Experimental Comparison of Routing Protocols for WSNs:

Routing Overhead & Asymmetric Links

Henry-Joseph Audéoud, Martin Heusse

ITC 29, Sept. 2017

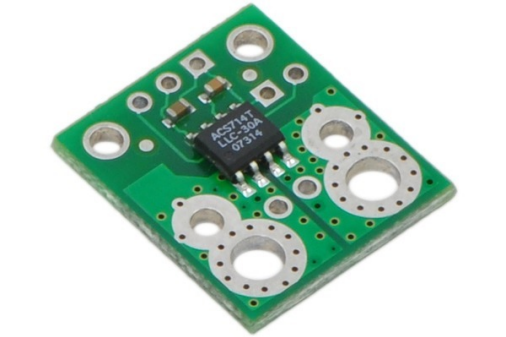
Introduction

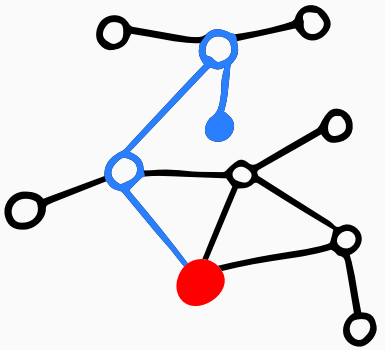
Intro. — Routing Protocols for Wireless Sensors Networks?

Constrained devices — memory, computation, energy

Radio links — lossy, variable, shared...

Multi-hops network — need a routing protocol





Intro. — Experimental Comparison?

Work on a real testbed instead of simulation

Platform FIT IoT-lab

Nodes M3 Open Node

OS ContikiOS 3.0

Radio channel Real radio interfaces
(*i.e.* with collisions, interference, variability...)

Routing protocol implementations

- LRP: `https://github.com/drakkar-lig/contiki/tree/lrp`
- RPL: included in ContikiOS' core

Intro. — RPL? LRP?

	RPL Routing Protocol for LLNs (RFC 6550 and many others)	**LRP** Lightweight Routing Protocol
Collection Tree construction	Bellman-Ford	
Host Routes construction	Proactive	
Collection Tree maintenance	Proactive (Trickle)	Reactive (local repair)
Host Routes maintenance	Proactive (continuous)	Reactive (flooding)
Loop avoidance	Per-packet header	Local rules

Intro. — Overview

Introduction

Routing Overhead

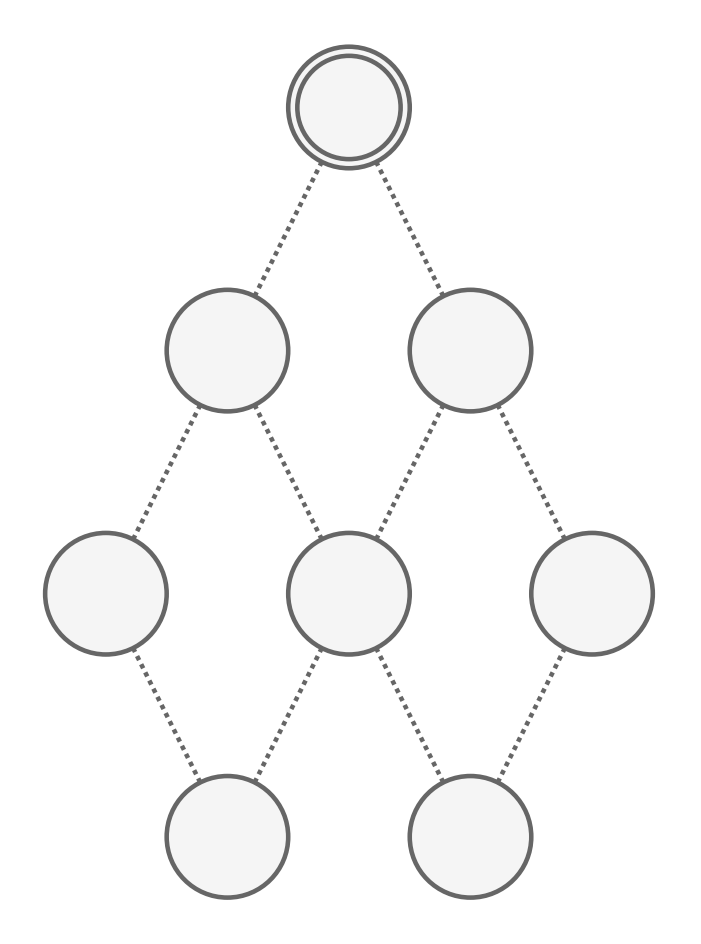
Asymmetric links

Conclusion

Routing Overhead

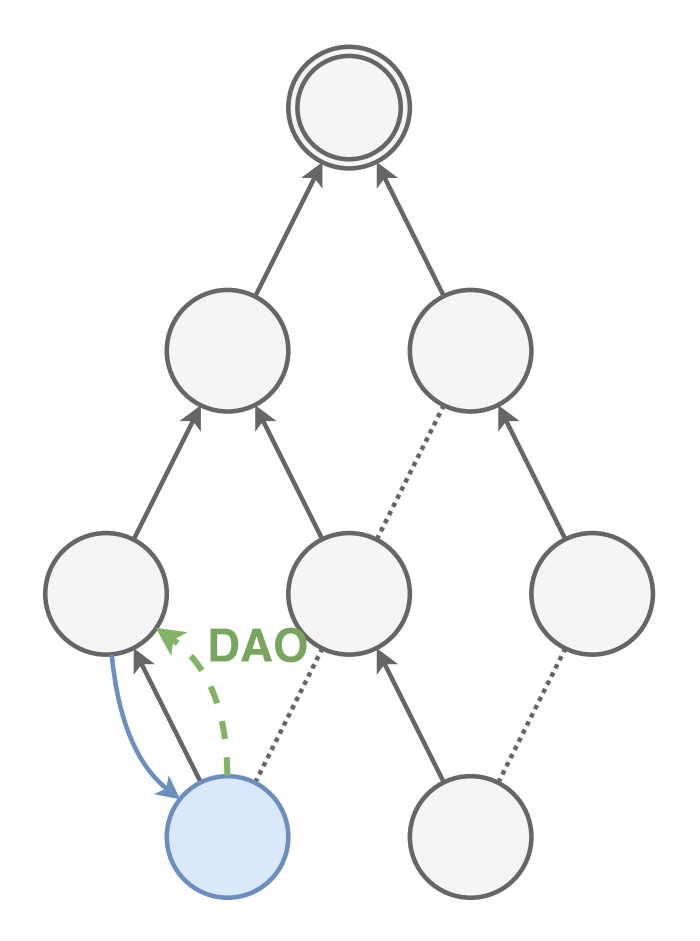
RPL Initialization — Collection Tree Construction

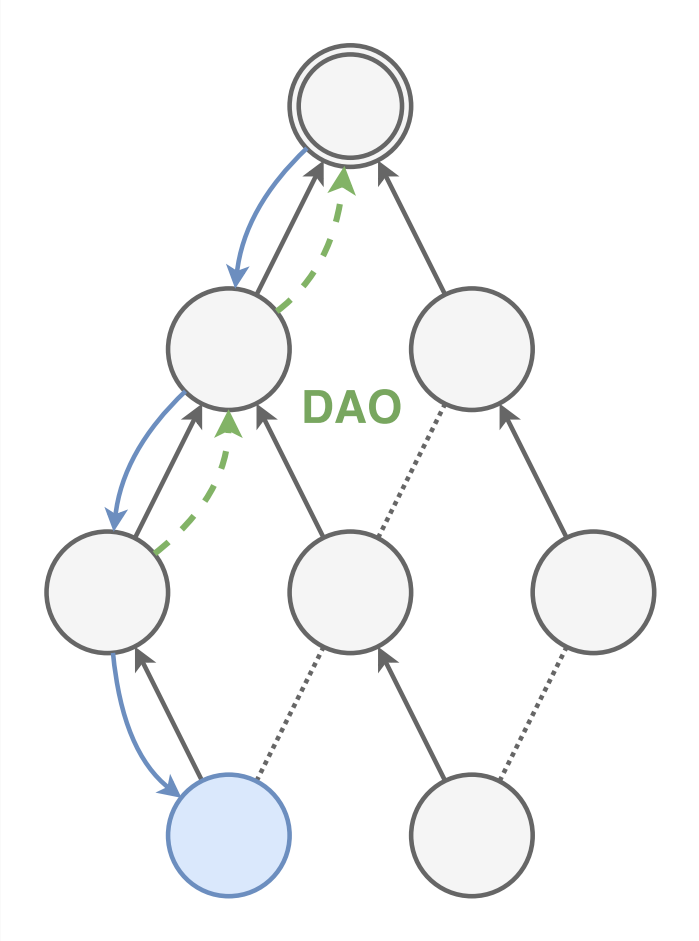
Distributed Bellman-Ford



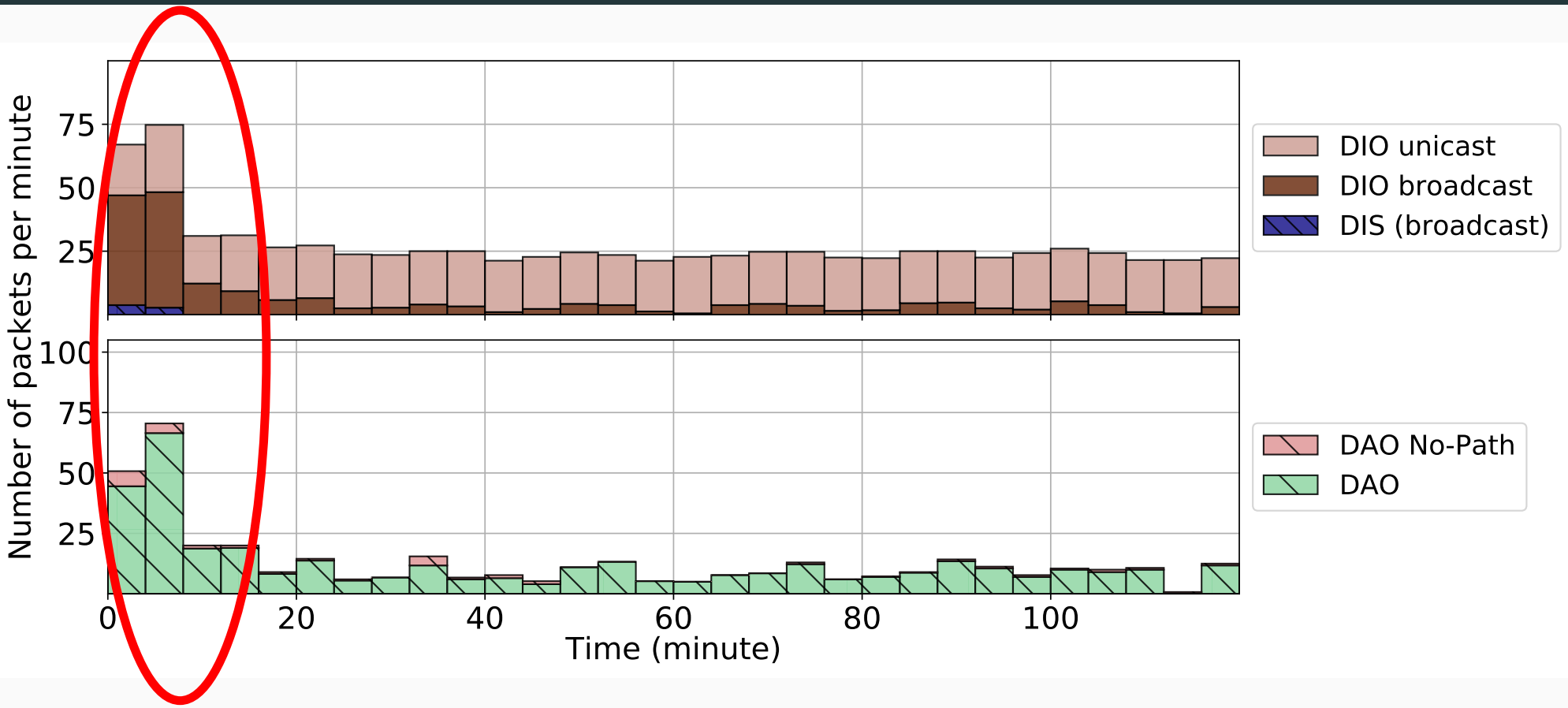
RPL Initialization — Host Routes Construction

Proactive construction





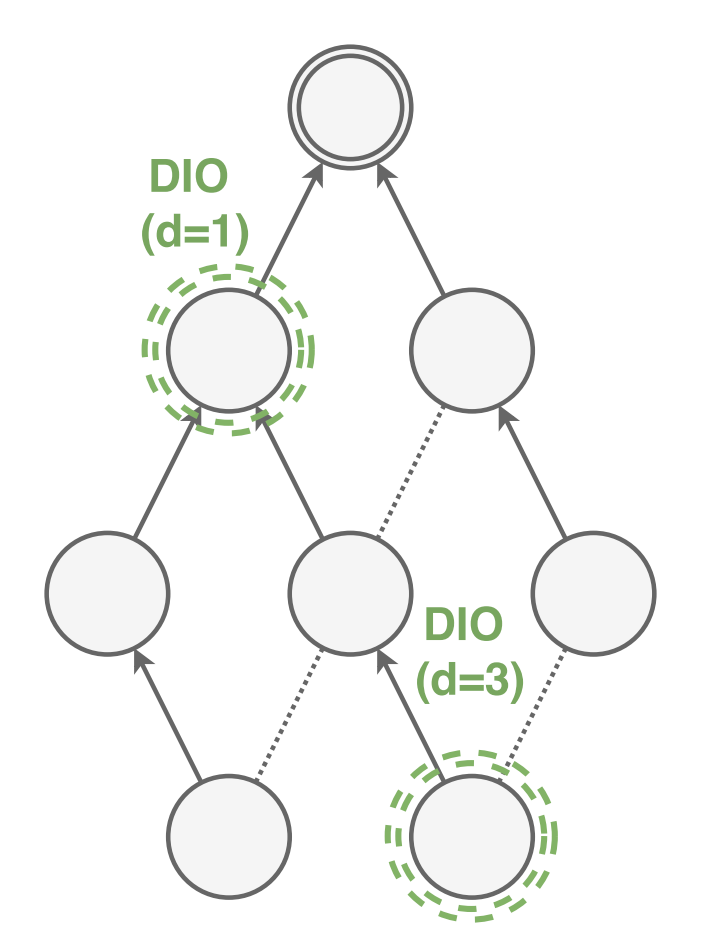
RPL Initialization — Experiment example



Experiment on FIT IoT-lab, 40 nodes, with RPL

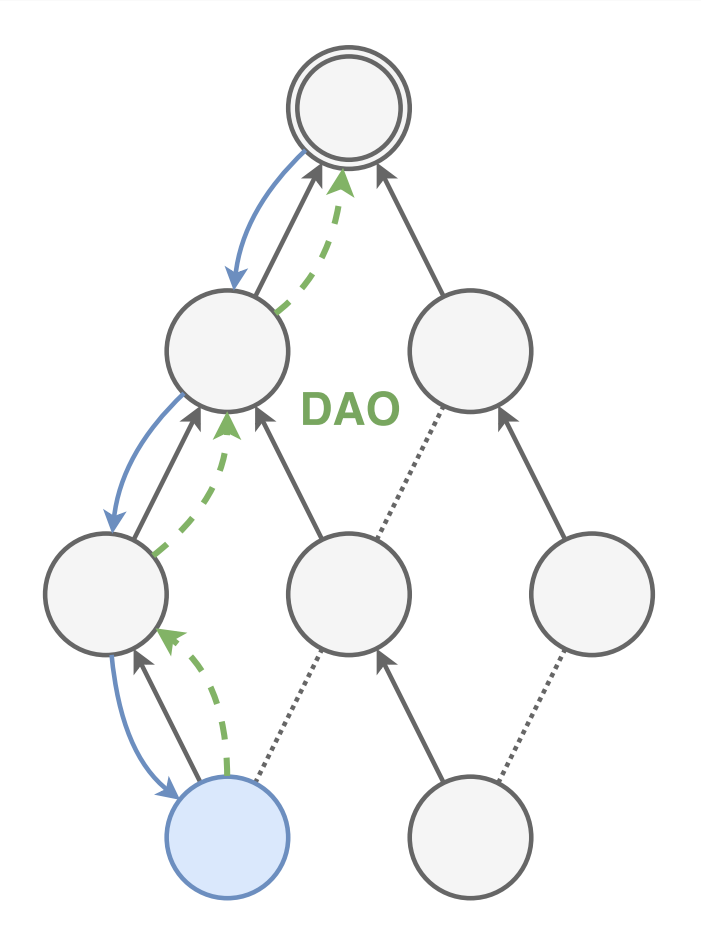
RPL Steady State — Collection Tree Maintenance

Trickle: Continuous maintenance traffic

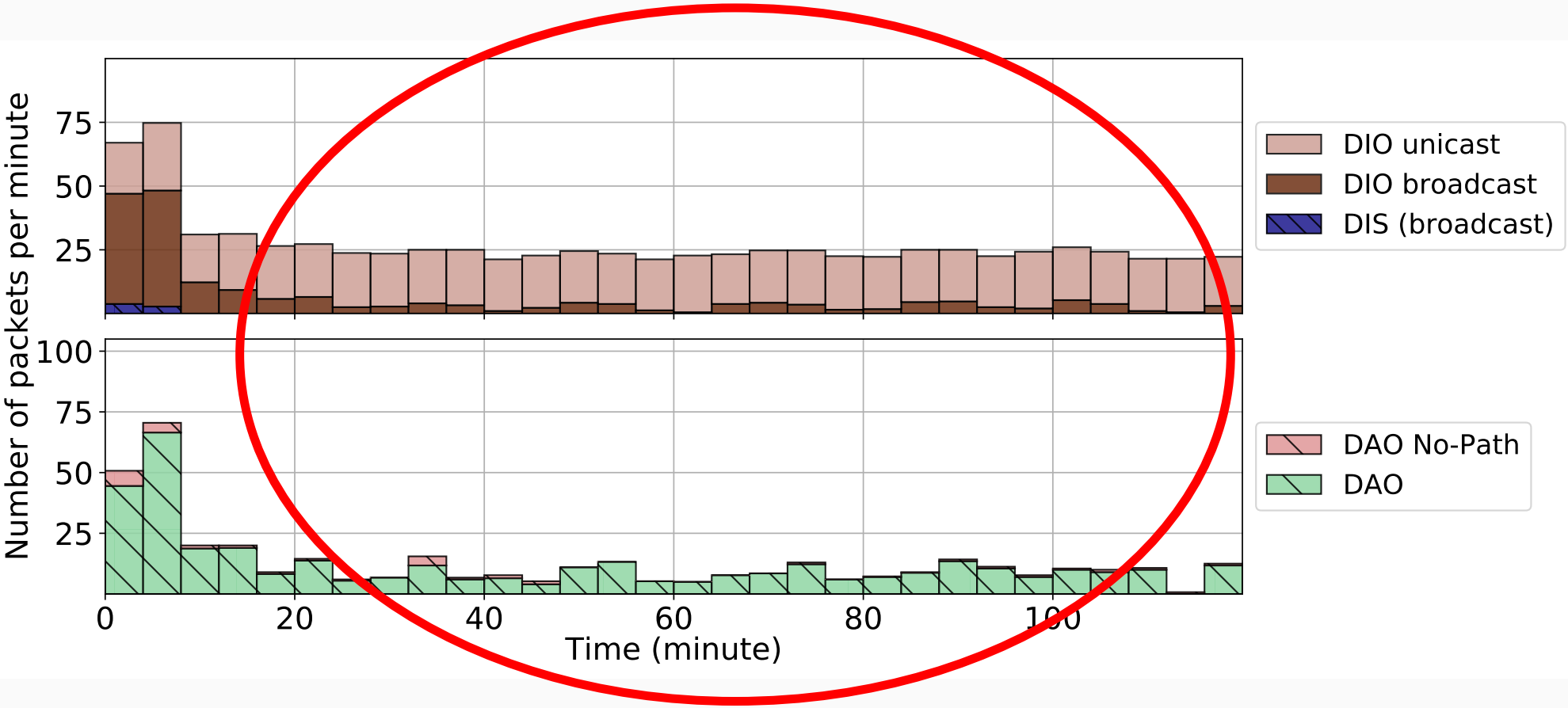


RPL Steady State — Host Routes Maintenance

Recurrent DAO traffic

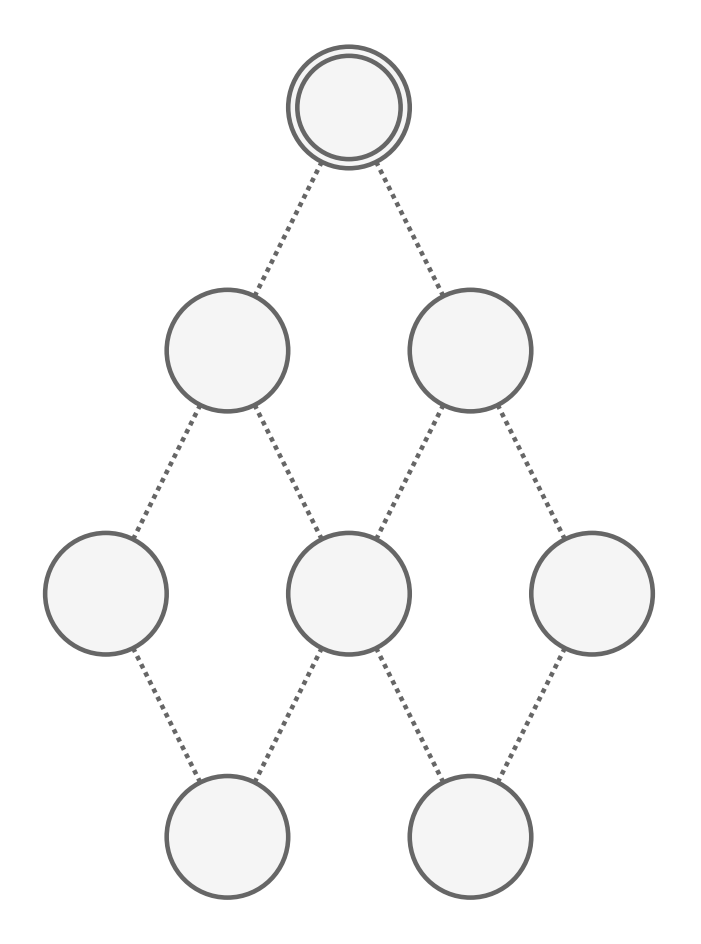


RPL Steady State — Experiment example



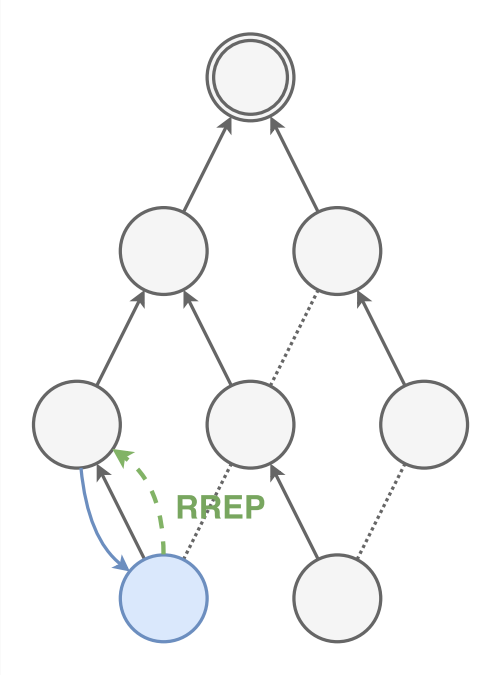
LRP Initialization — Collection Tree Construction

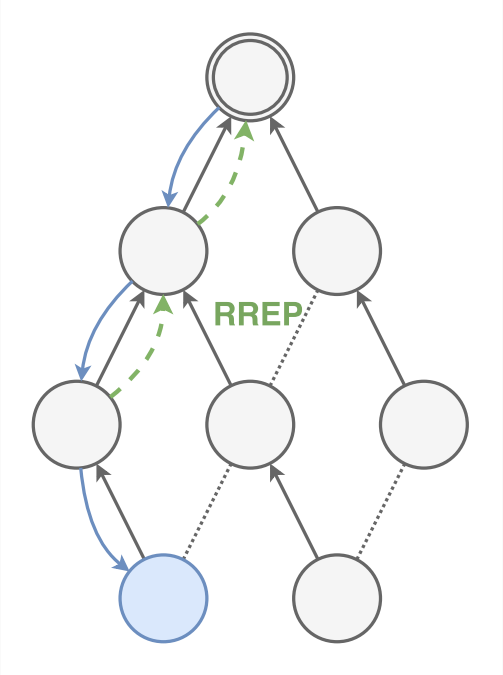
Distributed Bellman-Ford



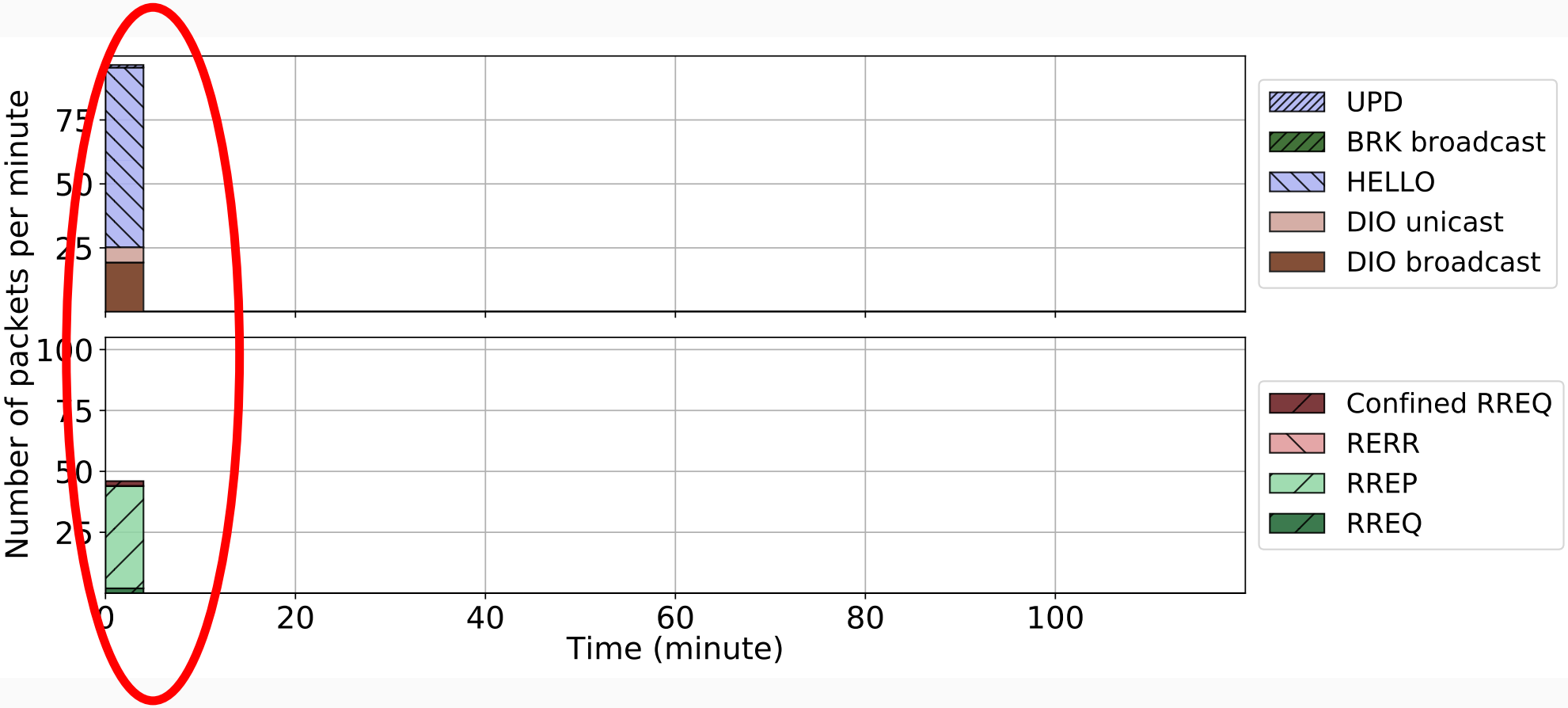
LRP Initialization — Host Routes Construction

Proactive construction



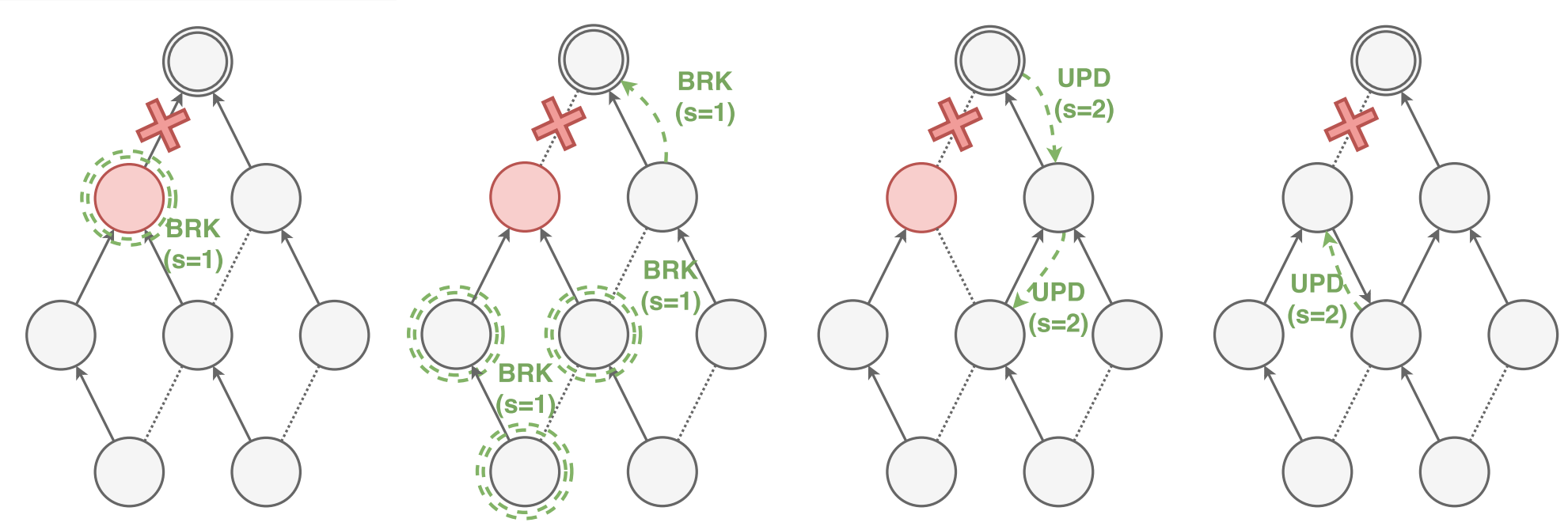


LRP Initialization — Experiment example



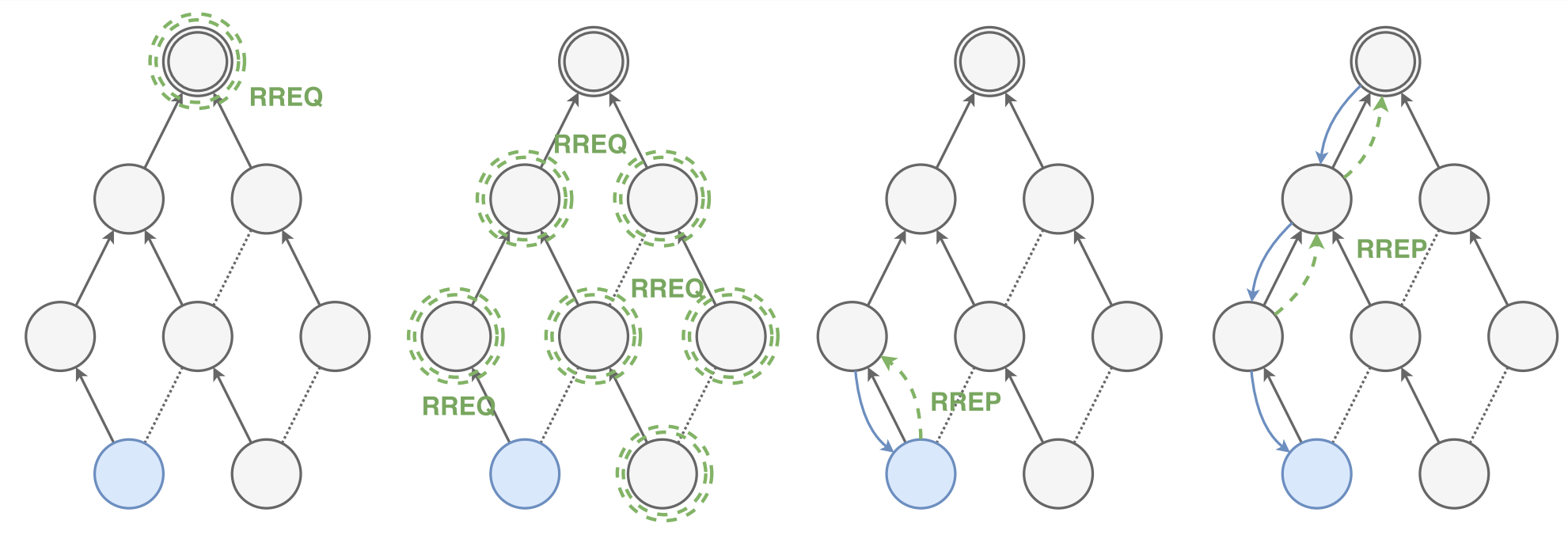
LRP Steady State — Collection Tree Maintenance

Local Repair mechanism

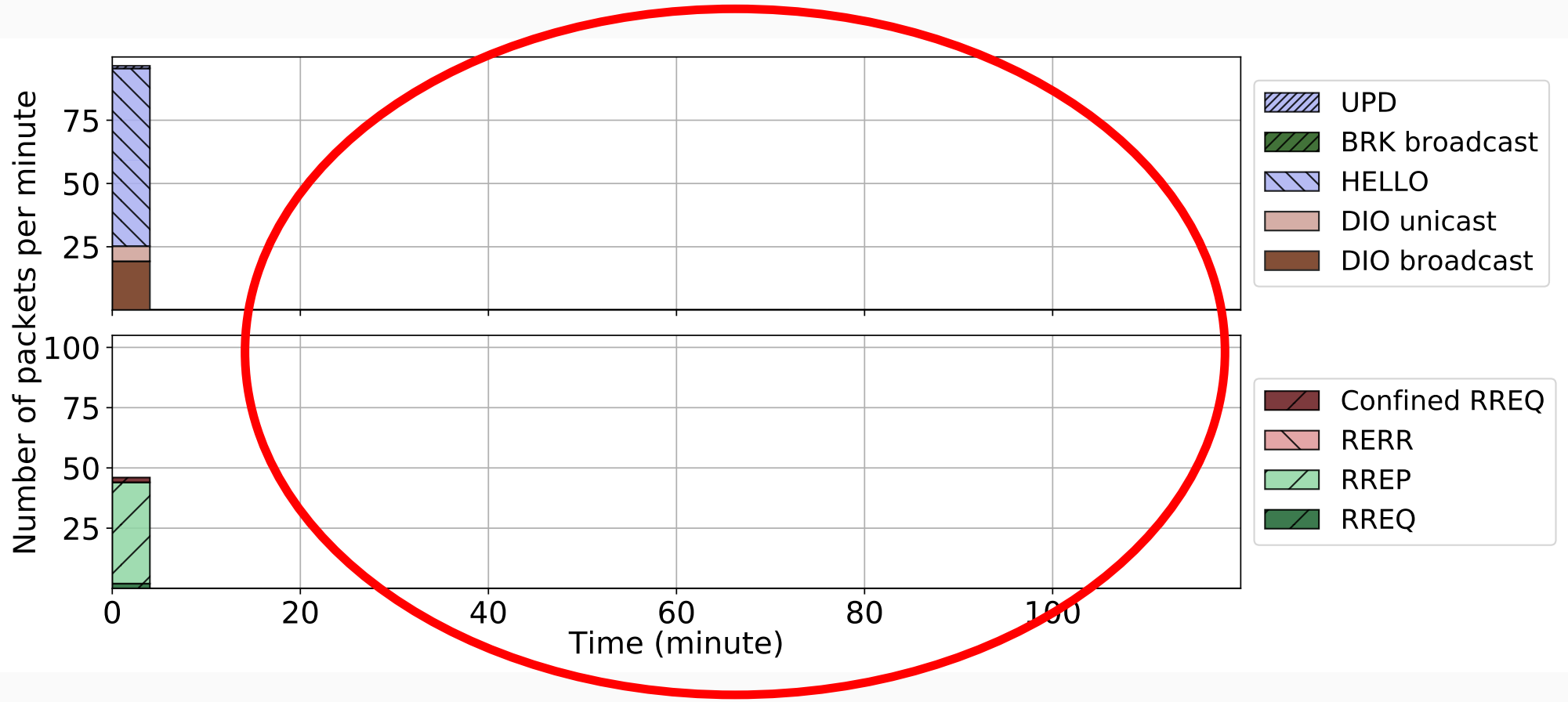


LRP Steady State — Host Routes Maintenance

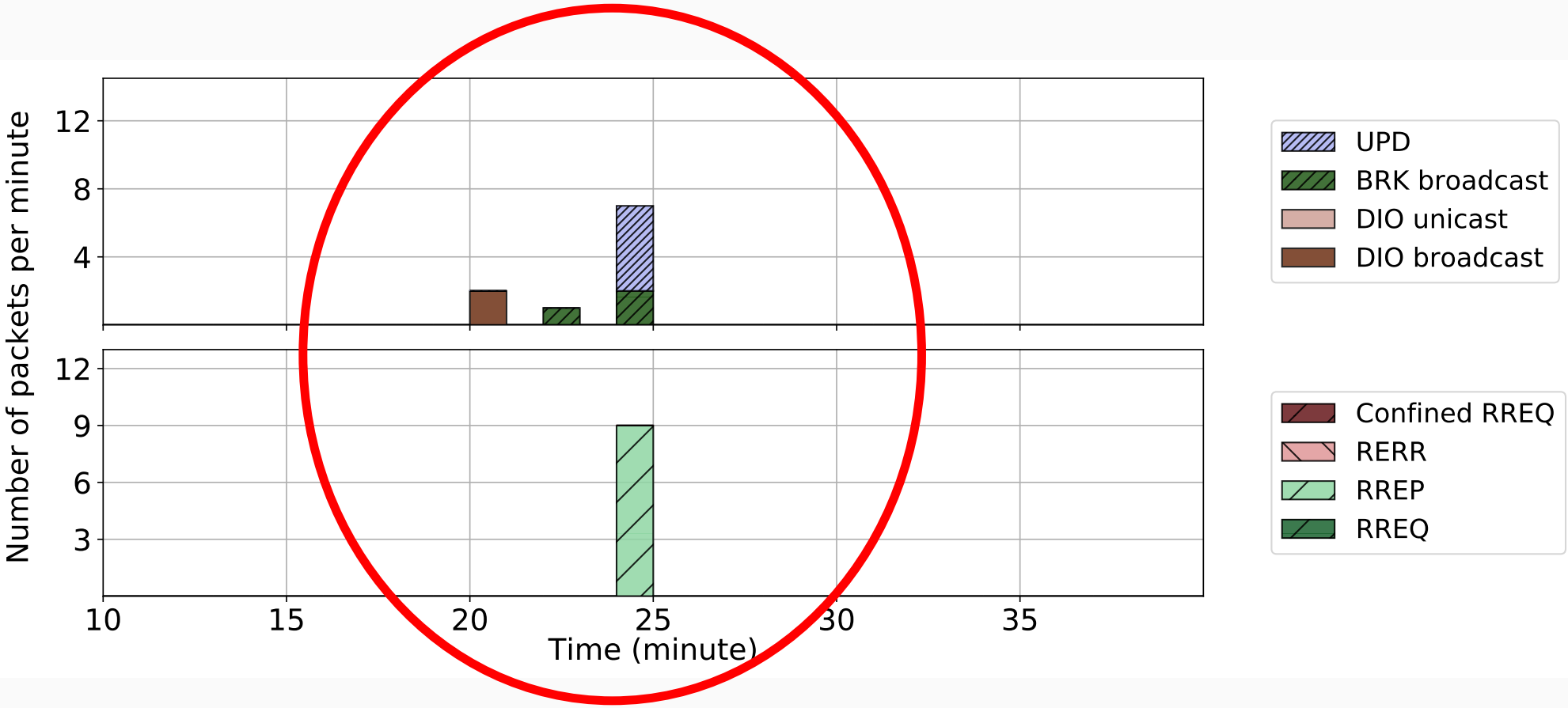
Reactive: only when it is needed



LRP Steady State — Experiment example

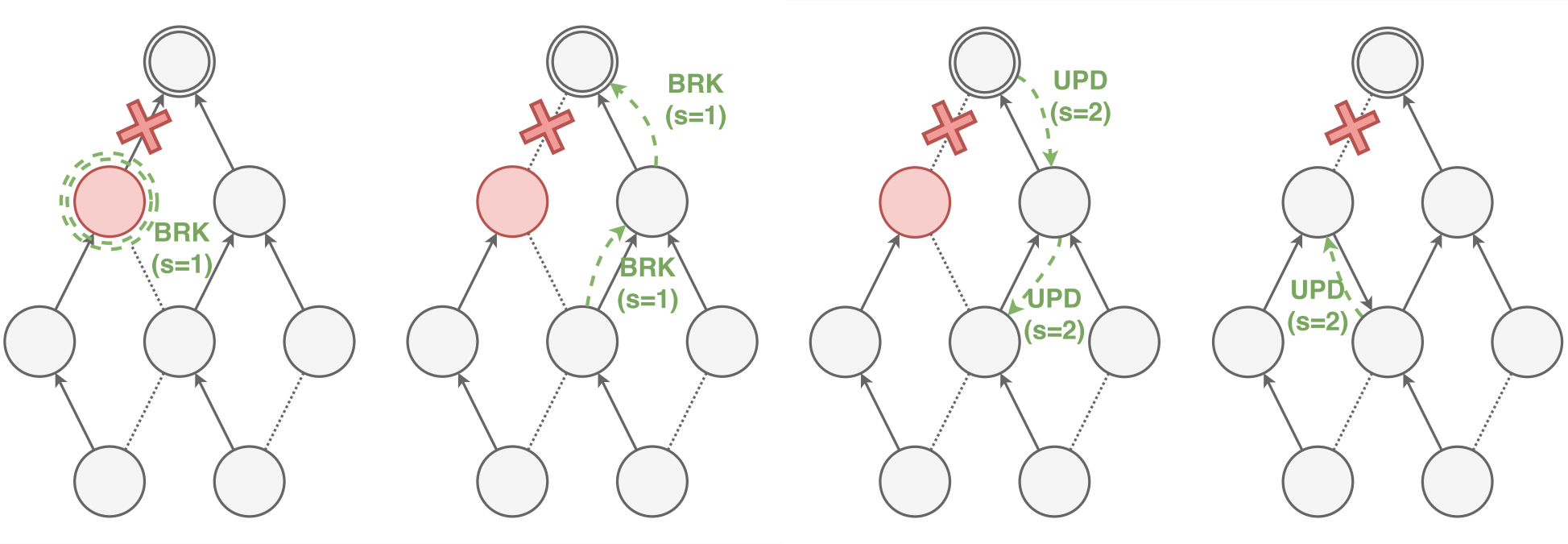


LRP Steady State — Experiment example



LRP — Improve Local Repair: Expending Ring Search

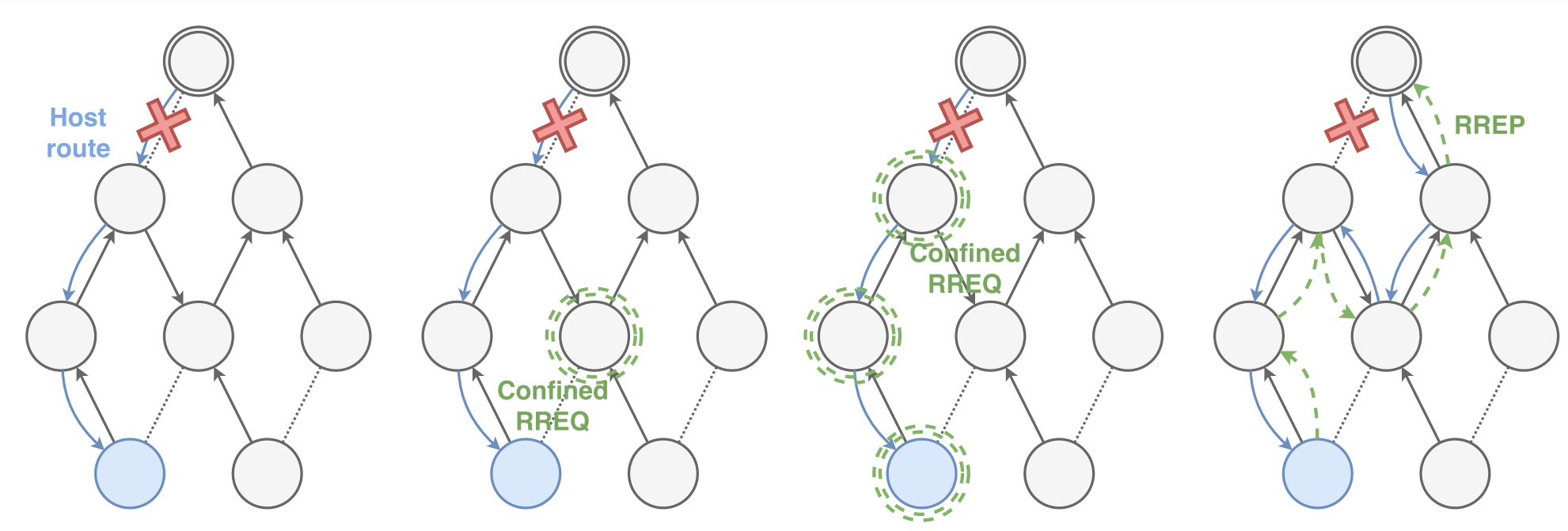
Idea: Limit the flooding to the top of the sub-tree



Proposed in the paper

LRP — Improve Local Repair: Confined Route Requests

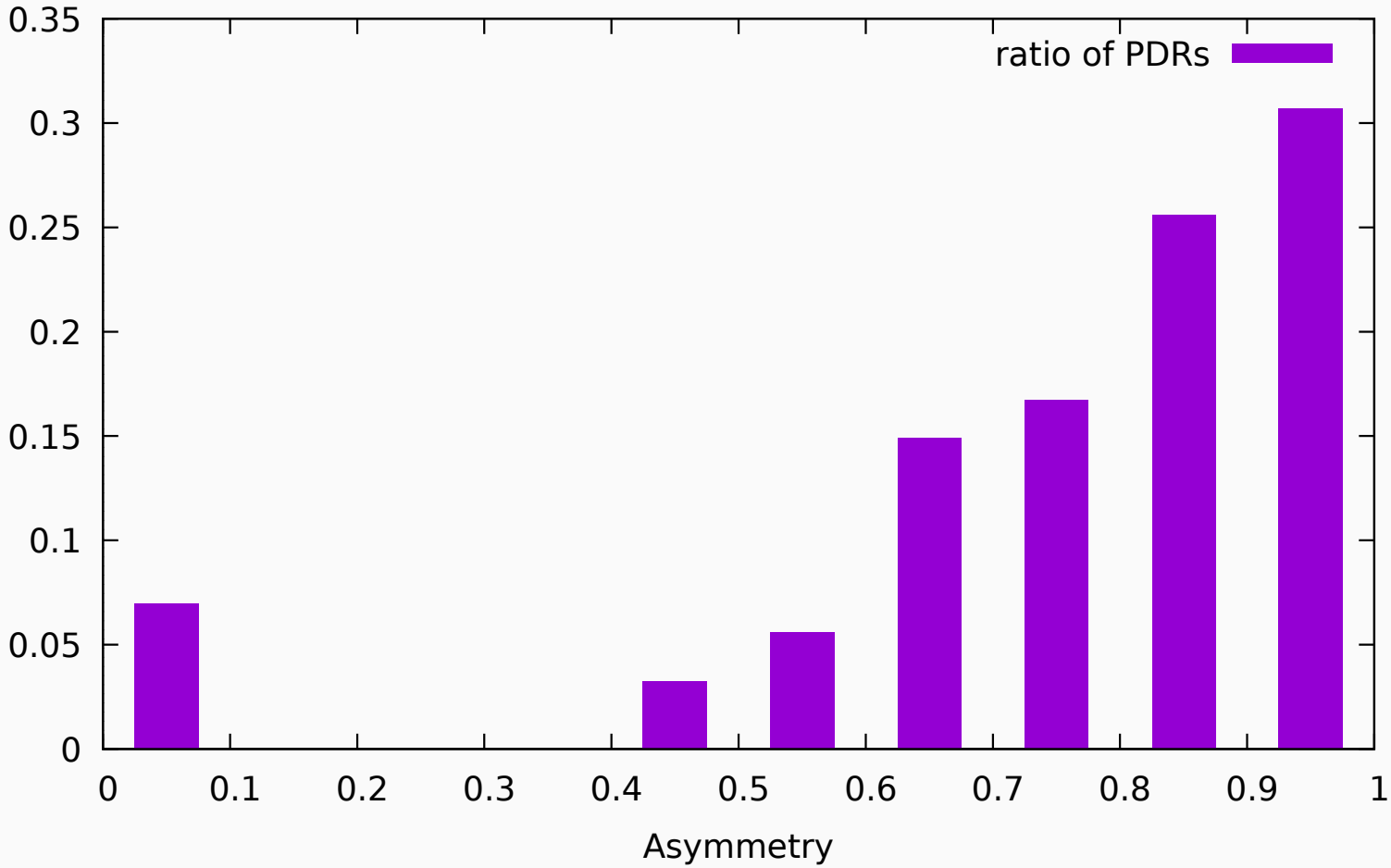
Idea: Also repair host routes!



Proposed in the paper

Asymmetric links

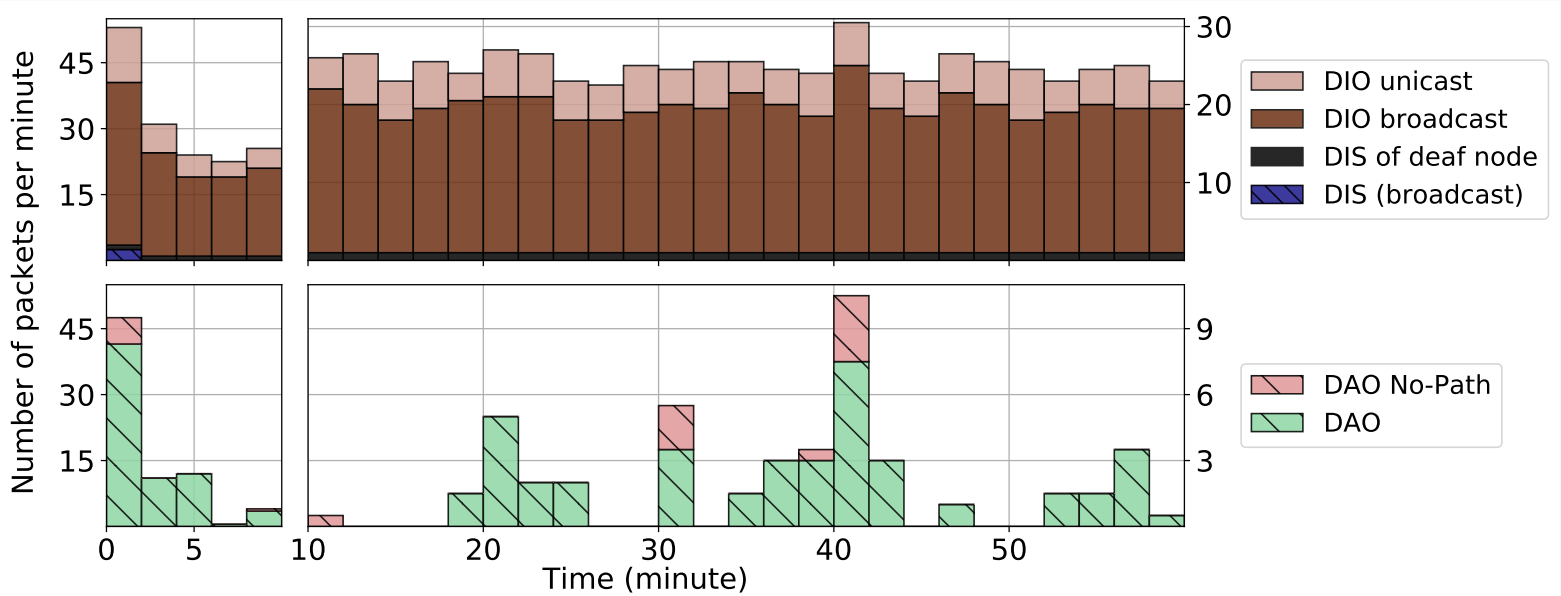
Does asymmetry really exist?



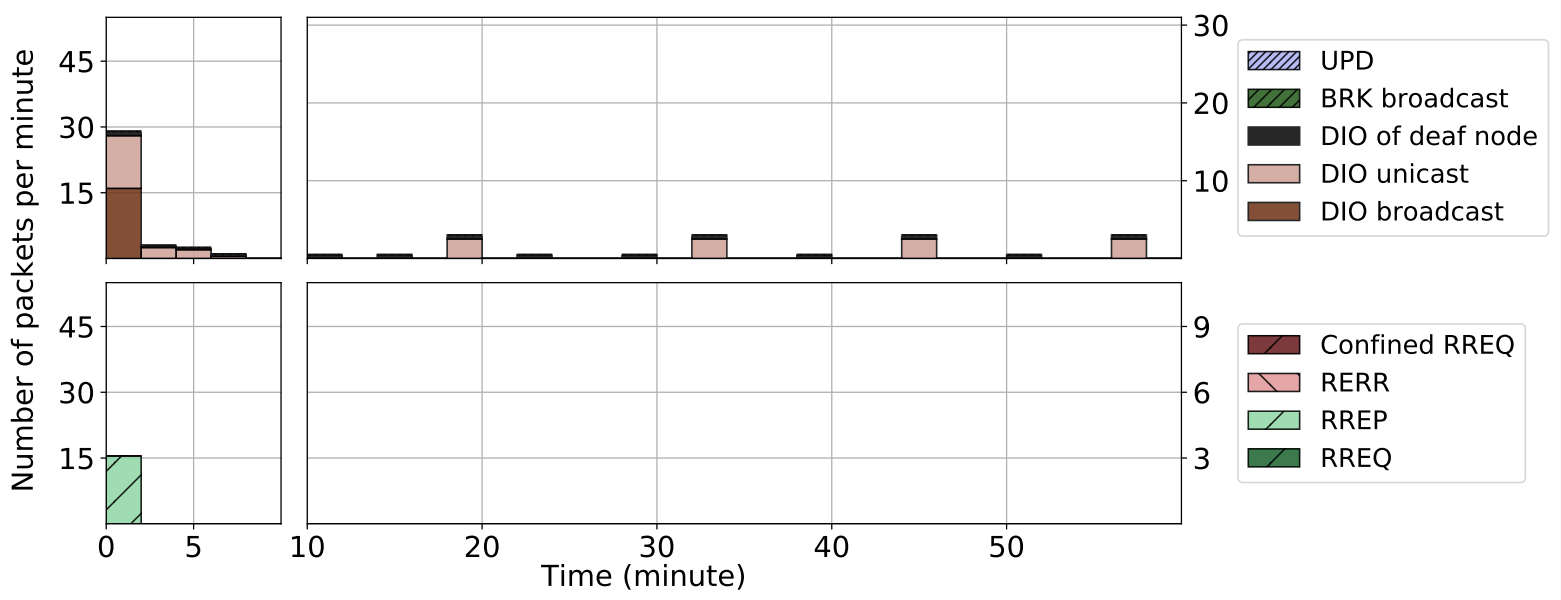
Tested on FIT IoT-lab, 28 nodes, 90 packets on each link.

The case of a deaf node

RPL

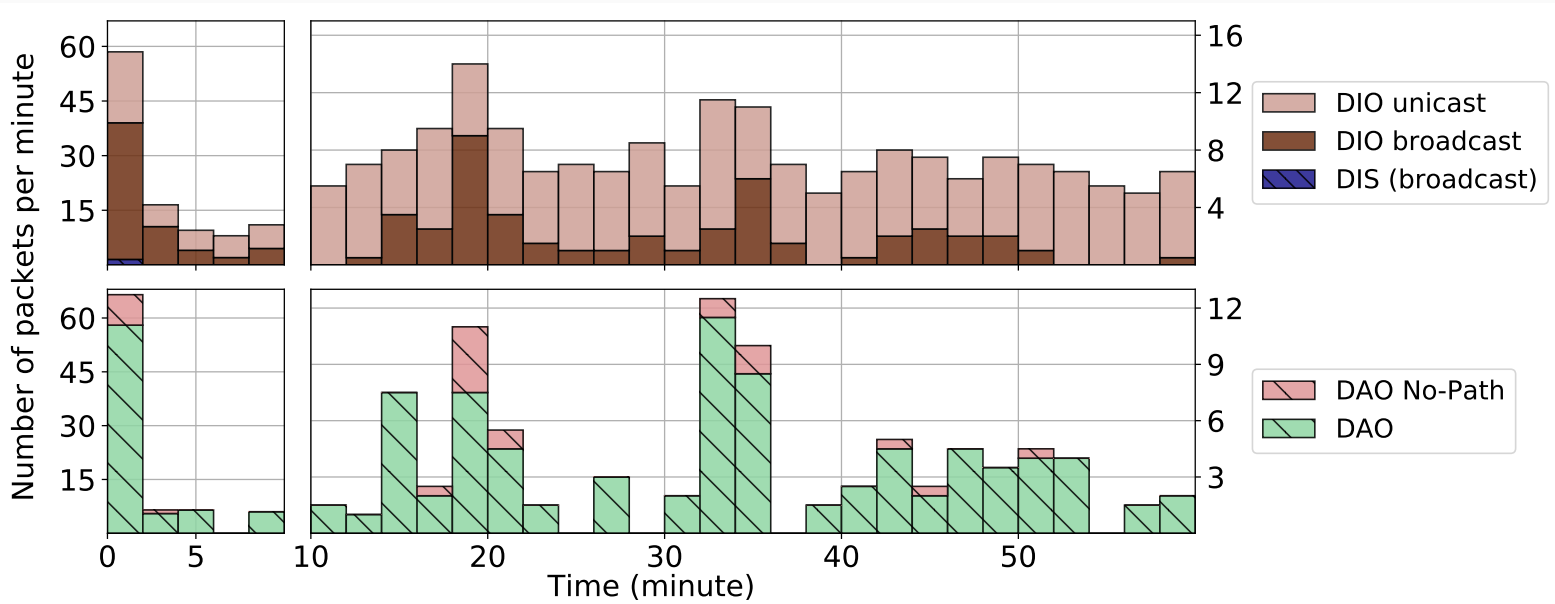


LRP

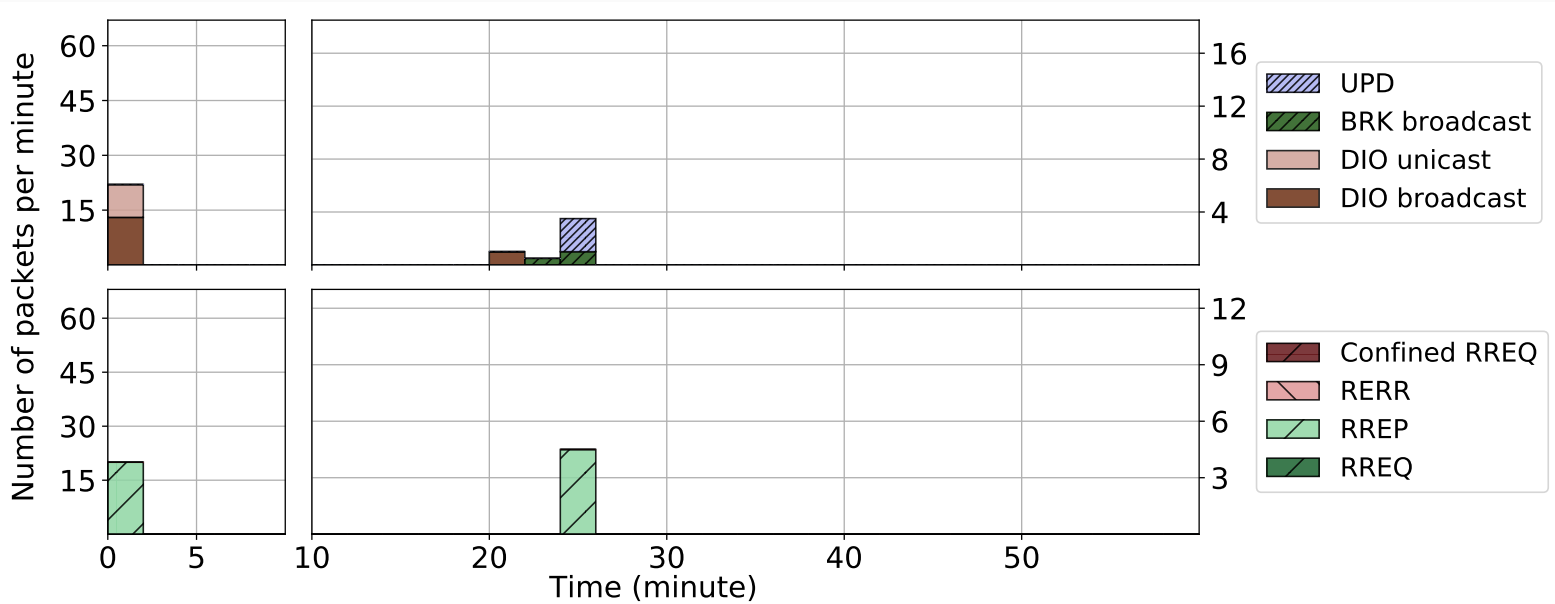


The case of a muted node

RPL



LRP



Conclusion

Conclusion

We have seen:

an **experimental comparison** of RPL and LRP

the **protocols overhead** in normal situation

their reaction in presence of **asymmetric links**

Next steps?

Balancing schemes (improve DODAG structure)

Multi-sink approach

Various prefix lengths

Thank you!