R/Kademlia: Recursive and Topology-aware Overlay Routing

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

- **k**-buckets as local routing table
  - Binary tree, **k**-buckets as leafs
  - All nodes in one bucket share a common node id prefix with the local node

- Routing metric **d**\(_{XOR}\)

- **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 \(\rightarrow\) merged into result vector \(V_y\)
  - Lookup terminates if no \(A_i \in V_y\) knows closer nodes to **y**
  \(\rightarrow\) 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, $\lambda = 5.000, k = 0.5$) [3][4]

- KAD churn: Up to a stabilization interval of $t_s \approx 2,500$ s, recursive mode is superior to iterative mode
- Idea: Kademlia using recursive routing and decreasing $t_s$ 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 $d_{XOR}$
  - ... 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 \( X \).
  - 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 $d_{XOR}$ replaced by $d_{KadPR} = d_{prefix} + d_{prox}$

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

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

- $d_{prox}$ is calculated from measured or estimated RTT
- Next hop $A_{i+1}$ is the physically closest to the current node $A_i$ 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
Related Work

  - 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
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

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


Thank you!

Any Questions?

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