Lecture 5

Internet Protokolle II

Routing in
Wireless Ad-hoc Networks

Part II

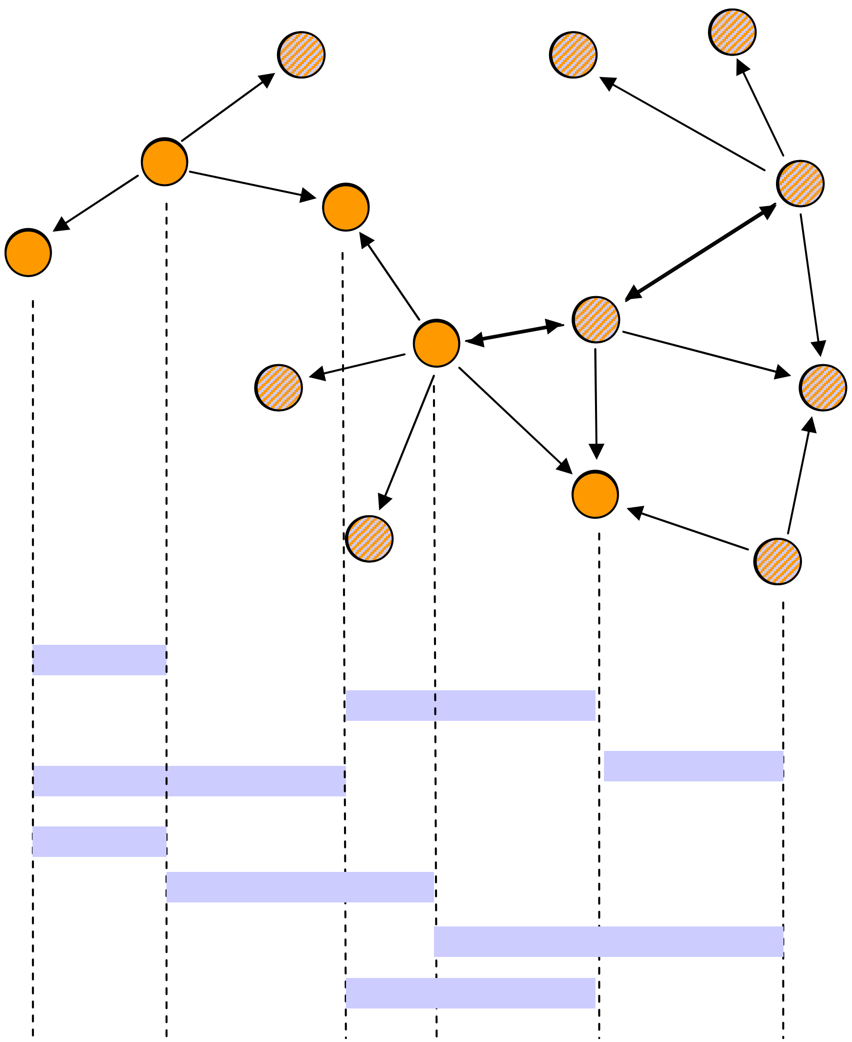
Thomas Fuhrmann

Network Architectures
Computer Science Department
Technical University Munich

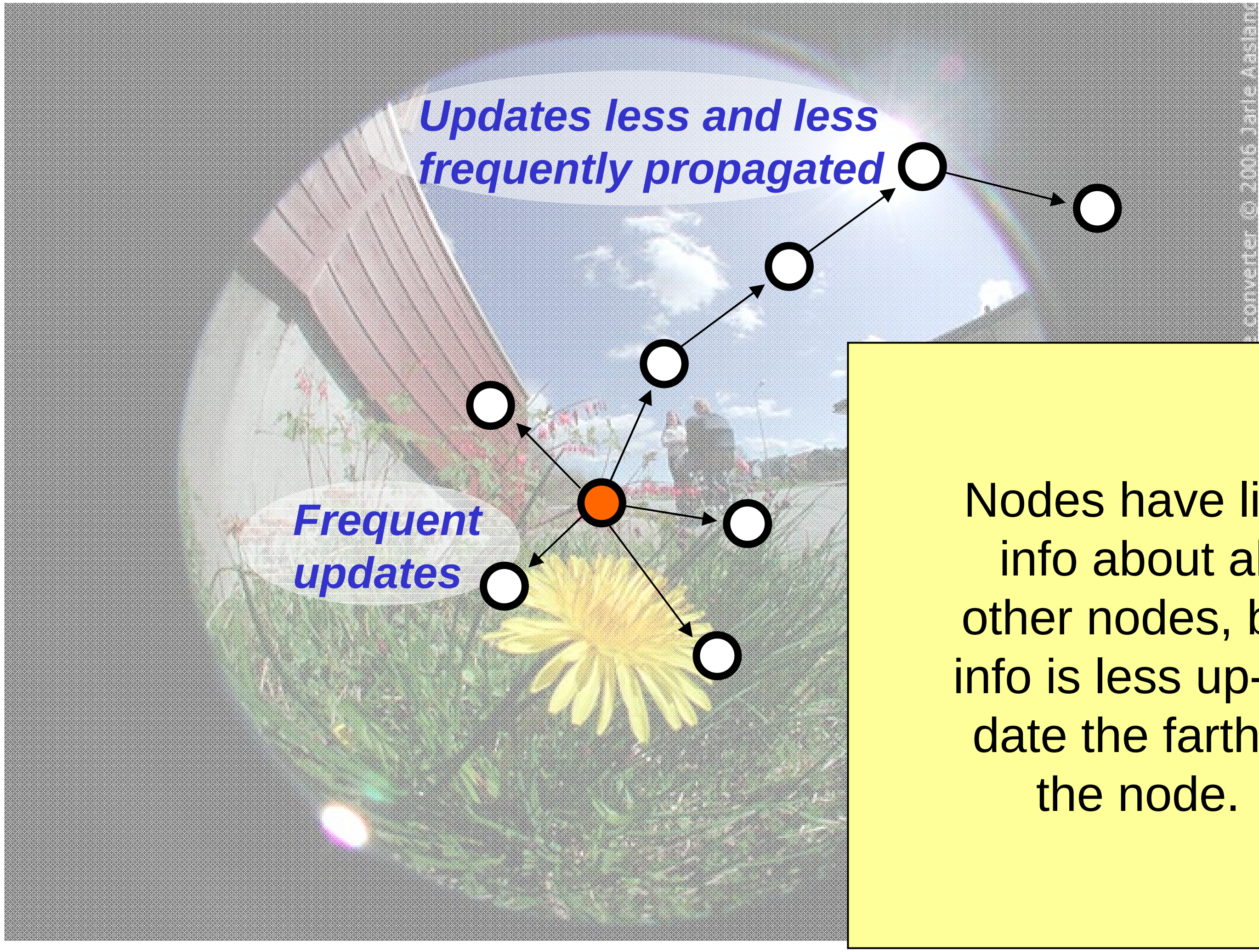
Routing in Mobile Ad-Hoc Networks

- Extension of fixed wire protocols
  - Avoid ‚count to infinity' by exchanging richer state info
  - Damp updates to limit traffic
- Flooding to acquire state
  - Do not proactively maintain state
  - Search reactively for the requested destinations
- Limited flooding – Scopes and landmarks
  - Different areas have more or less detailed routing information
- Geographic routing
  - Exploit the fact that nodes have coordinates
- Virtual structures
  - Establish virtual ‚coordinates'
  - Distribute routing info according to virtual coordinates

Global State Routing



Fisheye State Routing

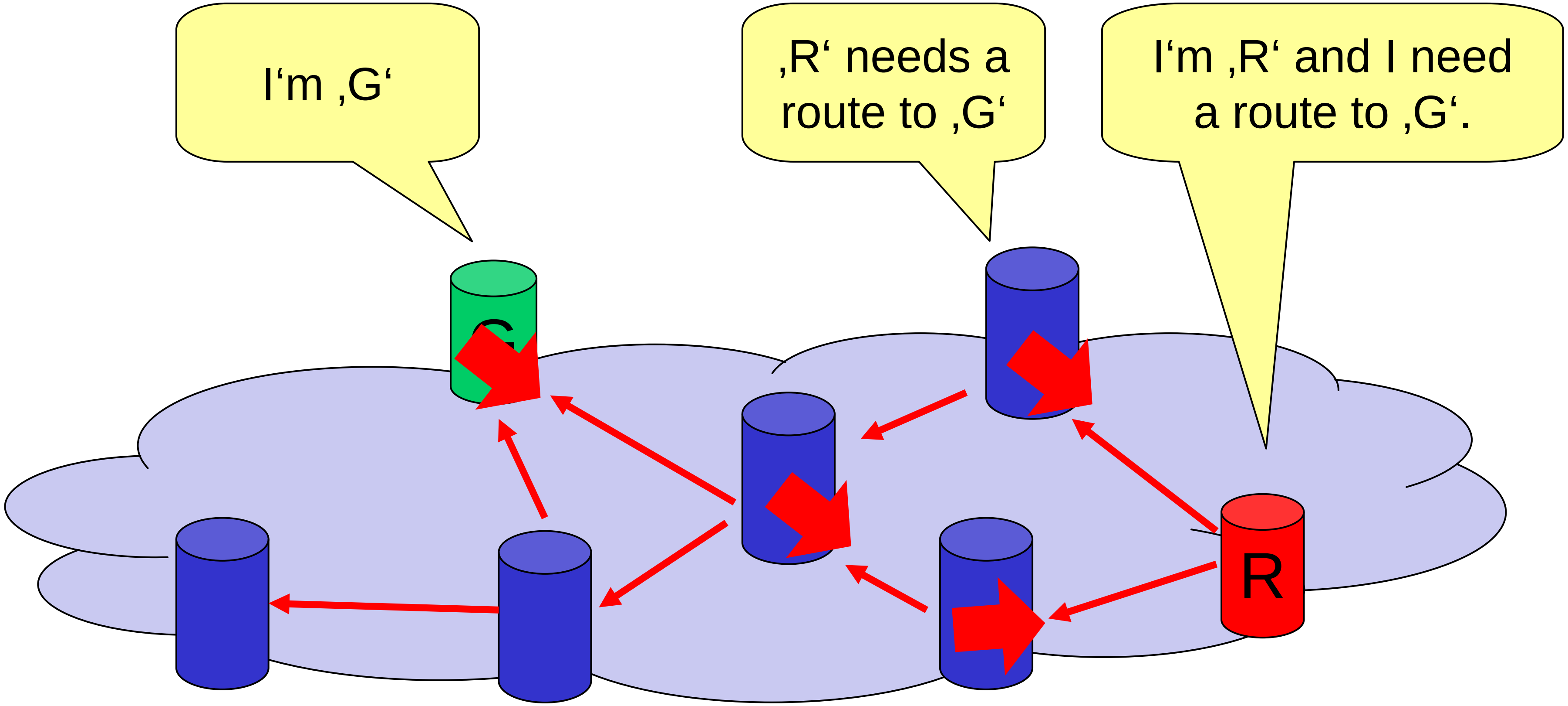


Nodes have link info about all other nodes, but info is less up-to-date the farther the node.

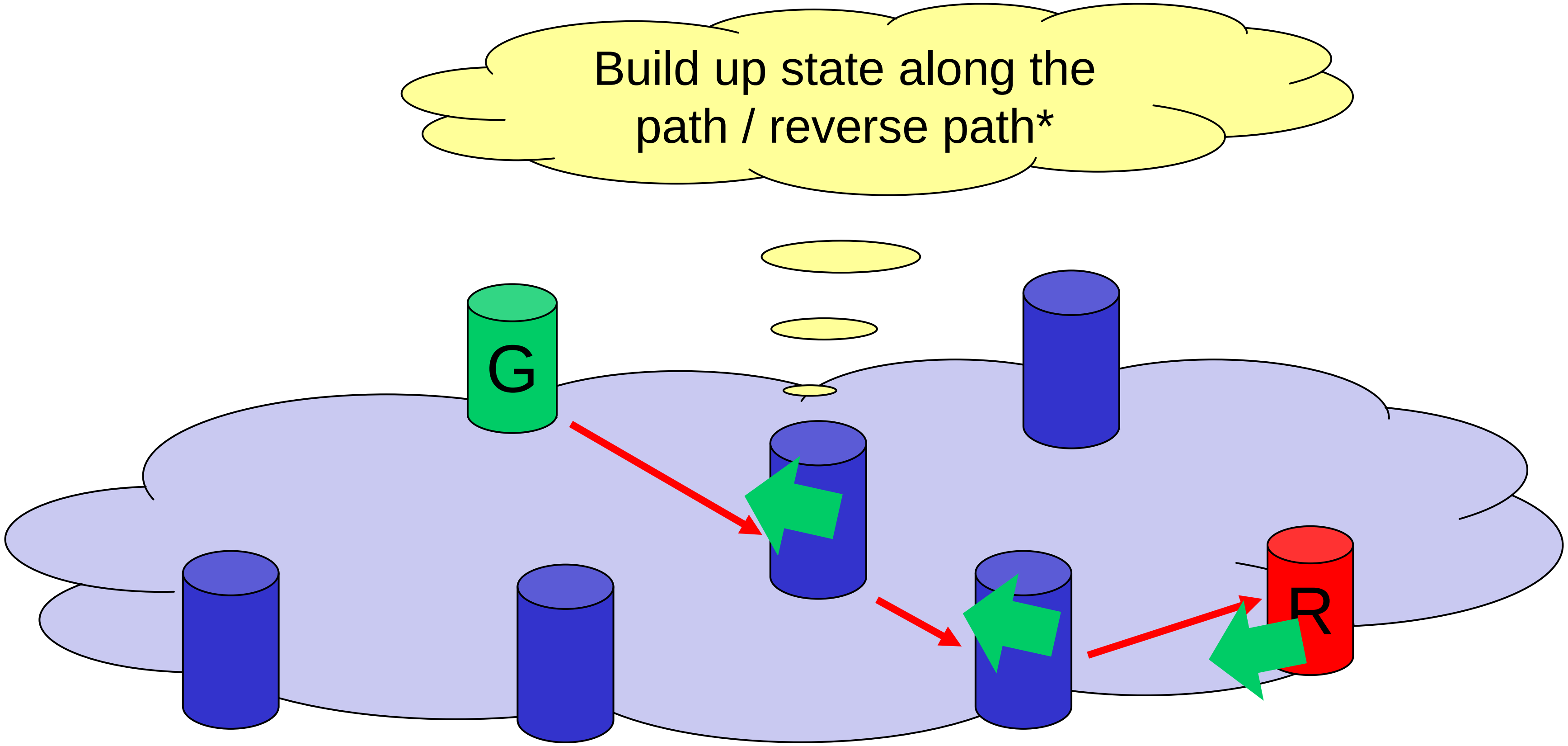
Global State & Fisheye State Routing

- Normally, link state routing would flood the network as soon as a link change was detected.
  - Topology changes are announced asap
  - Short fluctuations cause much traffic
- Global state routing exchanges info with direct neighbors, only.
  - Multiple topology changes accumulated and sent within one message
  - Damping of short fluctuations
  - Sub-optimal routing because not all nodes have up-to-date information
  - BTW, needs sequence numbers to avoid loops
- Fisheye state routing
  - Different intervals for sending updates (more often for closer nodes)
  - Same routing performance as GSR, but less update traffic (the farther the destination, the less accurate information needed)

Ad-hoc On Demand Vector Routing

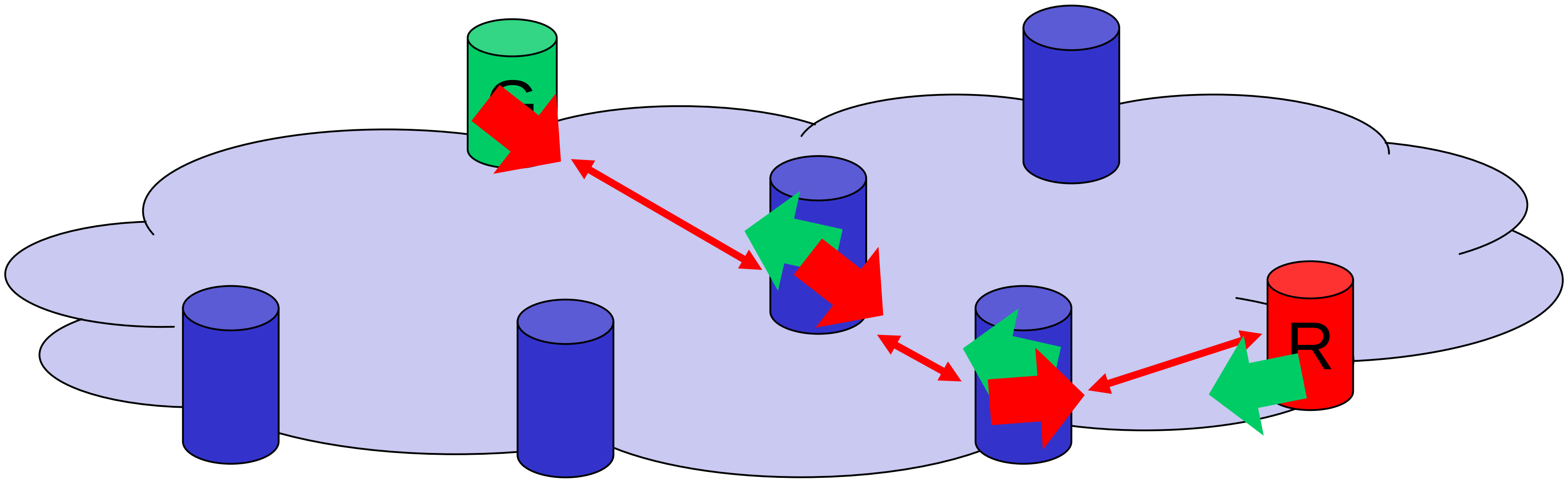


Ad-hoc On Demand Vector Routing



* Reverse path is defined by the nodes that were first to sent the request for the respective destination.

Ad-hoc On Demand Vector Routing



AODV – Summary

Basic Operation	Properties

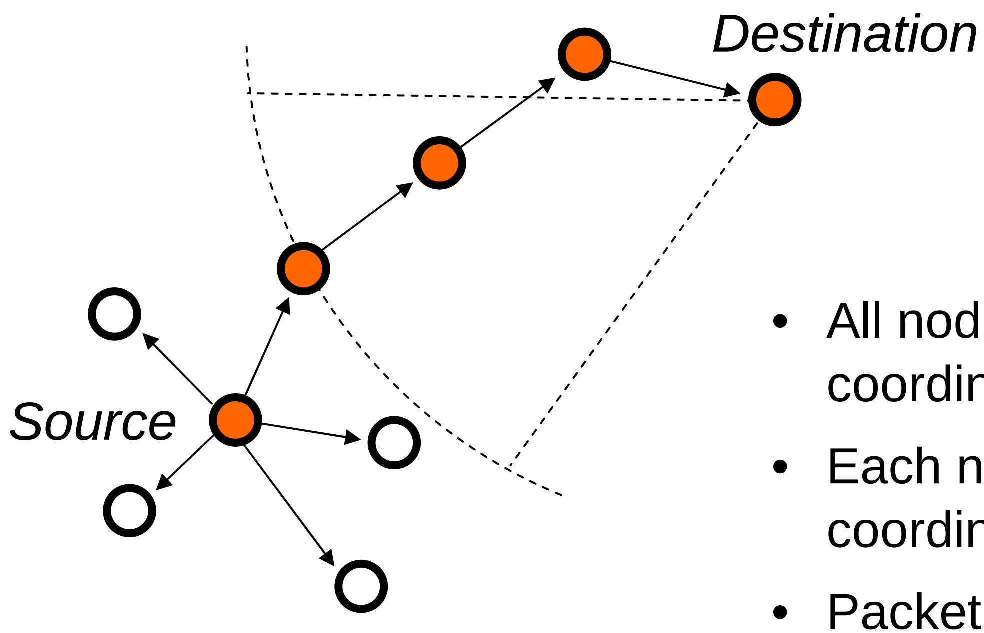
Basic Operation

- Receive route request?
  - Cache destination & sender of request
  - If already cache, do nothing.
  - Otherwise, if not destination, re-broadcast the request.
  - Otherwise, send route reply to sender of request.
- Receive route reply?
  - Cache destination & sender of reply
  - Forward reply to cached sender.
- Use timeout to tear down the path.

Properties

- Reactive routing
- Low processing demand,
- But still high bandwidth demand
  - if many nodes active, i.e. sending, receiving, or both
  - nodes are mobile
- Potentially high memory demand (e.g. if timeout wrongly chosen)

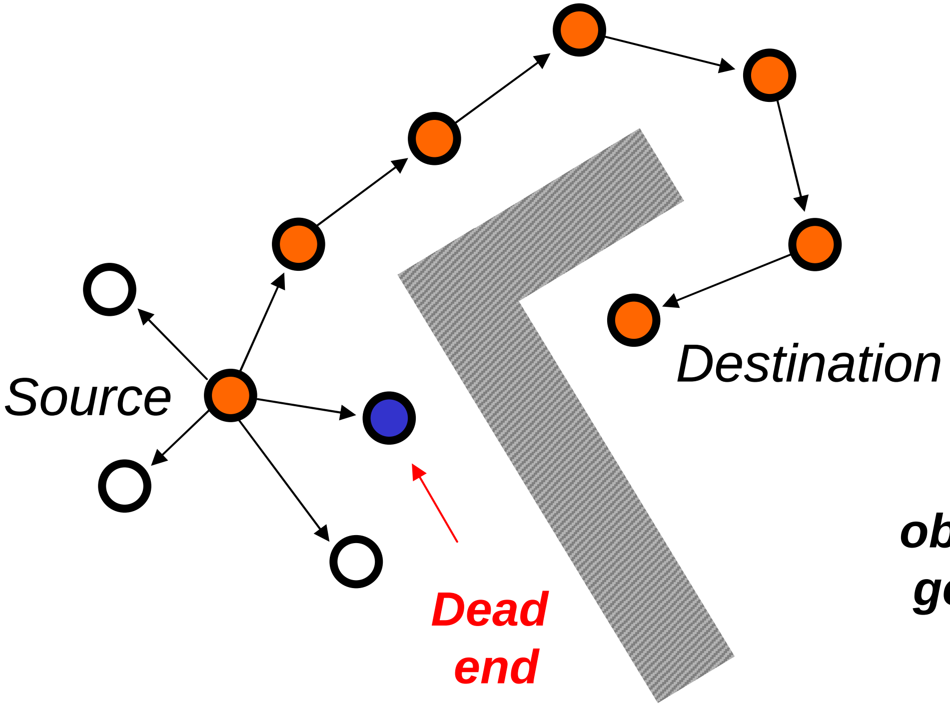
Geographic Routing



- All nodes know their geographic coordinates.
- Each node checks its neighbors‘ coordinates.
- Packet is forwarded to node closest to destination.

Geographic routing can achieve near-to-optimal routes without (large) routing tables and without (signfigicant) control traffic.

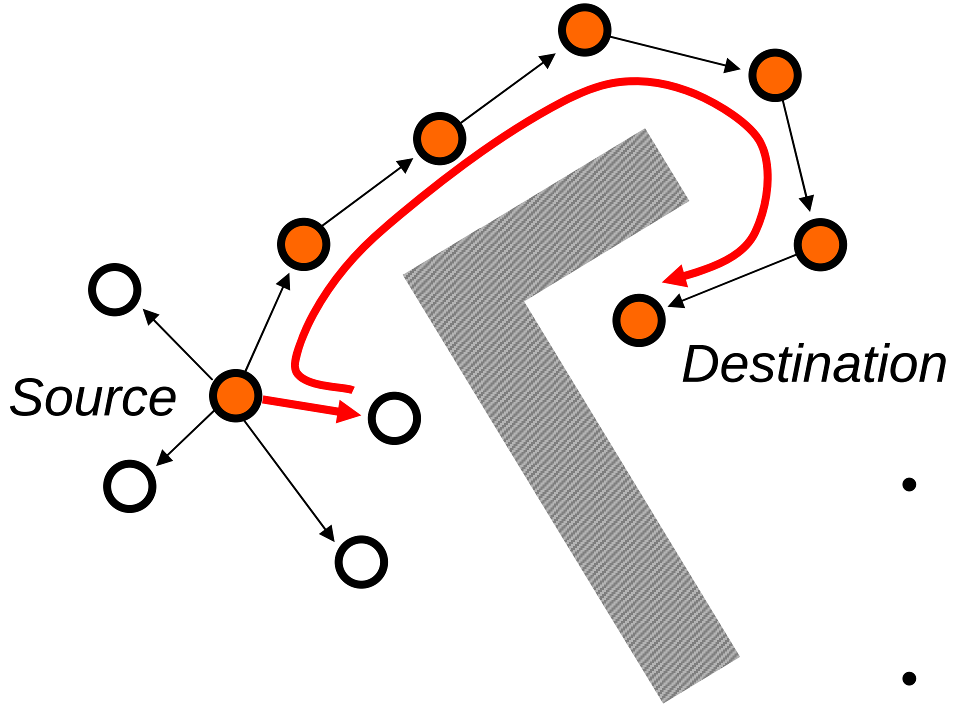
Problems with Geographic Routing



There is a path around the obstacle (red nodes), but greedy geographic routing leads into a dead end (blue node).

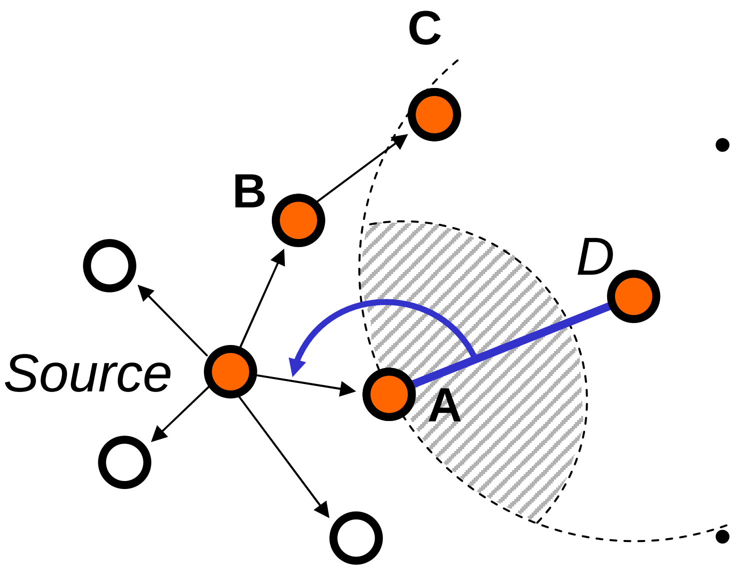
Geographic routing can lead into dead ends, especially in presence of obstacles.

Greedy Perimeter Stateless Routing



- Use geographic routing (=greedy mode) until you get stuck in dead end.
- Enter your node ID into the packet, ‚turn around‘ and switch to perimeter mode.
- Forward to the right-most node until ‘obstacle bypassed’.

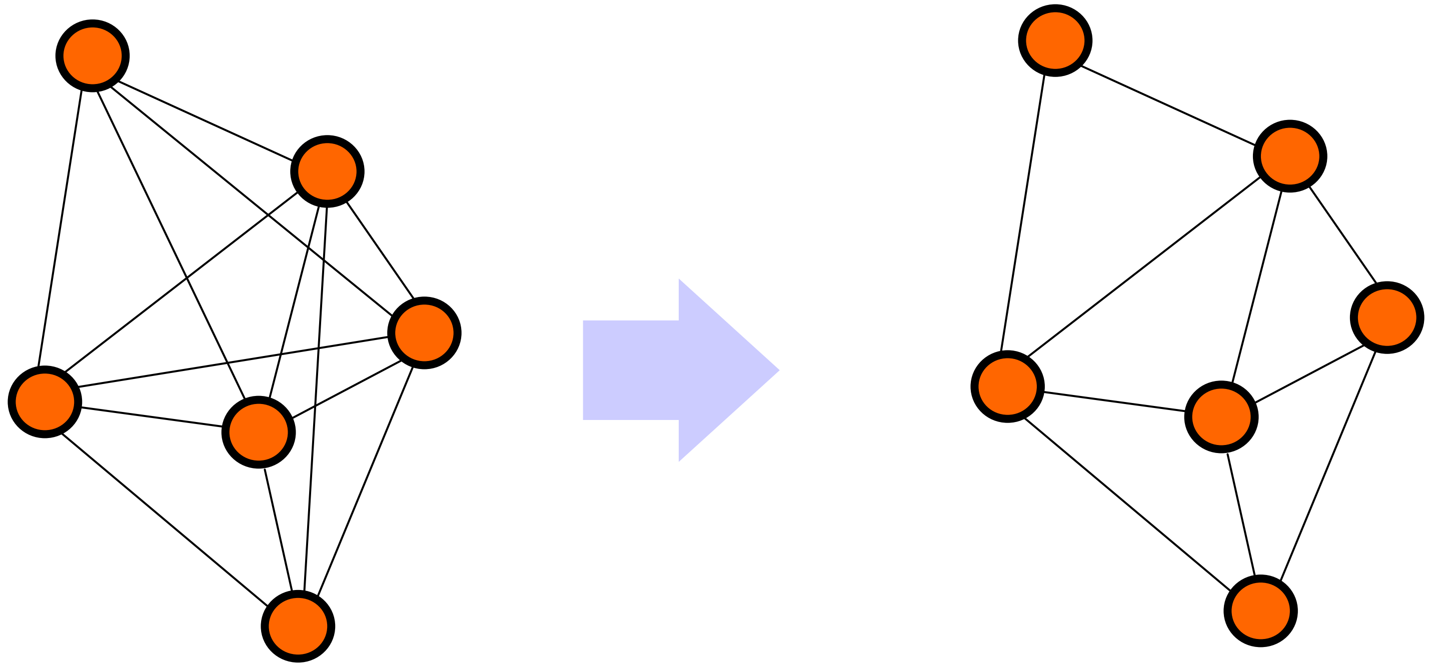
GPSR: Greedy and Perimeter Mode



- Source has several neighbor, for example A and B. It prefers A because A is closer to D than B.
- The intersection of
  - the region reachable from A, and
  - the region that is closer to D than A

  is empty. Thus switch to perimeter mode.
- Going back from A to S takes the 'right-most' edge.
- Node C is closer to D than A (where we switched to perimeter mode). Thus switch back to greedy mode.

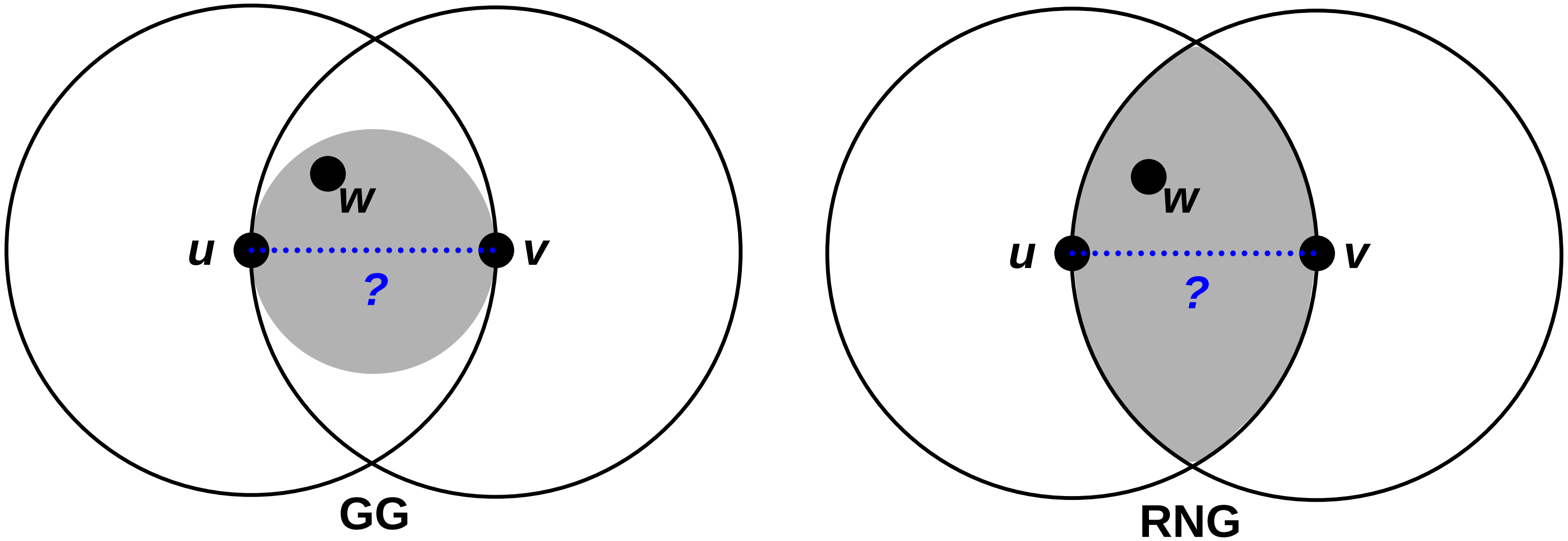
GPSR Needs Planar Graphs



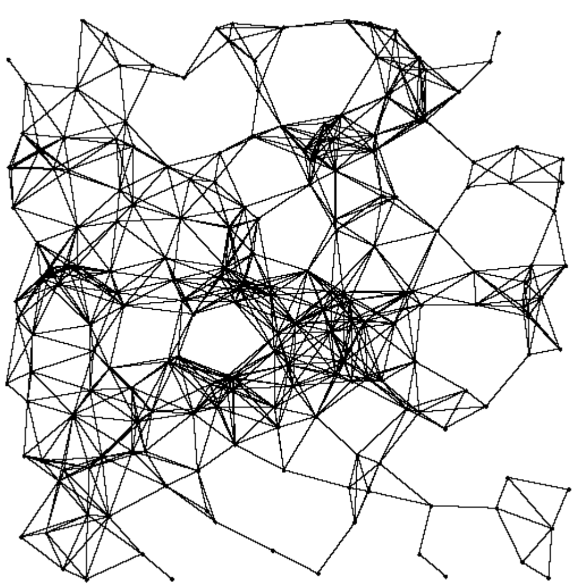
- Right-hand rule is only guaranteed to tour the entire perimeter if graph is planar.
- Thus we need to planarize the graph for GPSR!

Graph Planarization

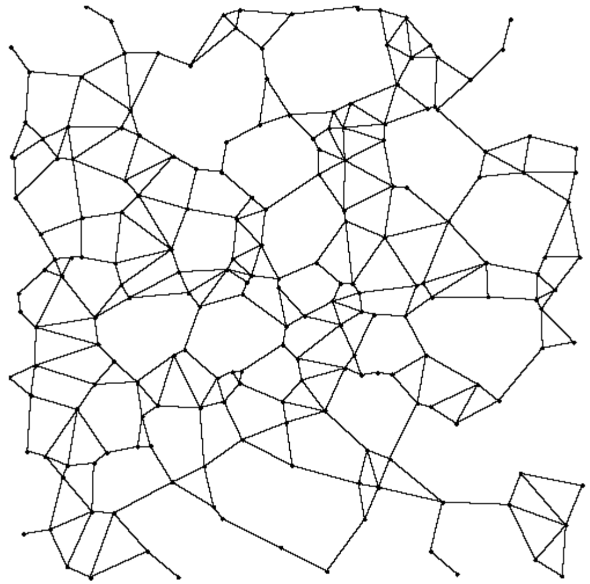
- The Gabriel graph is a subgraph of the Delaunay triangulation. It is obtained by ensuring that two points P and Q are connected iff the circle that has PQ as its diameter contains no other nodes.
- Relative neighborhood graph: … P and Q connected iff no other node in both the circles around P and Q.



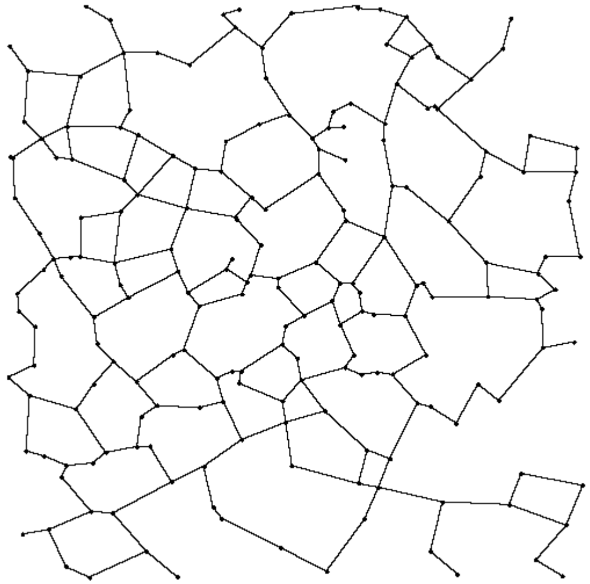
Planarized Graphs – Examples



Full Graph



GG Subgraph

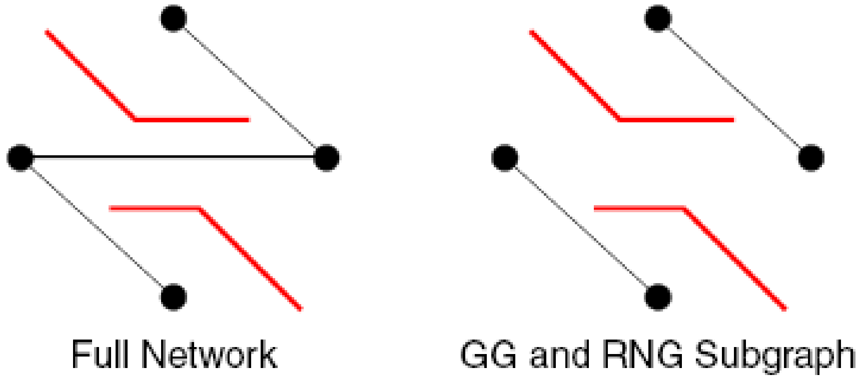


RNG Subgraph

200 nodes, placed uniformly at random on 2000-by-2000-meter region; 250-meter radio range.

Source: Brad Karp, UCL Computer Science

Planarization and Obstacles

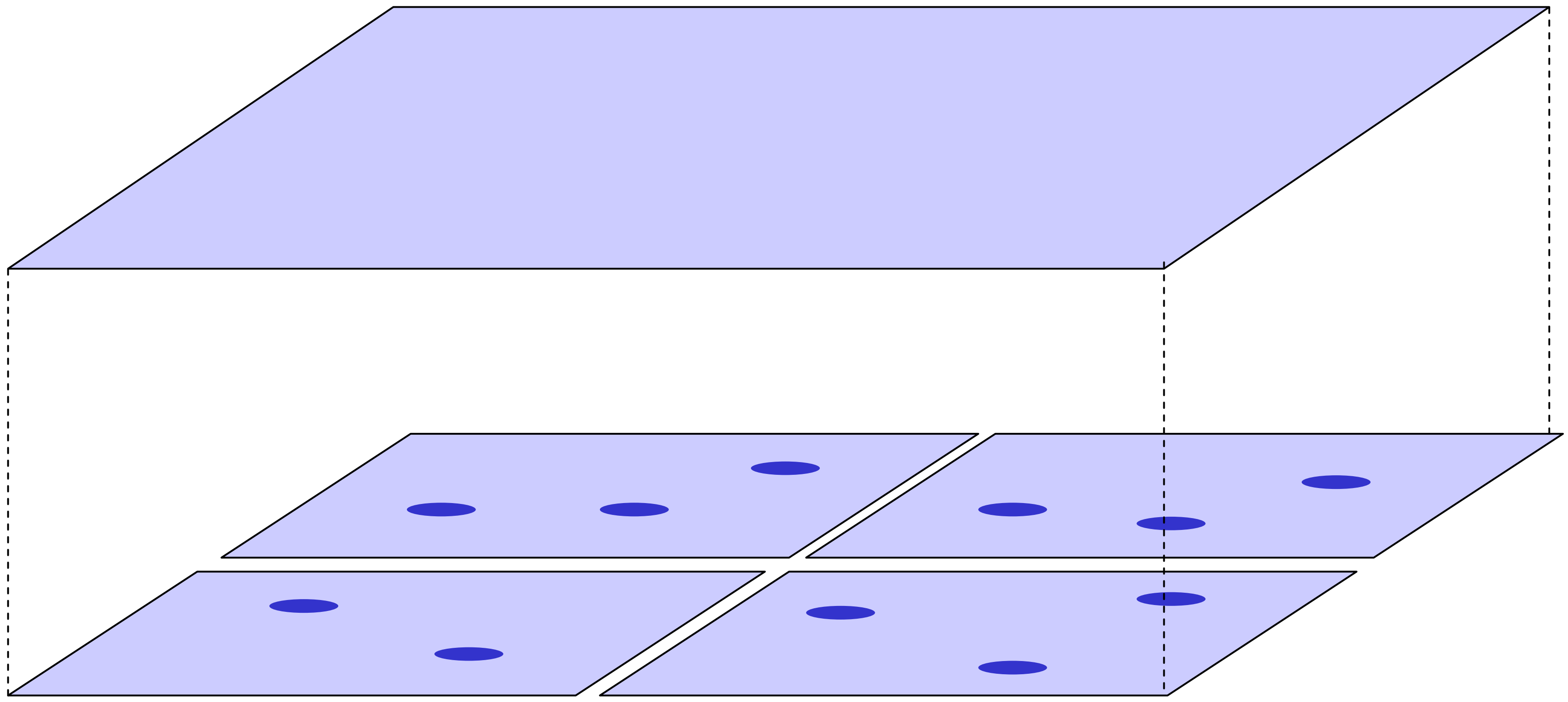


- Obstacles violate assumtions of GG and RNG planarization.
- This can lead to disconnected graphs.
- Solution: Mutual agreement of nodes before an edge is removed.
- But: This does not guarantee planarization any more.

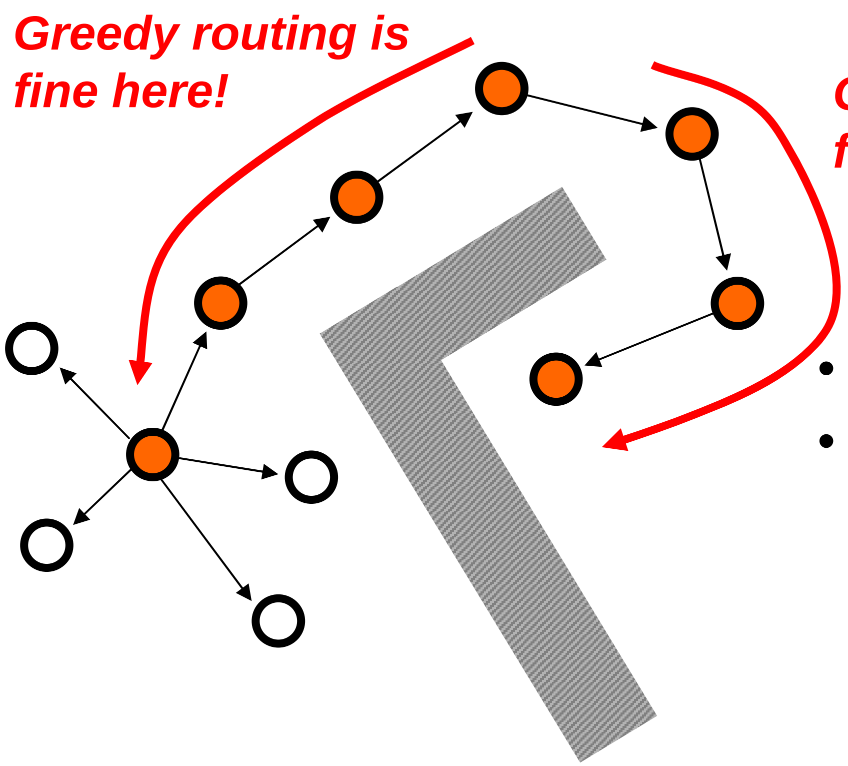
GPSR Summary

- GPSR requires all nodes to know their physical coordinates.
  - GPS or triangulation with RSSI, …
  - Location directory to look up destination coordinates
- The resulting graph must be planar
  - Planarization algorithm
  - Not guaranteed in presence of obstacles
- Then, switching between greedy and perimeter mode can reach destinations behind obstacles.

Grid Location Service



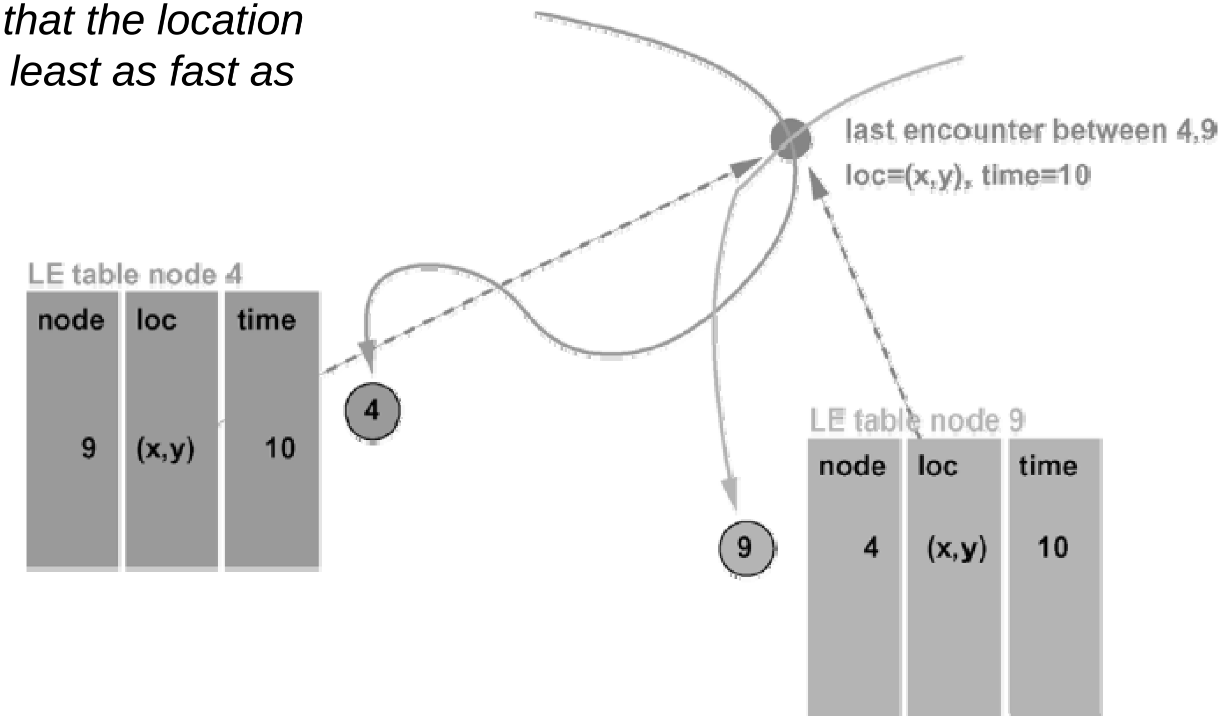
Ant Routing



- Use geographic routing, and
- Record direction to source while routing
  - If entry already present, enhance it by raising its weight
  - Entries' weights decrease over time unless they are enhanced
- Use routing table entries if their weight is sufficiently high

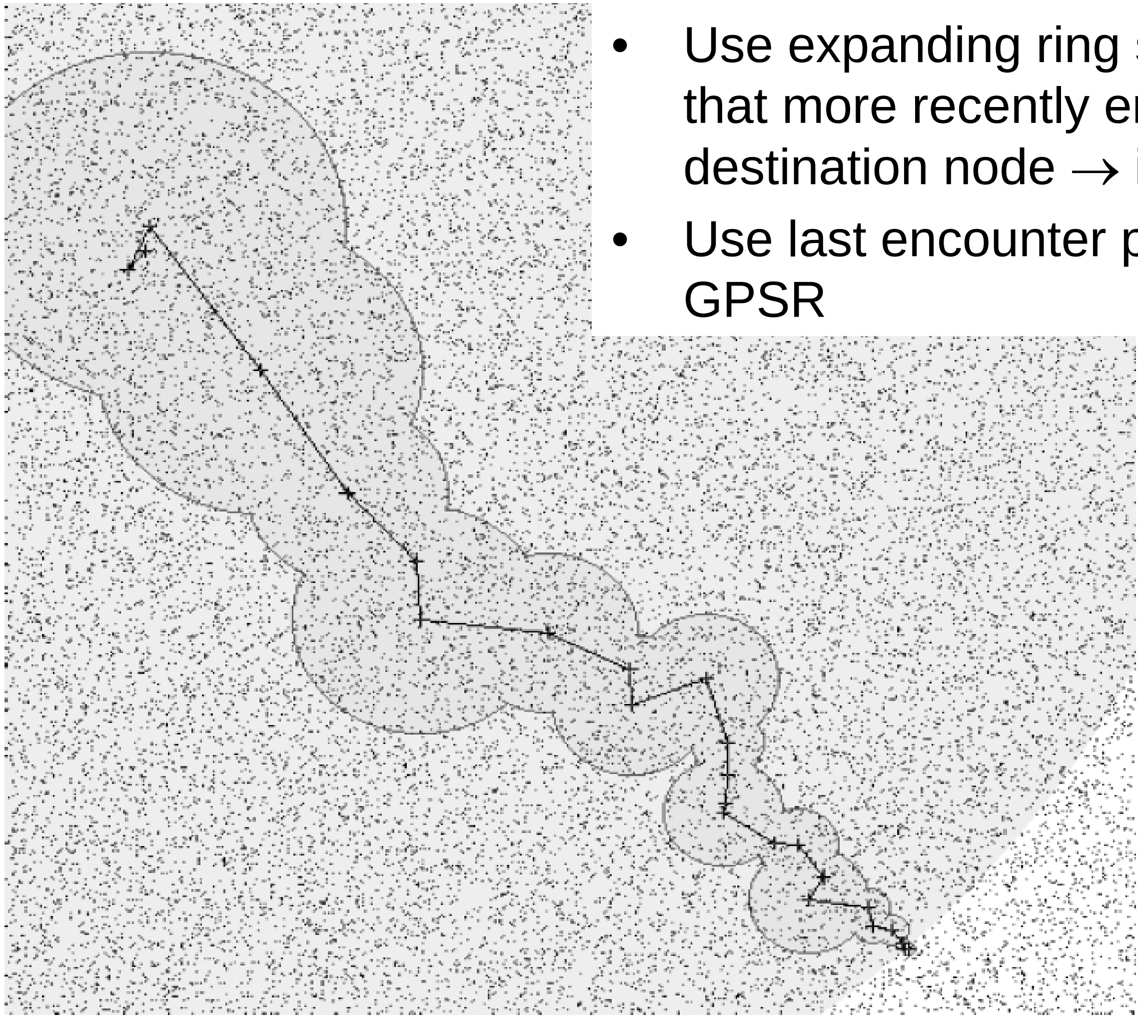
Last Encounter Routing (1)

Intuitively, mobility diffusion exploits three salient features of the node mobility processes: locality, mixing, and homogeneity… Locality ensures that aged information is still useful… Mixing of node trajectories ensures that position information diffuses around this destination node… Homogeneity ensures that the location information spreads at least as fast as the destination moves.



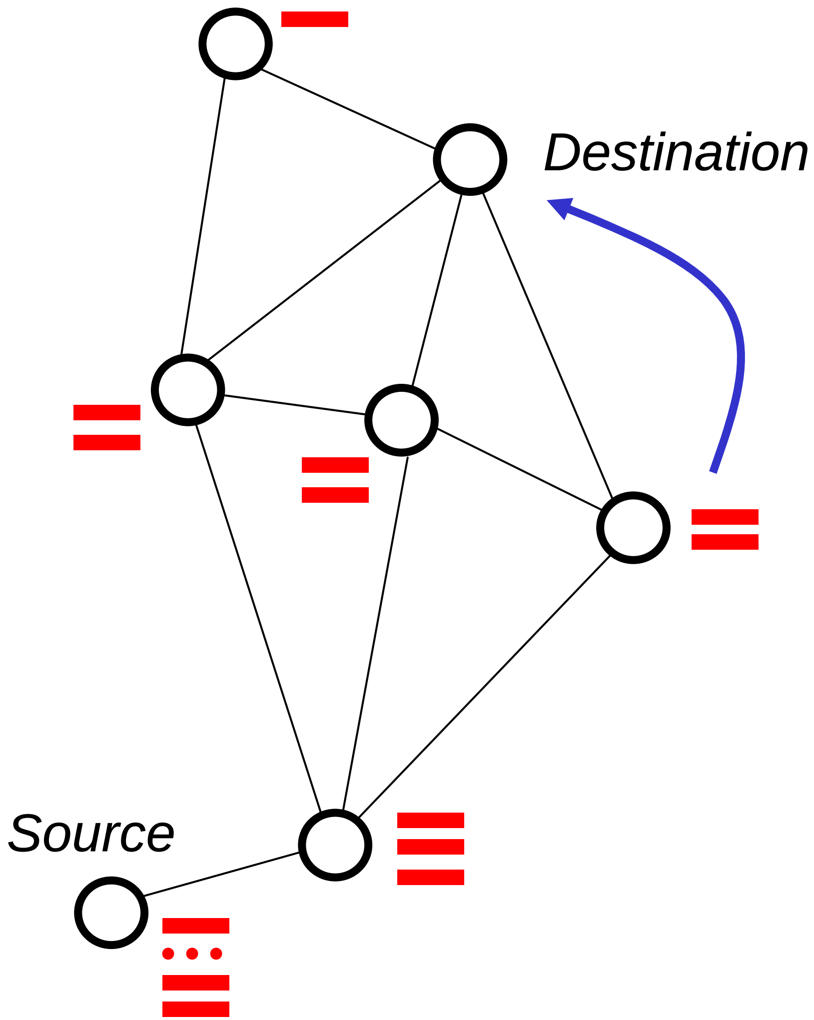
M. Grossglauser and M. Vetterli. Locating Mobile Nodes with EASE : Learning Efficient Routes from Encounter Histories Alone. IEEE/ACM Transactions on Networking, June 2006, 14(3), p. 457- 469

Last Encounter Routing (2)



- Use expanding ring search to find a node that more recently encountered the destination node → improves AODV, DSR
- Use last encounter positions to employ GPSR

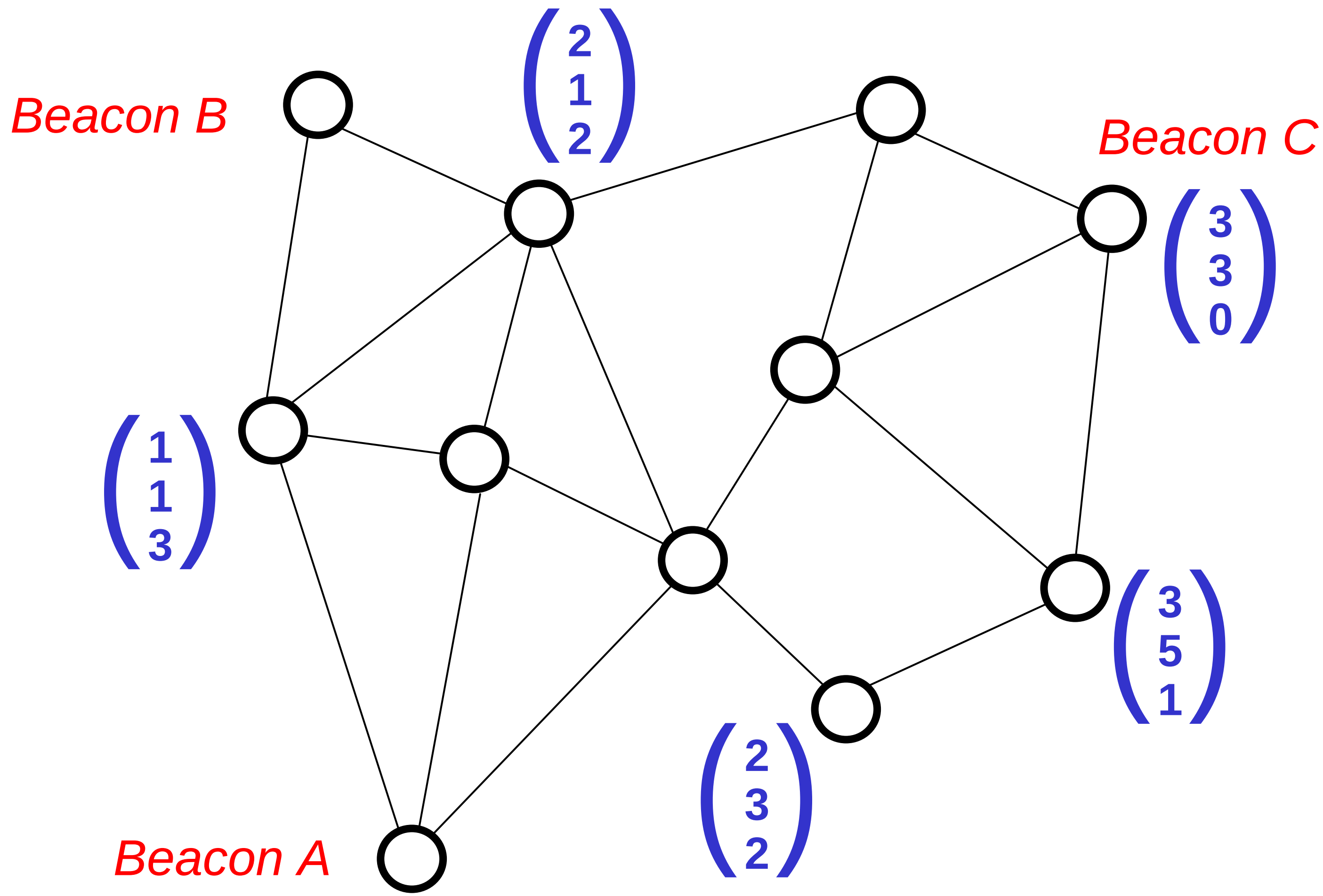
Volcano Routing



- Source generates „infinite“ packet stream
- Destination consumes all packets
- Each node routes packet to that neighbor that stores least packets of that stream

- Very robust (multipath)
- Packet loss (due to packets getting stuck) is normal
- Nodes need to store many packets, on the order of hop distance

Beacon Vector Routing (1)

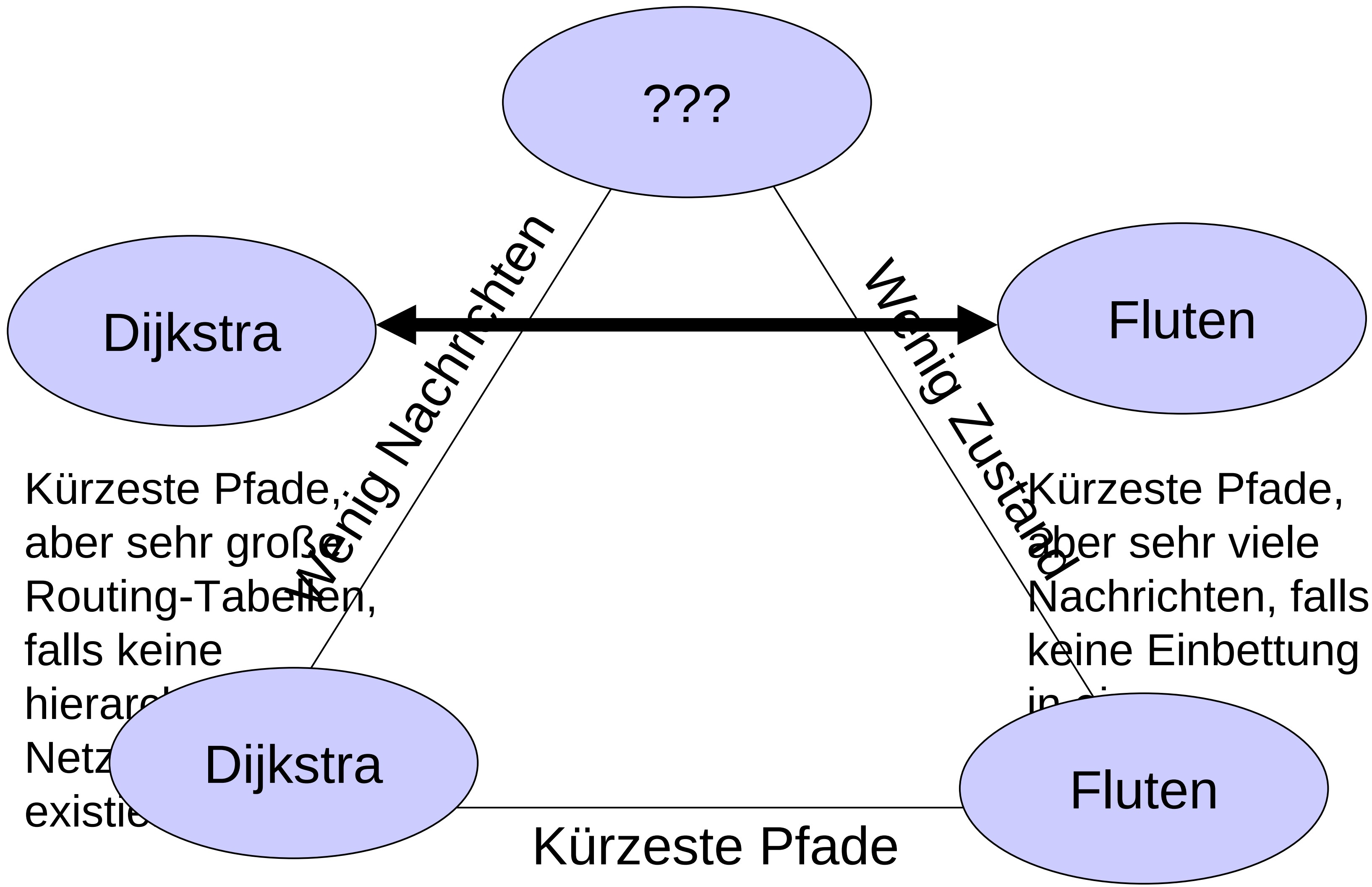


Beacon Vector Routing (2)

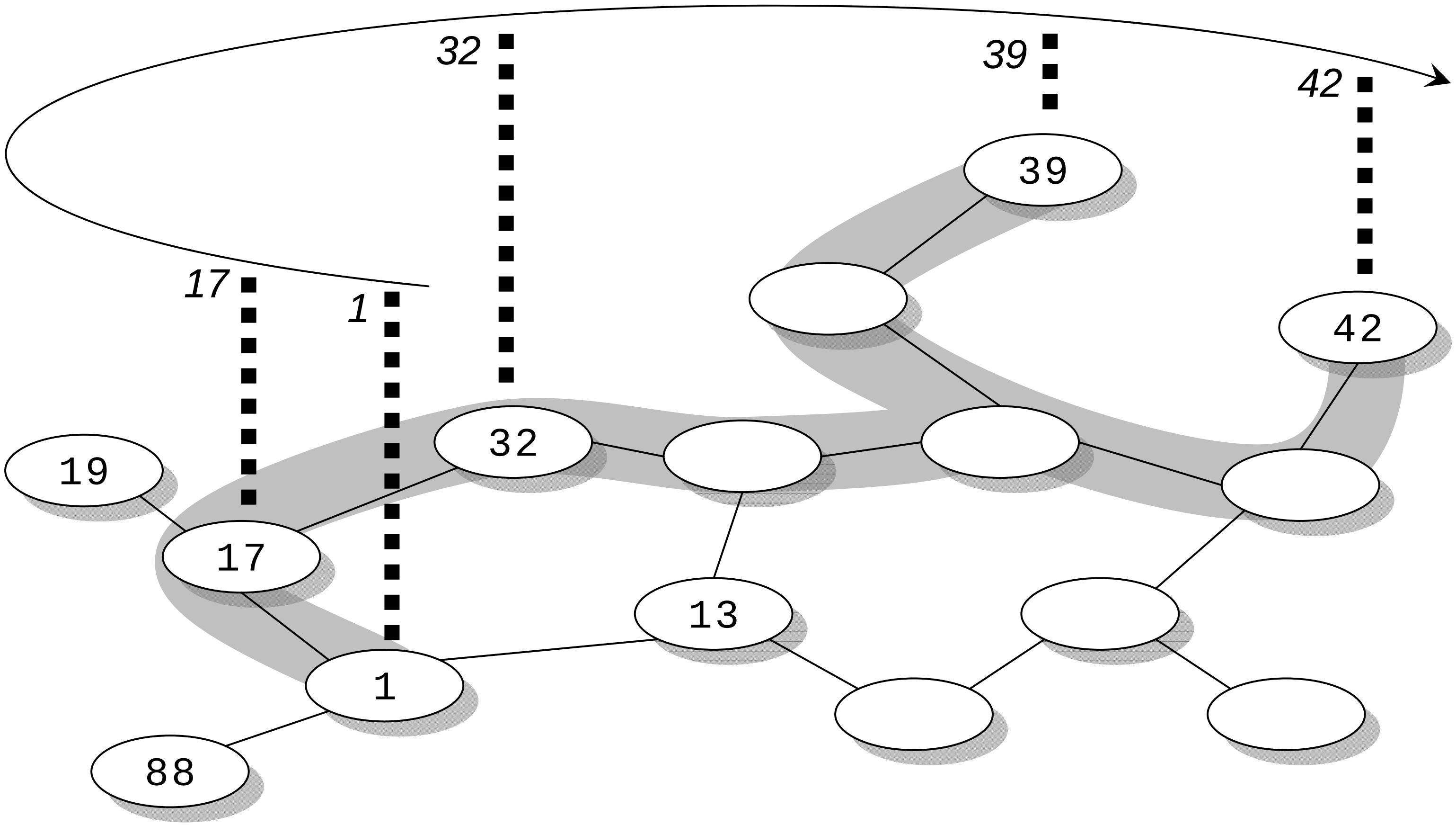
- Some nodes are elected as beacons.
- Beacons regularly broadcast their ID.
- When forwarding beacon messages, nodes update the messages‘ hop count.
- Nodes collect beacon messages to form a vector containing the beacons‘ hop count.
- When routing a packet to a given destination ID, nodes try to match the destination‘s beacon vector with their neighbors‘ beacon vectors.
- If beacon vectors are not unique, use local flooding and the destinations globally unique ID.

- Advantage
  - Proactive
  - Robust
  - Limited state
- Drawback
  - Beacon election difficult
  - Potentially not unique
  - Beacons flood the network

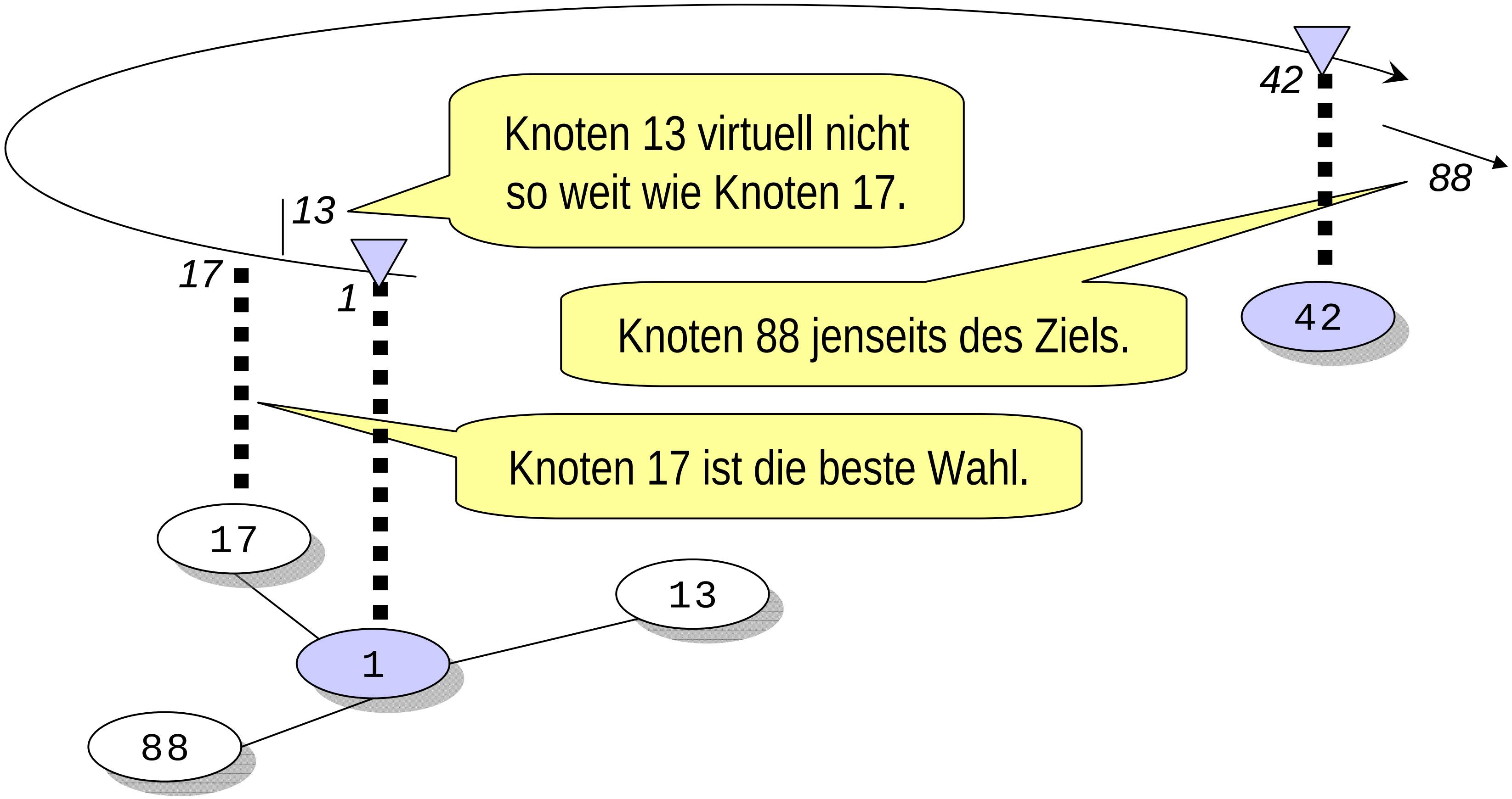
Wegewahlverfahren – Ein Überblick



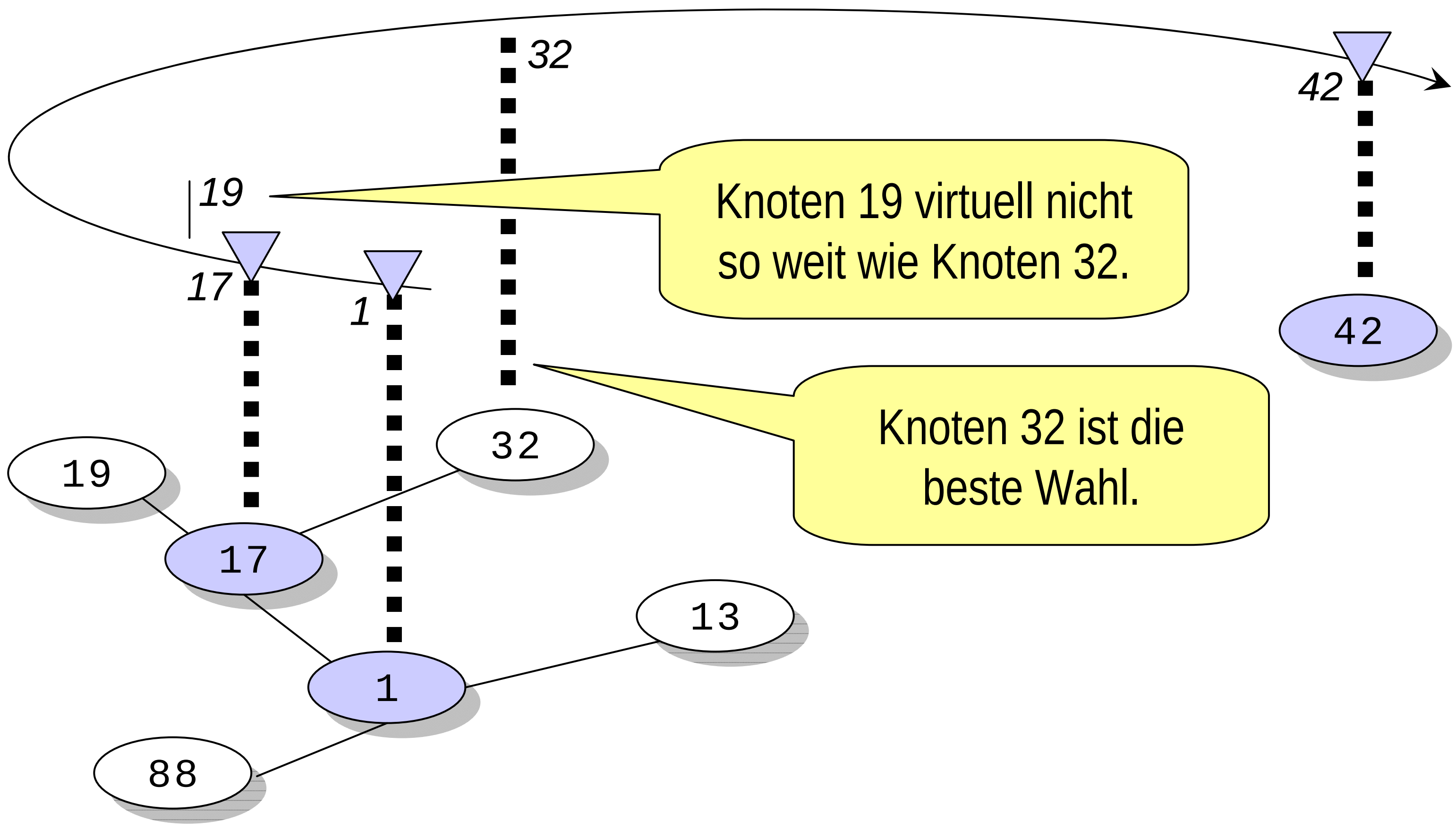
Virtual Ring Routing / Scalable Source Routing



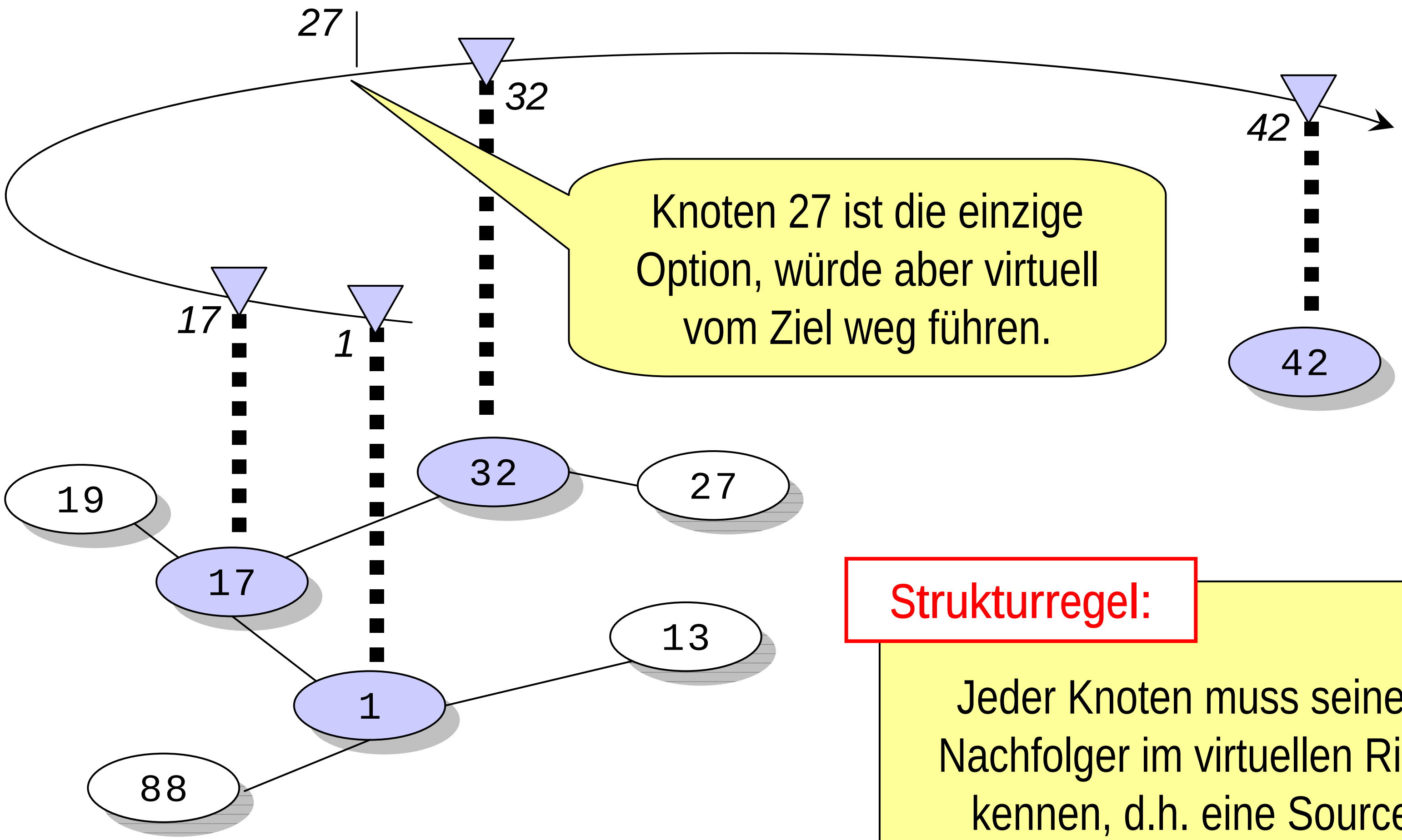
Scalable Source Routing



Scalable Source Routing



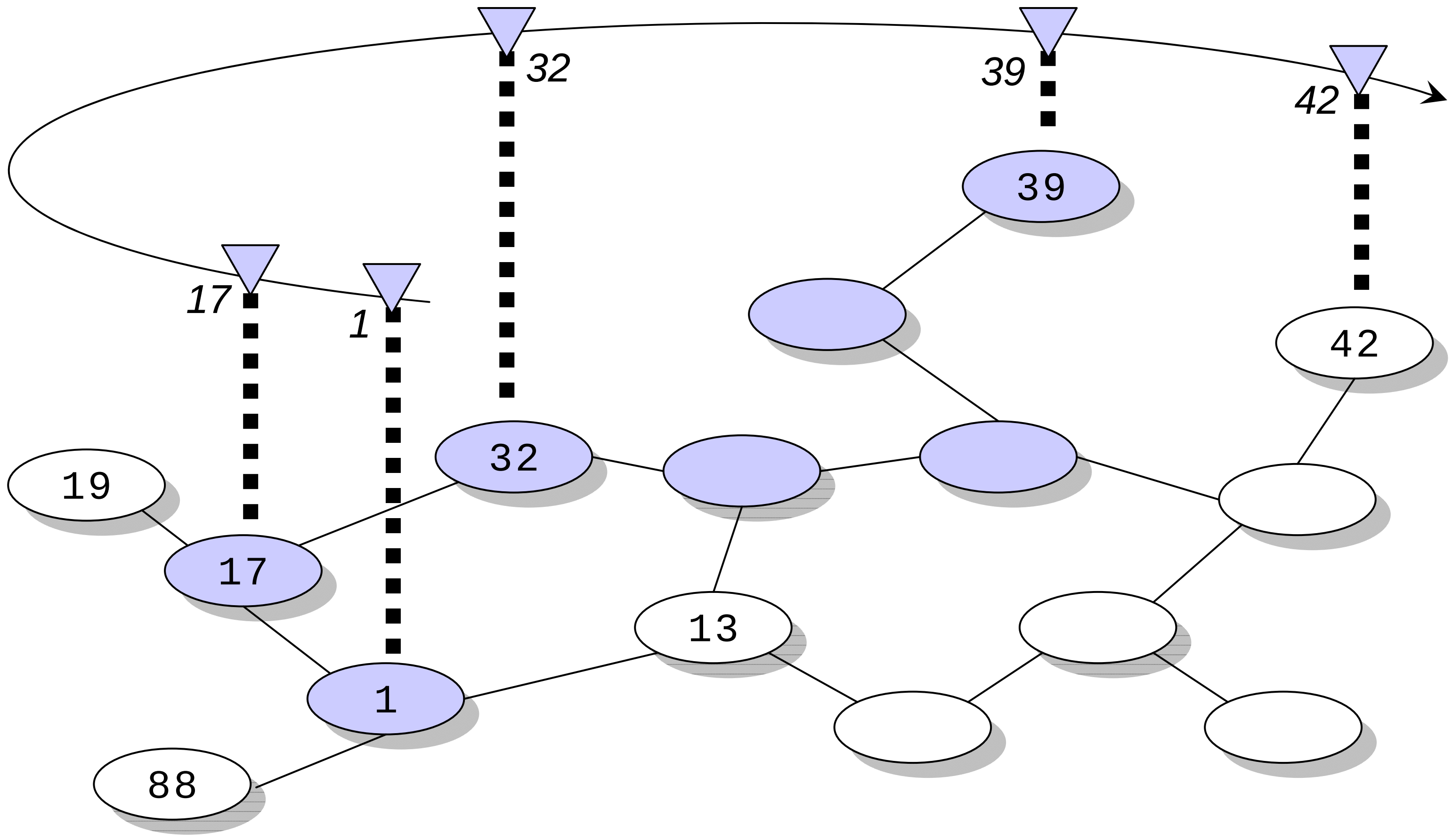
Scalable Source Routing



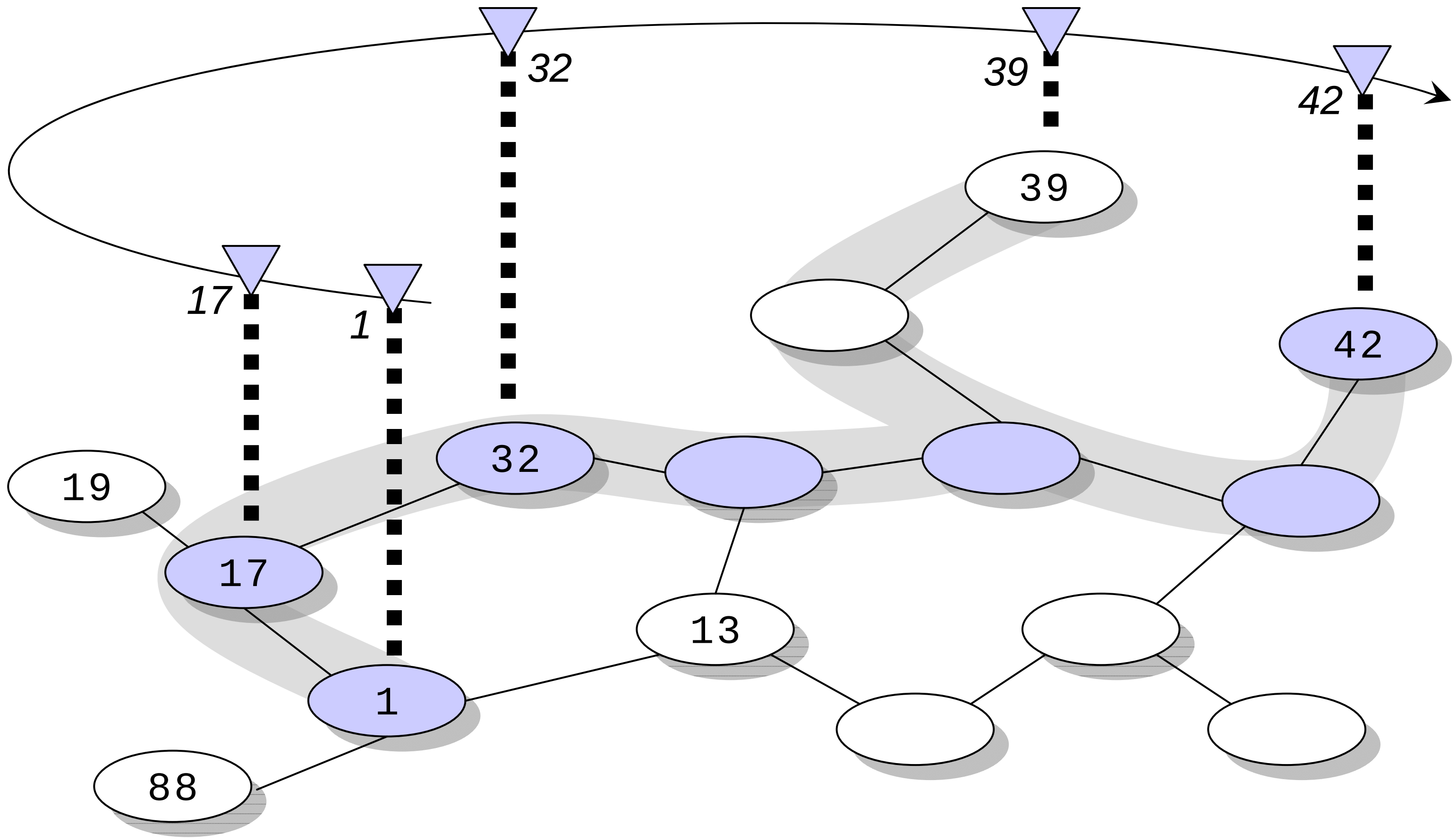
Strukturregel:

Jeder Knoten muss seinen Nachfolger im virtuellen Ring kennen, d.h. eine Source Route dahin haben.

Scalable Source Routing



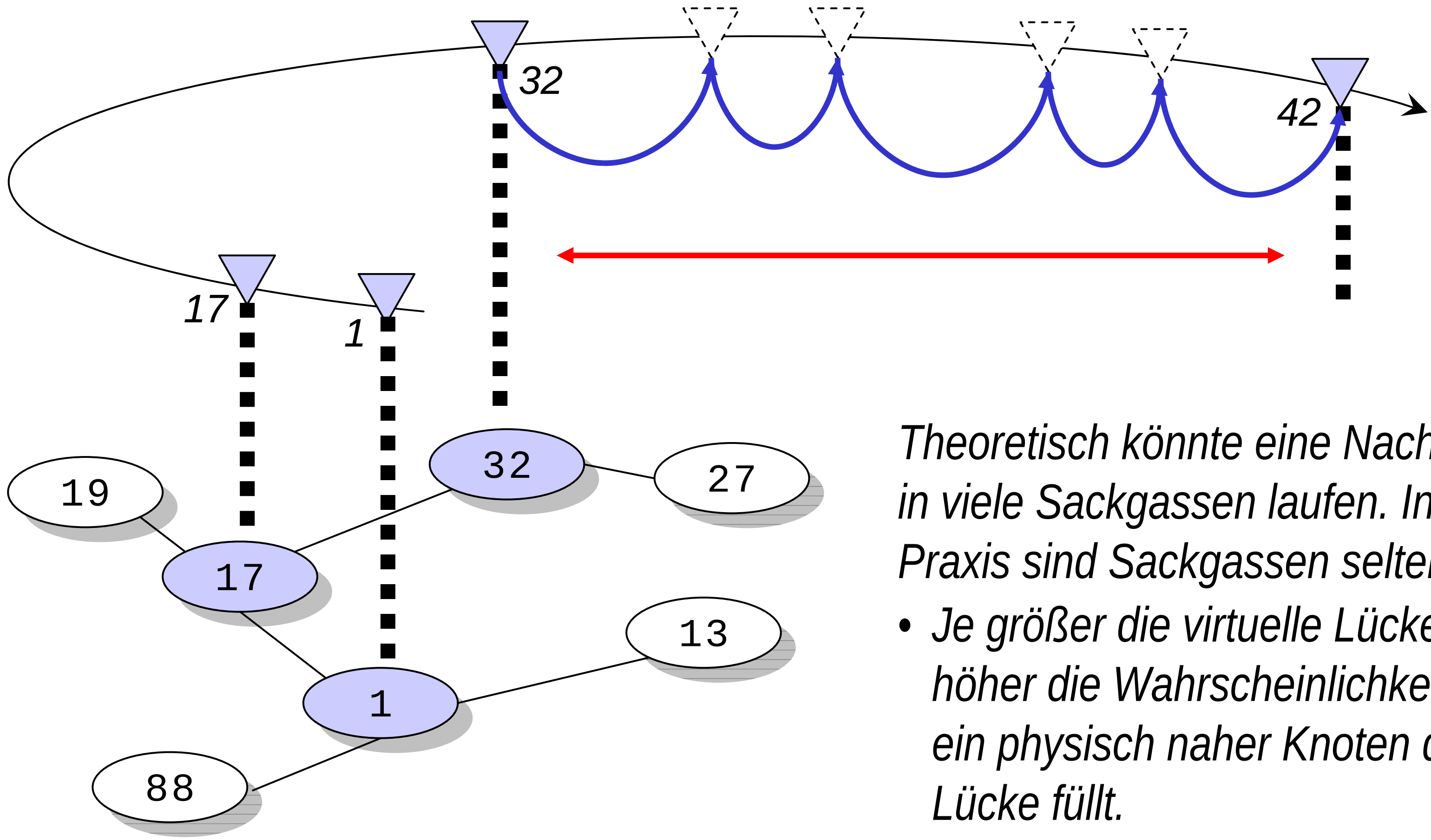
Scalable Source Routing



Fuhrmann, A Self-organizing Routing Scheme for Random Networks, Networking 2005.

Caesar et al. Virtual Ring Routing: Network Routing Inspired by DHTs, SIGCOMM 2006.

SSR – Selbstorganisation (1)

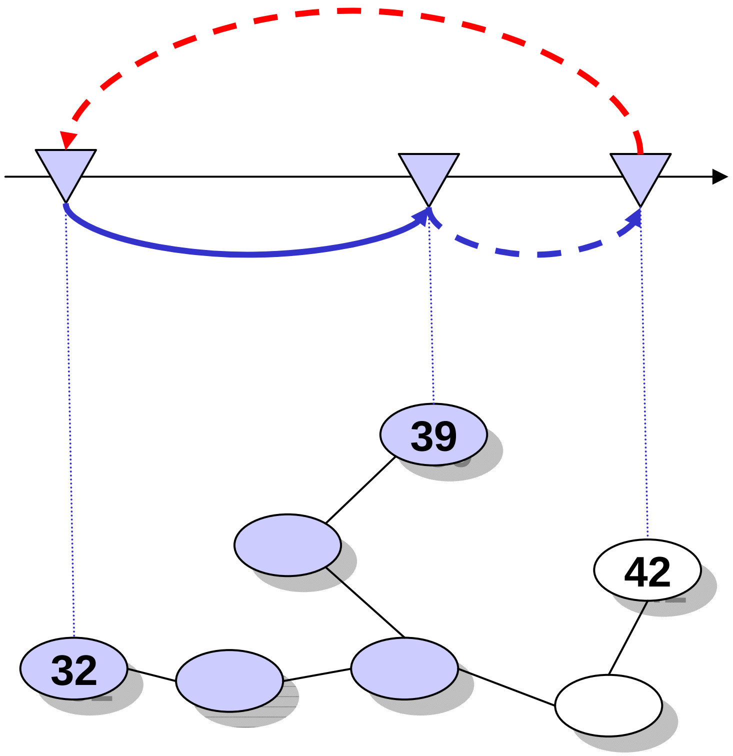


Theoretisch könnte eine Nachricht in viele Sackgassen laufen. In der Praxis sind Sackgassen selten:

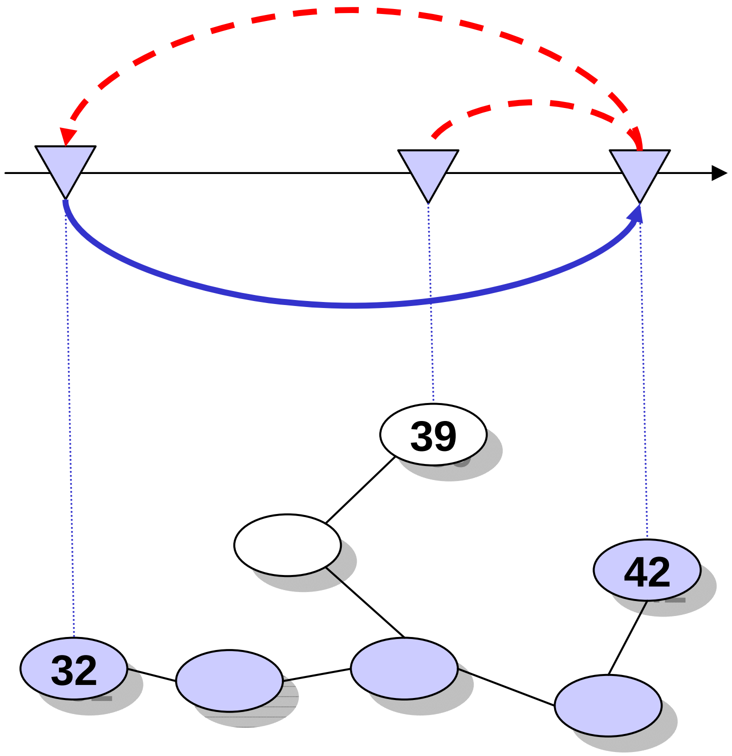
- *Je größer die virtuelle Lücke, desto höher die Wahrscheinlichkeit dass ein physisch naher Knoten die Lücke füllt.*
- *Knoten spezialisieren sich auf ihre virtuelle Nachbarschaft.*

SSR – Selbstorganisation (2)

Nachbarschaftsempfehlungen
(indirekte virtuelle Nachbarn)

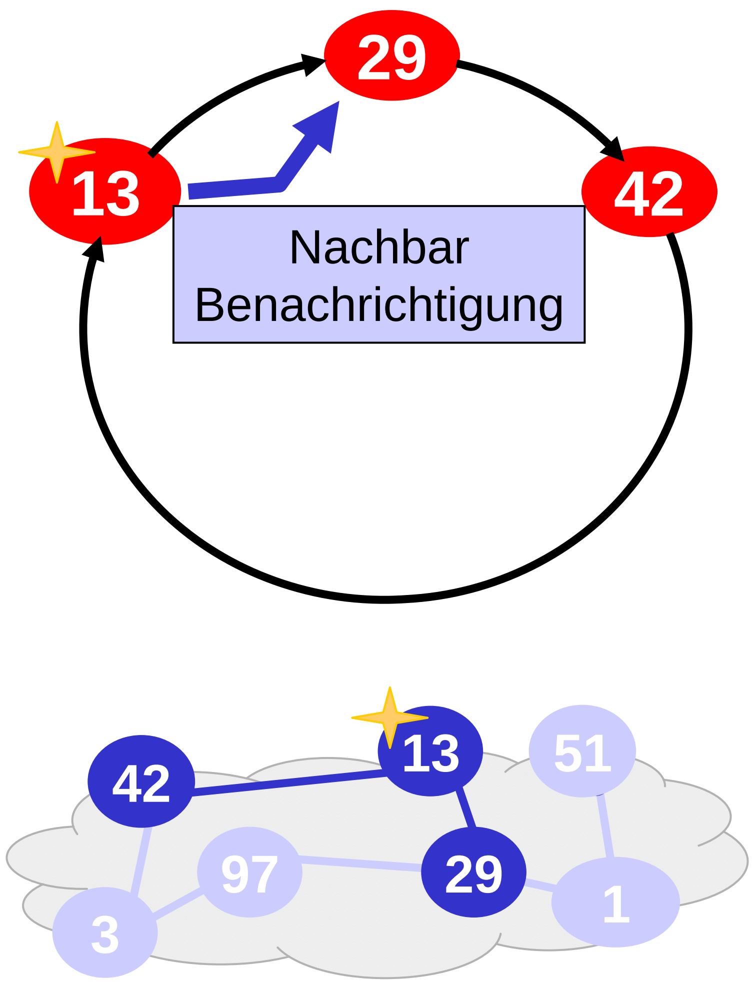


Nachbarschaftsberichtigungen
(direkte virtuelle Nachbarn)

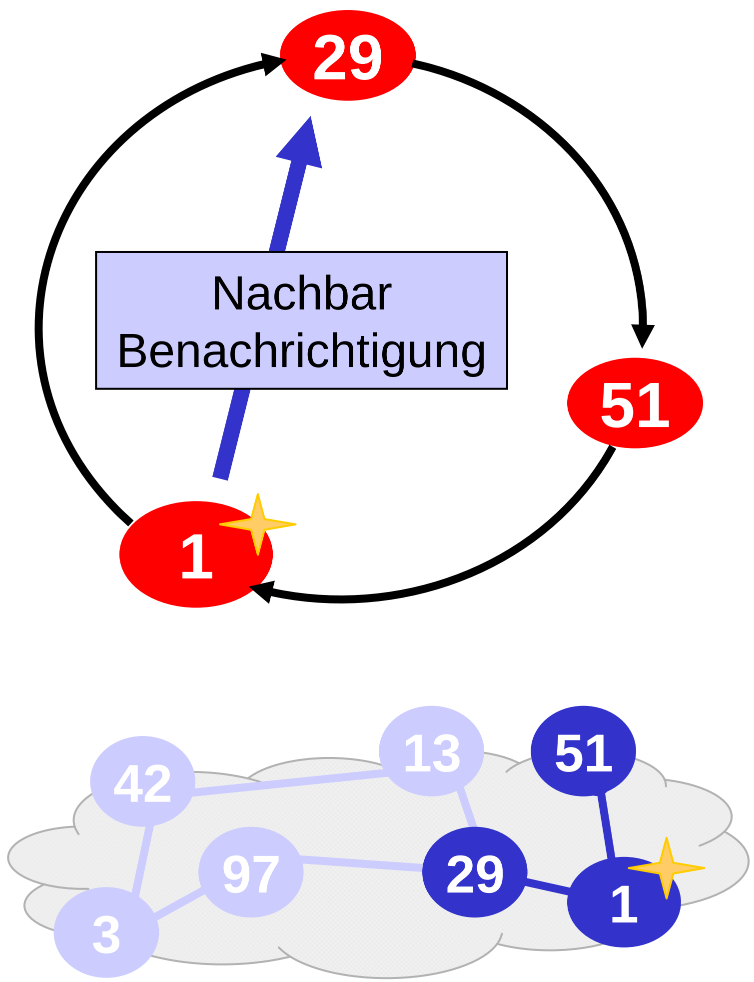


Iterative Nachbarsuche (1)

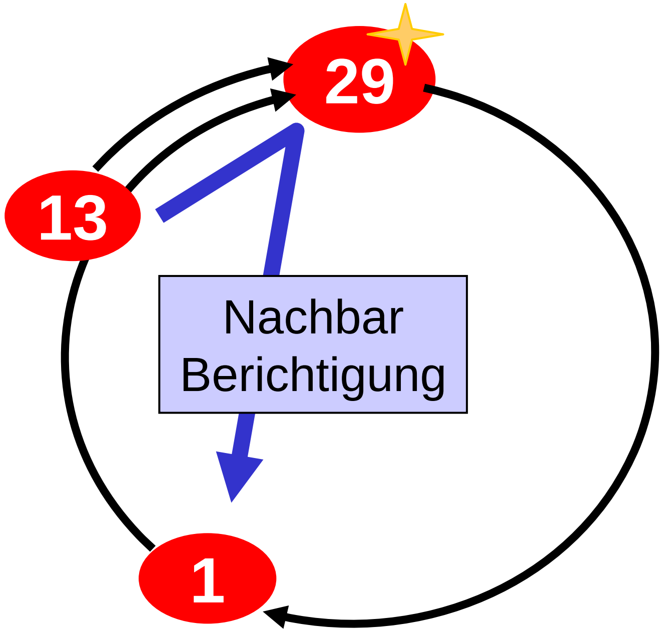
Sicht von Knoten 13:

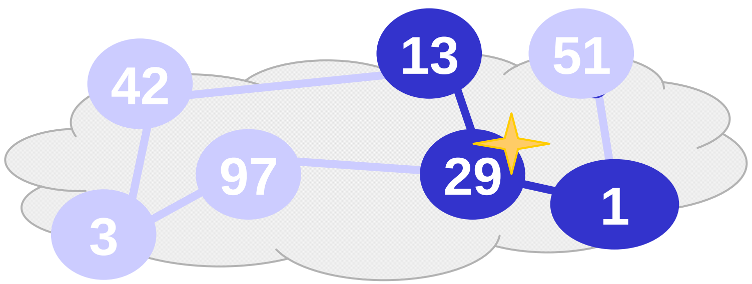


Sicht von Knoten 1:



Iterative Nachbarsuche (2)





Knoten 29 kann die Inkonsistenz auflösen:

Benachrichtigungen:

13	29	Notify

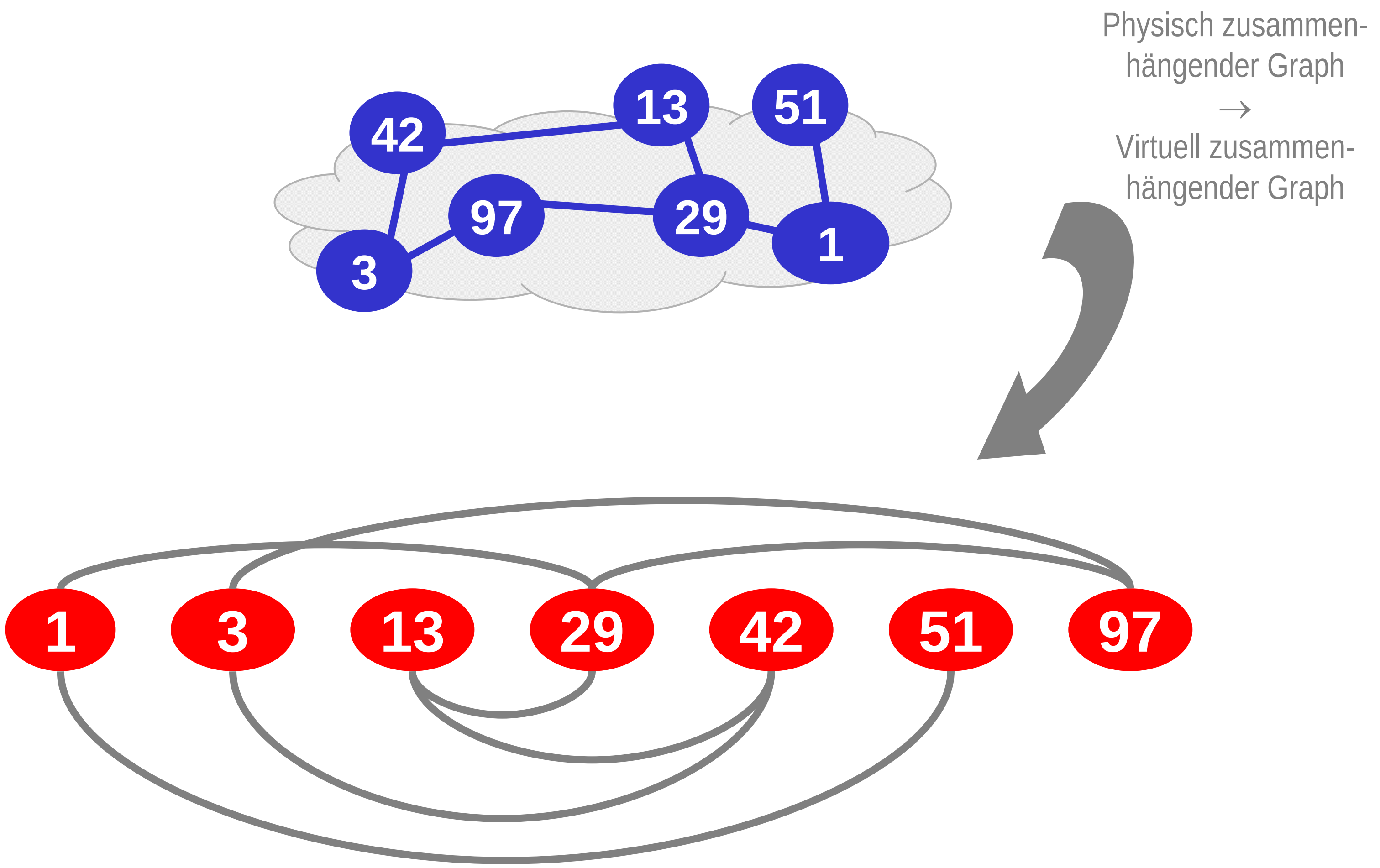
1	29	Notify

Berichtigung:

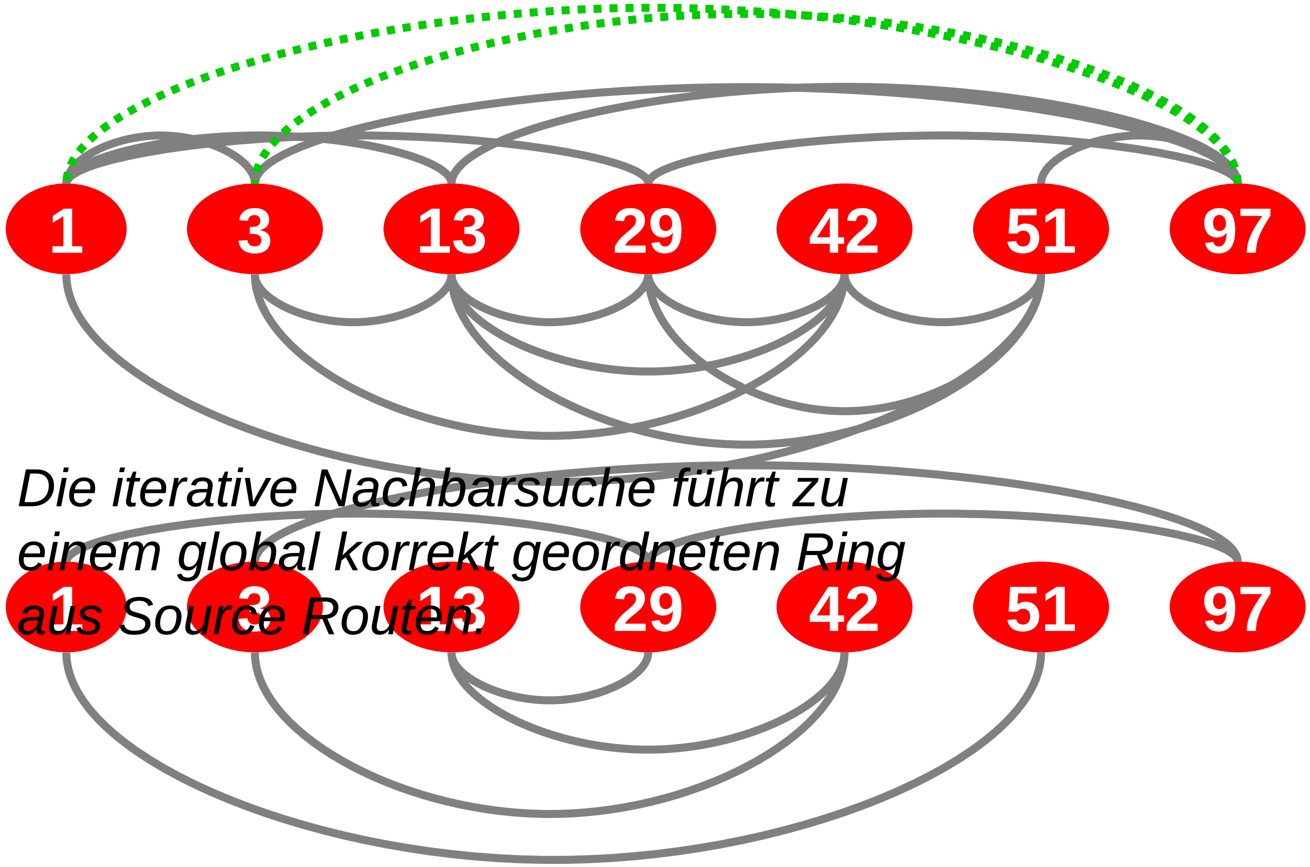
13	29

U	1	29	

Beweisskizze



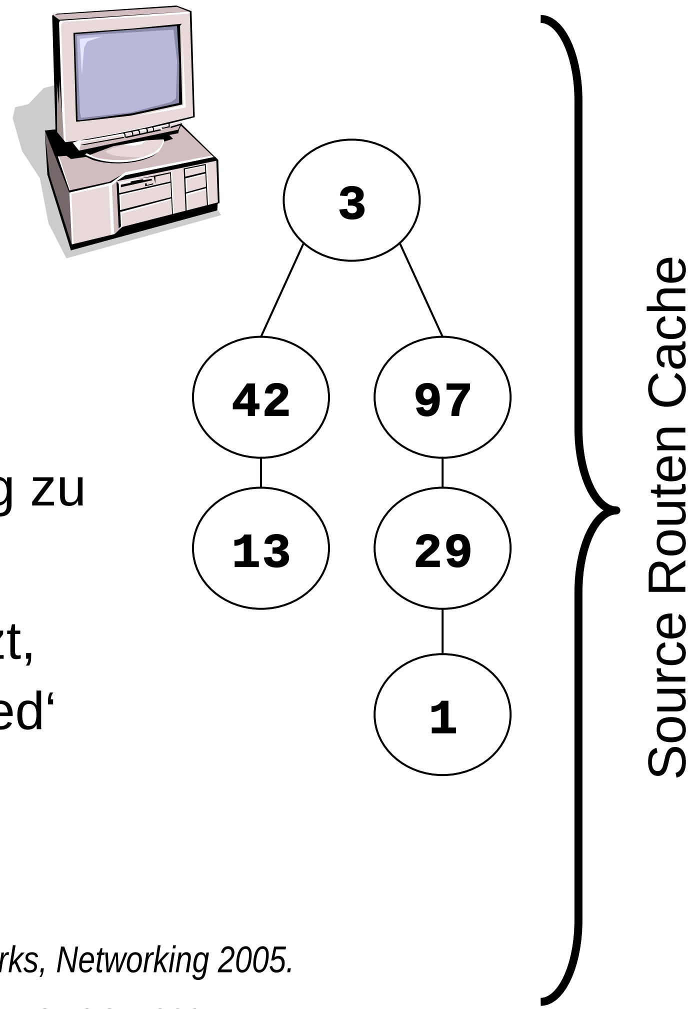
Beweisskizze



Die iterative Nachbarsuche führt zu einem global korrekt geordneten Ring aus Source Routen.

Ein Source Routen Cache

- Jeder Knoten speichert die Source Route zu seinem Nachfolger im virtuellen Ring (=Grundregel, vgl. Chord).
- Jeder Knoten speichert die Source Route zu seinem Vorgänger im virtuellen Ring (um ggf. eine Berichtigung zu senden).
- Weiterer Speicher wird als Cache genutzt, um Source Routen auf ‚least recently used' Basis zu speichern.



Fuhrmann, A Self-organizing Routing Scheme for Random Networks, Networking 2005.

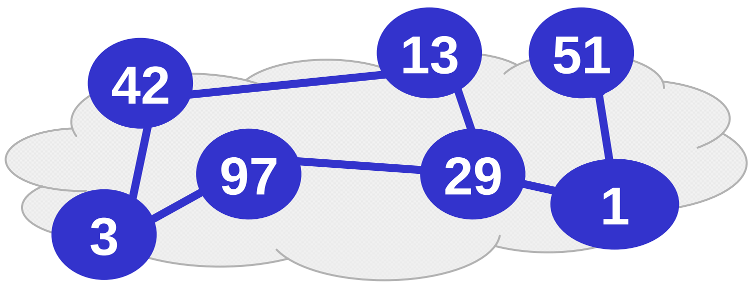
Fuhrmann, Scalable Routing for Networked Sensors and Actuators, SECON 2005.

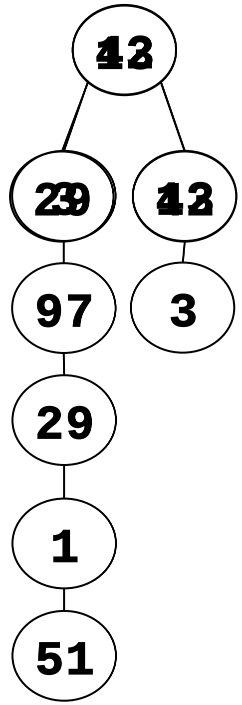
Anwendung der Chord-Routing-Regeln

- Mit Hilfe der Source Routen im Cache wird nach den Regeln des nähengewahren Chord-Netzes geroutet:
- Bei einem ‚cache miss' wird die Anfrage zu dem Knoten weitergereicht, der
  - in Ringrichtung noch vor dem Ziel liegt,
  - physisch am nächsten und
  - virtuell am weitesten liegt.

<u>Beispiel:</u> **13 → 51**

13	42	3	97	29	1	51





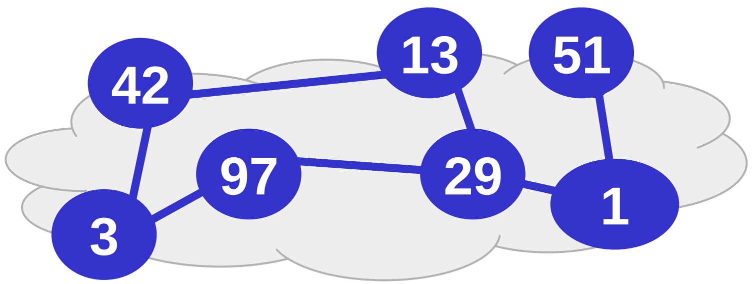
Beispiel (cont.):

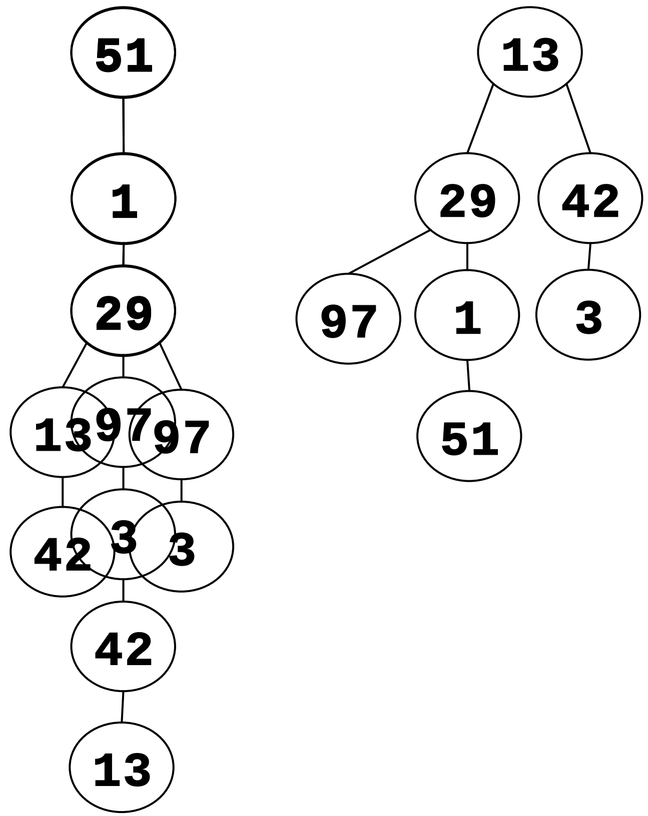
Im Cache sammeln sich allmählich gute Routen:

13	42	3	97	29	1	51

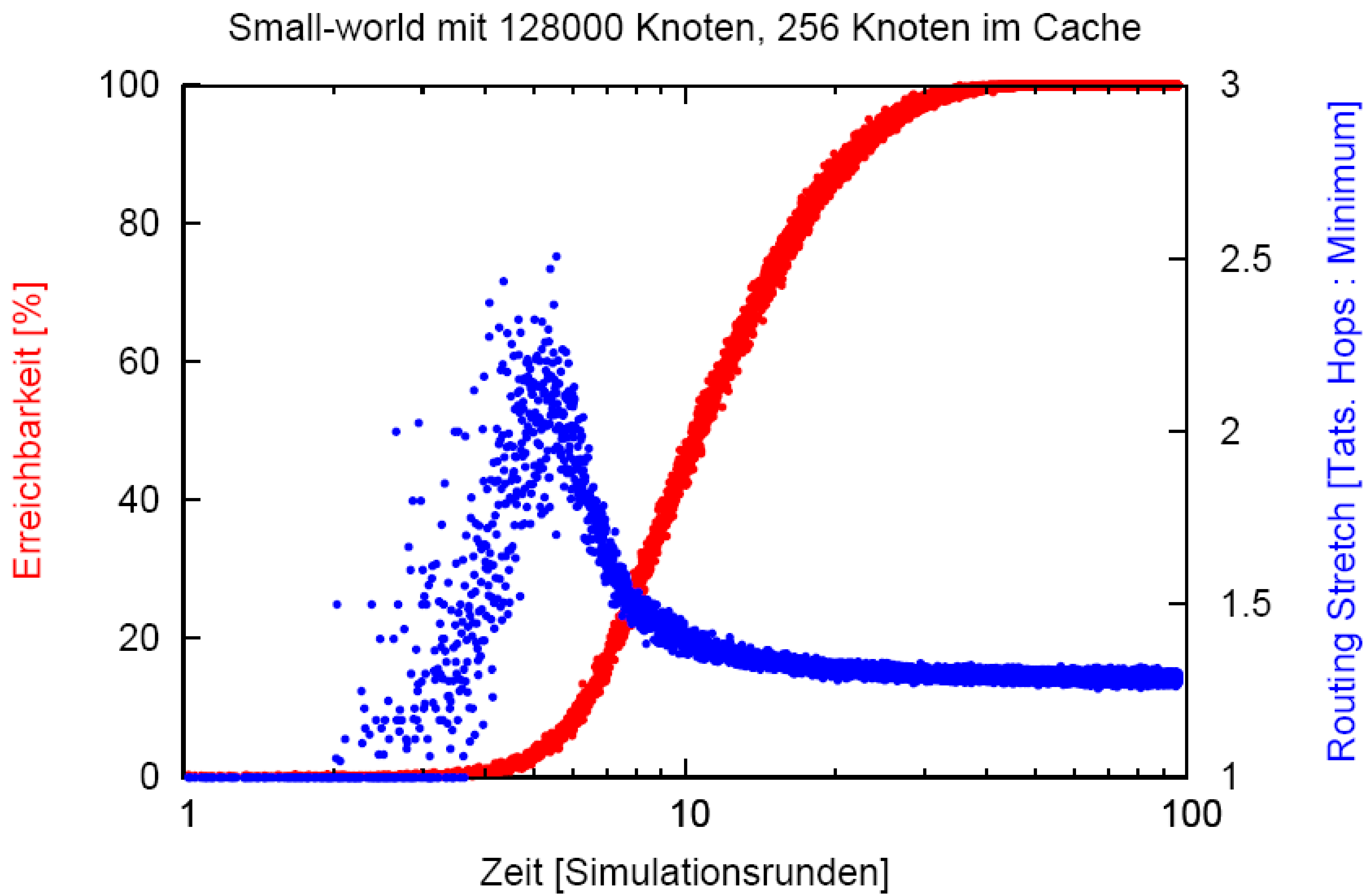
51	1	29	97	3	42	13

13	29	1	51

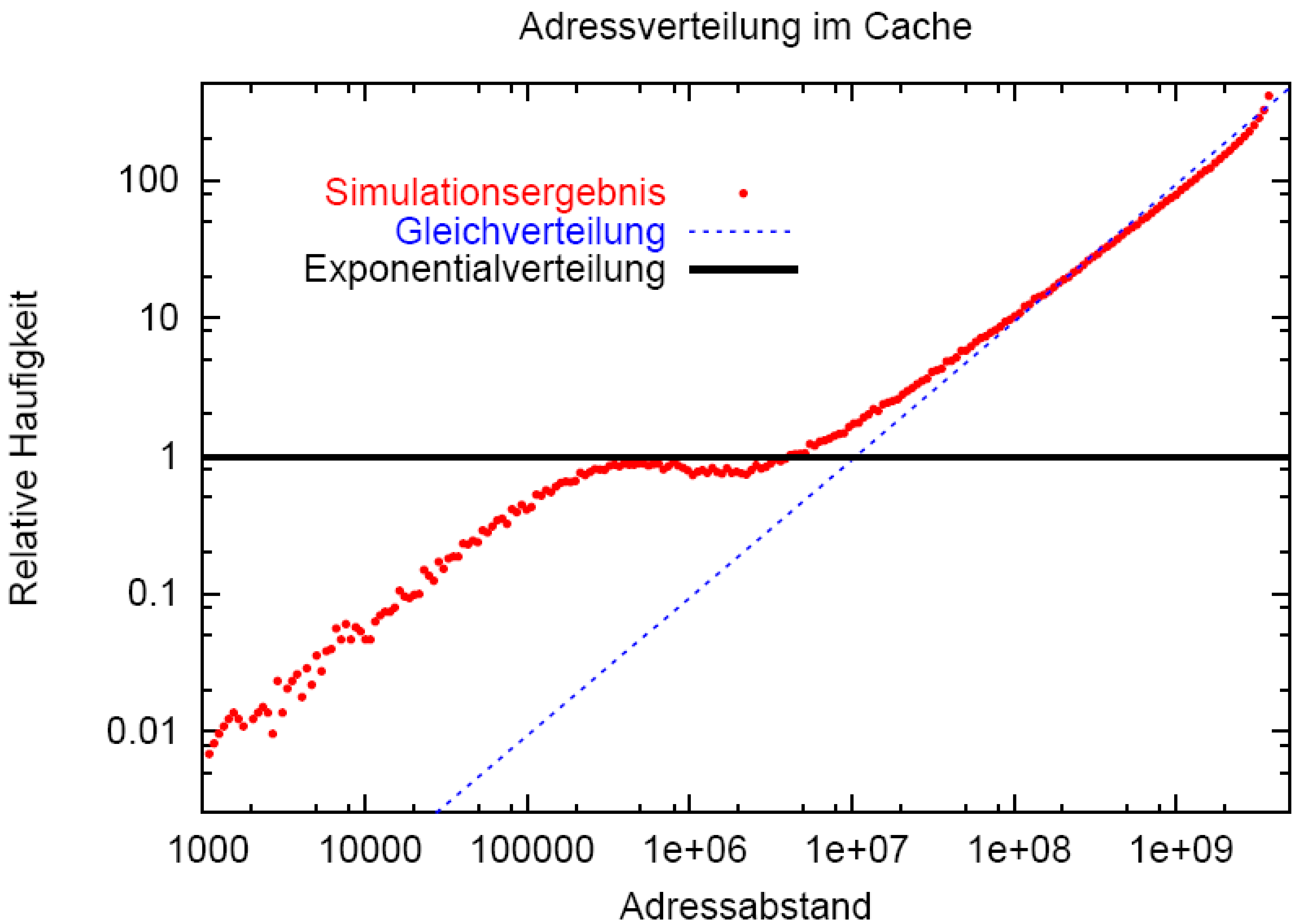




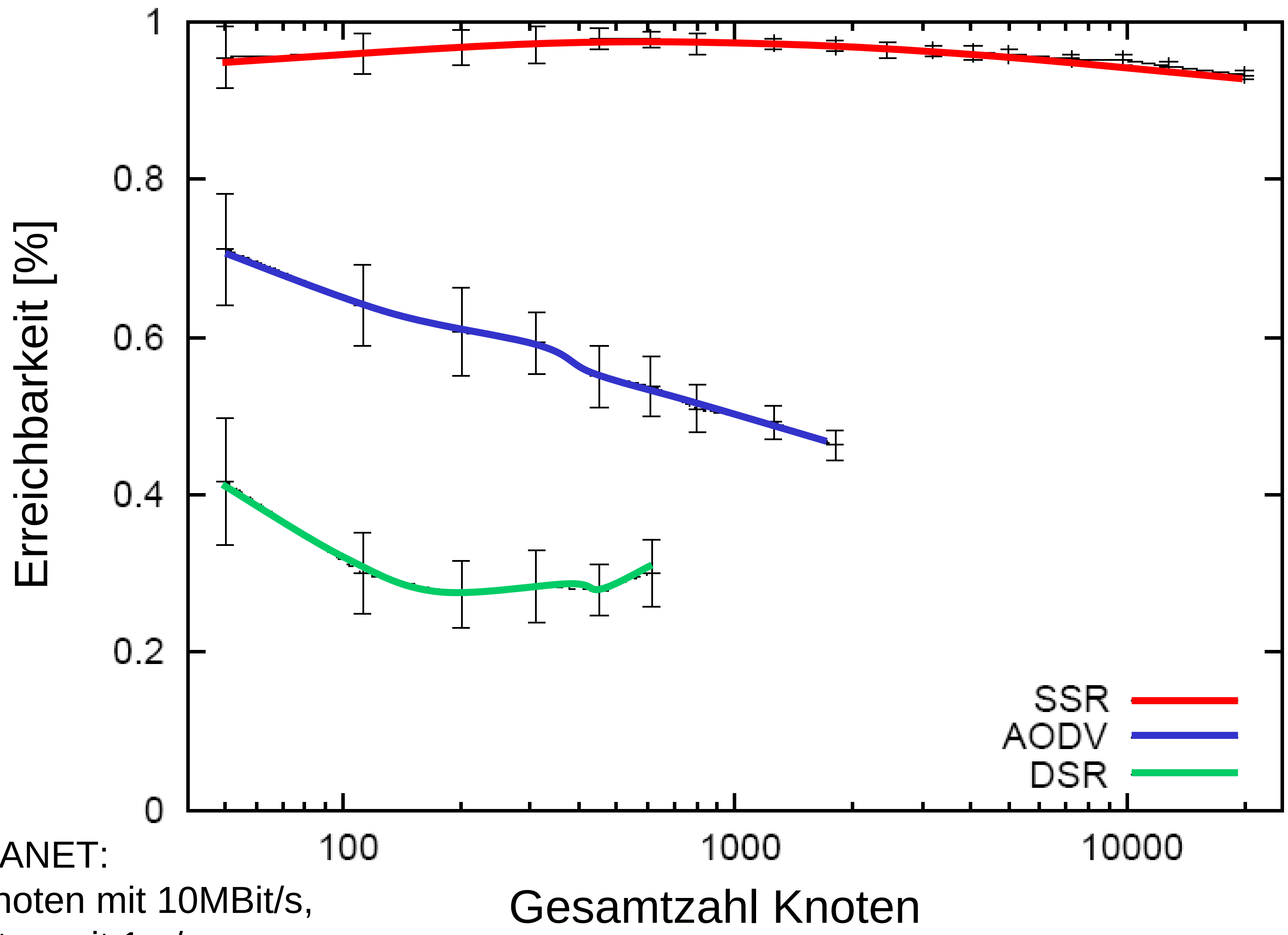
Simulationsergebnisse (1) - Bootstrapping



Simulationsergebnisse (2) - Adressverteilung

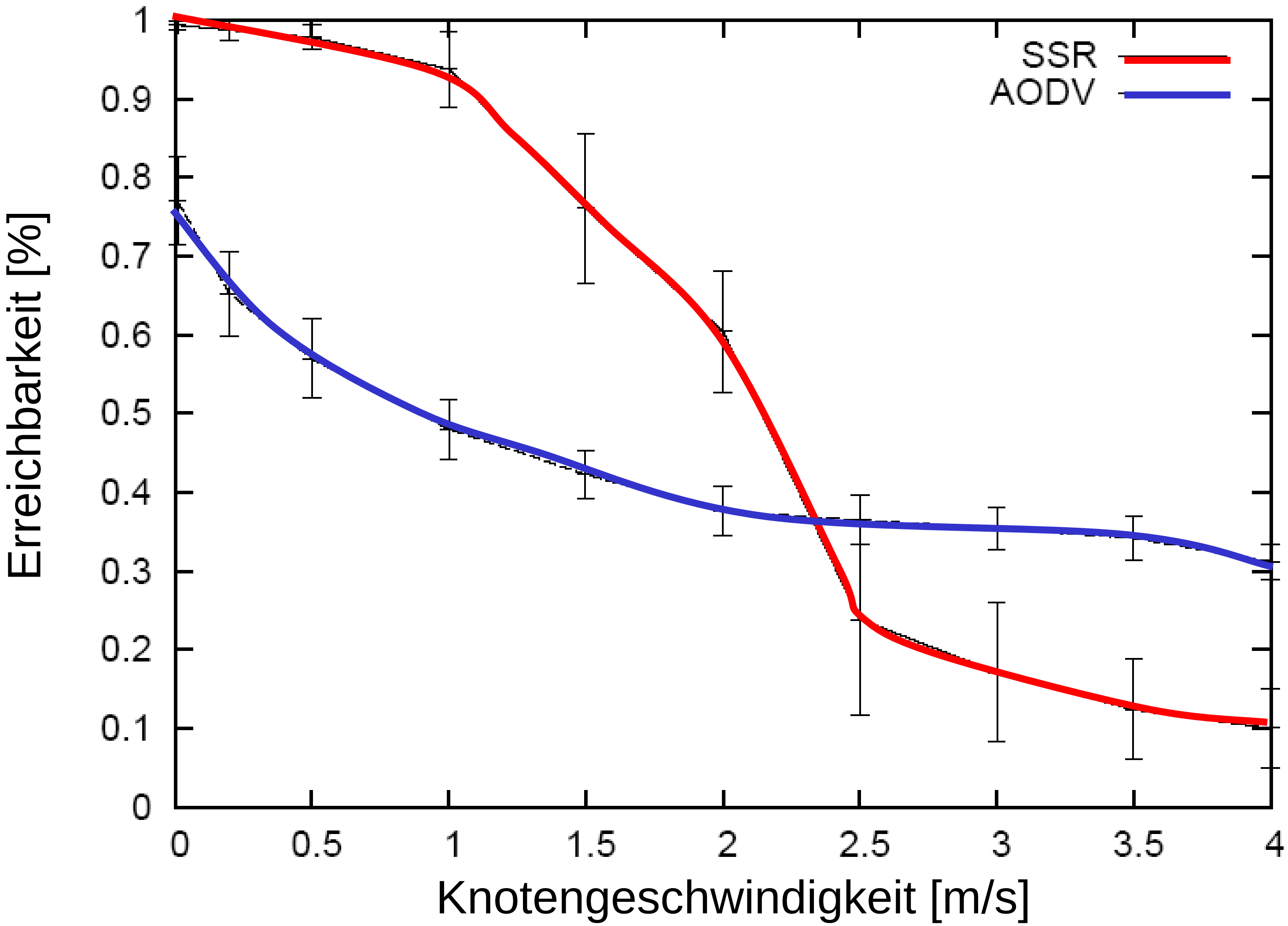


Simulationsergebnisse (3) - AODV und DSR



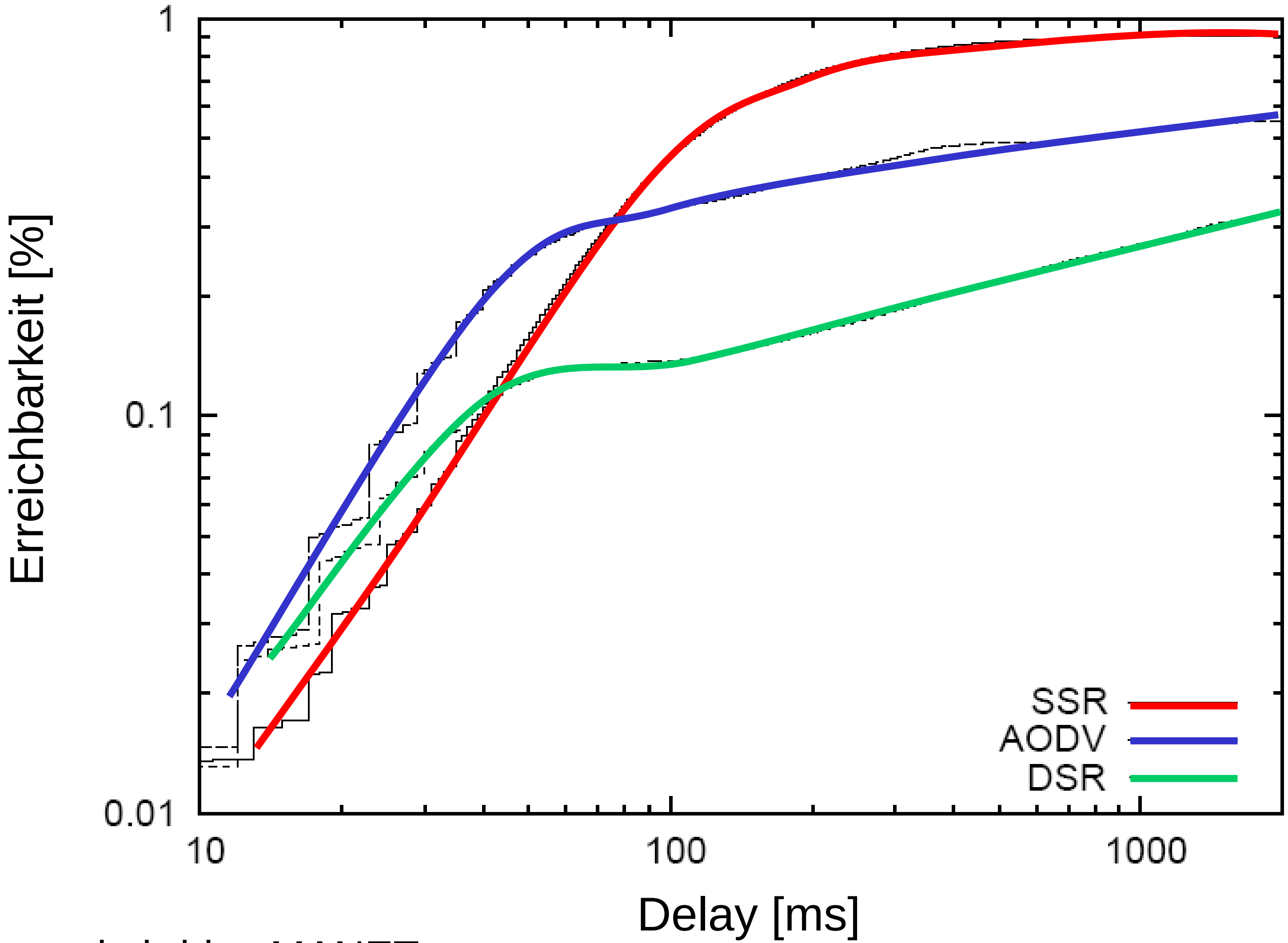
Hybrides MANET:
5% feste Knoten mit 10MBit/s,
mobile Knoten mit 1m/s

Simulationsergebnisse (4) – Mobilität



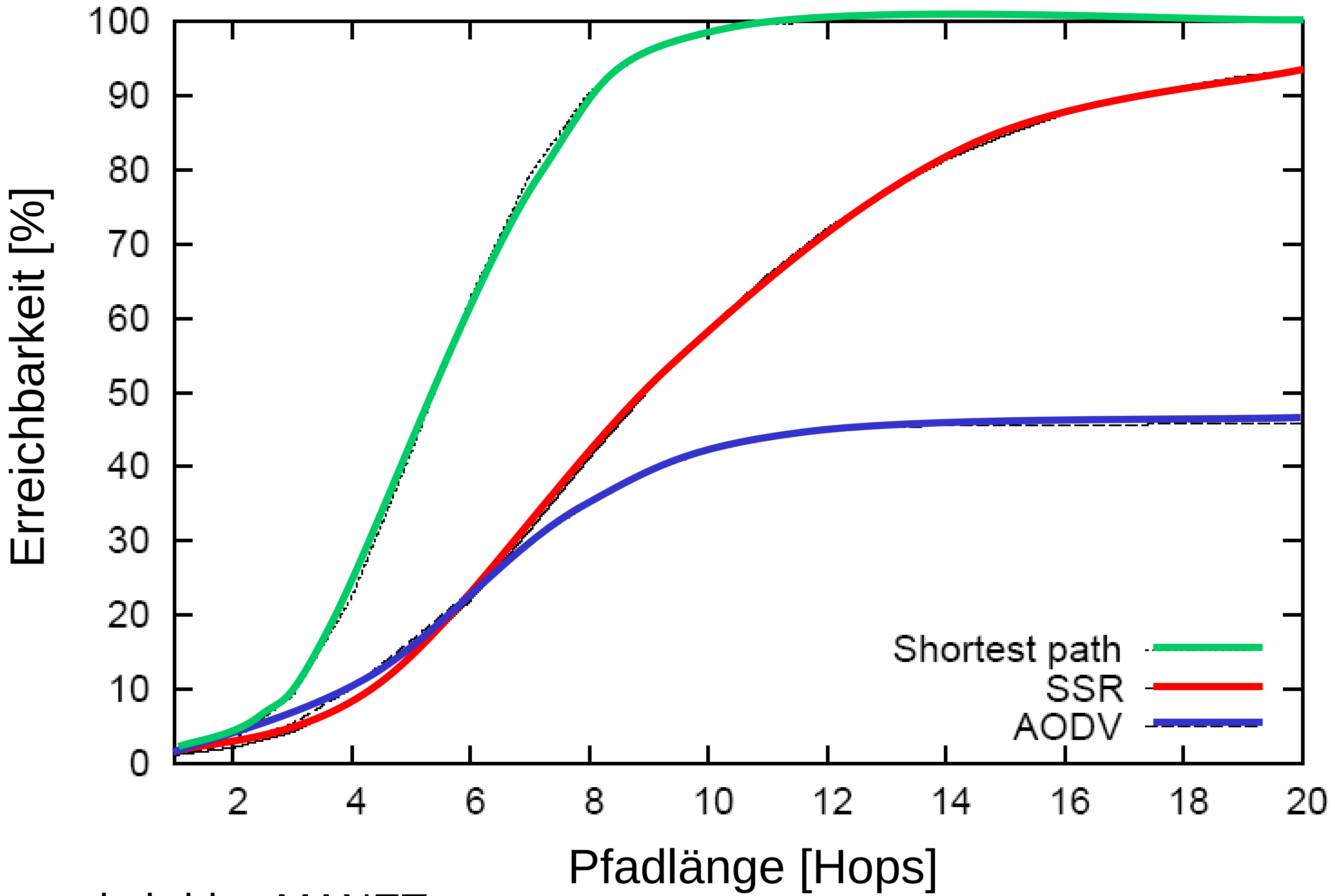
450 Knoten reines MANET

Simulationsergeb. (5) – Latenzverteilung



612 Knoten hybrides MANET

Simulationsergeb. (6) – Pfadlängenverteilung

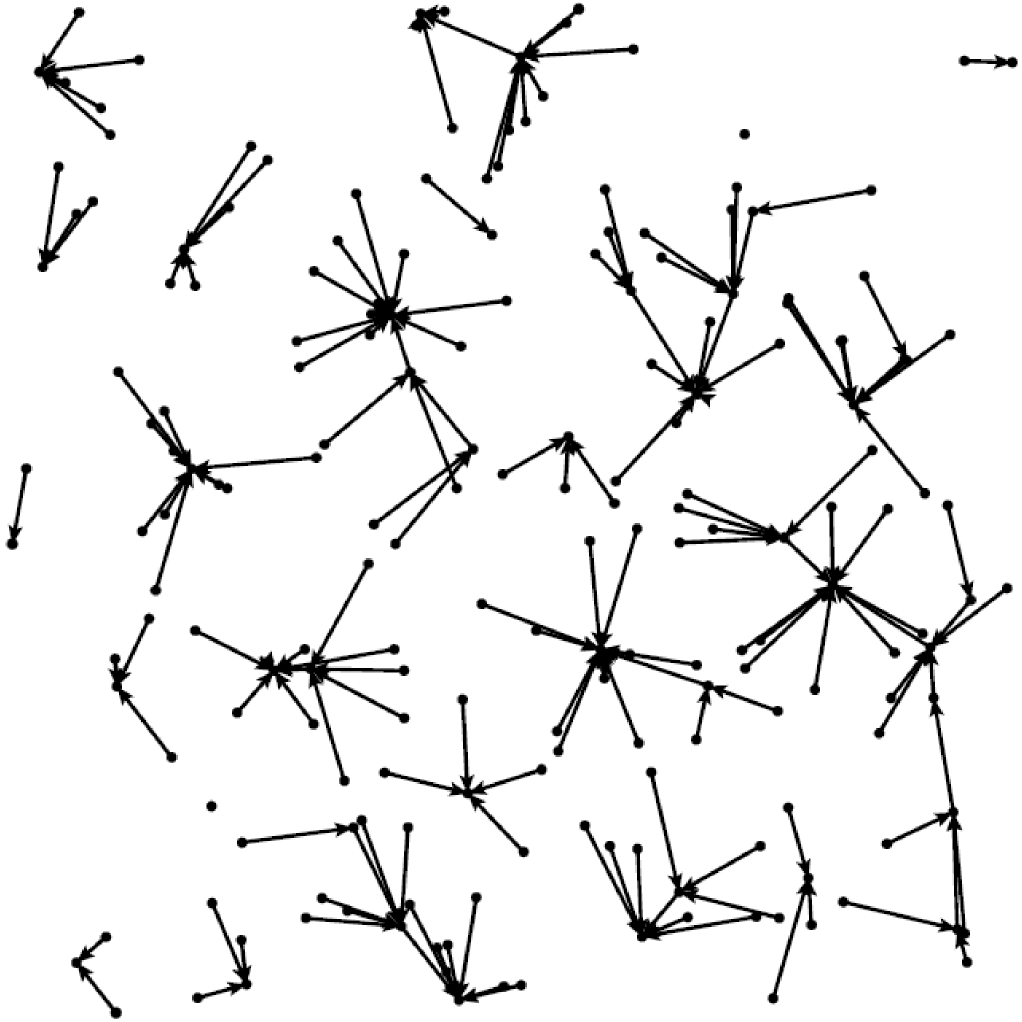


800 Knoten hybrides MANET

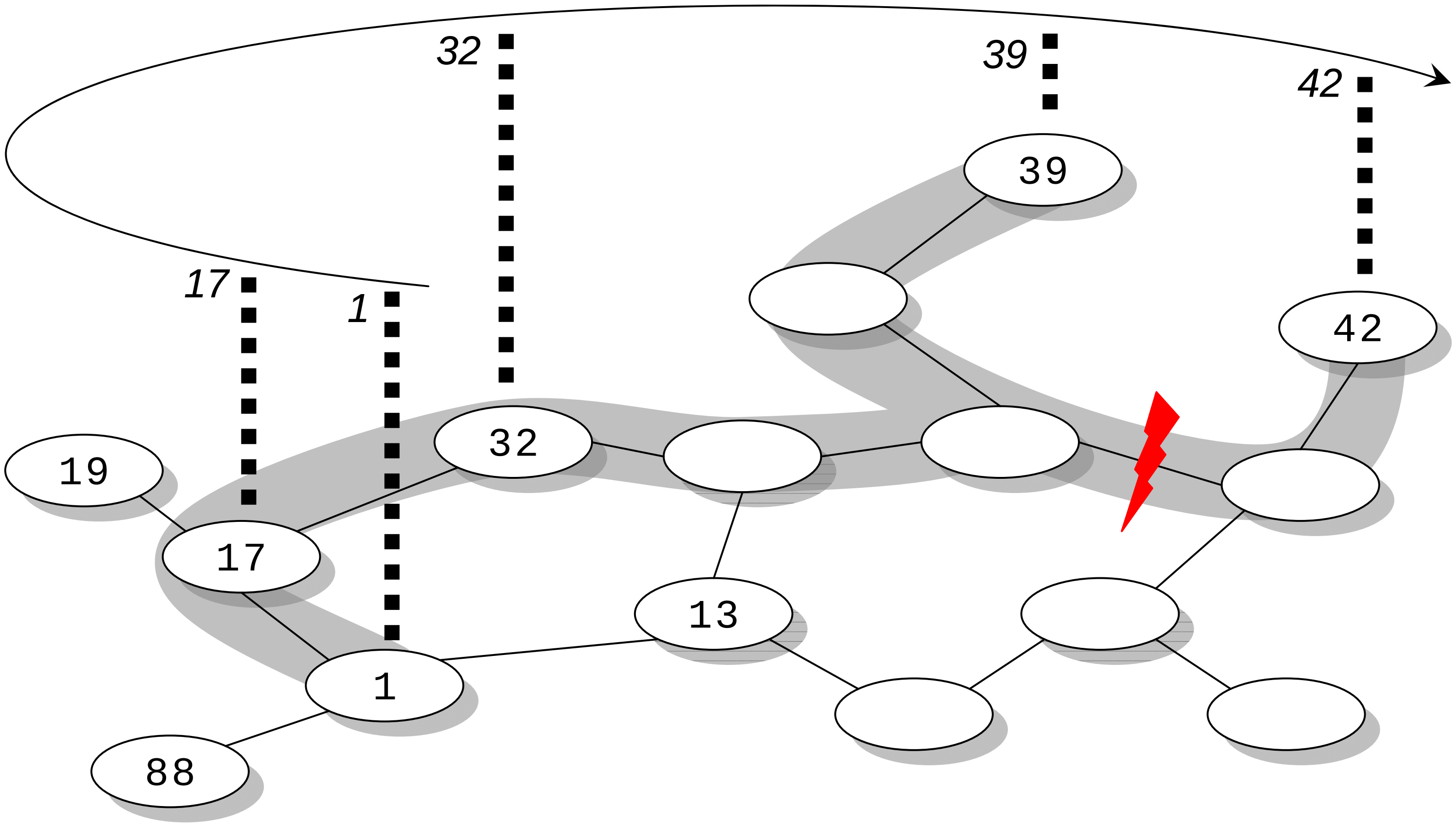
Simulation Results (7) – Cluster Formation

Unit disk graph with uniformly random node positions (critical density).

Forwarding to physical neighbor according to routing rule leads to cluster formation.



Unzuverlässige und Mobile Geräte …

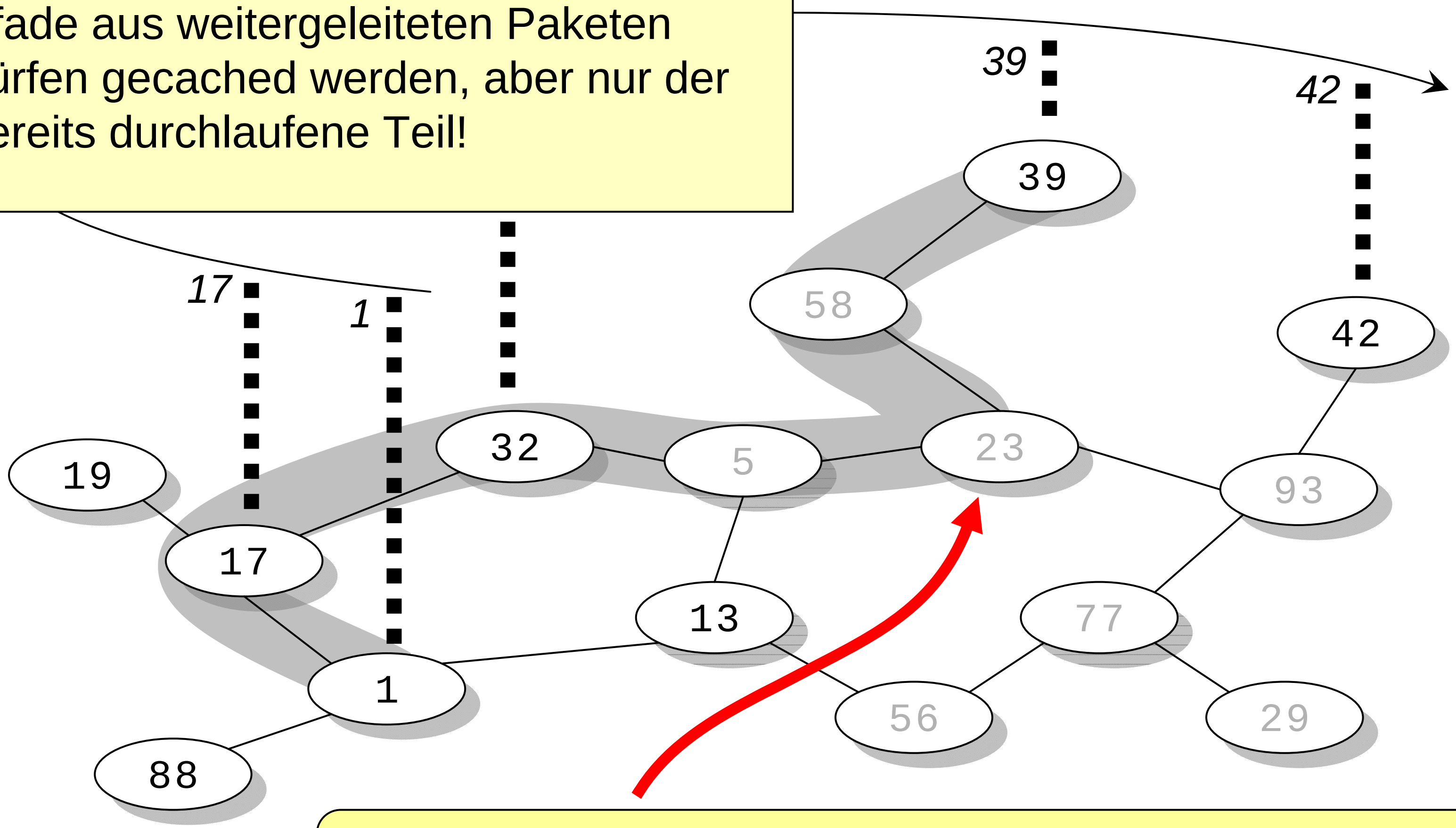


Fuhrmann, Scalable Routing for Sensor Actuator Networks with Churn, SECON 2006.

Fuhrmann et al. Scalable Routing for Hybrid MANETs, WONS 2007

Regel 1: Nur authentische Pfade cachen

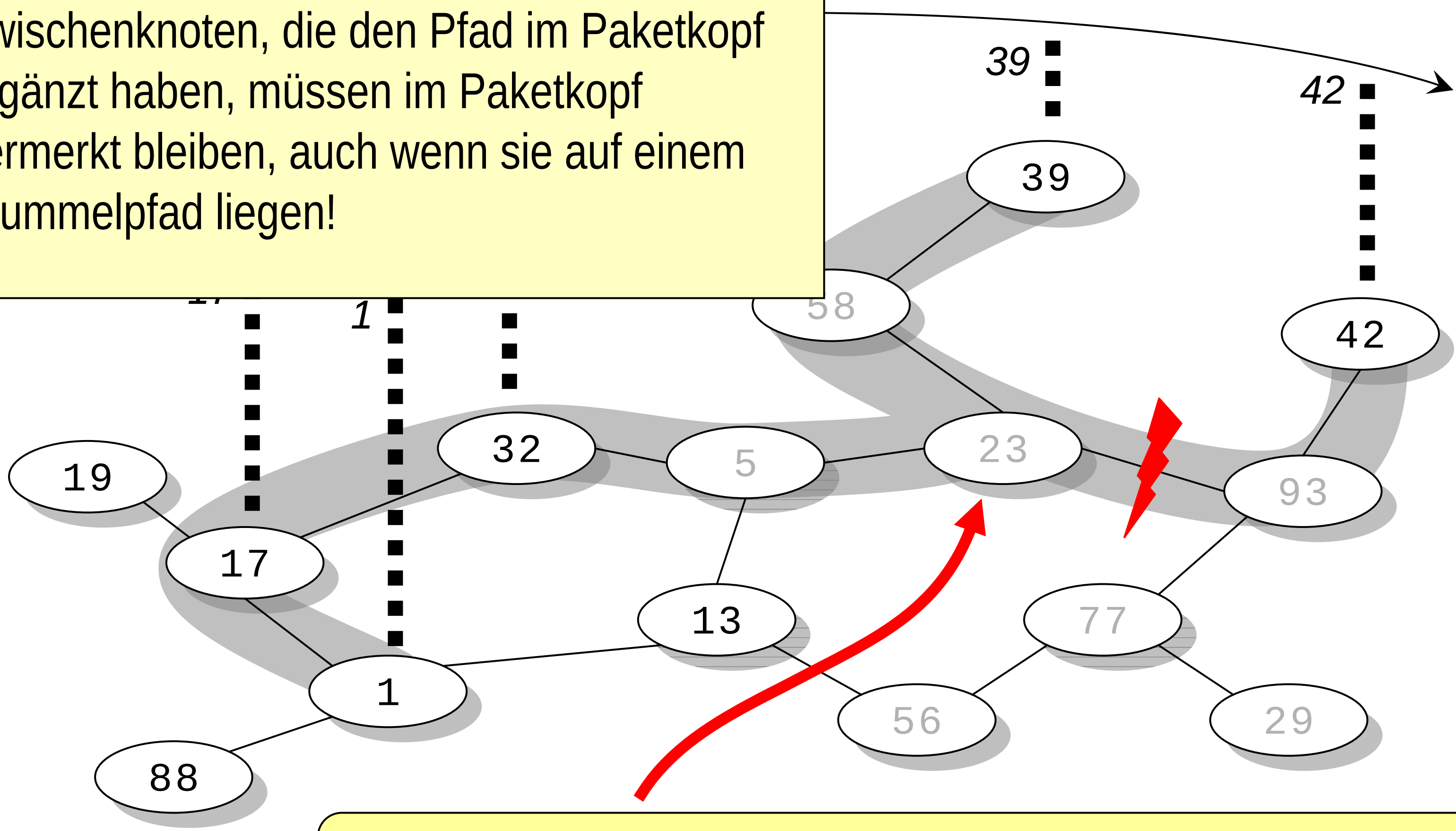
Pfade aus weitergeleiteten Paketen dürfen gecached werden, aber nur der bereits durchlaufene Teil!



1	17	32	5	23	58	39

Regel 2: Zwischenknoten behalten

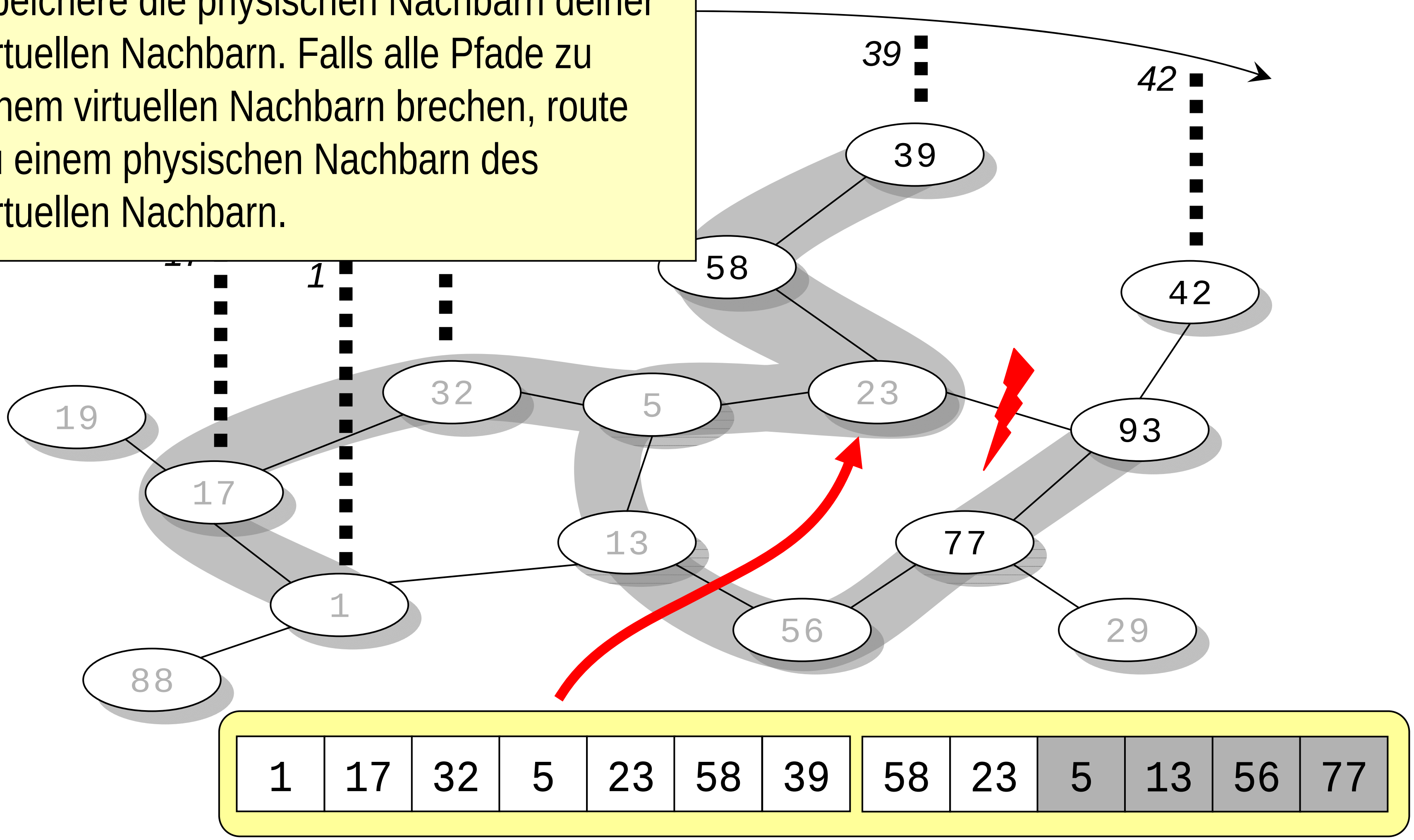
Zwischenknoten, die den Pfad im Paketkopf ergänzt haben, müssen im Paketkopf vermerkt bleiben, auch wenn sie auf einem Stummelpfad liegen!



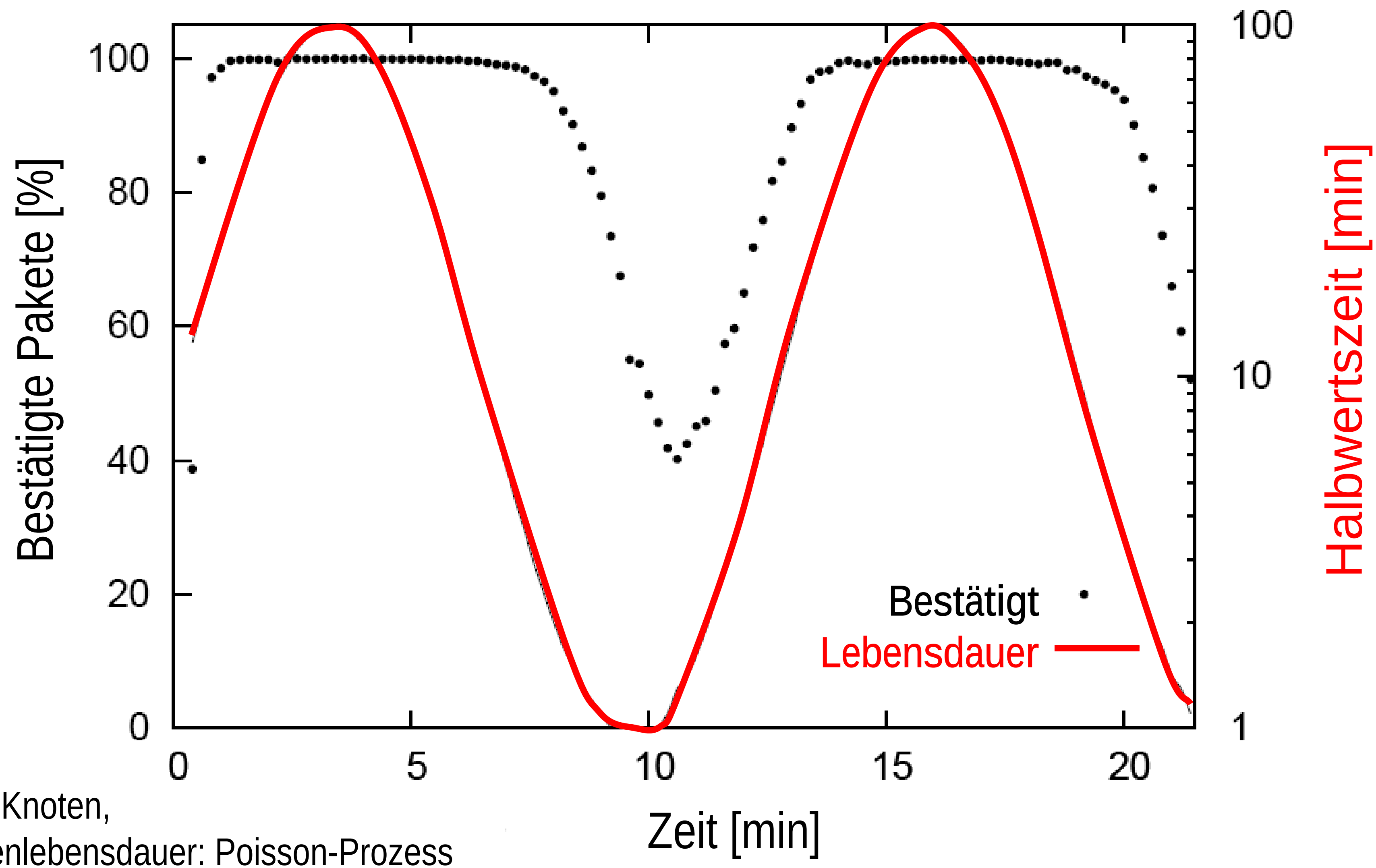
1	17	32	5	23	58	**39**	58	23	93	42

Regel 3: Physische Nachbarn mitbenutzen ...

Speichere die physischen Nachbarn deiner virtuellen Nachbarn. Falls alle Pfade zu einem virtuellen Nachbarn brechen, route zu einem physischen Nachbarn des virtuellen Nachbarn.

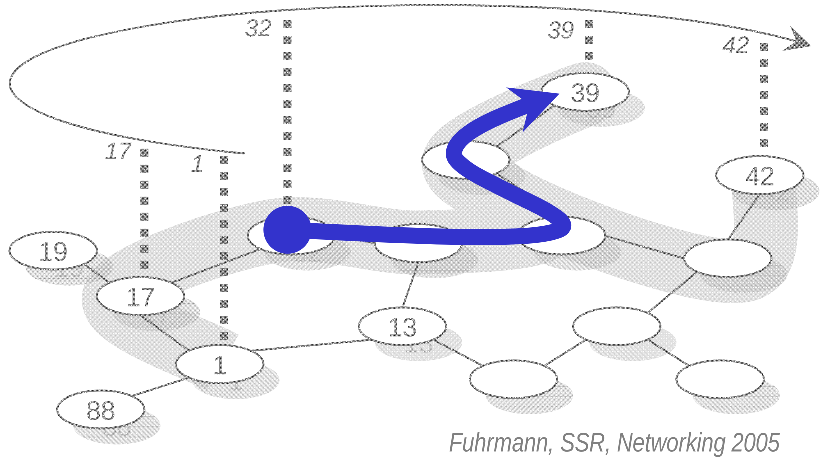


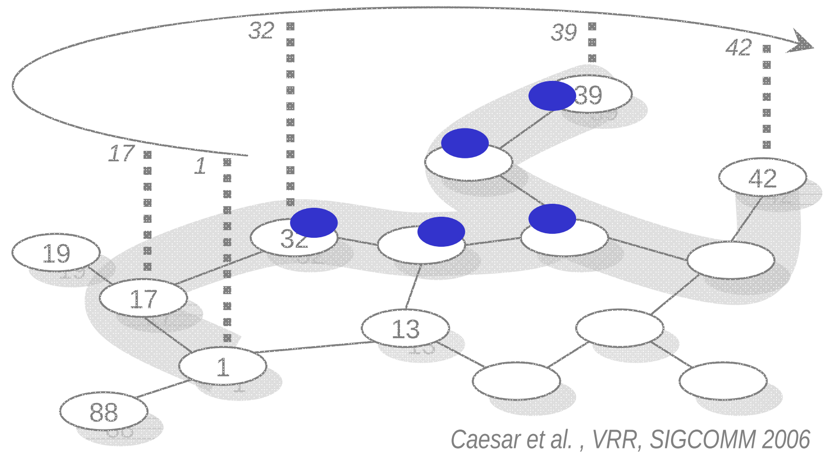
Simulationsergebnisse (2) – Node Churn



8000 Knoten,
Knotenlebensdauer: Poisson-Prozess

Vergleich SSR und VRR





- Source Routen
- Zustand pro Knoten beschränkt
- Große Paketköpfe durch Source Routen (Reduktion auf lokale Interface IDs mögl.)
- Konsistenz ohne Fluten (neues Verfahren, früher ISPRP vgl. VRR Rep.)

- Tabellen mit Pfadeinträgen
- Zustand pro Knoten potentiell unbeschränkt
- Kleine Paketköpfe
- Konsistenz durch Fluten von Repräsentanten IDs

Vergleichssimulationen zeigen leichten Vorteil von SSR solange Bandbreite nicht saturiert.
Zustandbeschränkung bei SSR könnte für Realisierung in Hardware vorteilhaft sein.

Questions?

Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de