

Routing for harvested nodes

Martin Heusse

and many others: Chi-Anh La, Michel Favre...

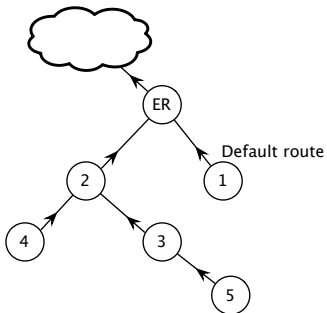


Context

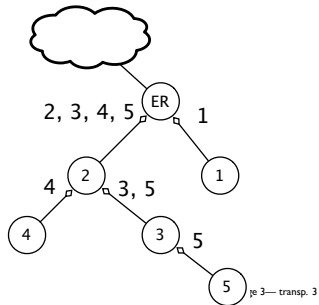
- 802.15.4
 - ✓ Beacon-enabled
- Super-low duty cycling nodes
- IP
- Small footprint (code, memory, traffic)

What's the problem?

- Collect data

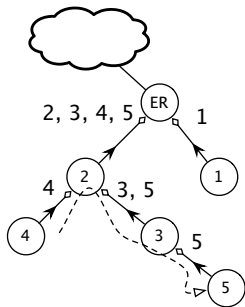


- Access the nodes



What's the problem? (cont.)

- Node to node traffic
(Ex.: Going from 4 to 5,
first on default route then on host route)



Build the Collection Tree

A node should **associate** with the **neighbor** that puts it the **closest** from the **ER**

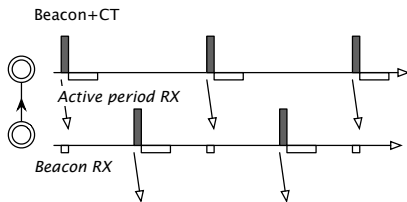
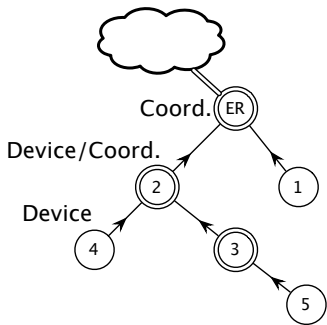
- Let's do three things at once:
 - ✓ Detect that a neighbor is there
 - ✓ Find its distance to the ER (rank)
 - ✓ Do periodic link check

Put the packets for CT construction in the beacons

(Or in some of the beacons... Or on demand...)

The **default** route next hop is the **coordinator**

Collection tree \Leftrightarrow cluster tree



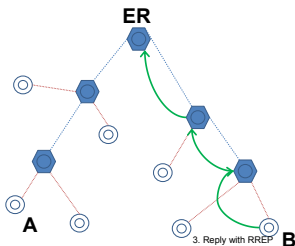
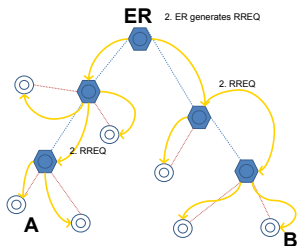
“Downward” routes

- Unsolicited (RPL DAO-style)

Sent when joining the network

- Reactive (on demand)

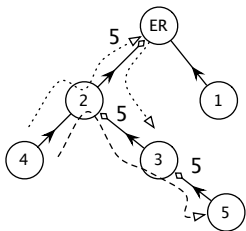
The edge router queries a specific node (RREQ/RREP LOADng)



Node to node communication

(Assuming the destination is unknown — worst case)

- The first packet will reach the ER
- The ER sends a RREQ, gets the RREP
- Subsequent packets will no necessarily go all the way up the tree



Why does this suit 802.15.4 beacon-enabled?

- Broadcasts/flooding go “down” only!

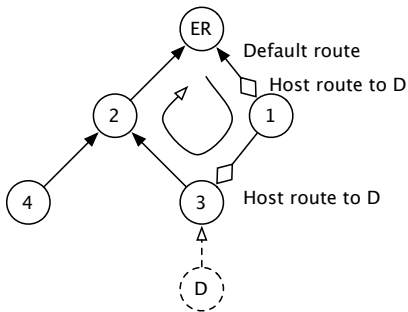
CT packets and RREQ

- **Up then down** sensor to sensor communication unavoidable anyway
- Only the ER sends RREQ
- Broken route \Rightarrow send RERR (The devil is in the details — see below!)

A general principle: Destination sequenced distance vector (RPL DIOs, LOADng, AODV, DSDV...)

- When sending a packet, the source marks it with a sequence number
 - Never use stale information
 - ⇒ **No routing loop**
- CT packets are piggybacked in beacon frames (Thus transmitted frequently – there is a tradeoff between latency and overhead)

A routing loop RERR sending



- No one sees any problem! (Except when the TTL reaches 0.)
- No one sends a RERR.
- Note that:
 - ✓ This can only happen for local¹ destinations
 - ✓ The issue is exacerbated because two nodes are “higher” than 3

¹Destined

What can we do about it?

- Quick fix:

With only one parent, never send back on the default route,
a packet I got from it

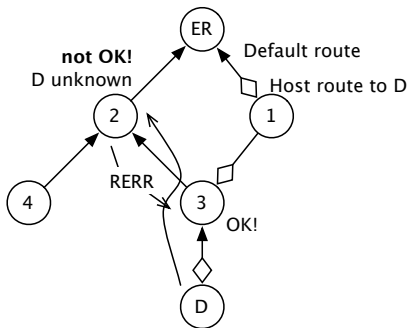
- RPL-like fix

Add a header to **each** packet, make sure it consistently goes
down or up the tree

Really problematic \rightsquigarrow IP in IP encapsulation?

Let's just check things are looking good

- Check that a packet (with a local destination) sent on the default route goes away from its source



- If not, send a RERR to the packet source.
If I get a RERR for me from my default router → resend RREP!
- So it's a good idea to send a spontaneous RREP before sending a packet to a local destination! Anyway, the target will likely need to respond.

What happens if a node loses memory?

- In the previous example, node 2 does not know who to send the RERR to.
- This packet could just as well be destined to a destination below, and 2 would have forgotten about it.

Broadcast the RERR when a problem is detected,
let the neighbors pick it if needed

The RERR will either:

- ✓ Go up on the default route
- ✓ Or go back to the source

ND in sensor networks

GreenNet does not use it

- NS/NA

Not necessary as link local addresses are **based** on EUI-64

The routing protocol gives the mapping between destination (global address) and next-hop (link-local address)

- DAD not necessary if global address derived from EUI-64 (thus ARO –Address Registration Option– not necessary)
- **No on-link prefix**, as the network is **ad hoc multi-hop**
- Auto-configuration at node insertion in the network (ABRO –Authoritative Border Router Option– not necessary)
- No NUD: there are beacons, association!

Issues / work in progress

- Start time (when does a relay send its beacon? relatively to parent, to neighbors)
(several approaches used/some in progress)
- Autoconf?
Prefix change etc.
- Standardization
Same network structure as RPL → Route redistribution relatively straightforward.
- Reduce idle listening (active period RX)