoglu, 1994



Distributed Topology Construction
T-Man



(a) active thread

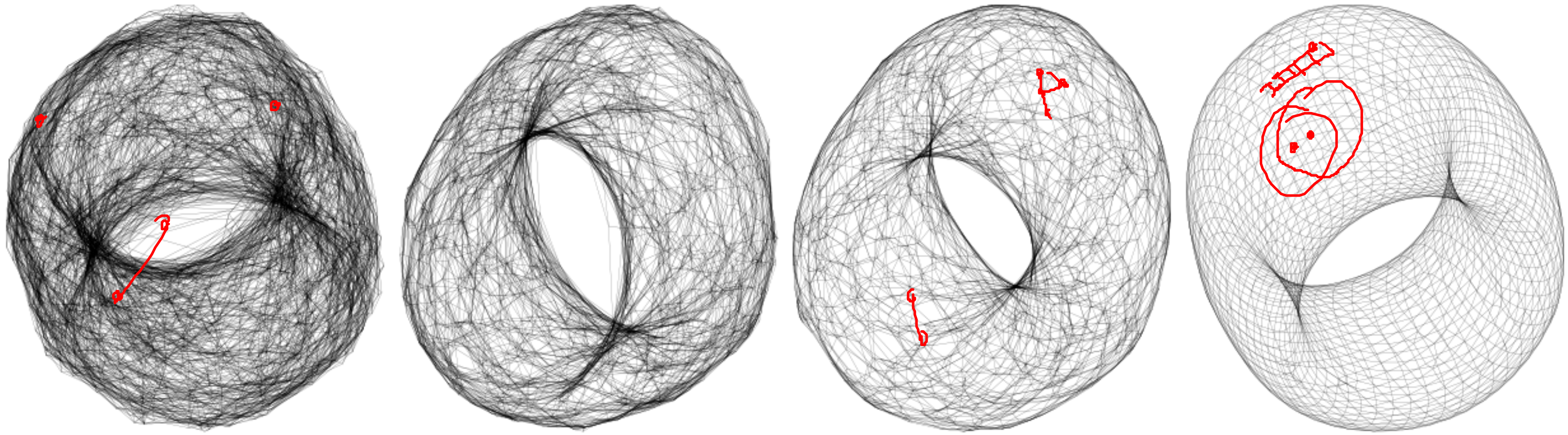
```
do at a random time once in each
consecutive interval of T time units
  p ← selectPeer()
  myDescriptor ← (myAddress,myProfile)
  buffer ← merge(view,{myDescriptor})
  buffer ← merge(buffer,rnd.view)
  send buffer to p
  receive buffer_p from p
  buffer ← merge(buffer_p,view)
  view ← selectView(buffer)
```

(b) passive thread

```
do forever
  receive buffer_q from q
  myDescriptor ← (myAddress,myprofile)
  buffer ← merge(view,{myDescriptor})
  buffer ← merge(buffer,rnd.view)
  send buffer to q
  buffer ← merge(buffer_q,view)
  view ← selectView(buffer)
```

Fig. 1. The T-MAN protocol.

Finding a Torus



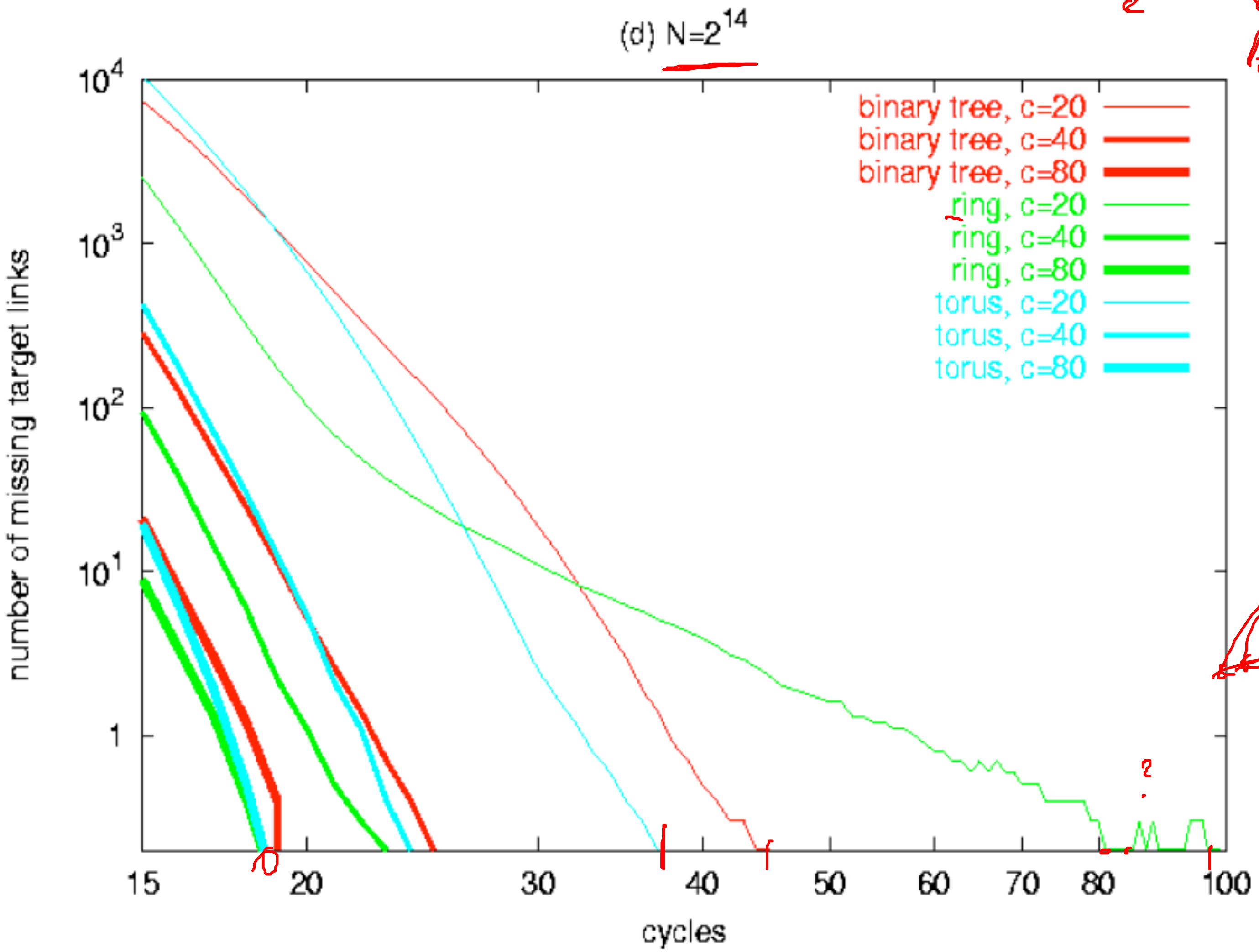
after 3 cycles after 5 cycles after 8 cycles after 15 cycles

Fig. 2. Illustrative example of constructing a torus over $50 \times 50 = 2500$ nodes, starting from a uniform random topology with $c = 20$. For clarity, only the nearest 4 neighbors (out of 20) of each node are displayed.

CoNe Freiburg

Convergence of T-MAN

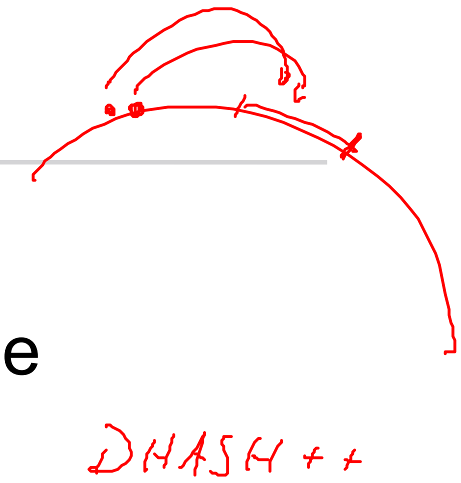
$2^4 \cdot 2^{10} = 16,384$



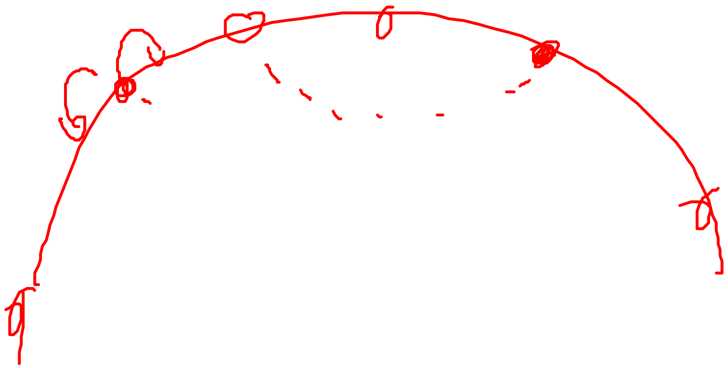
T-Chord

- Chord on demand, A Montresor, M Jelasity, O Babaoglu - Peer-to-Peer Computing, 2005
- Apply self-organization to Chord
  - compare insertion operation Pastry
- T-Chord
  - Apply T-Man
  - preferring Chord edges
- T-Chord-Prox
  - rank according to RTT

Ranking Function T-Chord



- 1st rank
  - nearest sucessor/predecessor on the ring $[0, 2^m - 1]$
- For each exponent $j \in [1, m-1]$
  - select from view the nodes nearest to $[\mathrm{ID} + 2^j \bmod 2^m, \mathrm{ID} + 2^{j+1} - 1 \bmod 2^m]$



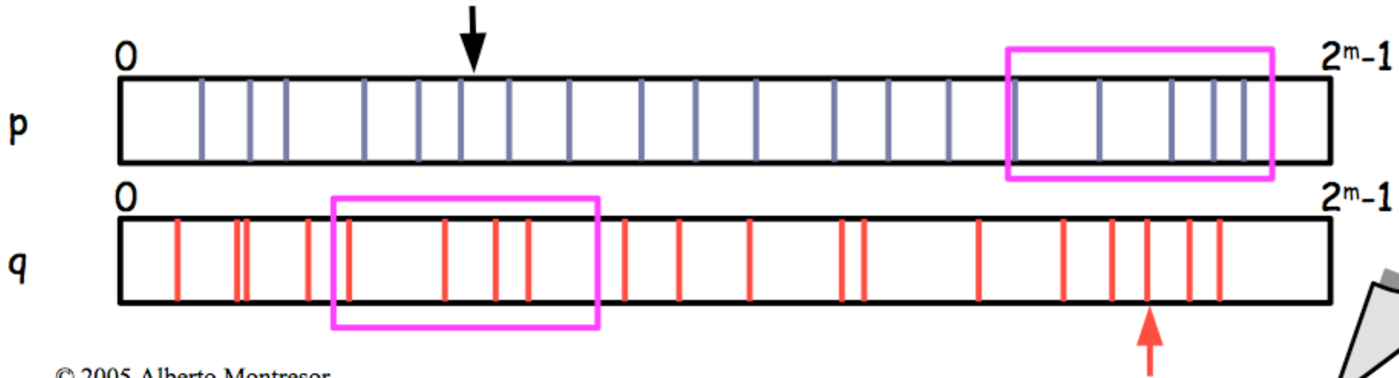
Ranking Function T-Chord-Prox

- 1st rank
  - nearest sucessor/predecessor on the ring $[0, 2^m - 1]$
- For each exponent $j \in [1, m-1]$
  - select from view the nodes nearest to
    $[\mathrm{ID} + 2^j \bmod 2^m, \mathrm{ID} + 2^{j+1} - 1 \bmod 2^m]$
  - measure latency (RTT) for p random nodes from view in such intervals and choose the closest

Adaption for Chord

T-Man for T-Chord

- selectPeer():
  - randomly select a peer ***q*** from the ***r*** nodes in my view that are *nearest to* **p** *in terms of ID distance*
- extract():
  - send to ***q*** the ***r*** nodes in local view that are *nearest to* **q**
  - ***q*** responds with the ***r*** nodes in its view that are *nearest to* **p**
- merge():
  - both ***p*** and ***q*** *merge* the received nodes to their view

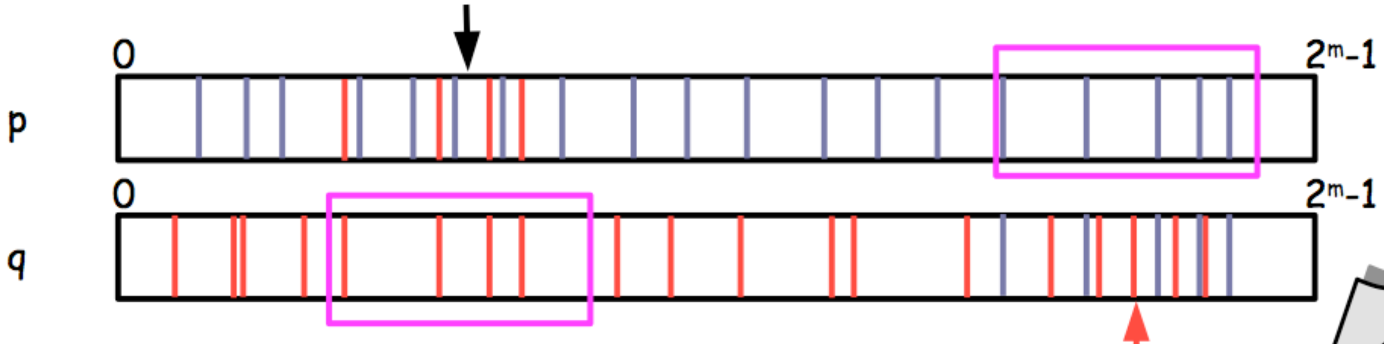


10

After Exchange of Links

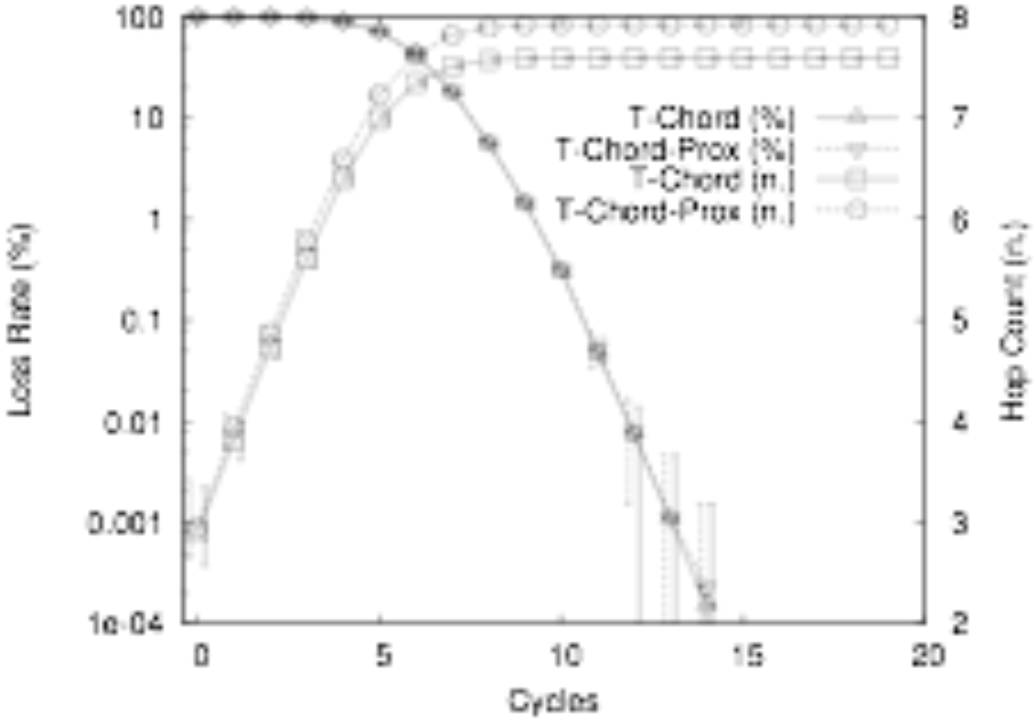
T-Man for T-Chord

- selectPeer():
  - randomly select a peer ***q*** from the ***r*** nodes in my view that are *nearest to* ***p*** *in terms of ID distance*
- extract():
  - send to ***q*** the ***r*** nodes in local view that are *nearest to* **q**
  - ***q*** responds with the ***r*** nodes in its view that are *nearest to* **p**
- merge():
  - both ***p*** and ***q*** *merge* the received nodes to their view



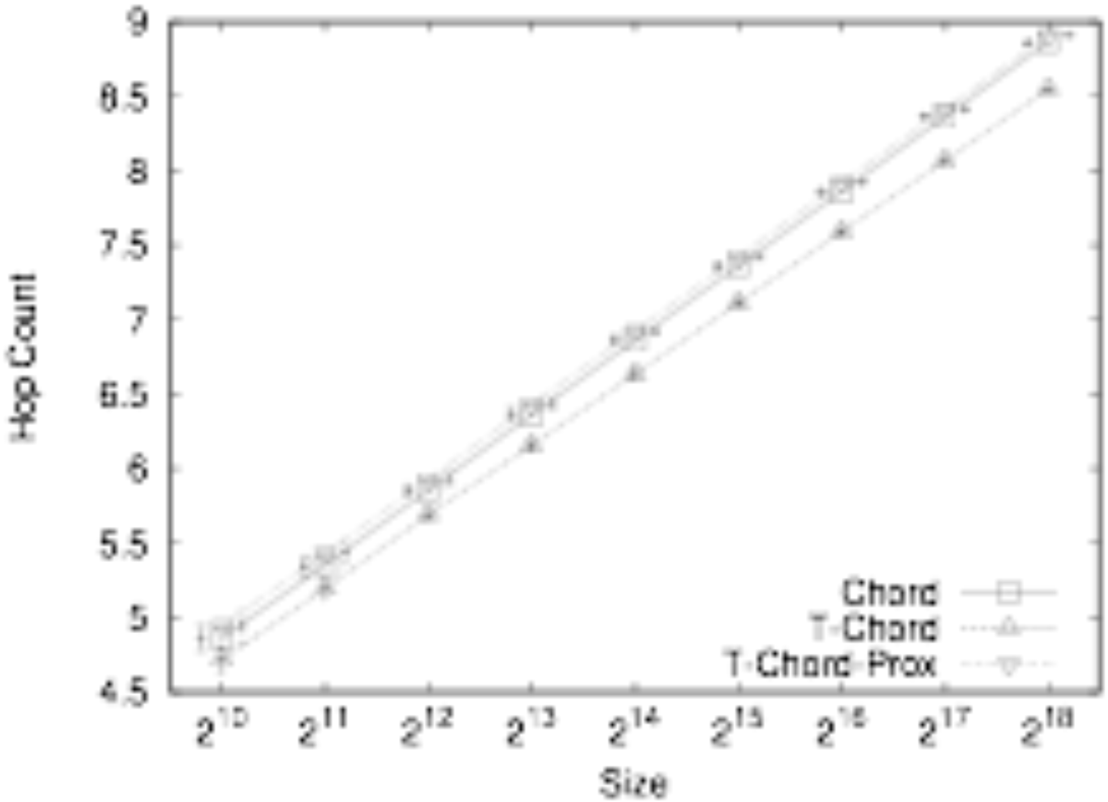
T-Chord Performance

- Starting with a neighbors on the ring
- Loss rate and hop count
  - experiments on a real-word dataset from 2002



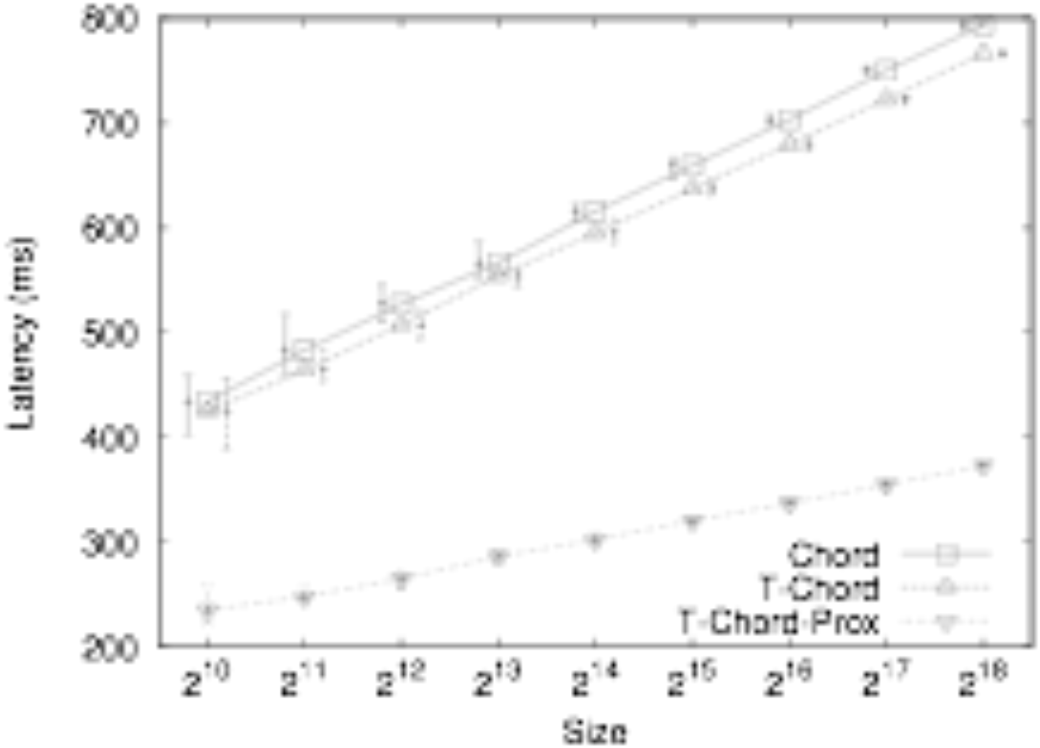
T-Chord Performance

- Starting with a neighbors on the ring
- Loss rate and hop count



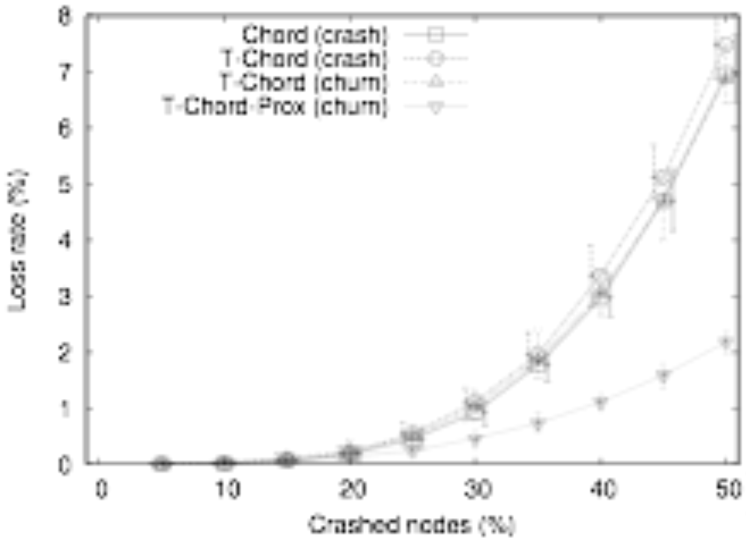
T-Chord Performance

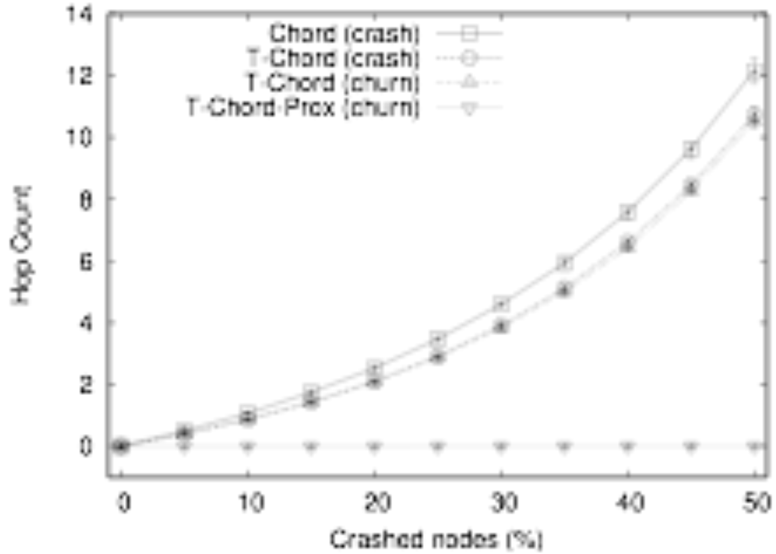
- Message Delay

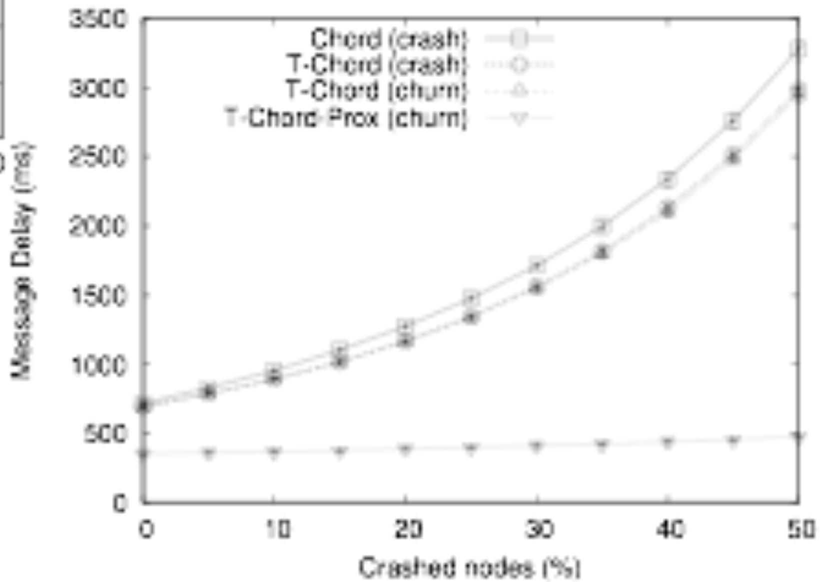


T-Chord Performance

- Robustness







Peer-to-Peer Networks
15 Self-Organization

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