Karlsruhe Institute of Technology

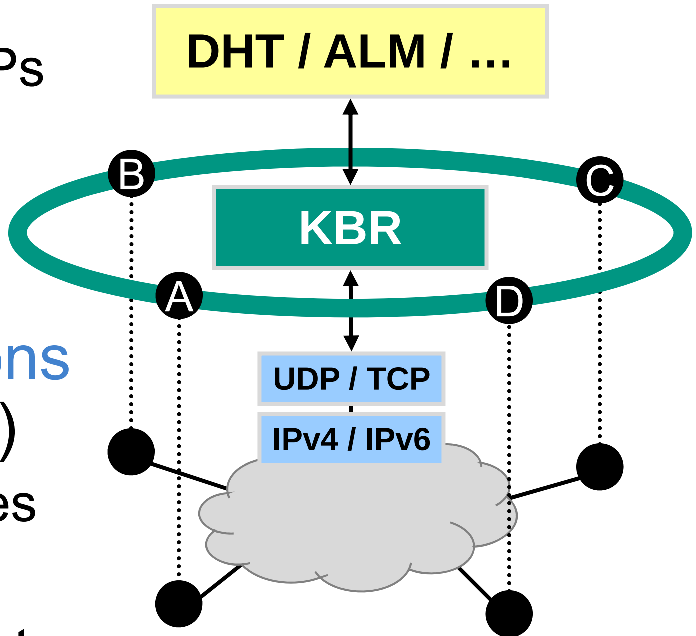
R/Kademlia: Recursive and Topology-aware Overlay Routing

Bernhard Heep
ATNAC 2010, Auckland, New Zealand, 11/02/2010

Institute of Telematics, Department of Computer Sciences

Motivation

- Structured P2P overlays like Kademlia [1] offer key-based routing (KBR) service
  - Messages are sent to keys instead of IPs
  - No central server needed
  - O(log N) routing hops per message



- Kademlia used by popular applications in the Internet (eMule, BitTorrent, …)
  - Could be basis for various other services

➔ Kademlia is scalable and robust, but…

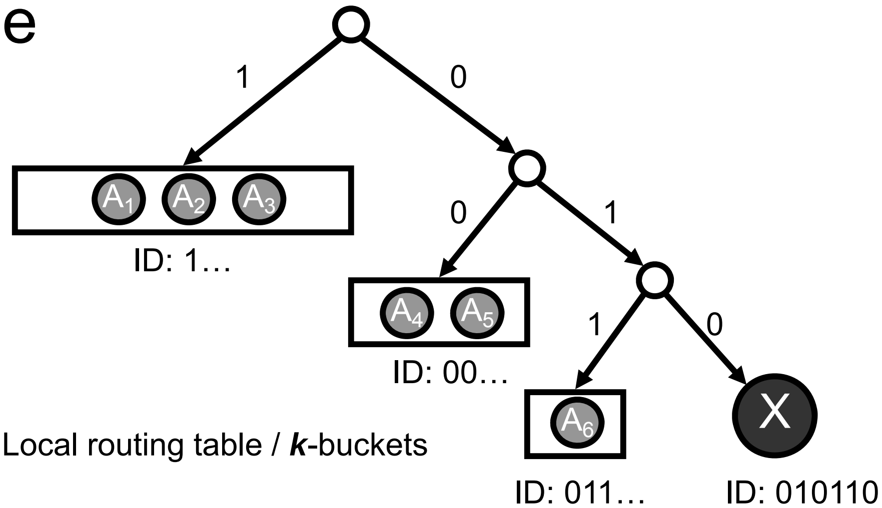
➔ Applications suffer from high routing latencies and problems with NAT/PAT gateways

Agenda

- Motivation
- The original Kademlia protocol
- Analysis
  - Key-based routing under churn
  - NAT / PAT
- R/Kademlia
  - Signaling modes
  - Topology adaptation
- Related work
- Implementation, simulation setup and evaluation
- Summary and outlook

Kademlia

- ***k*-buckets as local routing table**
  - Binary tree, ***k***-buckets as leafs
  - All nodes in one bucket share a common nodeId prefix with the local node
- Routing metric $\boldsymbol{d_{XOR}}$



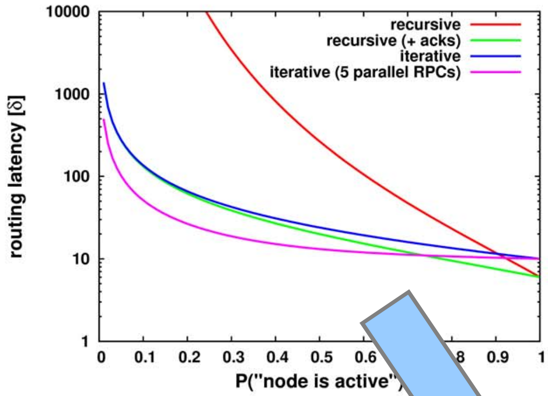
Local routing table / ***k***-buckets

- *Iterative lookups* to find close nodes to a destination key ***y***
  - Lookup initiator sends ***v*** parallel FIND_NODE requests to close nodes to ***y*** from the local ***k***-buckets
  - Responses with closer nodes ➔ merged into result vector $\boldsymbol{V_y}$
  - Lookup terminates if no $\mathbf{A_i} \in \boldsymbol{V_y}$ knows closer nodes to ***y***

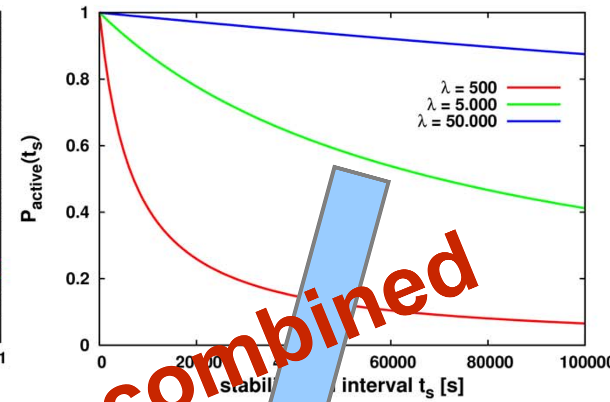
  ➔ New peers are met during lookups (FIND_NODE responses)

Analysis: Key-based Routing / Churn

- Analysis of routing modes under churn [2]
- Detected lifetime model in public KAD networks (Weibull distribution, λ = 5.000, k = 0.5) [3][4]

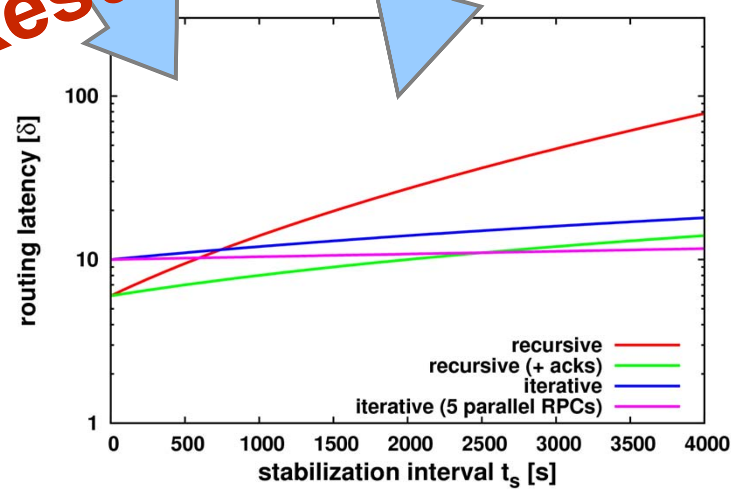


Routing latency depending on P("node is active"), min. 5 routing steps



P("node is active") depending on stabilization interval

Results combined



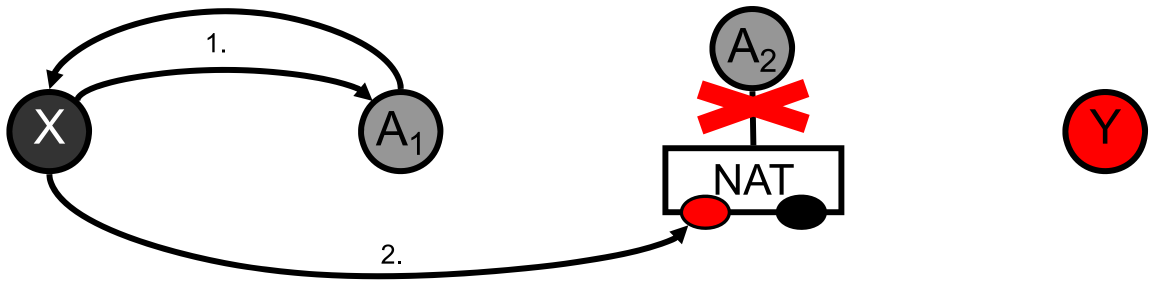
Routing latency depending on stabilization interval

➔ KAD churn: Up to a stabilization interval of $\mathbf{t_s} \approx 2{,}500$ s, recursive mode is superior to iterative mode

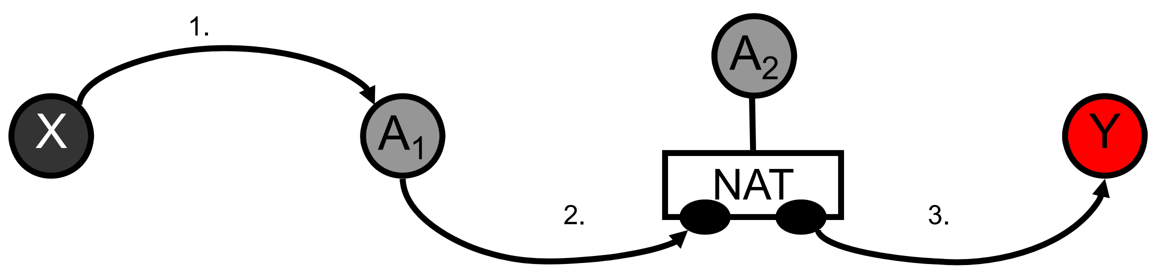
➔ Idea: Kademlia using recursive routing and decreasing $\mathbf{t_s}$ by exploiting application triggered routing traffic

Analysis: NAT / PAT

- During iterative lookups
  - Unknown nodes have to be contacted
  - Nodes might be inaccessible due to NAT/PAT gateways



- During recursive routing procedures
  - Only nodes from the local routing tables are contacted
  - Accessibility can be checked before



R/Kademlia: Demands

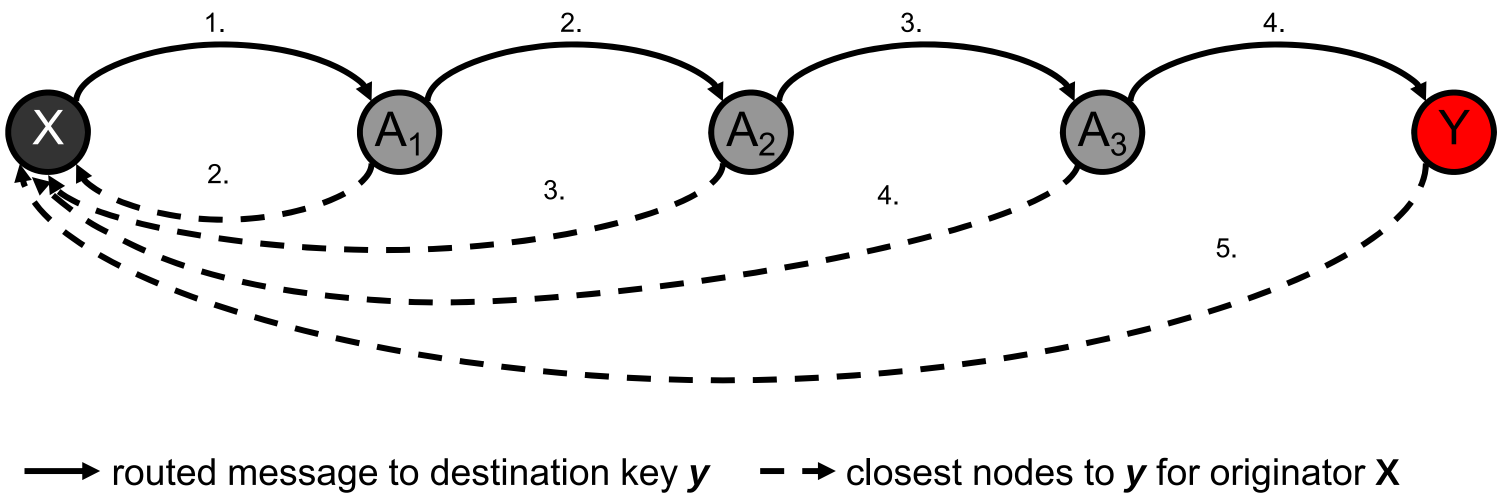
- Simple recursive routing and maintenance
  - Avoidance of connection problems in NAT/PAT scenarios
- Meeting new peers by application-triggered lookups
  - No periodic tasks needed
- Resilience against node failures
  - Hop-by-hop acknowledgements
  - Redundancy in routing tables
- Legacy support: Iterative lookups should still be supported
- Effective deployment of topology adaptation [5]
  - Proximity Neighbor Selection (PNS) and Proximity Routing (PR)
  - ➔ Low Key-based routing (KBR) latencies

R/Kademlia: Basic Operations

- Greedy recursive routing: Nodes on the routing path…
  - … forward a message to the closest node to the destination key ***y*** from the local ***k***-buckets according to $\boldsymbol{d_{XOR}}$
  - … use hop-by-hop acknowledgements
    - ➔ Failed nodes are removed from k-buckets
    - ➔ Messages are resent
- Meeting new peers by application-triggered routing procedures (like the original protocol)
  - Need for additional messages

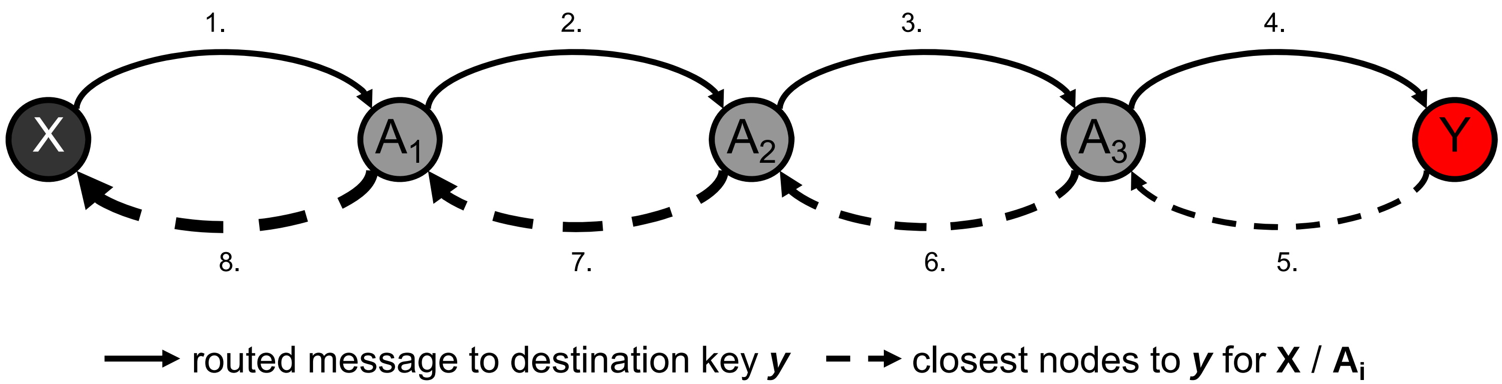
  ➔ 2 different signaling modes: *Direct* and *Source-routing*

Signaling: Direct Mode



- All nodes on routing path send back ***n*** nodes that are close to ***y*** back to the originator
  - ➔ Originator gets all information he would get in iterative mode

Signaling: Source-Routing Mode



- All nodes on routing path send back ***n*** nodes that are close to ***y*** back to last hop on routing path
  - ➔ Other nodes on routing path meet new peers and additionally merge their peers into signaling message
  - ➔ Originator gets all information he would get in iterative mode
  - ➔ Only mutually known peers communicate

PNS and PR for R/Kademlia

- Proximity Routing (PR)
  - Routing metric $\boldsymbol{d_{XOR}}$ replaced by $\boldsymbol{d_{KadPR} = d_{prefix} + d_{prox}}$

$$d_{\text{prefix}}(X,Y) = \begin{cases} 0 & , X_i = Y_i \, \forall \, 0 \leq i < m \\ m-n & , \exists n : X_i = Y_i, X_{n+1} \neq Y_{n+1} \\ & \;\forall \, 0 \leq i \leq n < m \end{cases}$$

$$d_{\text{prox}}(X,Y) \in [0;1) \qquad d_{\text{prefix}}(X,Y) \in [0;m] \subset \mathbb{N}$$

  - $\boldsymbol{d_{prox}}$ is calculated from measured or estimated RTT
  - ➔ Next hop $\mathbf{A_{i+1}}$ is the physically closest to the current node $\mathbf{A_i}$ of those nodes that share the longest common nodeId prefix with ***y***
- Proximity Neighbor Selection (PNS)
  - LRU-strategy used for buckets replaced
  - ➔ now filling up ***k***-buckets with ***k*** physically closest nodes that are met
    - Nodes must be probed to detect their proximity

Institute of Telematics
Department of Computer Sciences

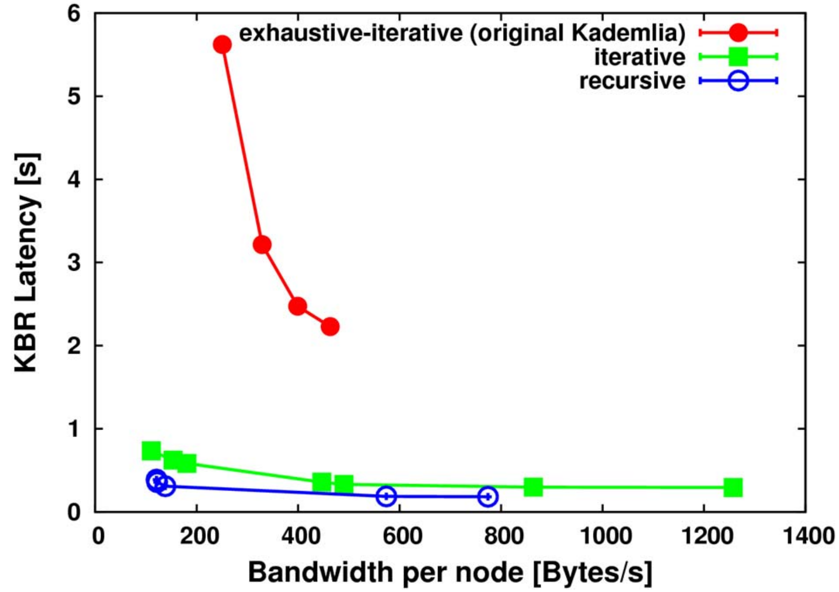
Related Work

- Rhea et al.: Bamboo [6]
  - Designed for high routing performance in dynamic networks
  - Recursive routing with PNS
  - New peers are only met by periodic tasks
  - Uses two different metrics / routing tables
  - Limited redundancy in the routing tables
    - ➔ Only one node per routing table entry is effected by PNS
- Kaune et al.: “Proximity in Kademlia” [7]
  - Noticeable decrease of routing latencies due to PNS
  - Only iterative lookups

Implementation & Simulation Setup

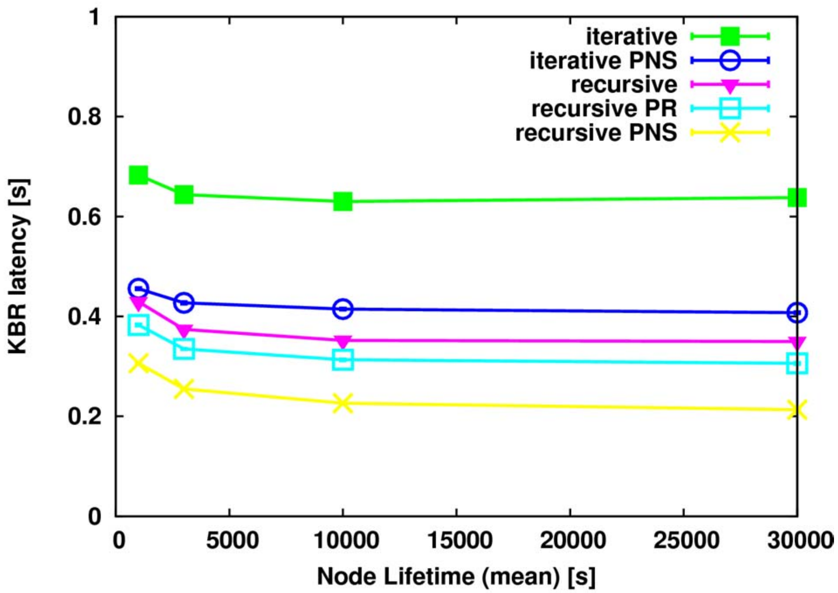
- R/Kademlia integrated into overlay framework OverSim [8]
  - Extension of available Kademlia implementation: recursive routing, PR, PNS
- Simulation:
  - Varied Parameters
    - Routing mode
    - PR, PNS, active probing
    - Signaling mode (rec.)
    - Number of parallel RPCs (in iterative mode)
  - 5000 nodes, 20 random seeds, 2h measurement time
  - Churn: weibull-distributed lifetime model [7][8]
    - Mean lifetime varied between 1,000 – 30,000 s
  - Test application: Every 60s RPCs to random nodeIds
- Evaluation with Performance vs. Cost framework (PVC) [9]

Evaluation: Comparison

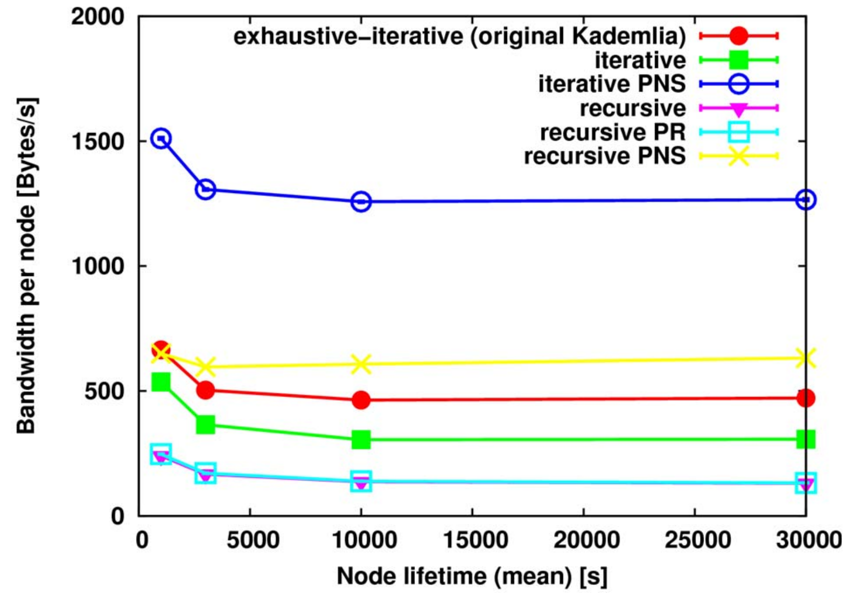


- PVC: Convex hulls of the original protocol, simple iterative mode and R/Kademlia (under KAD churn)
  - Original Kademlia cannot compete due to high bandwidth demands and high routing latencies
  - In all configurations, R/Kademlia achieves best performance/cost trade-off

Evaluation: Latency / Bandwidth



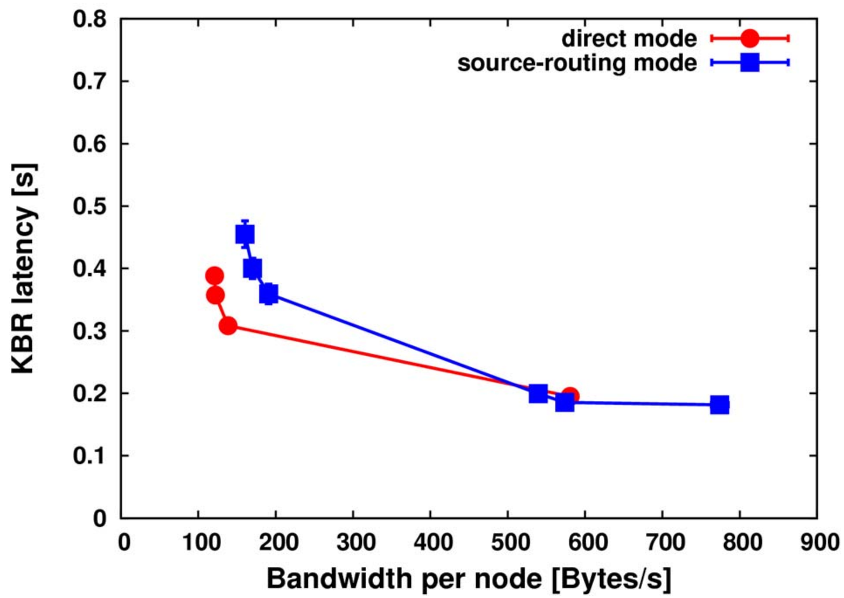
KBR routing latencies (RPCs to random destination keys)



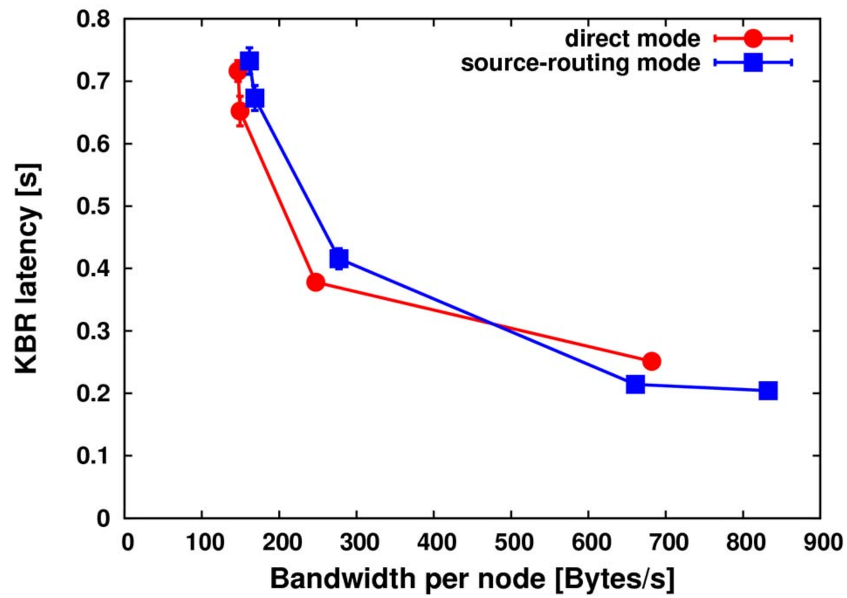
Bandwidth consumption

- R/Kademlia and R/Kademlia/PR achieve smallest bandwidth consumption
- R/Kademlia/PNS achieves lowest routing latencies under all churn rates
- Iterative mode shows average results
  - Low latencies, but high traffic with PNS enabled
- The original protocol has very high latencies (2-5 s) and average bandwidth consumption

Evaluation: Signaling Modes



Lifetime mean = 10,000 s

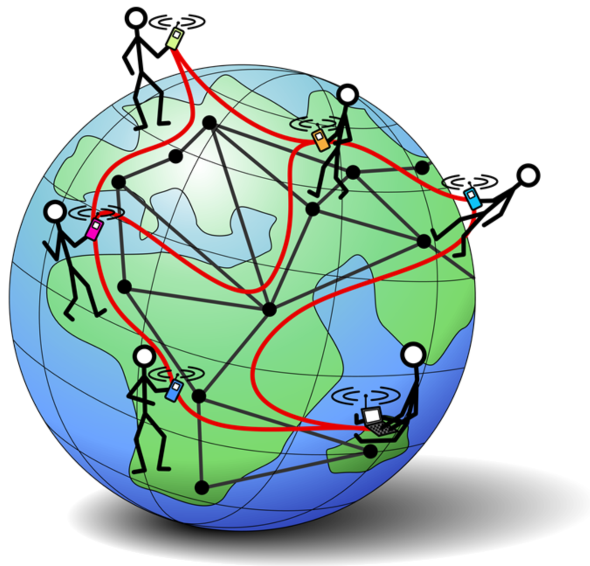


Lifetime mean = 1,000 s

- Comparison of both signaling modes under different churn rates (PVC convex hulls)
  - *Direct mode* has less bandwidth needs under moderate churn
  - *Source-routing mode* achieves lower routing latencies in high churn scenarios

Conclusion & Future Work

- Summary
  - R/Kademlia achieves better routing performance than the original
  - PR and PNS can be effectively applied
  - Different signaling modes for NAT/PAT compatibility
  - Iterative mode is still supported
  - Source code available at ***http://www.oversim.org/***



- Future Work
  - Comparison to other protocols like Bamboo
  - Evaluation in real networks and testbeds like PlanetLab and G-Lab
  - Usage of Topology-based NodeId Assignment

References

[1] Maymounkov and Mazières, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” in *Peer-to-Peer Systems: First International Workshop (IPTPS 2002)*, 2002

[2] Wu et al., “Analytical Study on Improving DHT Lookup Performance under Churn,” in *P2P ’06: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, 2006

[3] Stutzbach et al., “Understanding churn in peer-to-peer networks”, in *IMC’06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurements*, 2006

[4] Steiner et al., “Long Term Study of Peer Behavior in the KAD DHT”, in *IEEE/ACM Transaction on Networking*, 2009

[5] Castro et al., “Exploiting network proximity in distributed hash tables,” in *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, 2002

[6] Rhea et al., “Handling Churn in a DHT,” in *ATEC ’04: Proceedings of the annual conference on USENIX Annual Technical Conference*, 2004

[7] Kaune et al., “Embracing the Peer Next Door: Proximity in Kademlia,” in *Proceedings of the Eighth International Conference on Peer-to-Peer Computing (P2P’08)*, 2008

[8] Baumgart, Heep, and Krause, “OverSim: A flexible overlay network simulation framework”, in *Proceedings of 10th IEEE Global Internet Symposium (GI’07) in conj. with IEEE INFOCOM*, 2007

[9] Li et al., “A performance vs. cost framework for evaluating DHT design tradeoffs under churn,” in *INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005

… the end!

Thank you!

Any Questions?

http://telematics.tm.kit.edu/
http://www.oversim.org/