# Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks

Himabindu Pucha, Saumitra Das, Y. Charlie Hu

Distributed Systems and Networking Lab,
School of ECE, Purdue University

WMCSA 2004
6th IEEE Workshop on Mobile Computing Systems & Applications

PURDUE UNIVERSITY

# Mobile ad hoc networks (MANETs)

- Wireless networks in which wireless hosts act as forwarding nodes as well as end systems
- No base station or routing infrastructure
- Network topology changes frequently and unpredictably
- Challenge lies in routing packets with changing topology while minimizing overhead
  - Specialized routing protocols: DSR, AODV

# Structured peer-to-peer routing protocols in the Internet

- Operate in an overlay p2p network in which nodes act as clients as well as servers
- Rely on underlying Internet infrastructure to route packets between overlay hops
- Implement a DHT in a scalable, robust manner
- Challenge lies in routing packets in a network with changing membership while limiting state at each node
  - Specialized protocols: Pastry, Tapestry, Chord, CAN

# Motivation

- DHTs provide a useful platform for building scalable and robust distributed applications in the Internet
- DHTs can potentially provide an efficient way to construct distributed applications and services in MANETs
  - Applications such as file sharing, resource discovery could benefit from the insert/lookup convergence
- Main Challenge
  - Provide an efficient DHT abstraction in a MANET and demonstrate its usability

# Outline

- How to support an efficient DHT abstraction in an ad hoc environment?
  - Can off-the-shelf protocols be used?
  - If not, what is an efficient architecture to provide the DHT abstraction?
- How should the DHT abstraction be used in an ad hoc environment ?
  - Can a MANET application benefit from the DHT?

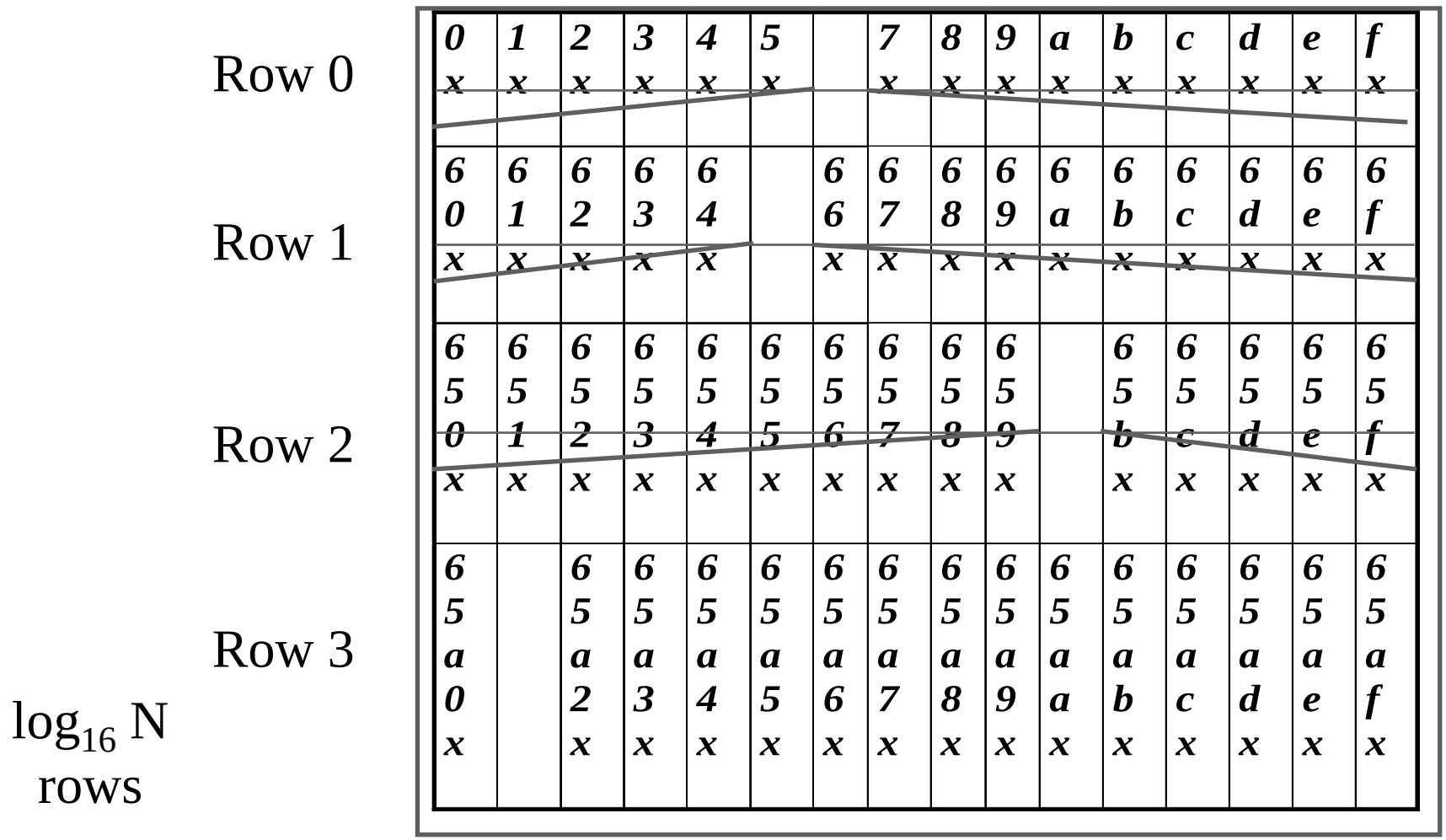
# Implementing DHT: Layered Approach

- Layer Pastry (structured p2p protocol) on top of DSR (MANET routing protocol)
- Pastry operates in the application layer similar to the Internet
- DSR used as underlying routing protocol

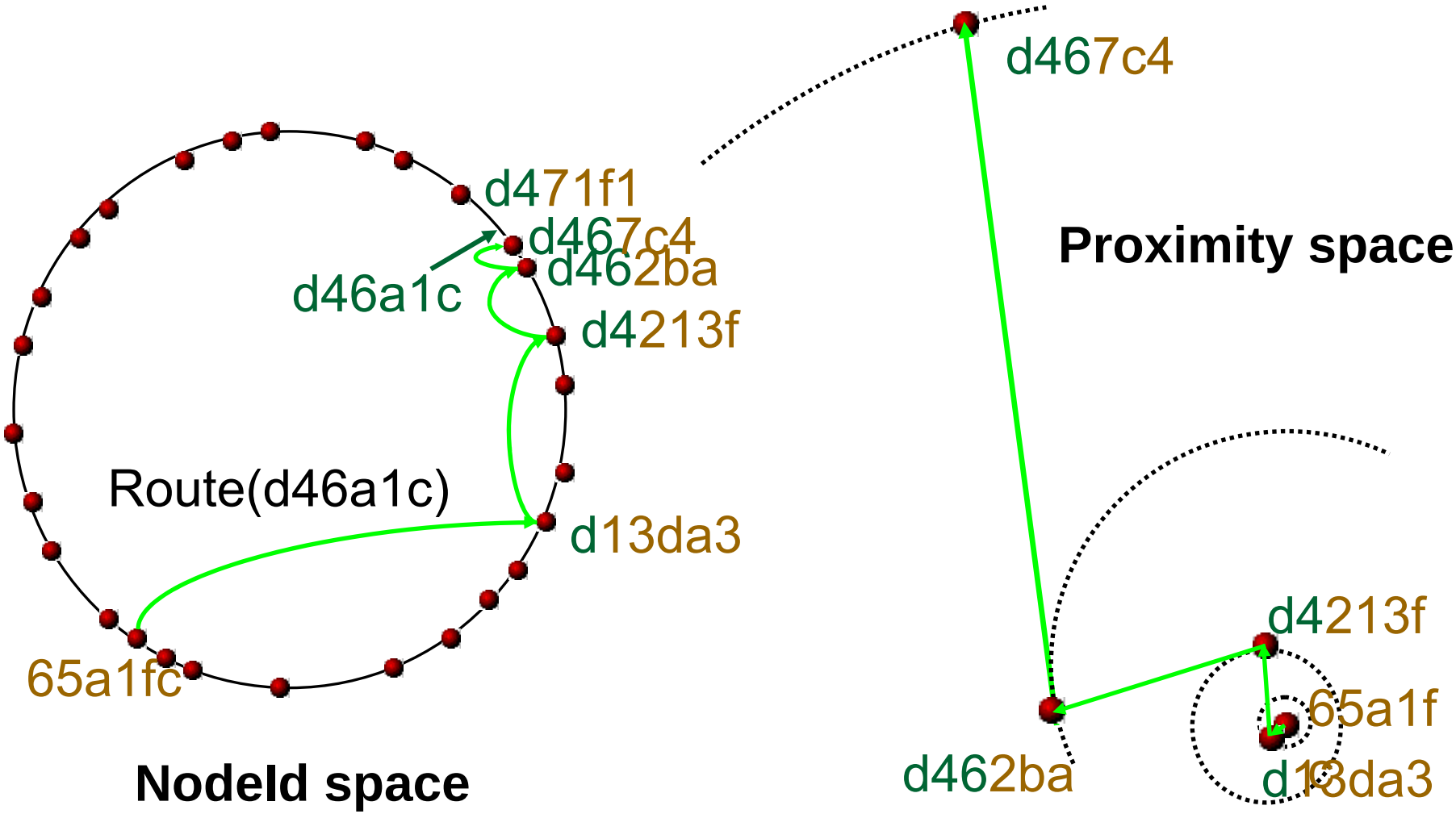
# Pastry : Features

- Nodes have unique Id, messages have keys
  - Typically 128 bits long
- Primitive: **`Route(msg, key)`**
  - Delivers **`msg`** to the currently alive node whose Id is numerically closest to **`key`**
- Scalable, efficient
  - Per node routing table contains O(log(N)) entries
  - Routes in O(log(N)) steps
- Fault tolerant
  - Self-fixes routing tables when nodes are added, deleted or fail

# Pastry : Routing table (# 65a1fcx)

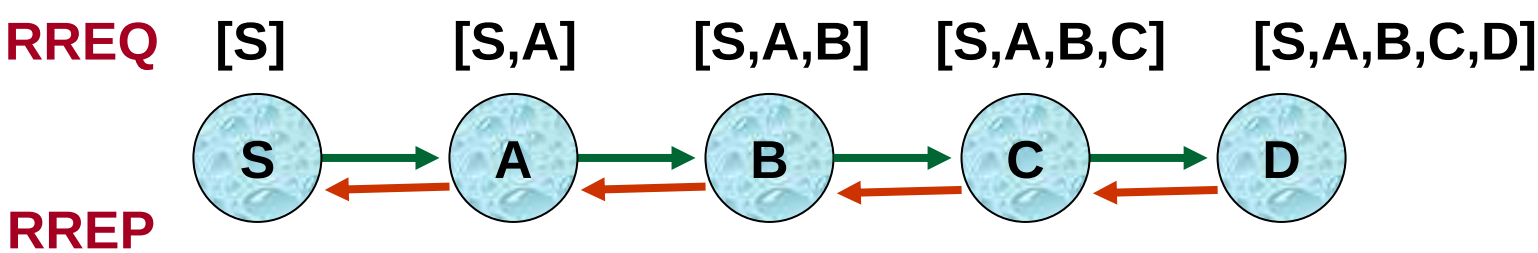


# Pastry: Routing Example



# DSR

- Reactive routing protocol based on source routing
- Operation: route discovery and maintenance
- Route caching: path and link cache



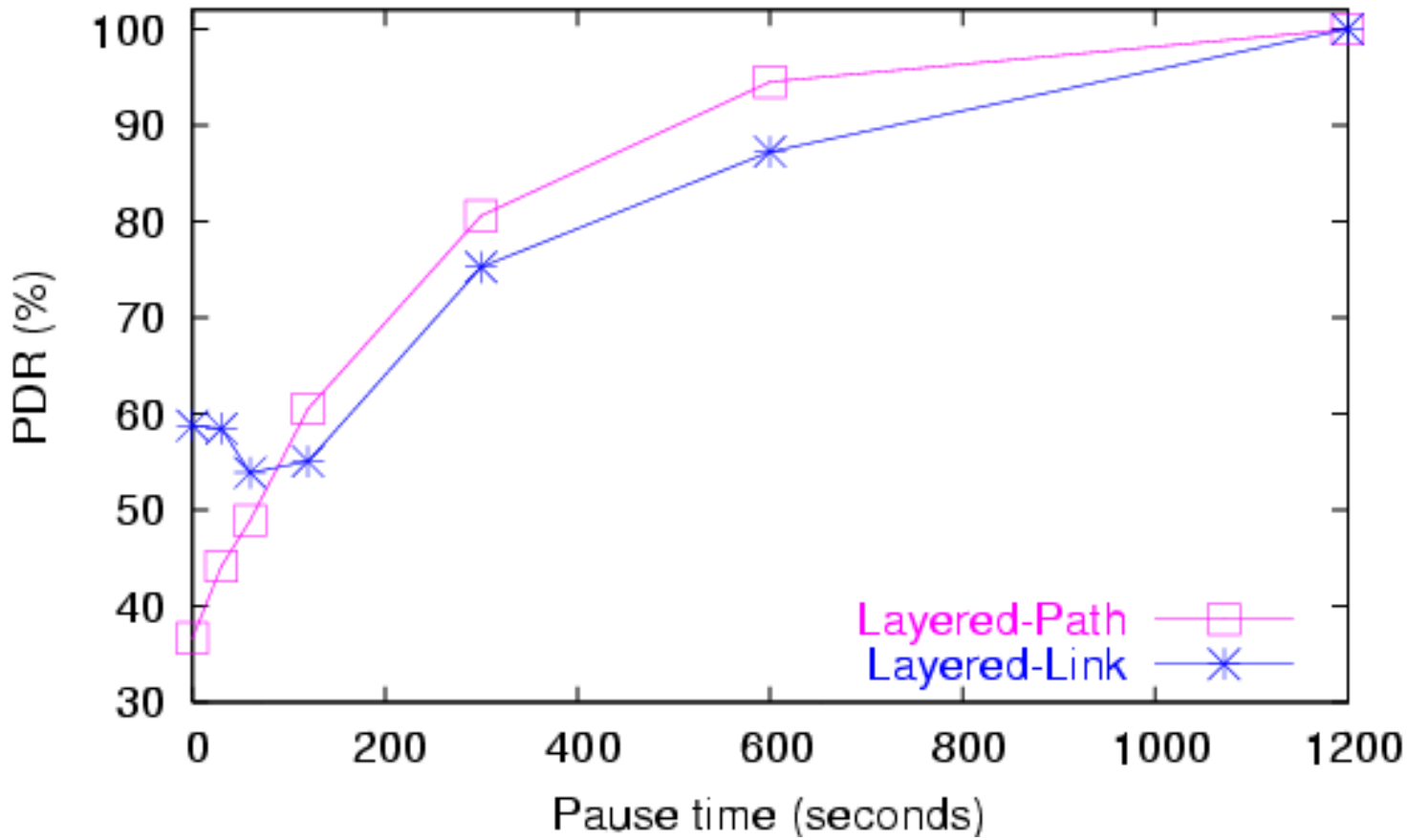
# Layered approach

- Direct layering is not practical
- Pastry uses ‘ping’ to measure delay for proximity
  - Proximity of Pastry routing tables uses hop count instead of delay from ‘ping’
  - DSR exports API to answer proximity probes from Pastry using its route cache
- Expanding ring search for bootstrap node

# Simulation setup

- Pastry implemented in ns-2 on top of DSR
- Parameters: 50 nodes, 1500mx300m, 2Mbps, 250m, 1-19m/s
- Traffic: 40 sources, 3 pkts/sec, random keys generated
- Metrics
  - Packet delivery ratio (PDR): ratio of successfully and correctly delivered packets to packets sent
  - Overhead
  - Delay

# Layered approach performance



# Problems with layered approach

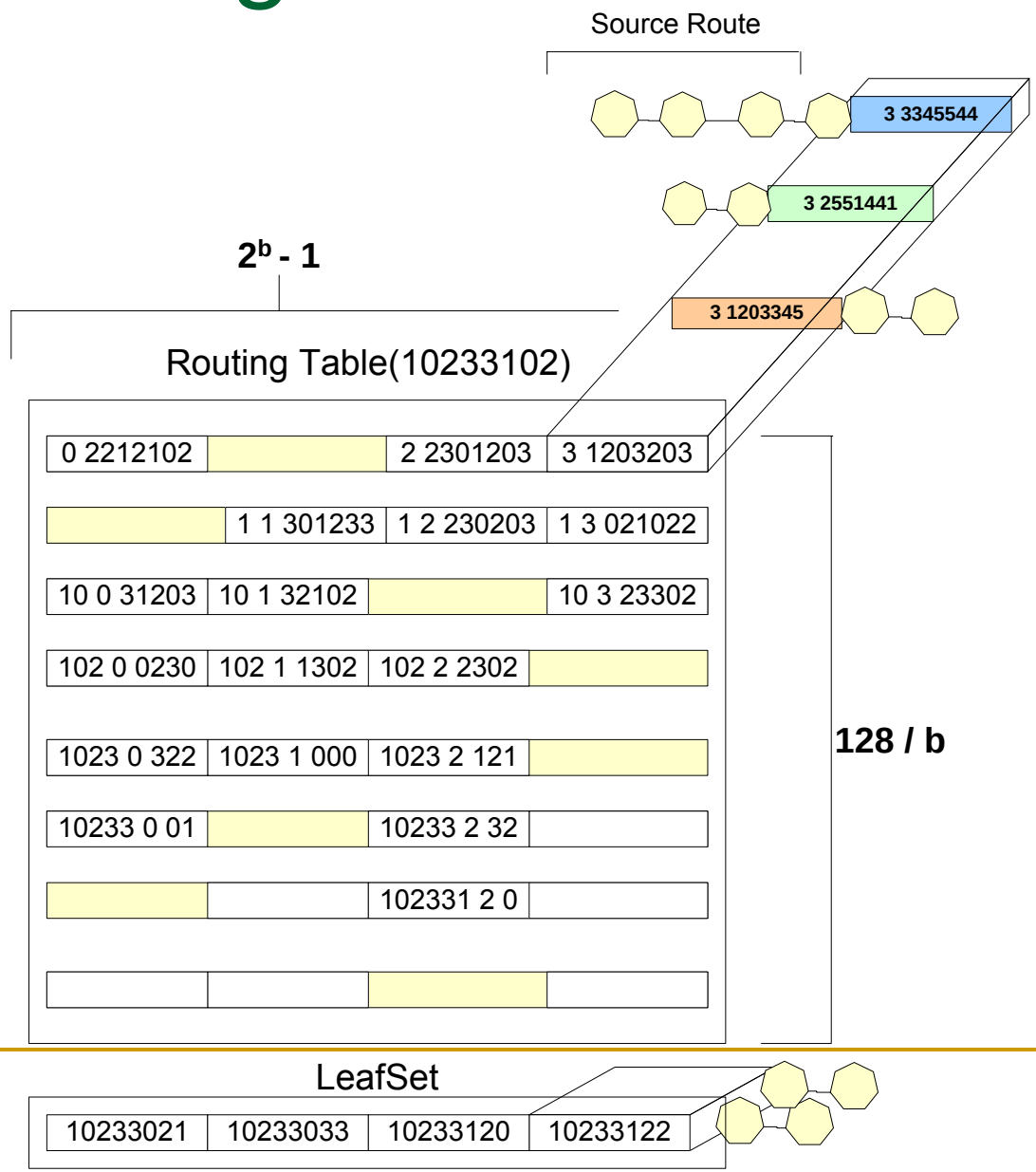
- High overhead
  - Periodic maintenance, proximity probing
- Choice of next logical hop independent of DSR
- Stale proximity information with Pastry

- Mismatch between routing state of Pastry and DSR

# Implementing DHT: Integrated Approach

- Integrates Pastry and DSR
- A unified DHT substrate at the network layer
- Referred to as *Ekta* (unity)

# Ekta: Routing



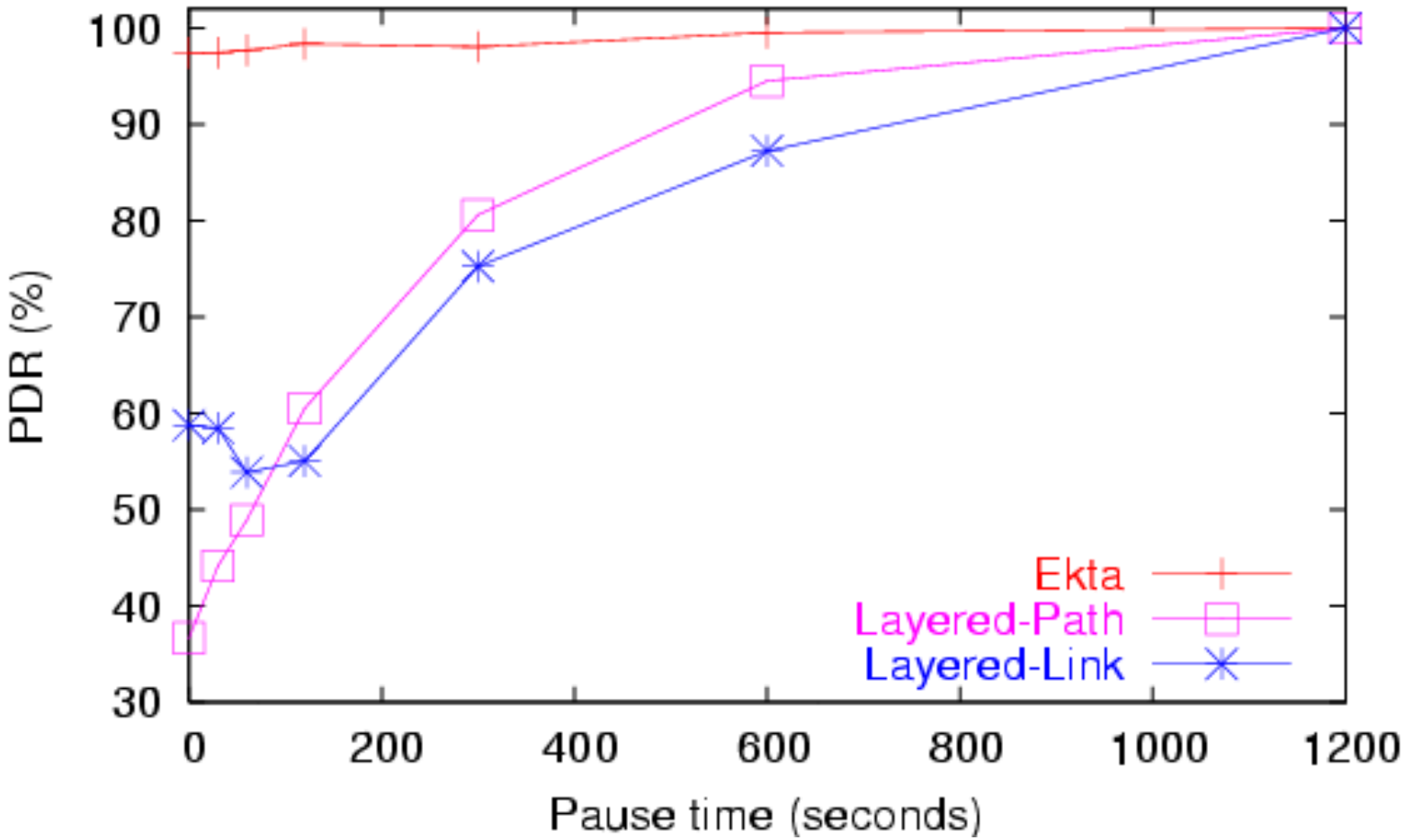
# Ekta: Join and leave

- Join
  - Flood “JOIN REQUEST”
  - Potential leafset members send “JOIN ACK”
  - Node closest sends “JOIN COMPLETE”
- Graceful leave
  - Flood “LEAVE”
  - Leafset members send “LEAVE ACK”
  - Exchange leafsets
- Node failure
  - Reactive failure handling
  - Node sends “proxy leave”

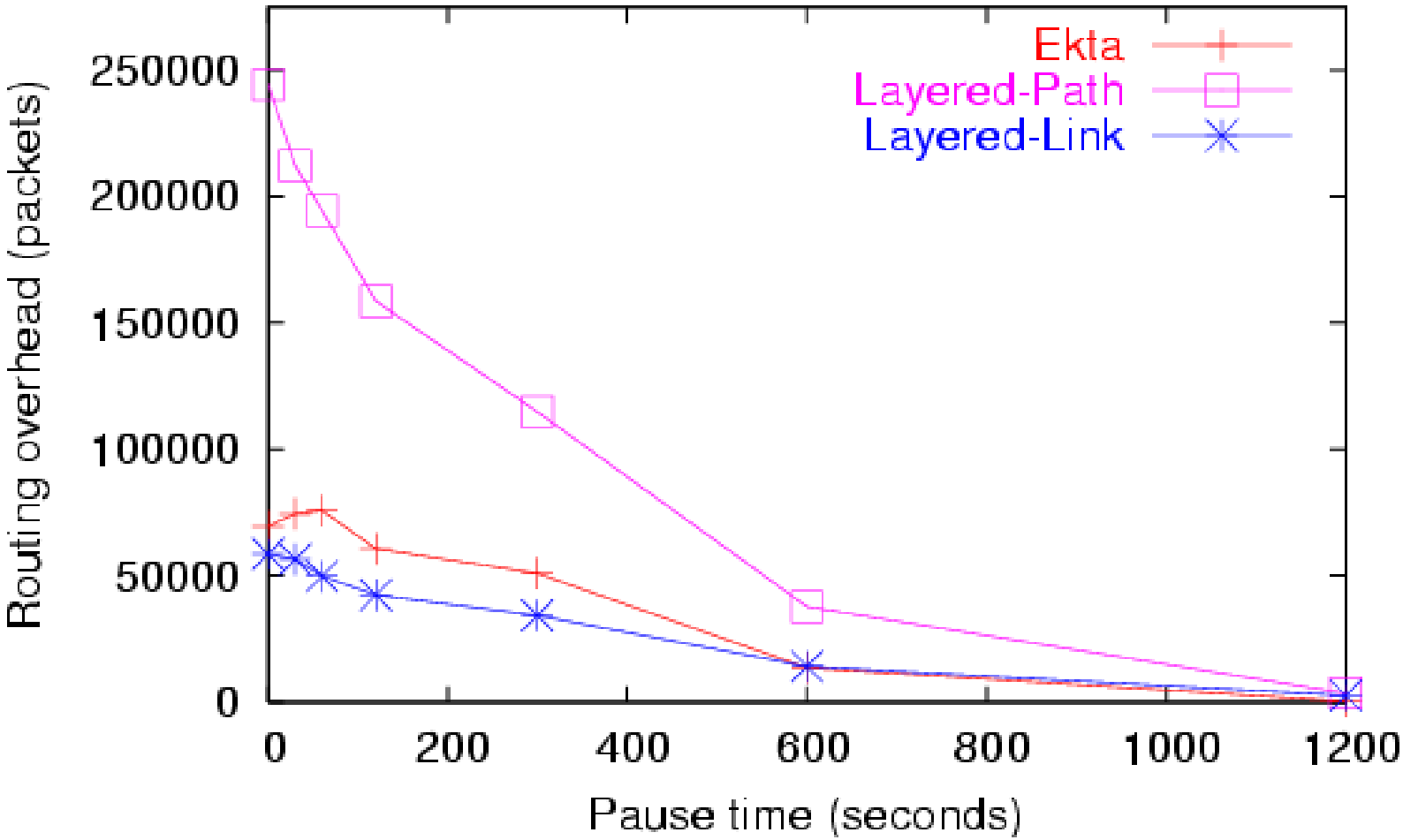
# Ekta: Optimizations

- Prefix based route requests
- Routes updated using snooping and overhearing

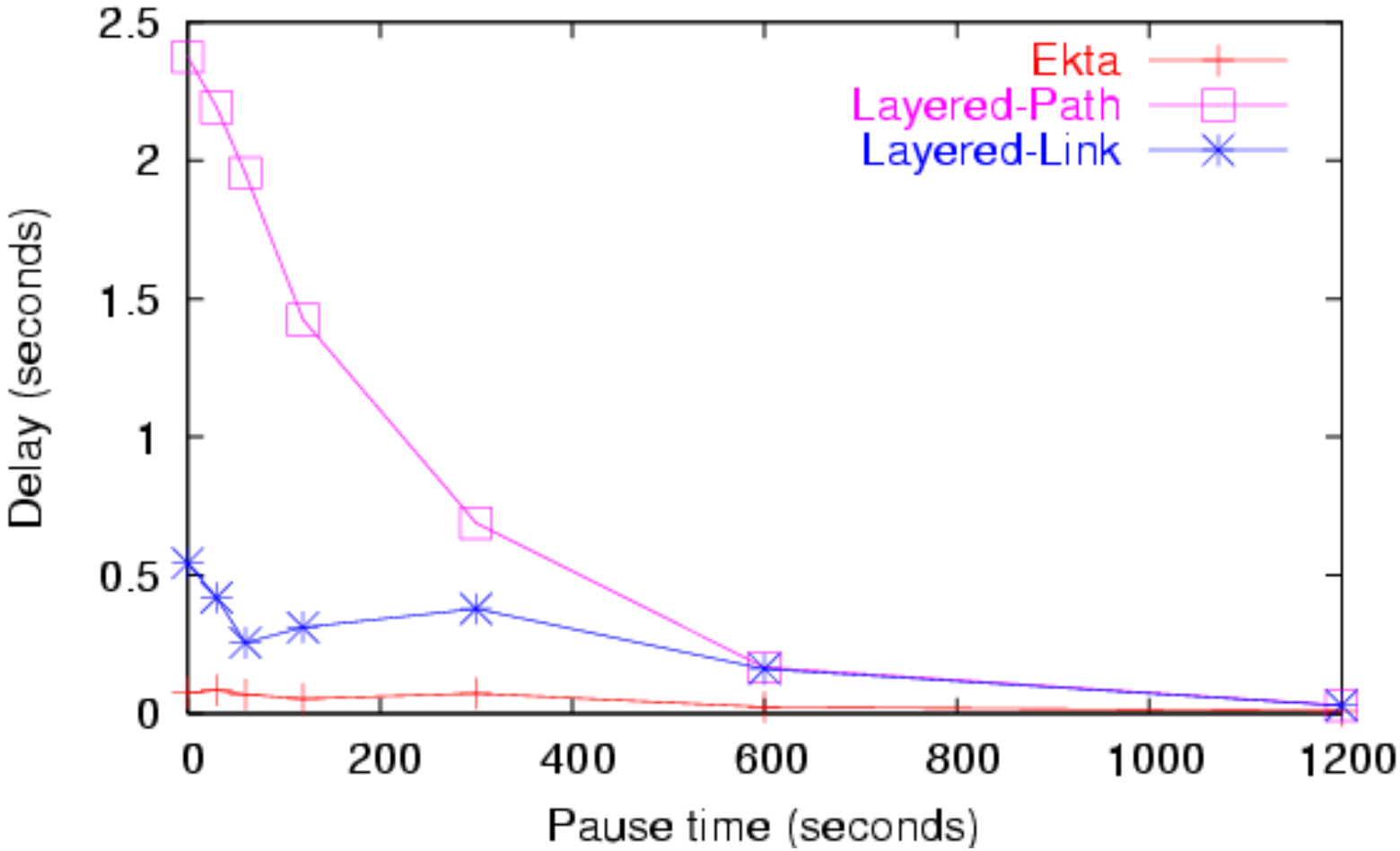
# PDR


100
90
80
70
60
50
40
30
PDR (%)
0
200
400
600
800
1000
1200
Pause time (seconds)
Ekta
Layered-Path
Layered-Link

# Overhead


Ekta
Layered-Path
Layered-Link
Routing overhead (packets)
250000
200000
150000
100000
50000
0
0
200
400
600
800
1000
1200
Pause time (seconds)

# Delay


Ekta
Layered-Path
Layered-Link
Delay (seconds)
Pause time (seconds)
2.5
2
1.5
1
0.5
0
0
200
400
600
800
1000
1200

# Summary

- Integrated approach is an efficient architecture for implementing a DHT in MANETs
- Ekta is superior
  - No proximity probing and periodic maintenance
  - Better coordination between routes available in routing table and choice of logical hops
  - Prefix route requests
  - Fresher proximity information from snooping and overhearing

# Can a MANET application benefit from the DHT?

Resource discovery using Ekta

# Problem

- Nodes in a MANET possess heterogeneous capabilities and resources
- Cooperative resource sharing is useful in MANETs
  - Requires resource discovery
- Two schemes
  - Ekta-RD
  - DSR-RD

# Ekta-RD vs. DSR-RD

- Simulations in ns-2 of both protocols
- Number of unique resources = number of nodes
- Each resource replicated on average on 10% of nodes
- Traffic: Poisson arrival of resource requests at each node, each request chooses random resource, varying λ
- Metrics
  - Success ratio: resource requests successfully satisfied
  - Overhead: control overhead for routing

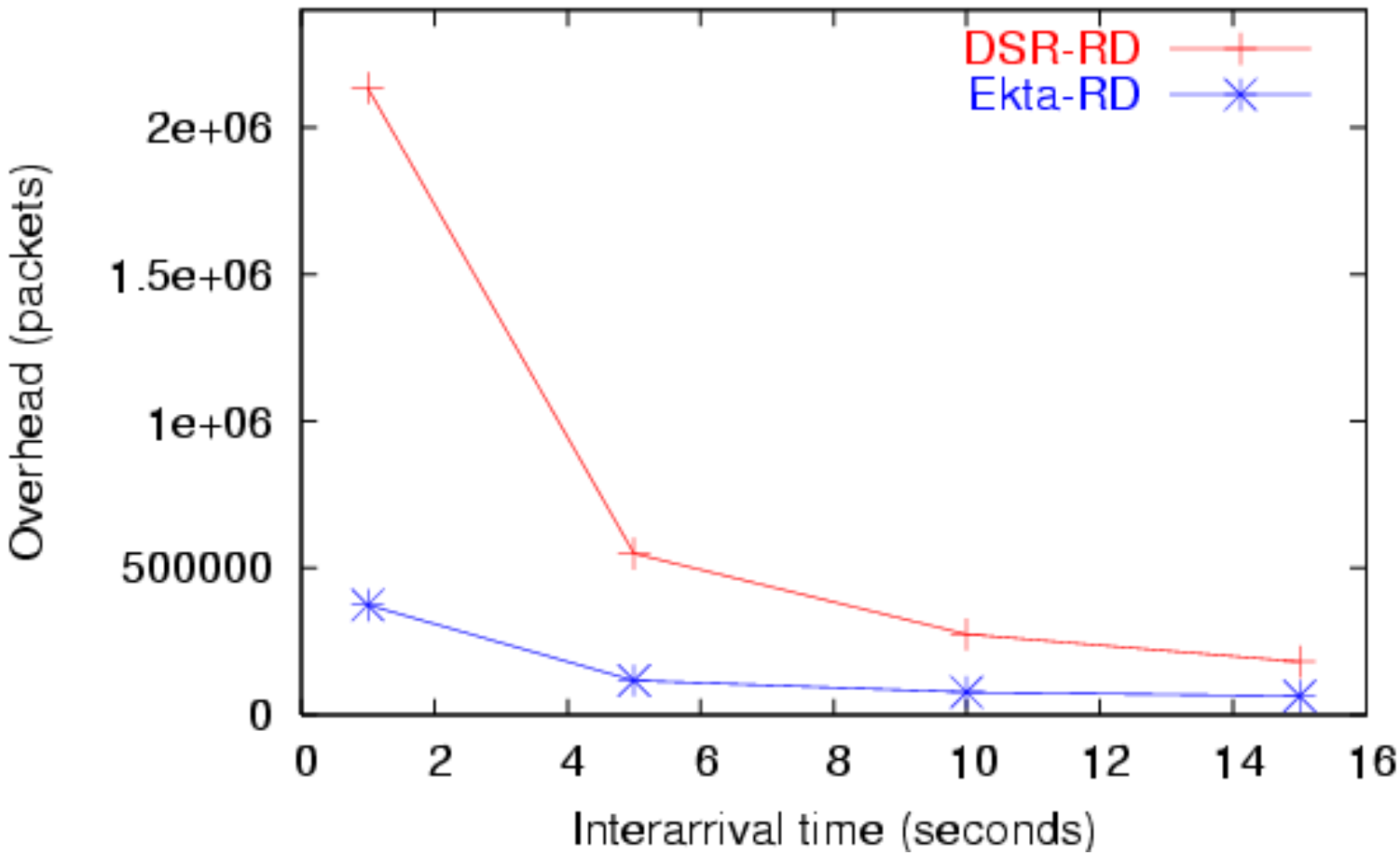
# Overhead analysis

- N = network size
- λ = average num of resource requests/node
- q = average degree of replication of any resource
- P = average hops between 2 nodes
- Pb = probability a route is not cached or cached but stale

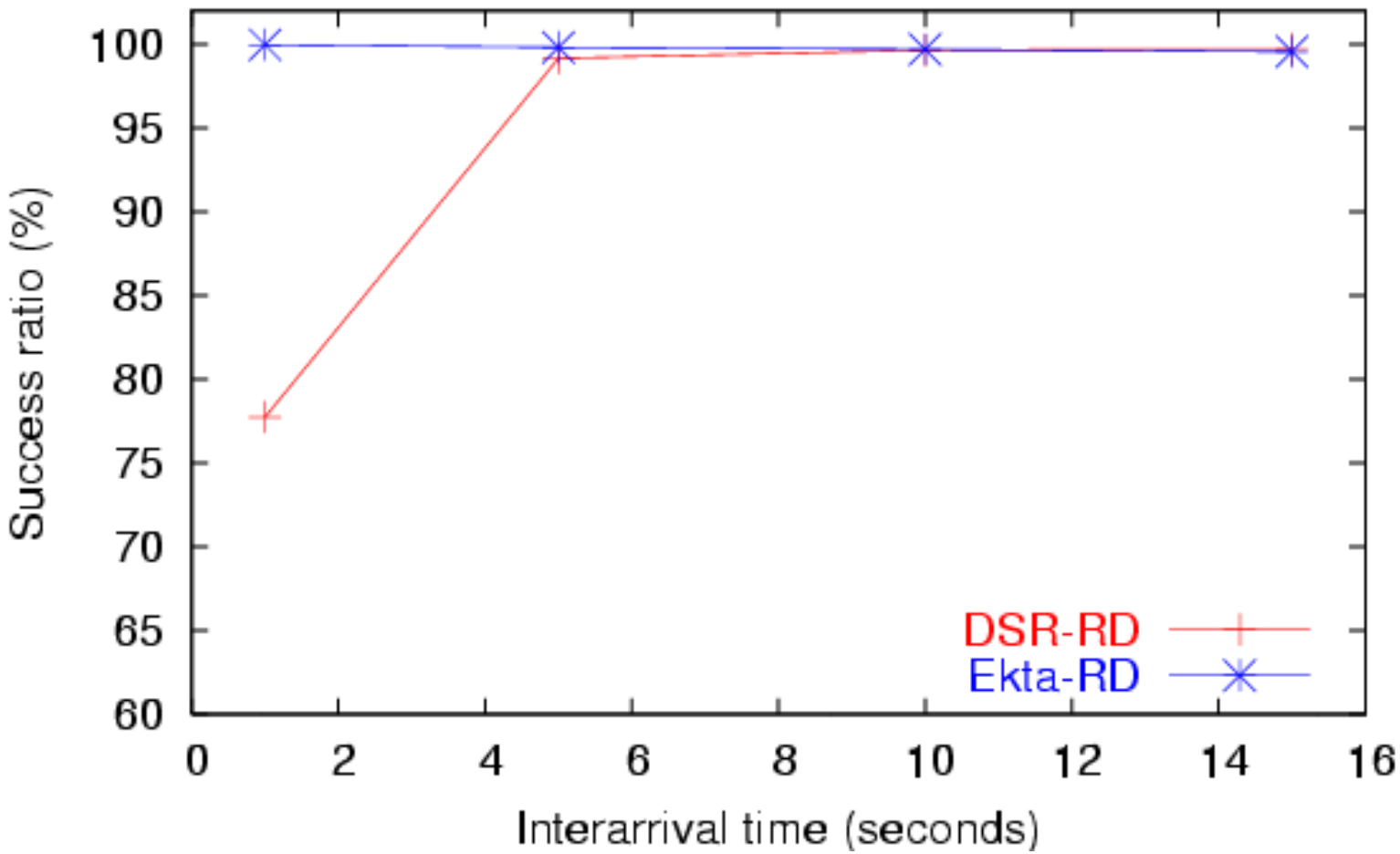
- Overhead (DSR-RD) = $\lambda . N^2 + \lambda . q. N^2 . P$
  - Independent of mobility
  - Grows as $O(N^2)$
  - Decreases with increasing λ

- Overhead (Ekta-RD) = $N ( \lambda . \log_{2^b} N . P + \lambda . P ) + N^2. Pb . \lambda (\log_{2^b} N + 1)$
  - Increases with mobility
  - Grows as $O(N . \log_{2^b} N )$
  - Decreases with increasing λ

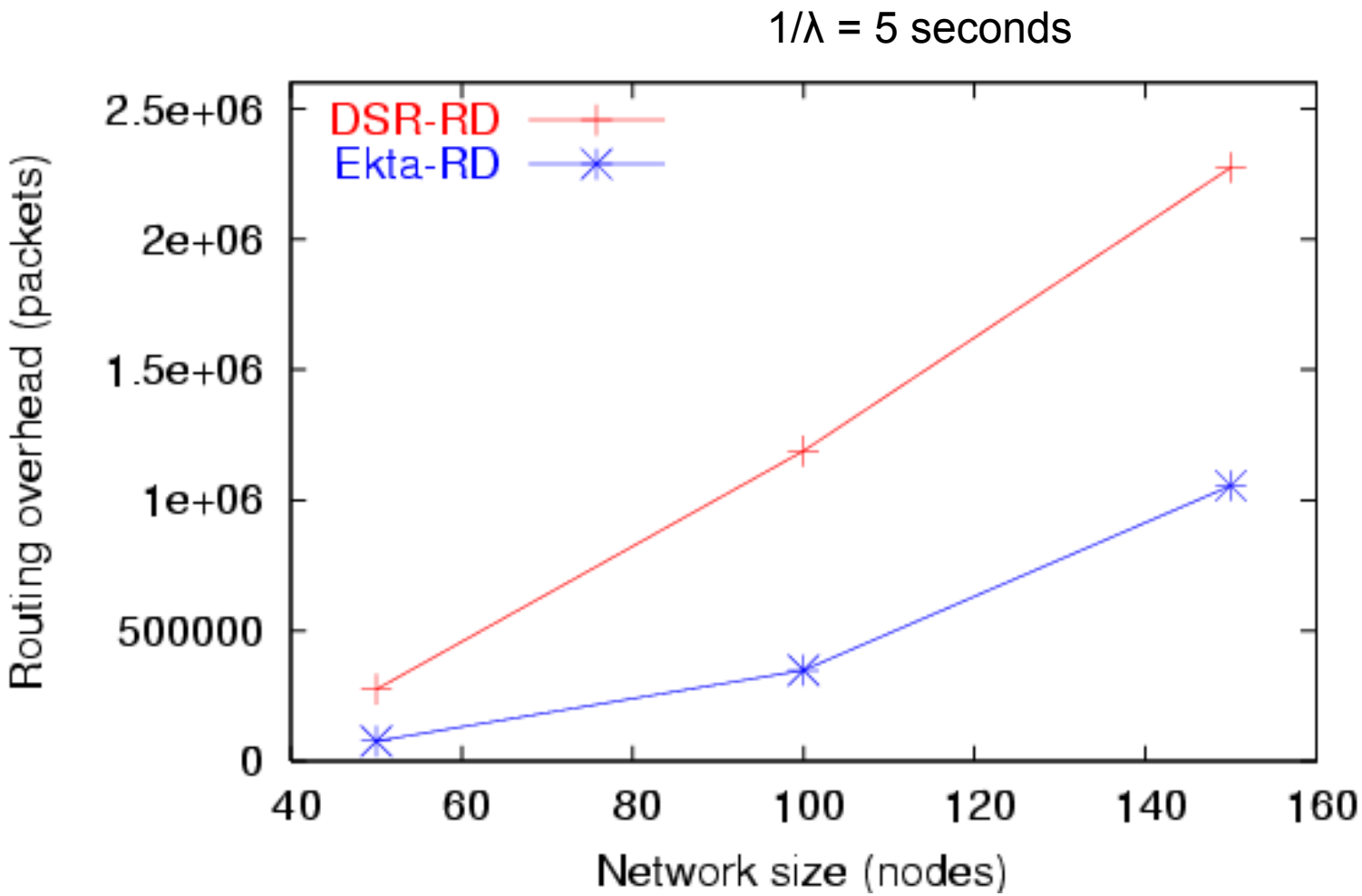
# Overhead: Varying λ



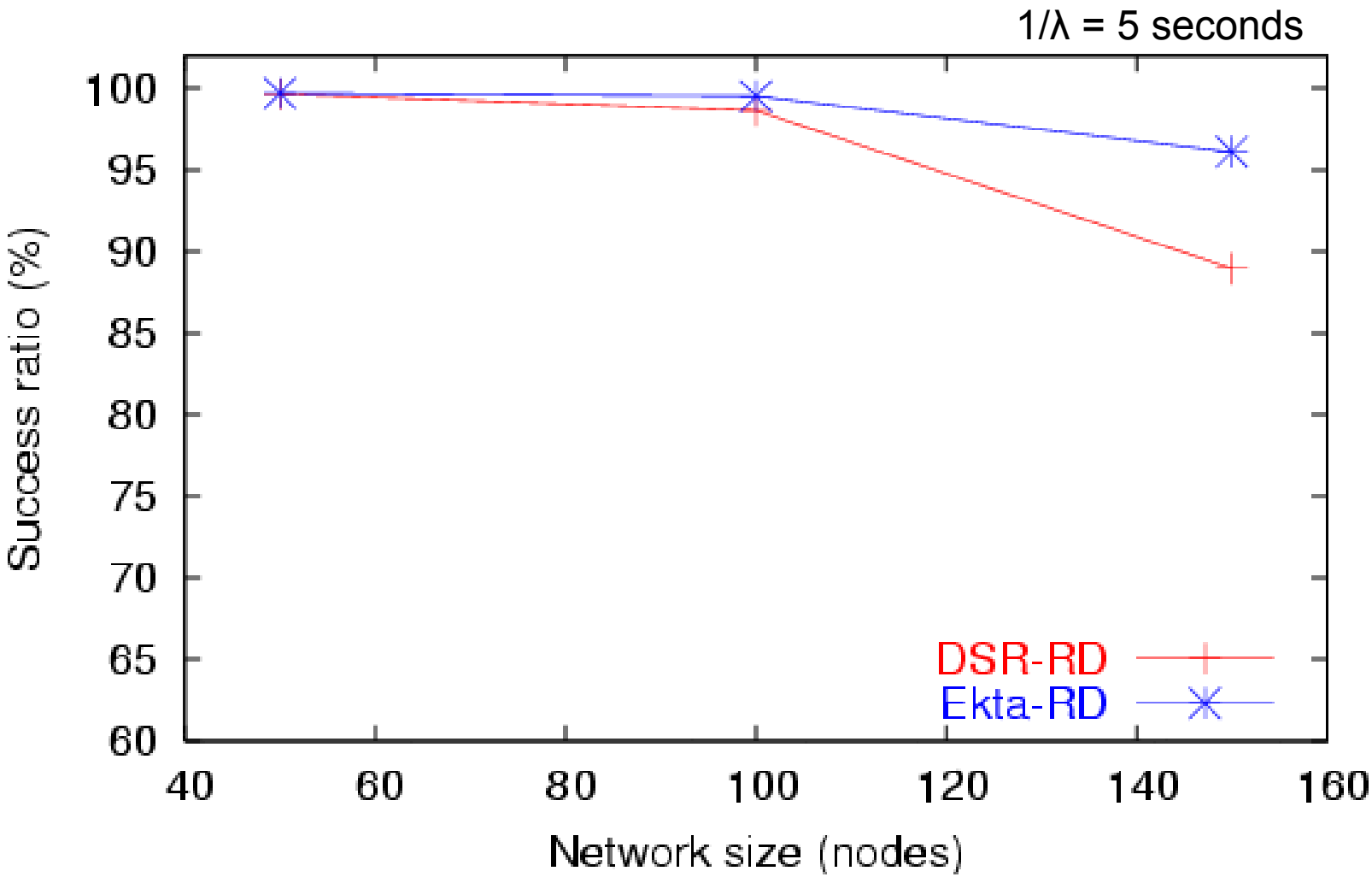
# Success ratio: Varying λ



# Overhead: Varying N



# Success ratio: Varying N



# Conclusions

- Integrated approach is an efficient architecture for implementing a DHT in MANETs
- MANET applications can benefit from DHTs as demonstrated by the resource discovery application
- Ekta can potentially be used as an efficient substrate for other applications

# Q & A