Peer-to-Peer Networks
15 Self-Organization

Christian Ortolf
Technical Faculty
Computer-Networks and Telematics
University of Freiburg
Toplogy-Management

- T-Man: Fast Gossip-based Construction of Large-Scale Overlay Topologies Mark Jelasity Ozalp Babaoglu, 2004
Distributed Topology Construction
T-Man

**do** at a random time once in each consecutive interval of \( T \) time units

\[ p \leftarrow \text{selectPeer}() \]

\[ \text{myDescriptor} \leftarrow (\text{myAddress}, \text{myProfile}) \]

\[ \text{buffer} \leftarrow \text{merge}(\text{view}, \{\text{myDescriptor}\}) \]

\[ \text{buffer} \leftarrow \text{merge}(\text{buffer}, \text{rnd.view}) \]

send buffer to \( p \)

receive buffer\(_p\) from \( p \)

\[ \text{buffer} \leftarrow \text{merge}(\text{buffer}_p, \text{view}) \]

view \leftarrow \text{selectView}(\text{buffer})

(a) active thread

**do** forever

receive buffer\(_q\) from \( q \)

\[ \text{myDescriptor} \leftarrow (\text{myAddress}, \text{myprofile}) \]

\[ \text{buffer} \leftarrow \text{merge}(\text{view}, \{\text{myDescriptor}\}) \]

\[ \text{buffer} \leftarrow \text{merge}(\text{buffer}, \text{rnd.view}) \]

send buffer to \( q \)

receive buffer\(_q\) from \( q \)

\[ \text{buffer} \leftarrow \text{merge}(\text{buffer}_q, \text{view}) \]

view \leftarrow \text{selectView}(\text{buffer})

(b) passive thread

**Fig. 1.** The T-MAN protocol.
Finding a Torus

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.
Convergence of T-MAN

(d) $N=2^{14}$

number of missing target links

cycles

- binary tree, $c=20$
- binary tree, $c=40$
- binary tree, $c=80$
- ring, $c=20$
- ring, $c=40$
- ring, $c=80$
- torus, $c=20$
- torus, $c=40$
- torus, $c=80$
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
1st rank

- nearest successor/predecessor on the ring $[0, 2^m - 1]$

For each exponent $j \in [1, m - 1]$

- select from view the nodes nearest to $[\text{ID} + 2^j \mod 2^m, \text{ID} + 2^{j+1} - 1 \mod 2^m]$
1st rank

- nearest successor/predecessor on the ring $[0, 2^m - 1]$

For each exponent $j \in [1, m - 1]$

- select from view the nodes nearest to $[\text{ID} + 2^j \mod 2^m, \text{ID} + 2^{j+1} - 1 \mod 2^m]$

- 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.

© 2005 Alberto Montresor
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 random network
- Loss rate and hop count

![Graph showing T-Chord Performance](image)
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 Ortolf
Technical Faculty
Computer-Networks and Telematics
University of Freiburg