



# Efficient and Guaranteed Routing in Wireless Sensors Networks

Henry-Joseph Audéoud

## ► To cite this version:

Henry-Joseph Audéoud. Efficient and Guaranteed Routing in Wireless Sensors Networks. LIG PhD's day, Mar 2017, Grenoble, France. hal-02379648

**HAL Id: hal-02379648**

**<https://hal.univ-grenoble-alpes.fr/hal-02379648>**

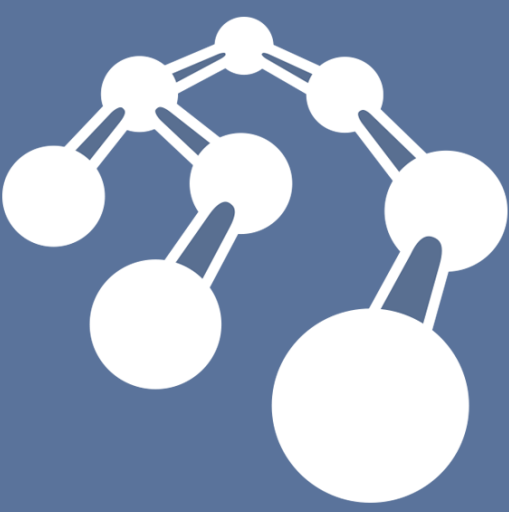
Submitted on 25 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Efficient and Guaranteed Routing in Wireless Sensors Networks

Henry-Joseph Audéoud



### The Constraints in a Wireless Sensors Networks

- The routes
  - Creation** — how to compute them?
  - Reliability** — the problem of loops
  - Maintenance** — it's a dynamic problem
- The network
  - Losses** — temporary perturbations
  - Saturation** — low traffic load
  - Variability** — changing environment
- The platforms
  - Energy** — battery-powered
- QoS — not reliable in time
- Asymmetry — not exactly the same devices

A routing protocol, at a time light, reliable, and tolerant?

## The Lightweight Routing Protocol (LRP)

### Collection Tree — traffic extraction out of the network

- Distributed Bellman-Ford (Fig. 1)
  - Sequence number** (temporal indication) ; **metric** (link cost)
- Avoid loops in the tree
  - Never go backwards** (get rid of count-to-infinity situations)
- Tree maintenance
  - Global** repair (sink initiative) v.s. **local** repair (host initiative, cf. Fig. 2)

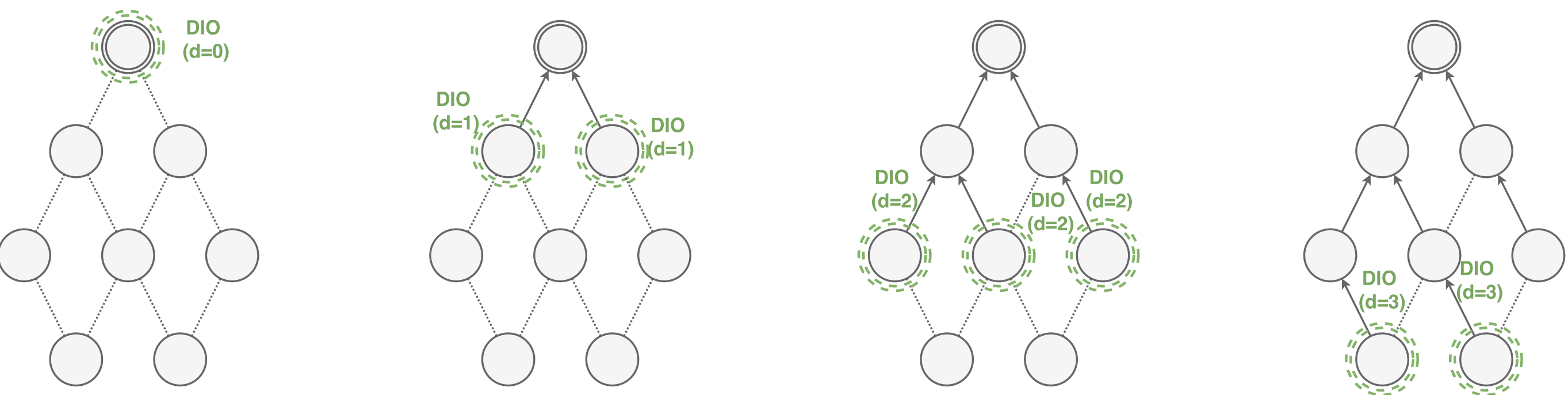


Figure 1: Collection tree construction using the Bellman-Ford algorithm. This mechanism is used in global repair, by the sink (highest node).

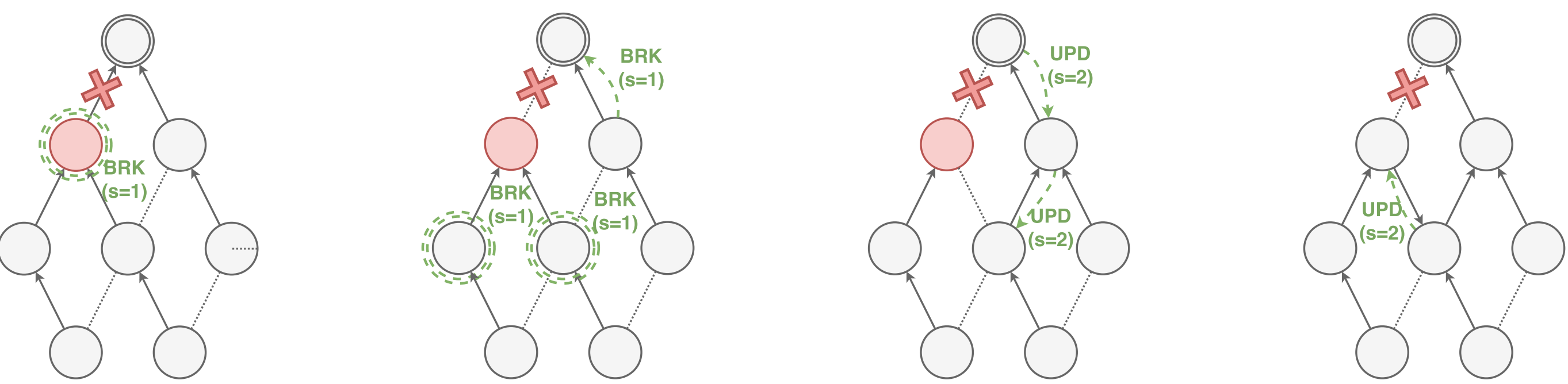


Figure 2: Local repair, after a link was lost. The detached node will reverse a link to one of its predecessors, to be able to reach the sink again.

### Host Routes — traffic distribution inside the network

- Host route establishment (Fig. 3)
  - Proactively** — host initiative, spontaneous creation
  - Reactively** — sink initiative, looking for the host
- Avoid the loops when routing
  - Detection** — more and more precise routes (Fig. 4)
  - Elimination** — whole destruction to the sink

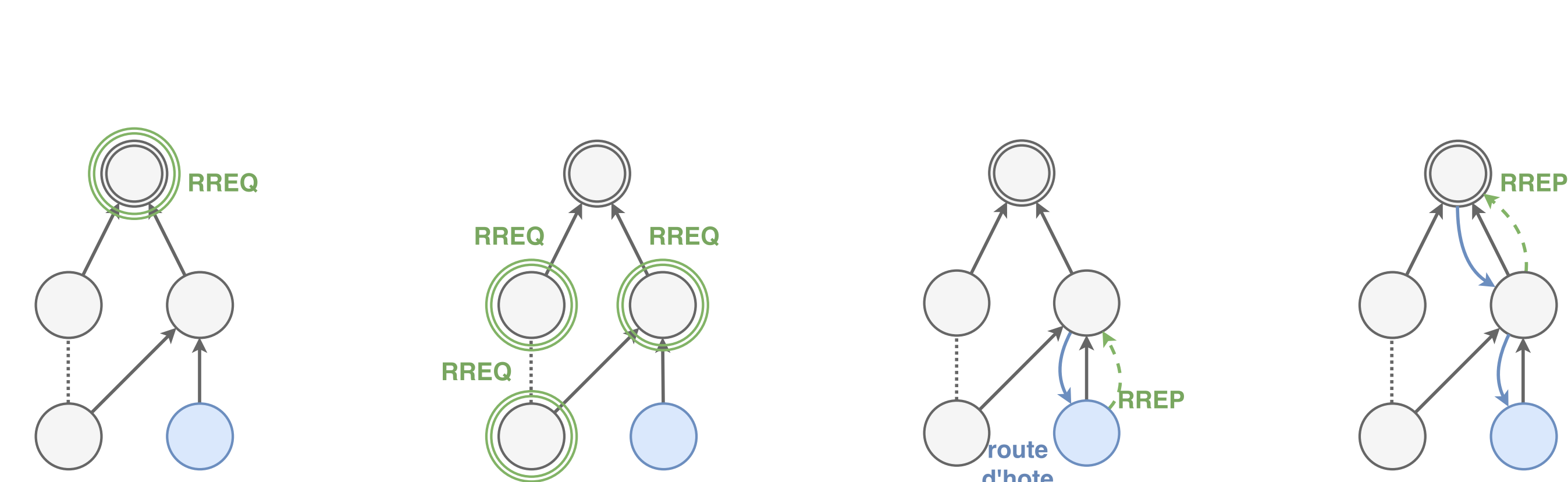


Figure 3: Host route establishment. Looking for the host into the network; then host route establishment. When a host proactively builds its host route, only the two last steps occur.

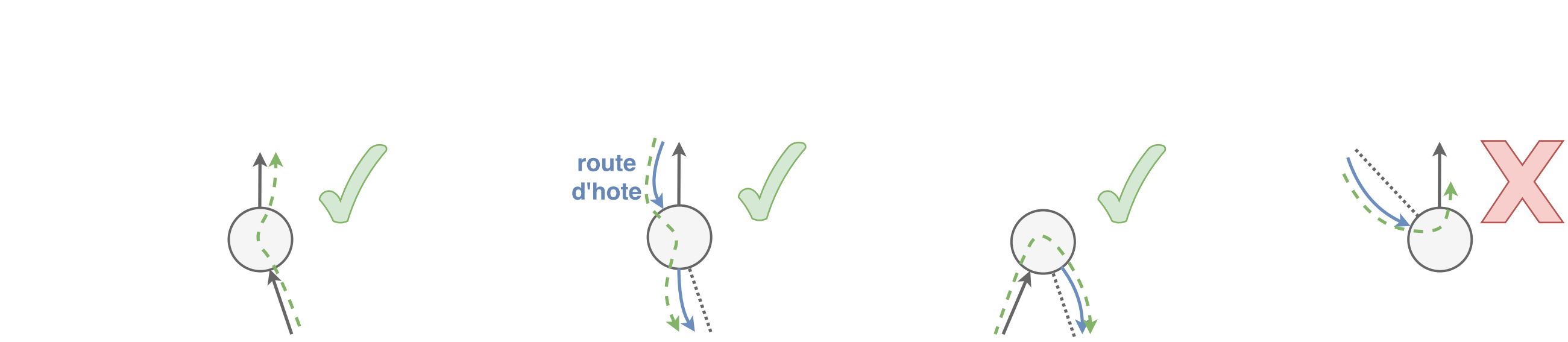


Figure 4: Routing loop detection. In the last case, the node should use a route which is less precise than the previous one, that is forbidden.

## Experimentations — FIT IoT-lab

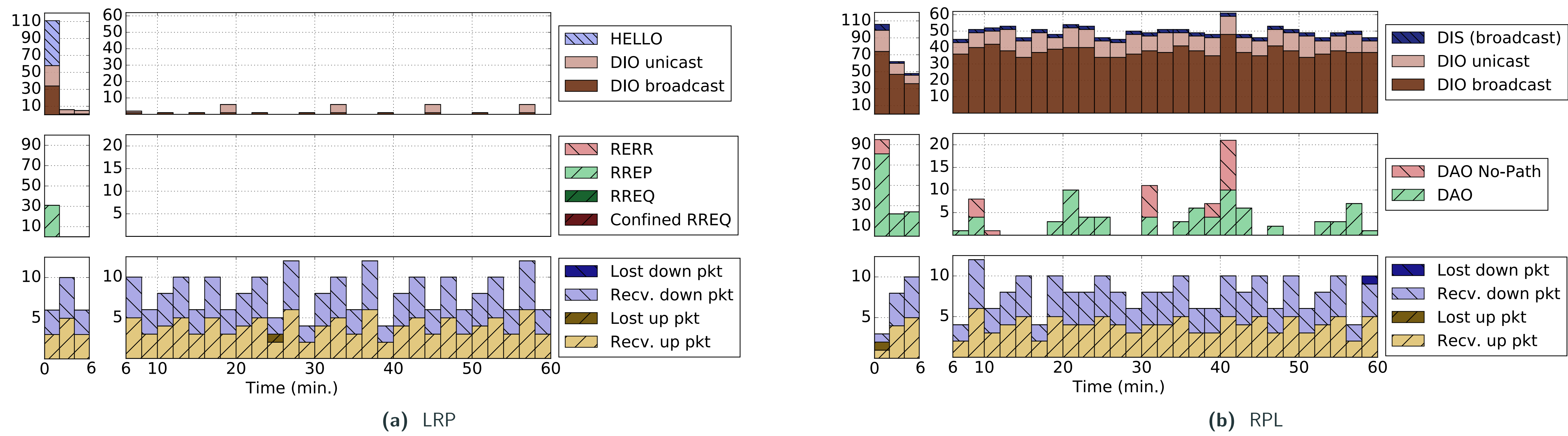


Figure 5: Messages exchanged between 12 nodes, one among them is deaf (it does not receive the other's messages). A comparison is done between LRP and RPL (RFC 6550). RPL handles this problem very badly.

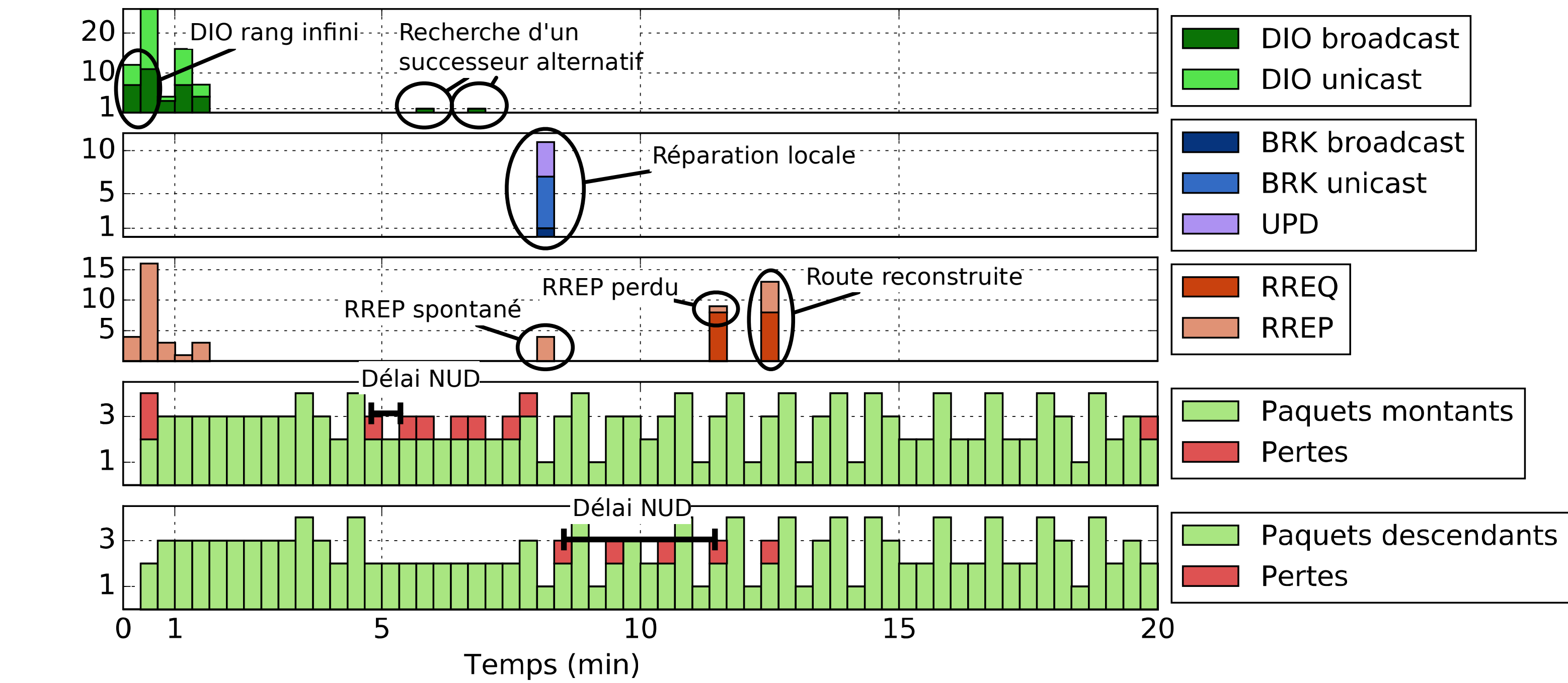


Figure 6: Messages exchanged between 10 nodes. At 4'30, a node among them is shut down.

## And now?

- Improve **metric**
  - A **local problem**... — which measure is important?
  - ... and a **global problem** — local choices influence the whole tree
- Use many **prefix sizes** — only one route for many hosts
  - By aggregating routes? Not efficient for random addresses...
  - With a subnet? Not really ad-hoc...
- Use a **backbone** to lighten nodes
  - How sinks may cohabit on the same network?
  - How to ensure connectivity even if backbone is not audible?