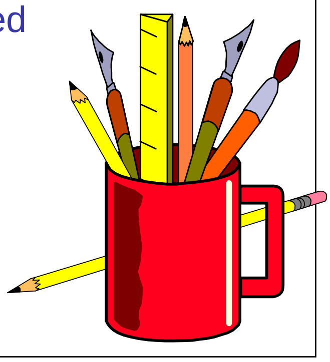
Scalable Routing for Unstructured Networks of Low-Resource Devices

Thomas Fuhrmann

IBDS Systemarchitektur
Universität Karlsruhe (TH)
Germany

Motivation: Moore's Law

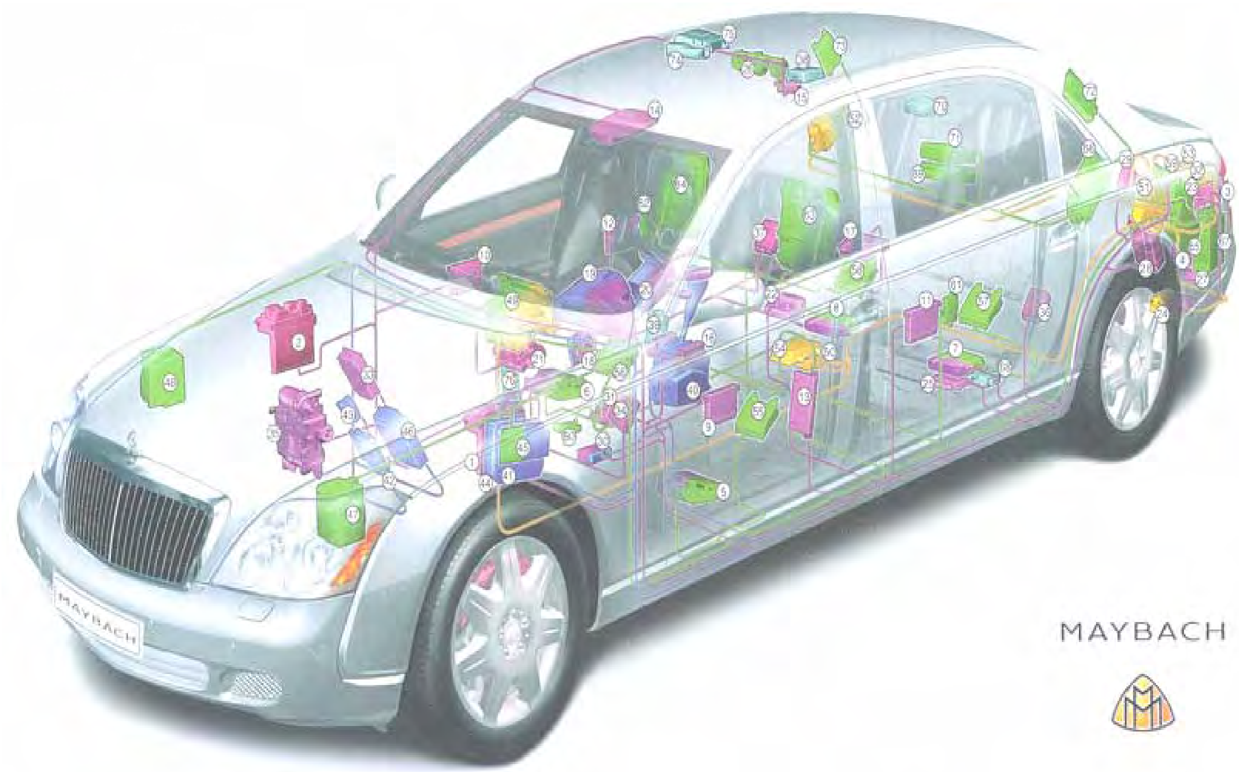
Swiss Army Knife	Toolbox
1. Moore's law makes electronic devices more powerful.	1. Moore's law makes computation and communication devices smaller and cheaper.
2. Software updates provide the devices with ever increasing capabilities.	2. Physical tools can be enhanced with electronics.



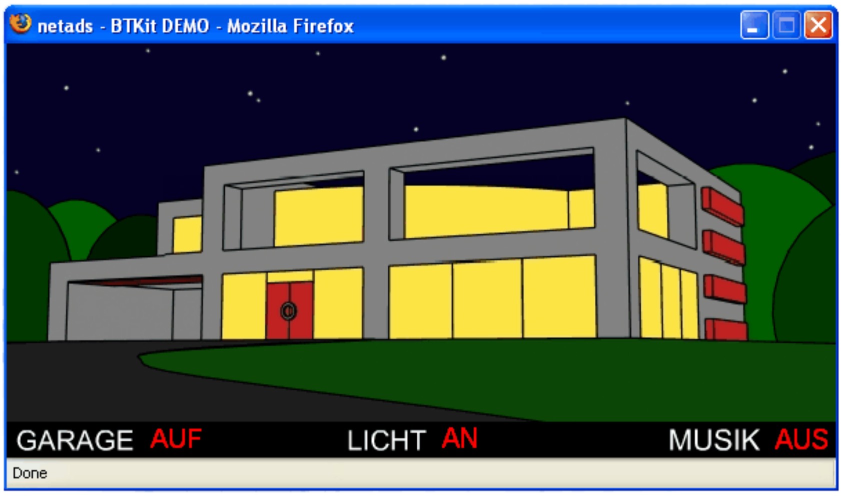
Our field of research:

Algorithms and protocols for distributed systems in the toolbox scenario.

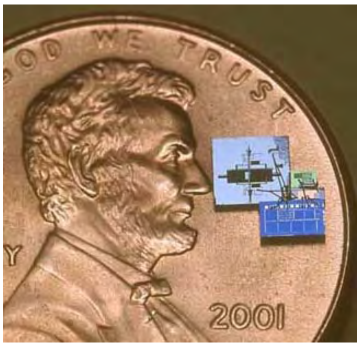
Cars already have plenty of electronics …



… are everyday environments next?



Self-Organizing Distributed Systems

Sensor Networks	Mobile Ad-hoc Networks	Mesh Networks
		
Low-resource wireless devices	Mobile devices without infrastructure	Combining various access infrastructures

SmartDust: J. M. Kahn and B. A. Warneke, Univerisity of California at Berkeley. Soldier: US-Army. Mesh-Netz: Netkrom Technologies Inc.

Problem Statement

- How can we route resource efficiently in self-organizing network?
  - Flooding is too expensive!
  - Holding routing tables is infeasible (no aggregation possible without structure)
  - Geographic routing does not work (no 2D setting)
  - Network planning is infeasible (no network administrator)

This talk presents a novel routing algorithm that is capable to route memory and message efficiently in networks that have arbitrary (random) topology.

Agenda

1. Motivation

2. Structured Routing Overlays

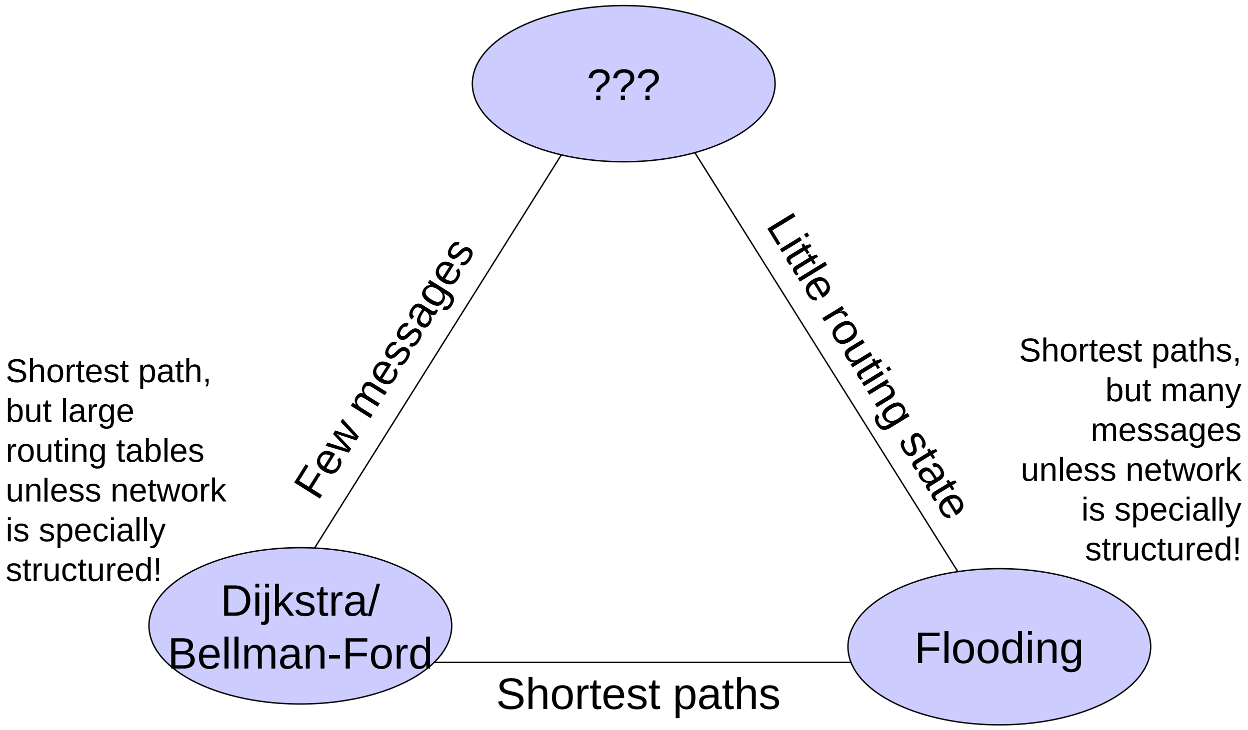
3. Scalable Source Routing

4. Summary

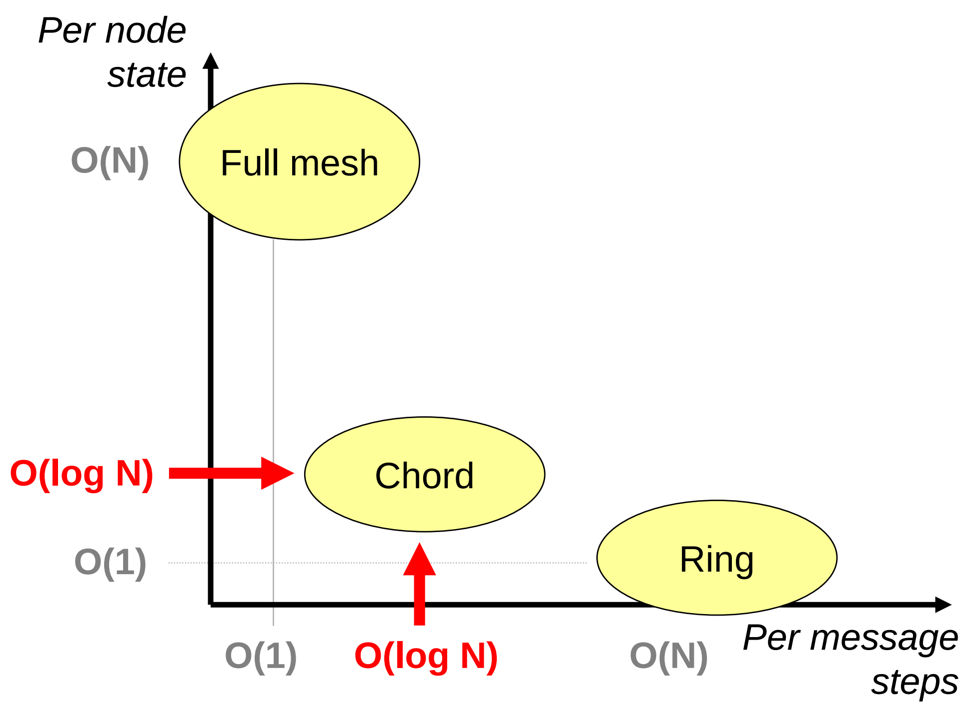
Routing – An overview (1)

Based on (static) Topology	Typically ad-hoc
1. **Dijkstra:** Each node stores the entire network graph in order to calculate spanning tree. 2. **Bellman-Ford:** Neighboring nodes exchange routing tables. Converges in stable networks.	1. **Flooding:** All nodes get all the messages. 2. **Geographic:** Nodes are embedded into vector space. Forwarding according to destination direction. Special care taken to avoid dead ends.
Network structure and addresses need to match. Otherwise routing tables explode.	Network structure needs embedding into vector space. Otherwise, too many messages sent.

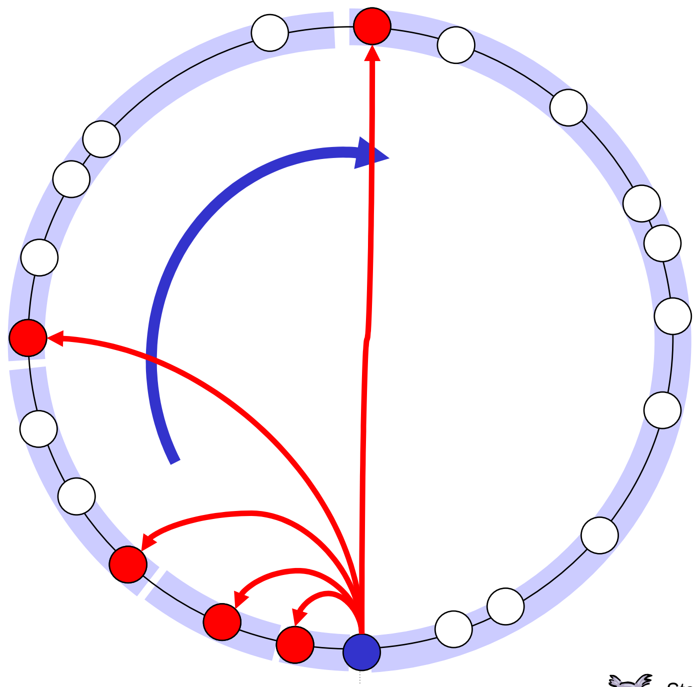
Routing – An Overview (2)



Trade-Off with Structured Routing Overlays

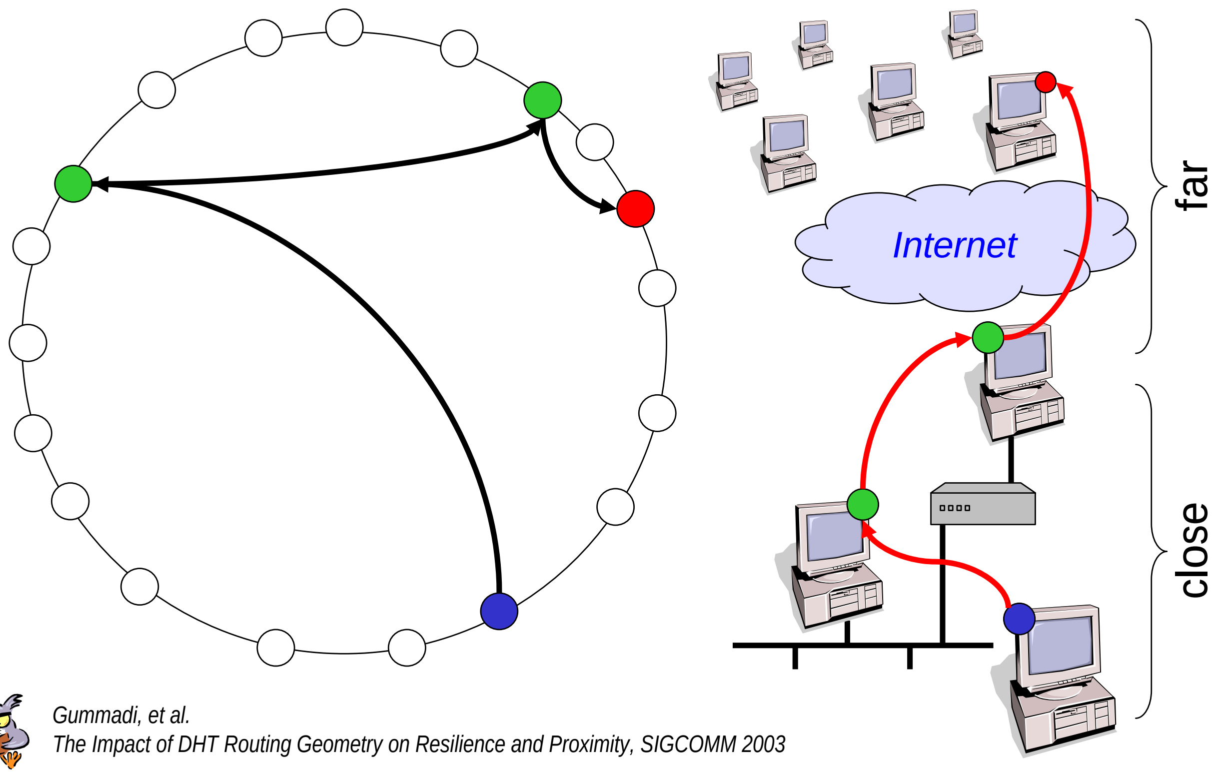


Chord – A Structured Routing Overlay



Stoica, et al. Chord: A scalable P2P lookup service, SIGCOMM 2001.

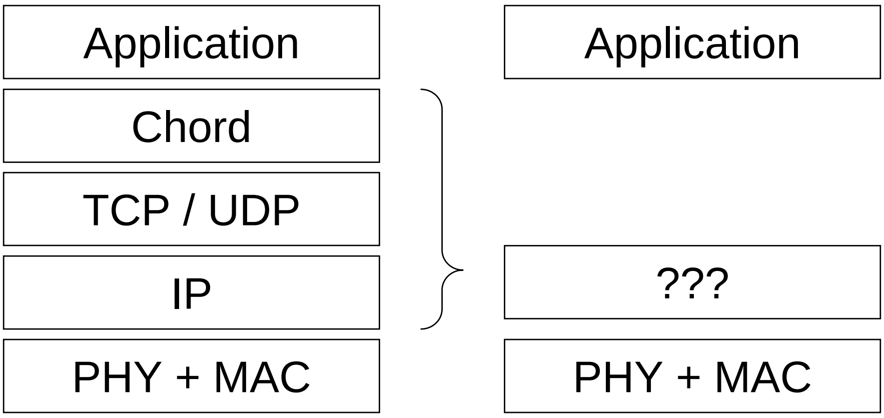
Proximity Awareness with Chord



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

Can we Push Chord into the Underlay?

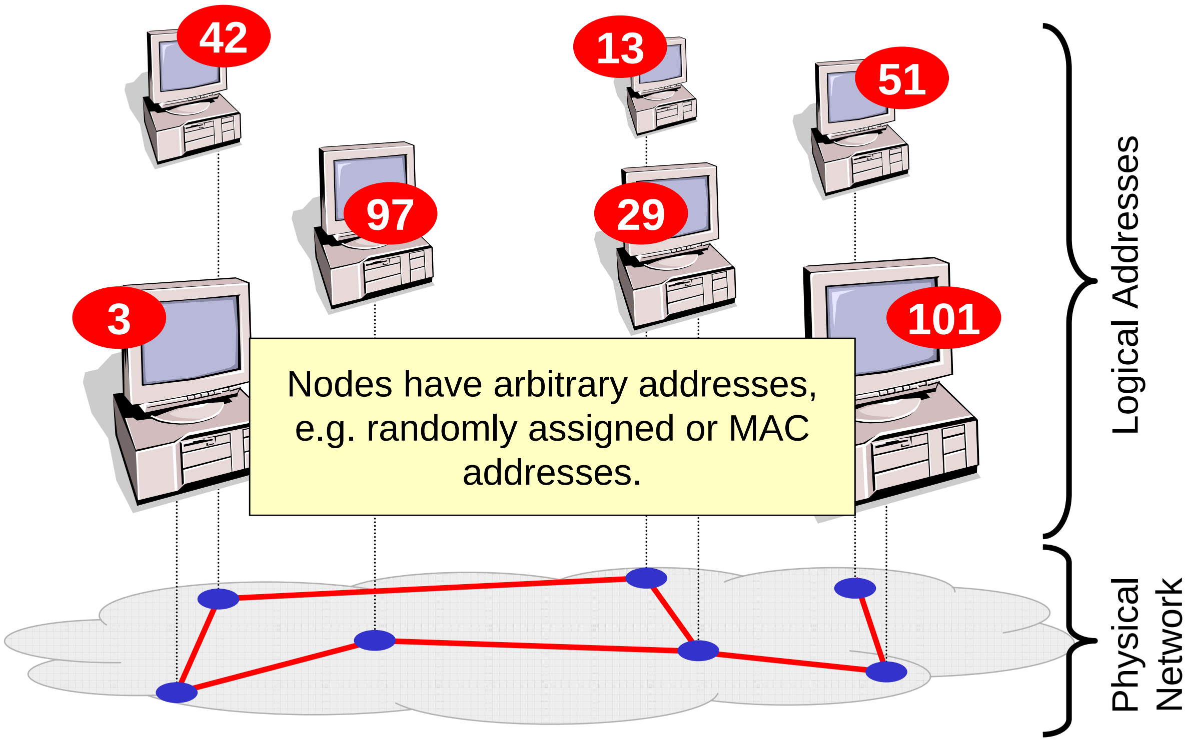
Does this Chord idea help
us with our sensor actuator vision
of thousands of houses with
hundreds of nodes each?



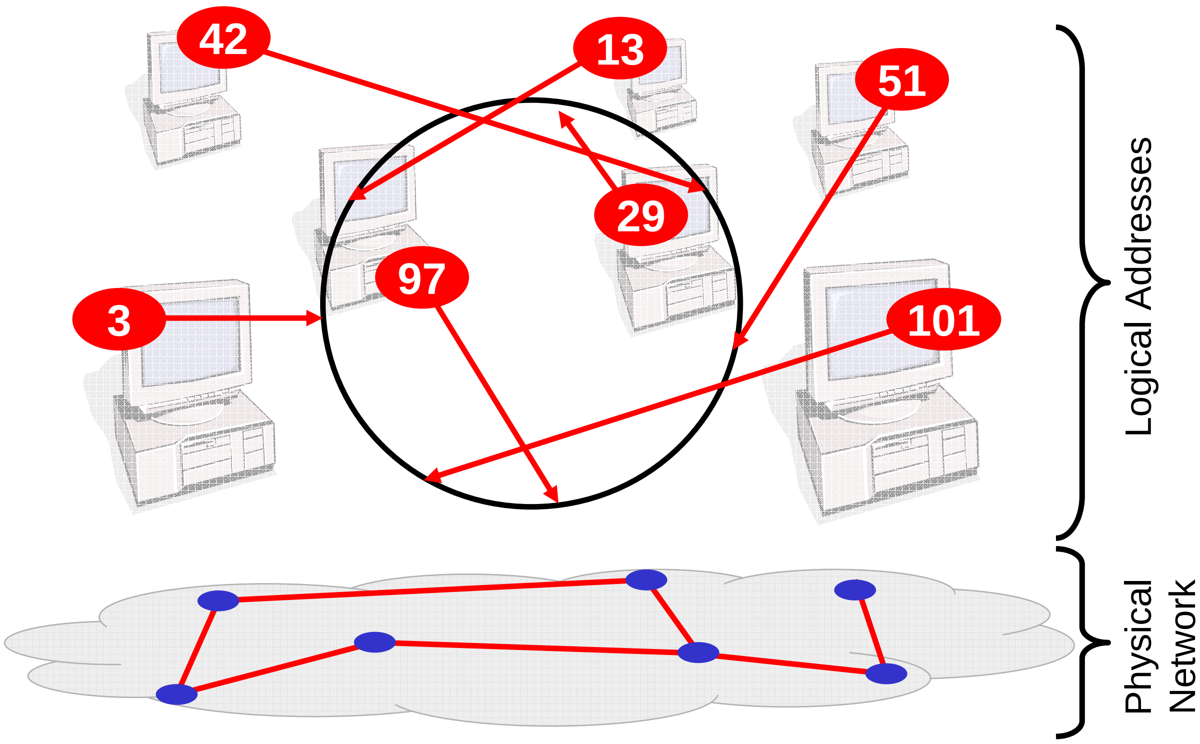
Agenda

1. Motivation
2. Structured Routing Overlays
3. Scalable Source Routing
4. Summary

A Physical Network with Any Topology …

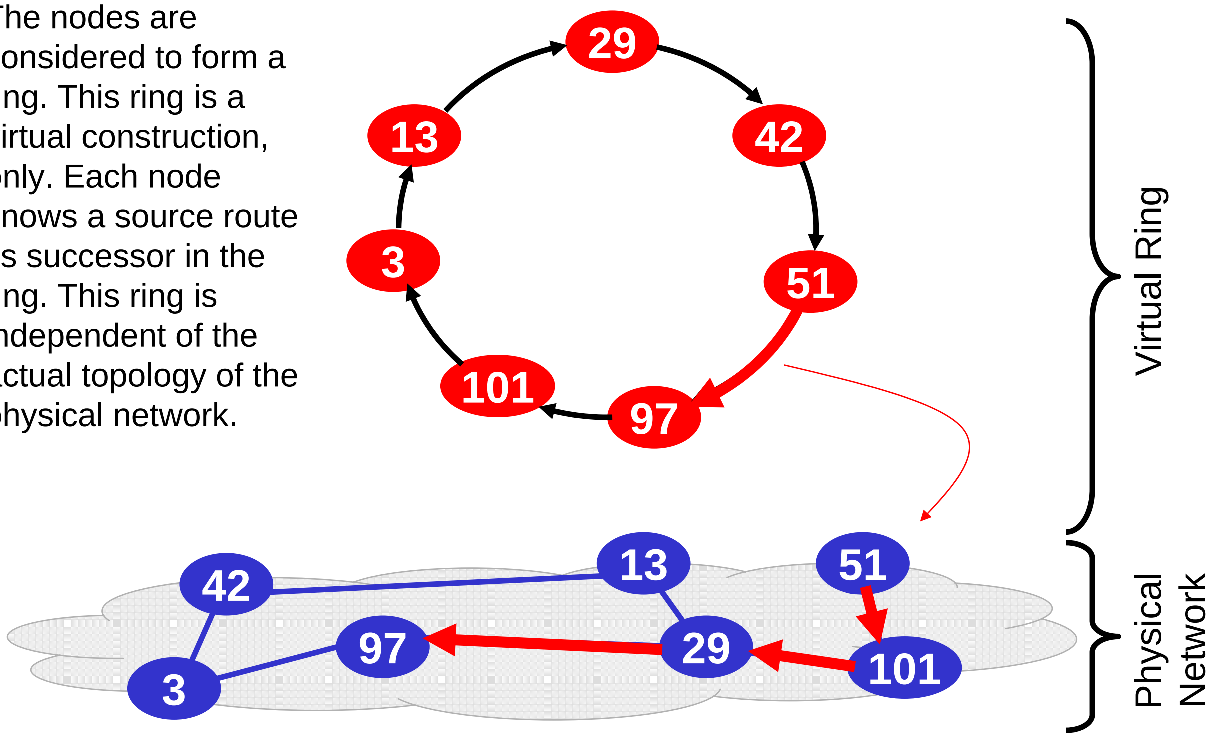


… Organized Into a Virtual Ring …



… Connected by Source Routes

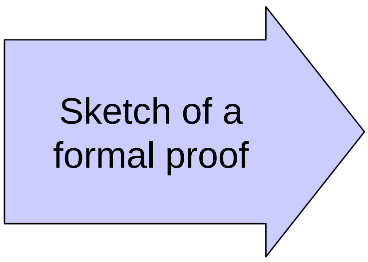
The nodes are considered to form a ring. This ring is a virtual construction, only. Each node knows a source route its successor in the ring. This ring is independent of the actual topology of the physical network.



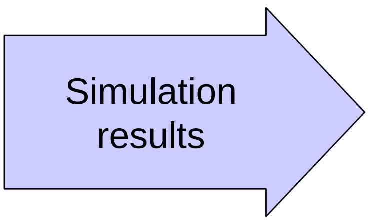
A Virtual Ring of Nodes Connected by Source Routes

Claim 1: If each node knows a source route to its successor (in the virtual ring), any node can reach any destination.
Proof: cf. routing along the ring

Claim 2: These source routes can be obtained without flooding.



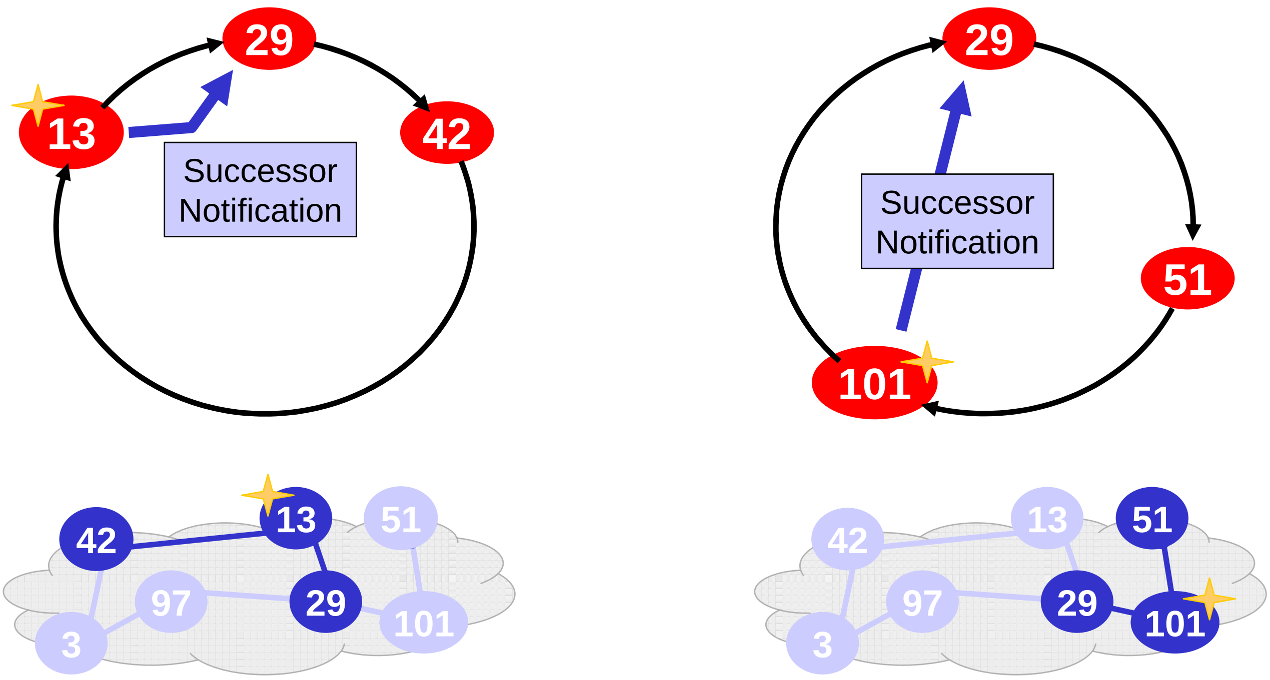
Claim 3: A small per node cache suffices to achieve efficient routing.



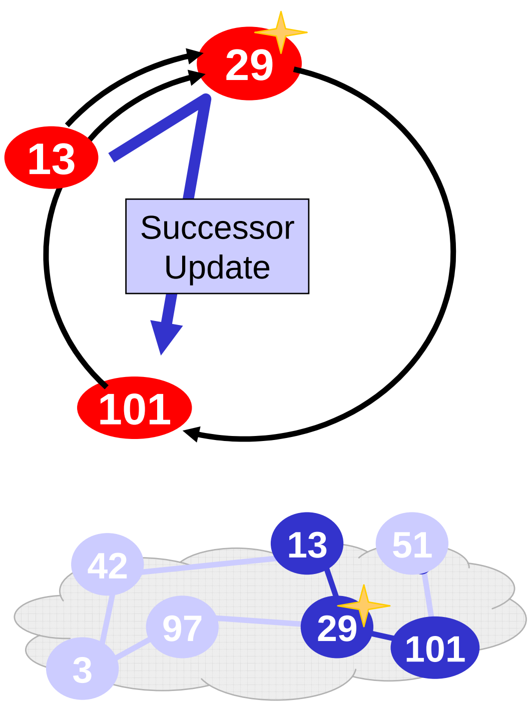
Iterative Successor Search (1)

Viewed from node 13:

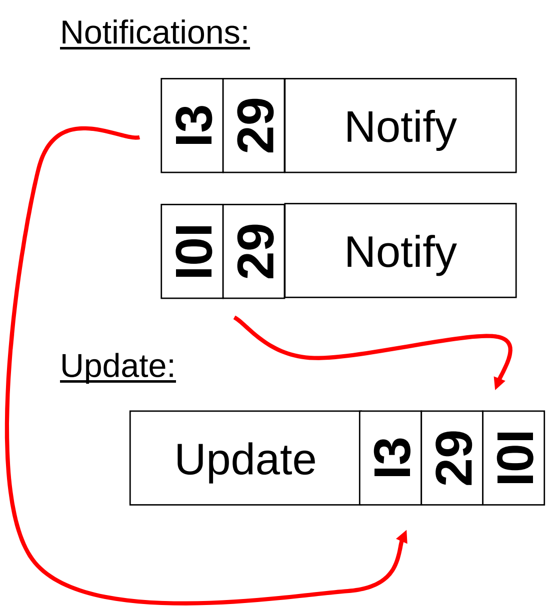
Viewed from node 101:



Iterative Successor Search(2)



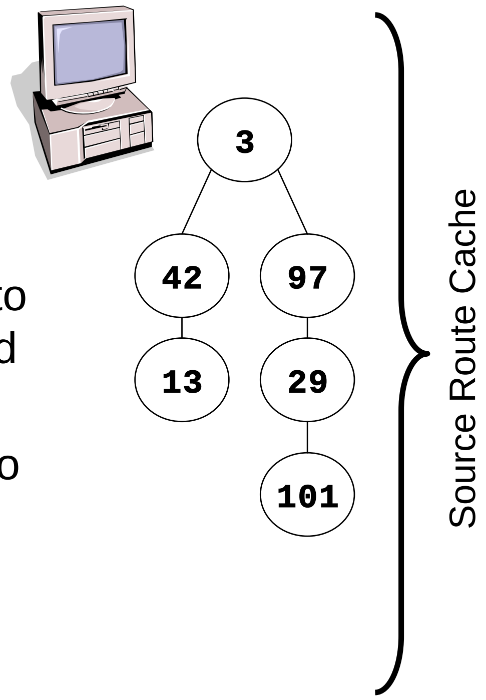
Node 29 can resolve the inconsistency



Source Route Cache (1)

Nodes use static memory:

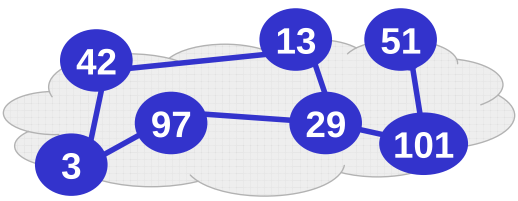
- Each node stores its direct physical neighbors.
- Each node stores a source route to its successor (cf. Chord).
- Each node stores a source route to its predecessor (to be able to send updates).
- All remaining memory (assigned to routing) is used to cache source routes in a LRU manner.



Fuhrmann, Scalable Routing in Random Networks, Networking 2005.

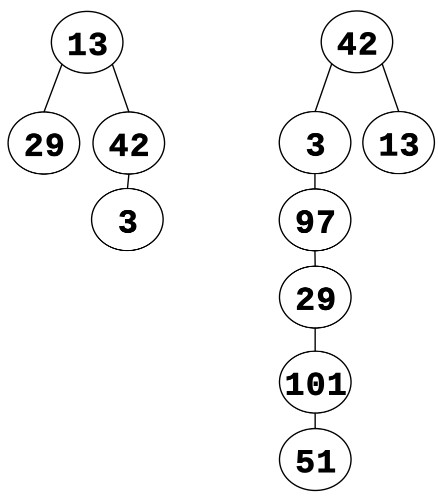
Source Route Cache (2)

- Upon a ‚cache miss' the message is forwarded to the node that
  - Lies before the target (in direction of the ring),
  - is physically closest to the forwarding node, and that
  - virtually closest to the target.



Example: 13 → 51

13	42	3	97	29	101	51

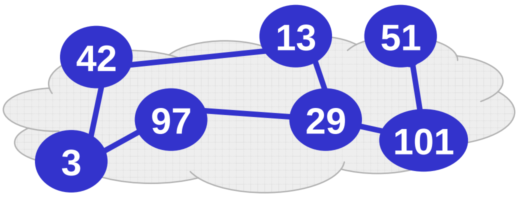


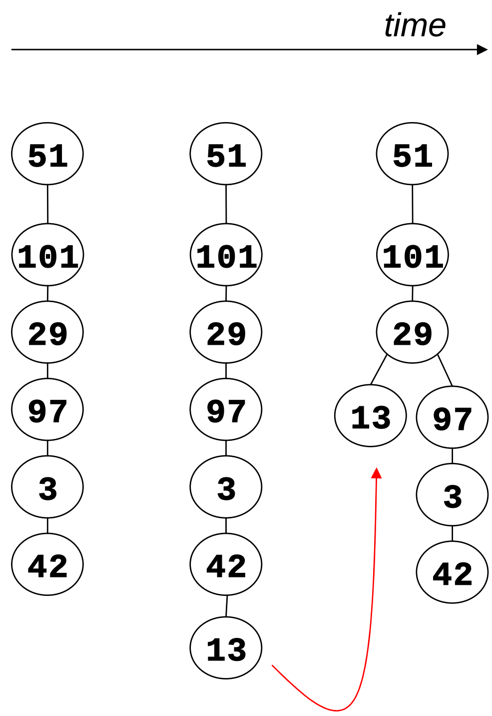
Source Route Cache (3)

The cache accumulates short routes:

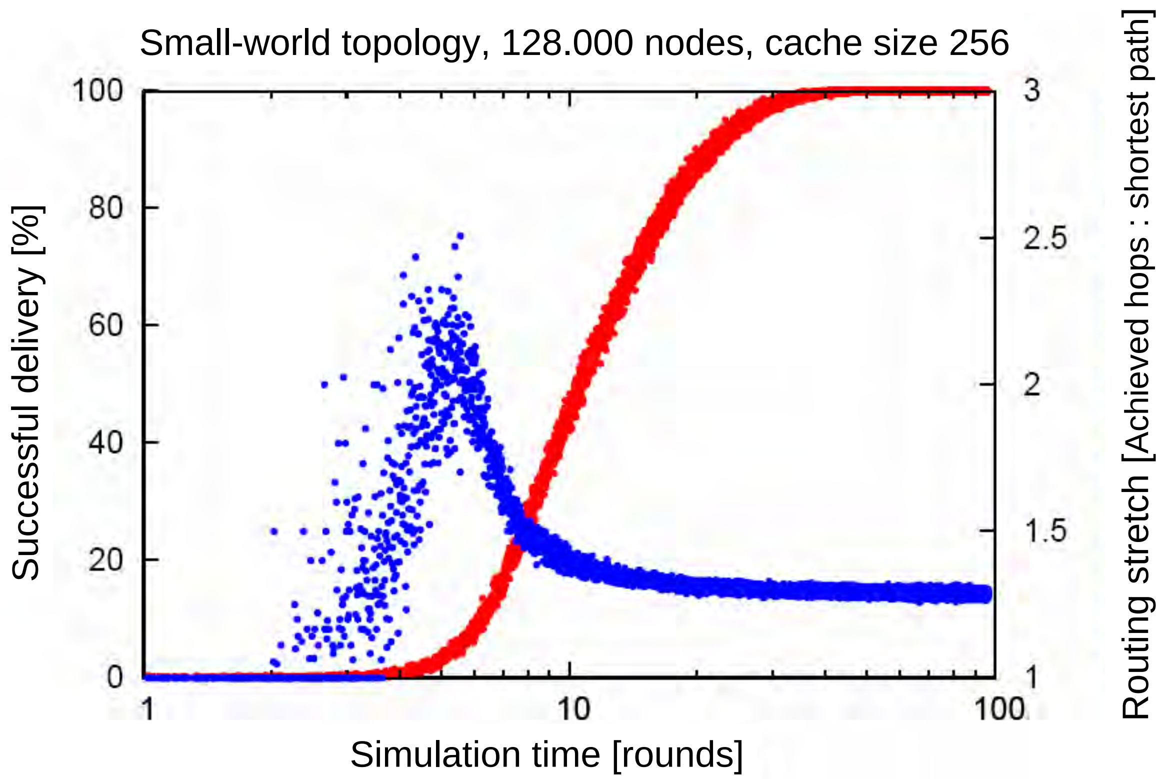
time

13	42	3	97	29	101	51
51	101	29	97	3	42	13
13	29	101	51			

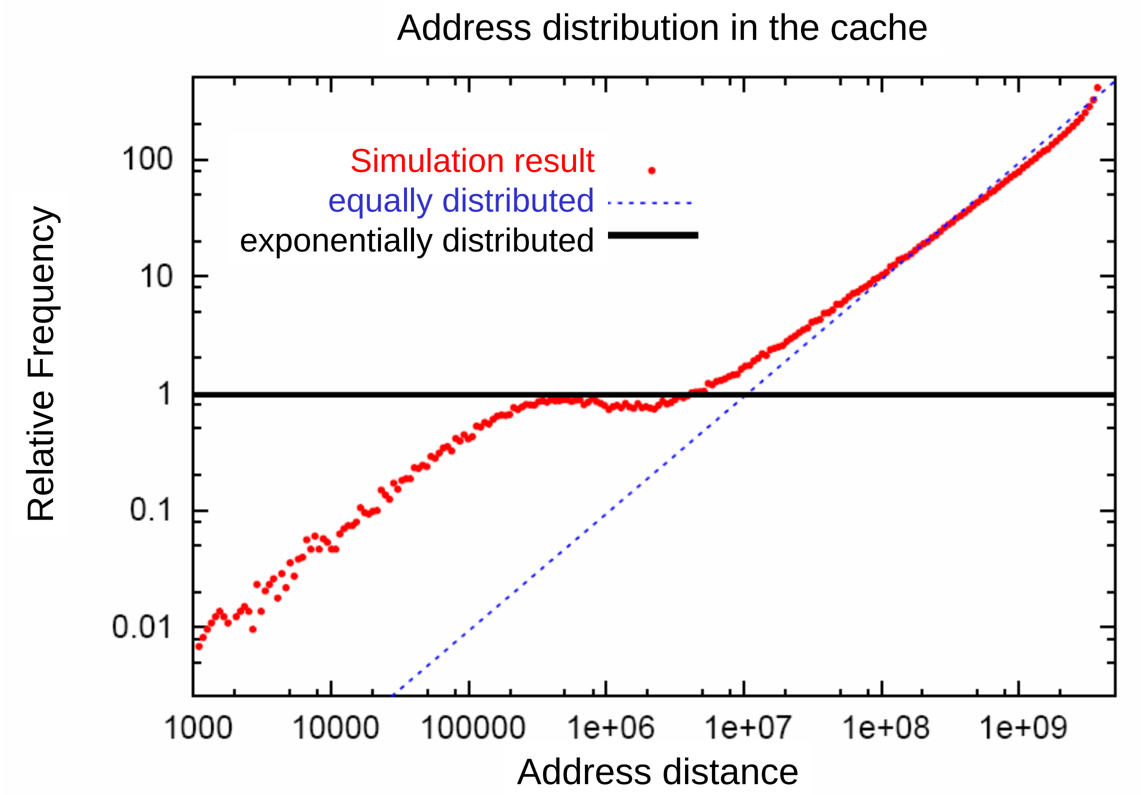




Simulation Results (1) – Consistency



Simulation Results (2) – Node Specialization



Agenda

1. Motivation
2. Structured Routing Overlays
3. Scalable Source Routing
4. Summary

Scalable Source Routing – Protocol Overview

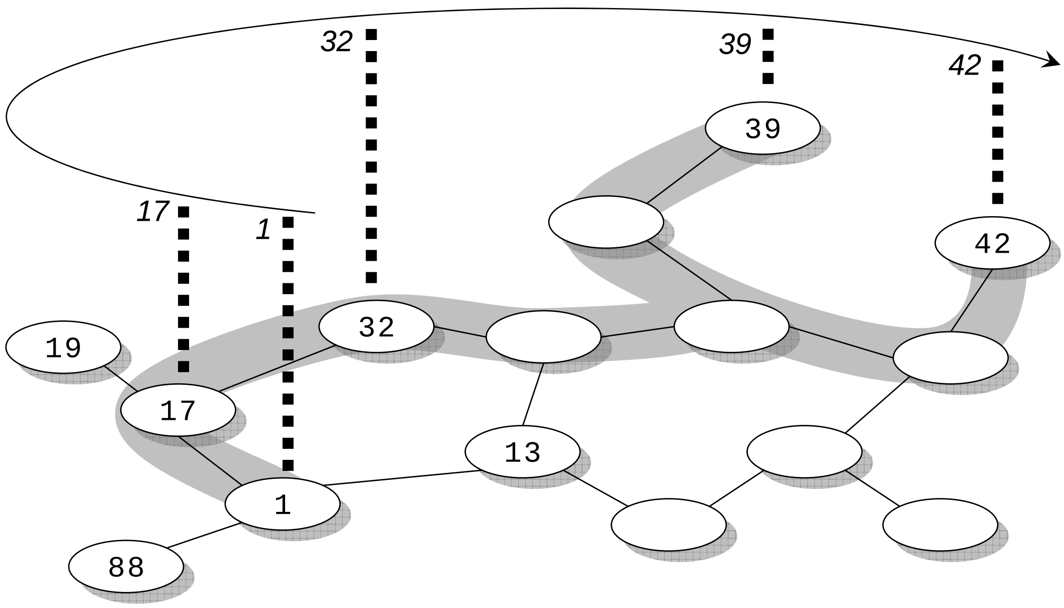
Seven simple rules lead to self-organized routing:

- Register with your successor in the virtual ring.
- Update your predecessor if necessary.
- Flood only if you think to have the globally greatest address.
- Prune paths when appending paths.
- Cache paths in a LRU manner.
- Keep shorter path variant in the cache.
- Upon cache miss, forward to the node that is physically closest and virtually farthest.

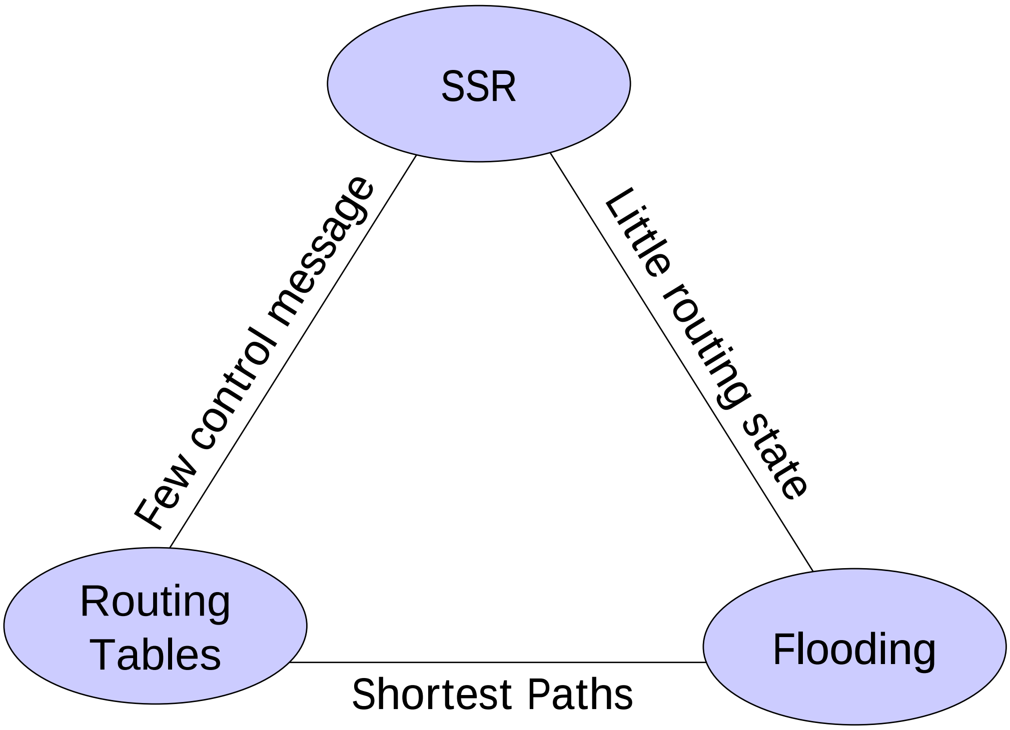
This leads to

- Much smaller memory requirement compared to Dijkstra and Bellman-Ford,
- Much less control messages compared to flooding,
- Works in arbitrary topologies (unlike geographical routing).

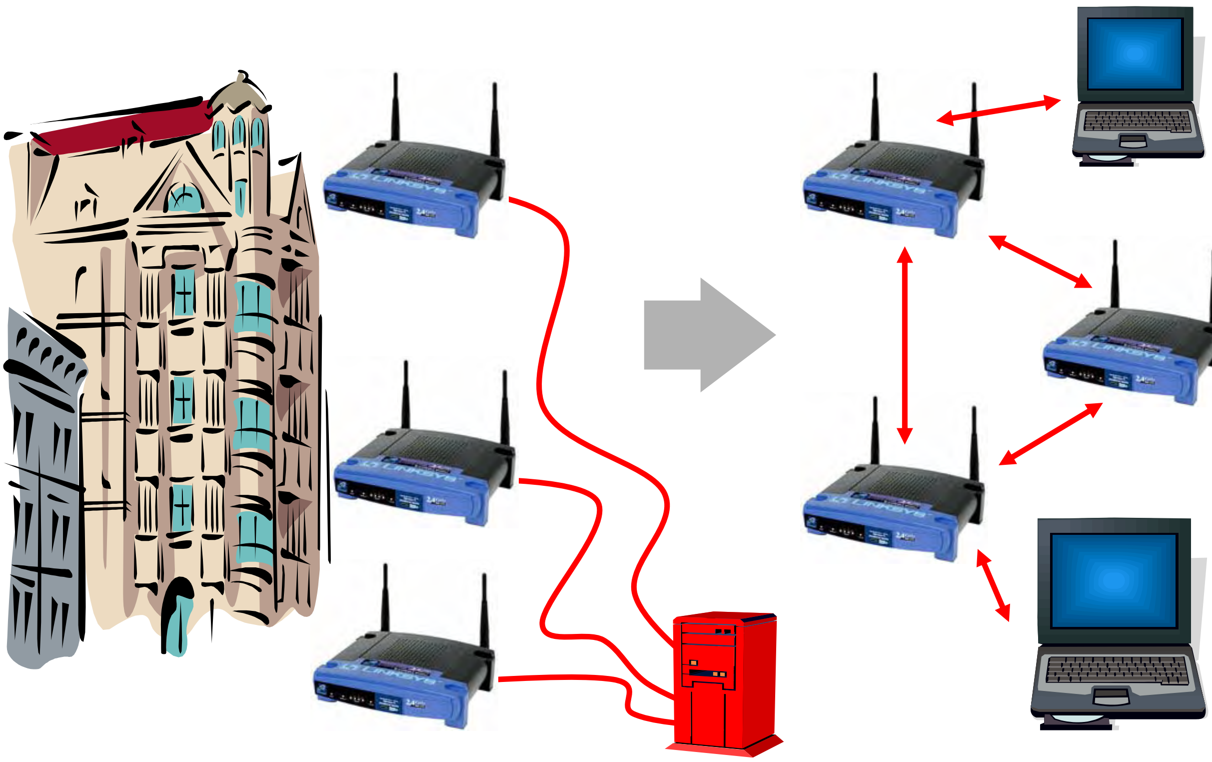
Scalable Source Routing – Protocol Overview



Scalable Source Routing breaks the Trade-Off



The Linyphi Mesh Network for IPv6



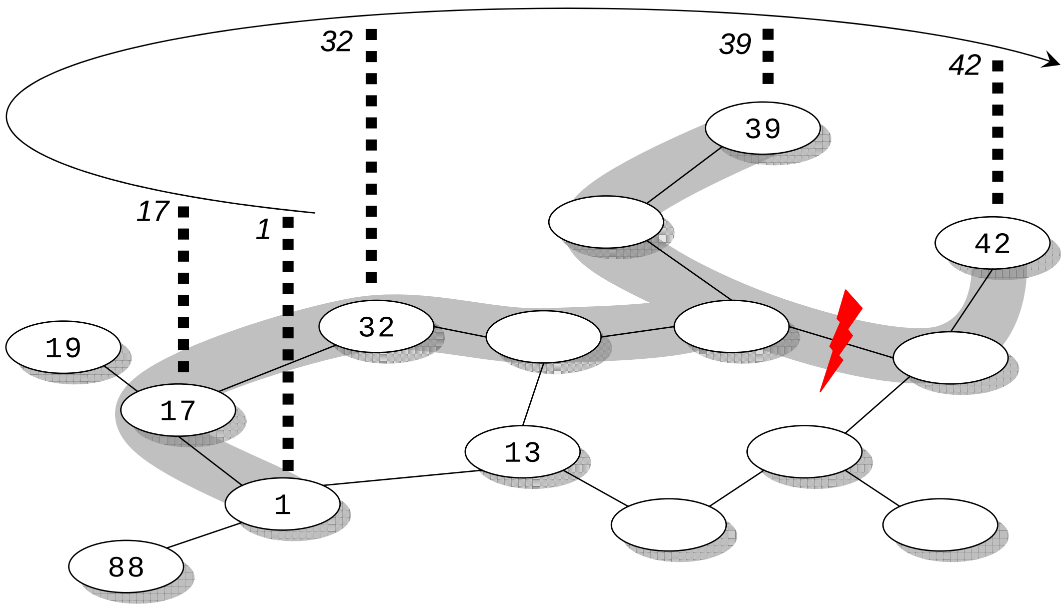
Thank you!

Questions?

Thomas Fuhrmann
IBDS System Architecture
University of Karlsruhe, Germany

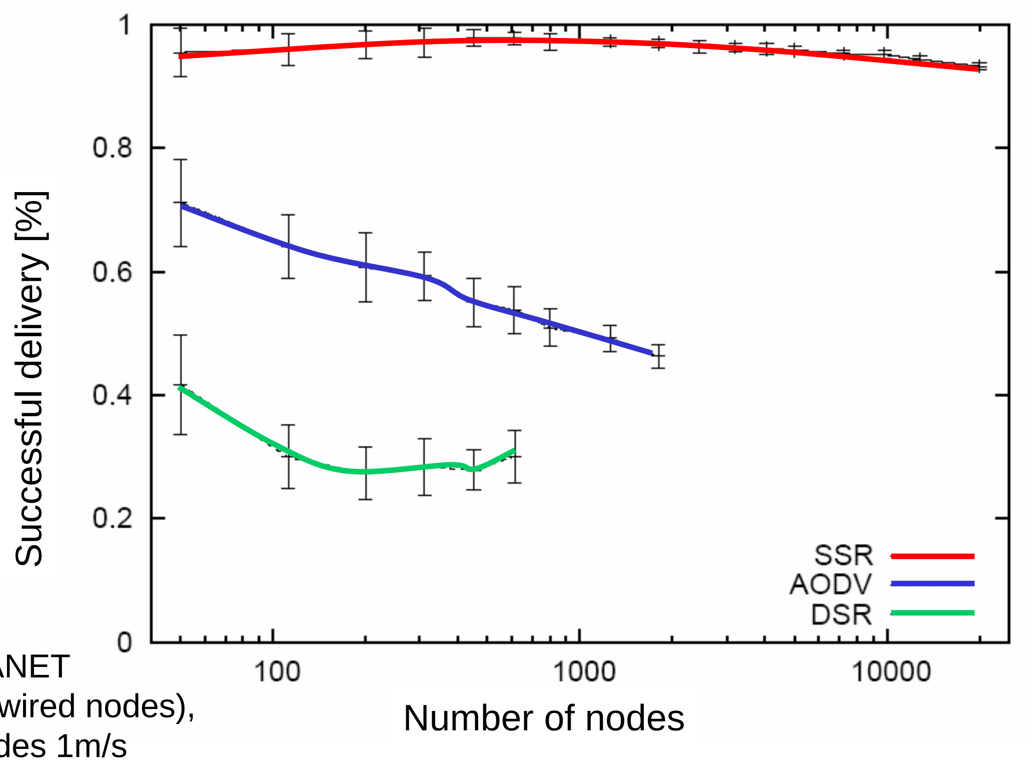
fuhrmann@ira.uka.de

Dealing with Churn and Mobility



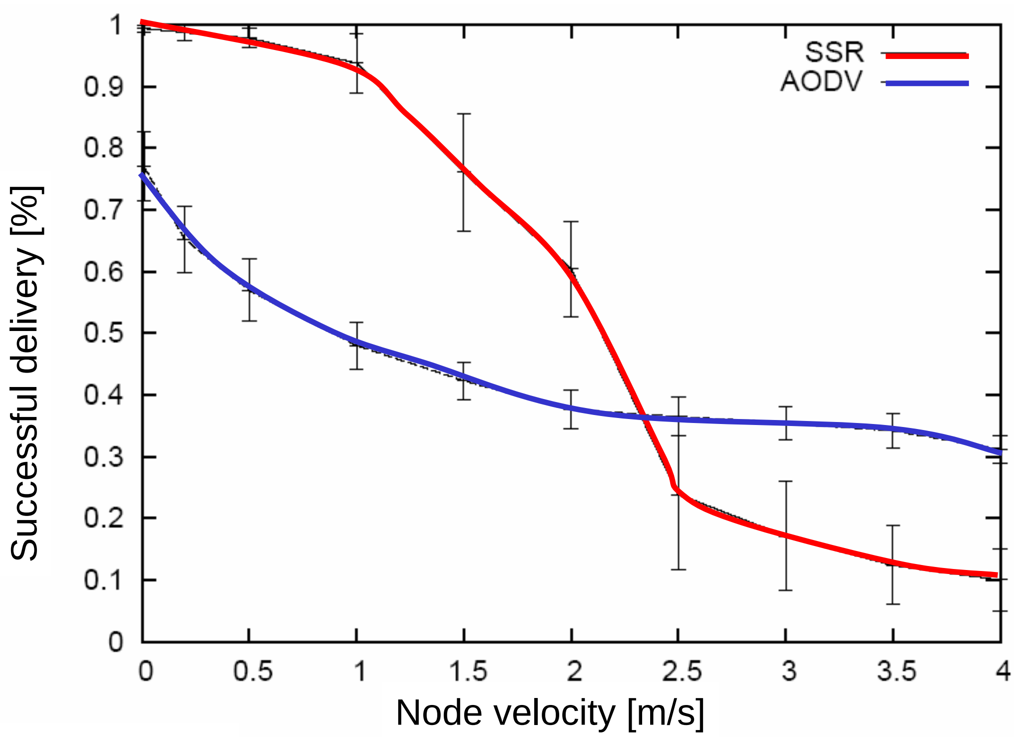
Thomas Fuhrmann, Pengfei Di, Kendy Kutzner, and Curt Cramer. Pushing Chord into the Underlay: Scalable Routing for Hybrid MANETs. Universität Karlsruhe, Fakultät für Informatik, Technical Report 2006-12, June 2006

Simulation (3) – Compare to AODV and DSR



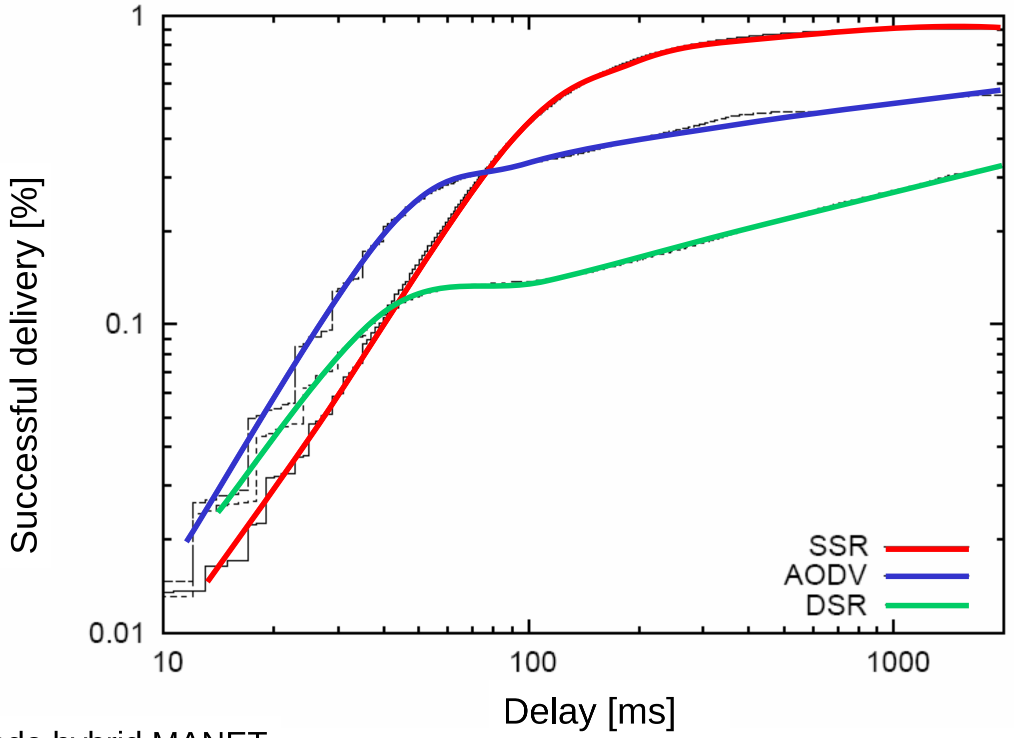
Hybrid MANET
(5% fixed wired nodes),
Mobile nodes 1m/s

Simulation Results (4) – Node velocity



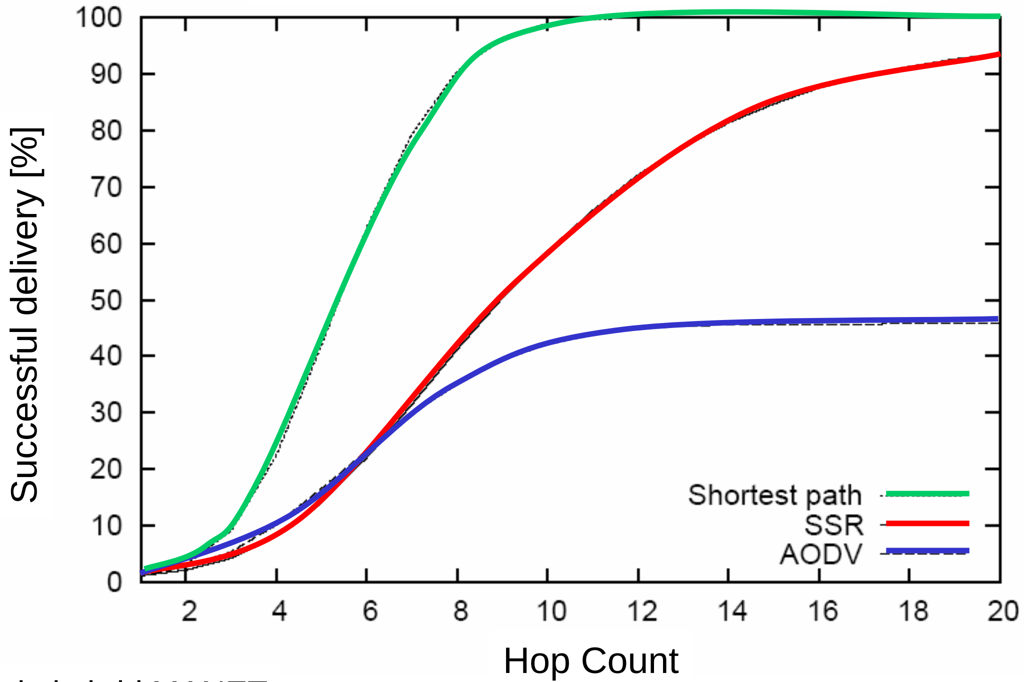
450 node pure MANET

Simulation Results (5) – Delay distribution



612 node hybrid MANET

Simulation Results (6) – Hop distribution



800 node hybrid MANET