RPL Routing Protocol Overview

Yevhenii Shudrenko

Institute of Communication Networks
Hamburg University of Technology

April, 2020

Definition

RPL — proactive distance-vector routing protocol for Low Power and Lossy Networks.

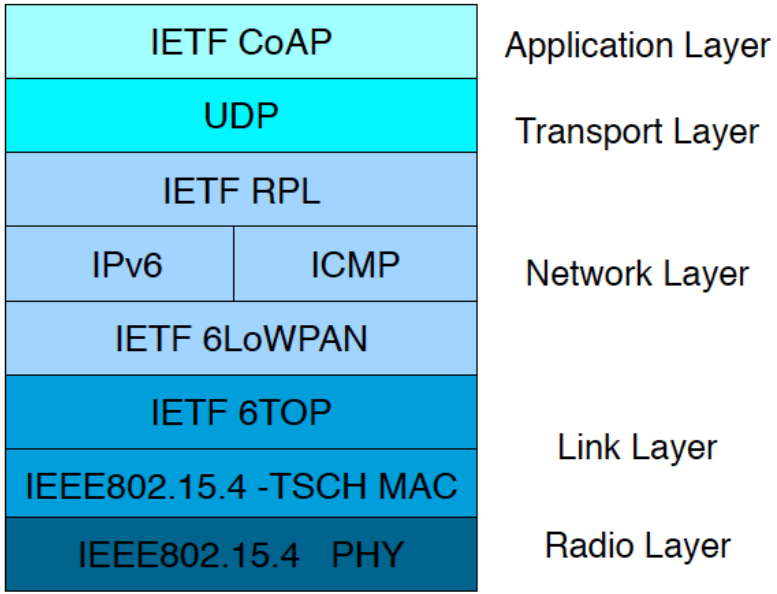

Figure 1: RPL in IETF standardized stack[1]

[1] Iova et al., "Rpl: The routing standard for the internet of things... or is it?"

Main Features

- Support of point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P) communication modes
- Configurable objective function for path computation
- Two operating modes for different resource constraints
- *Trickle* mechanism to leverage energy-efficiency for topology reactivity
- IPv6 compatibility
- High scalability for MP2P communication

DODAG

- Destination Oriented Directed Acyclic Graph
- Network topology & Data flow

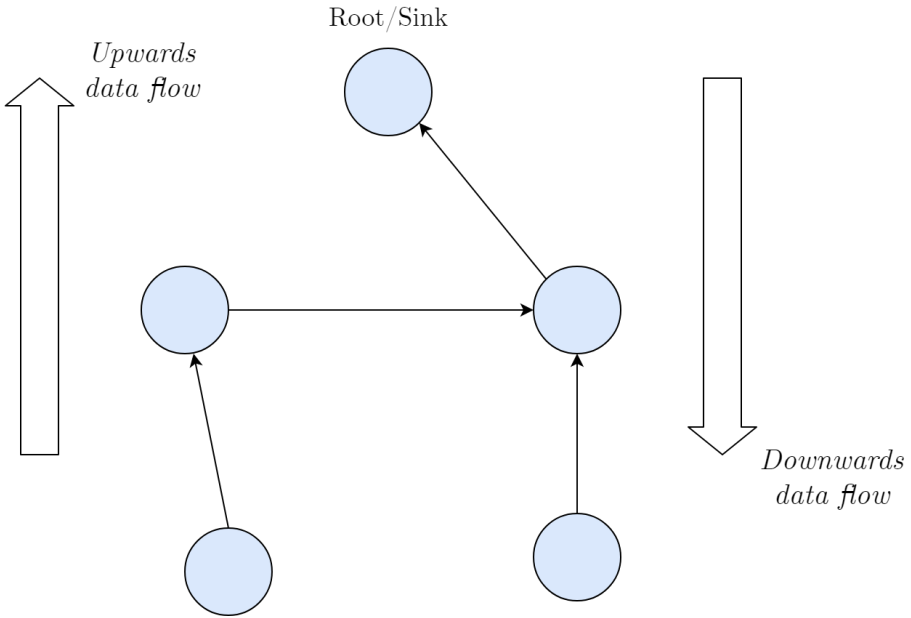

Figure 2: Example of a DODAG

Node Rank

Calculated using objective function, represents distance to the root.

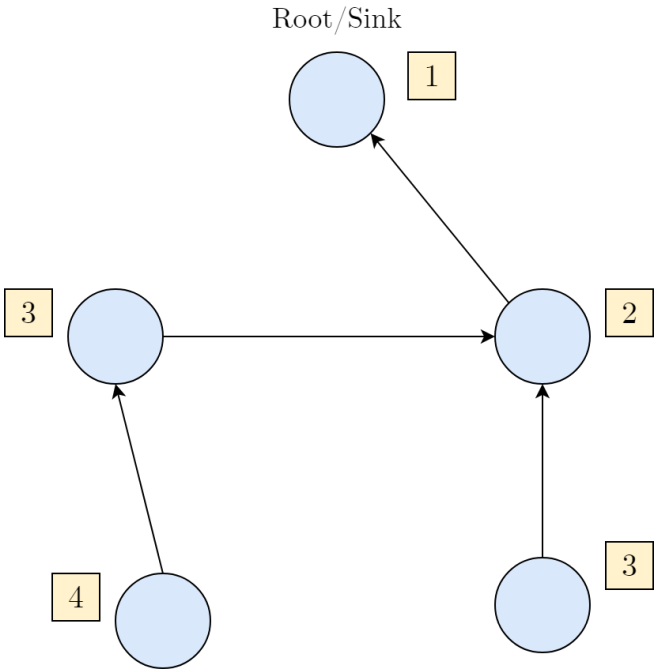

Figure 3: DODAG with nodes' ranks

Objective Function

Defines a metric for each node to choose preferred parent

Table 1: Commonly used metrics for objective function in LLNs[2]

Node Metric	Link Metric
State: CPU, Memory Load etc.	Throughput
Energy	Latency
Hop Count	Reliability

[2] Zhao et al., "A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities".

Control Messages

- **DIO** (DODAG Information Option) — sent periodically by all nodes to maintain view of dynamic topology
- **DAO** (Destination Advertisement Object) — announces node's availability, required for (Point-to-point) P2P, (Point-to-multipoint) P2MP routing
- **DIS** (DODAG Information Solicitation) — multicast request for a DIO from a new node joining the DODAG

DIO Message

Disseminates control data through the DODAG, includes:

- Node's rank
- Objective function
- DAGID
- Trickle timers
- DODAG sequence number (incremented by root to trigger rebuild procedure)

DODAG Construction

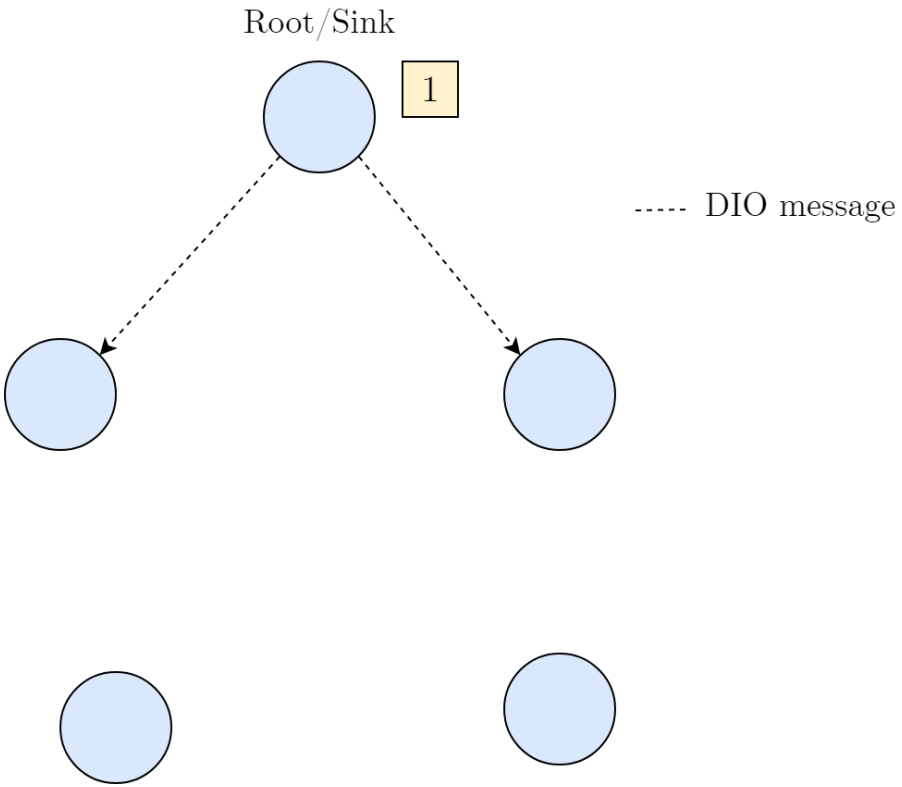

Figure 4: DODAG construction process

DODAG Construction

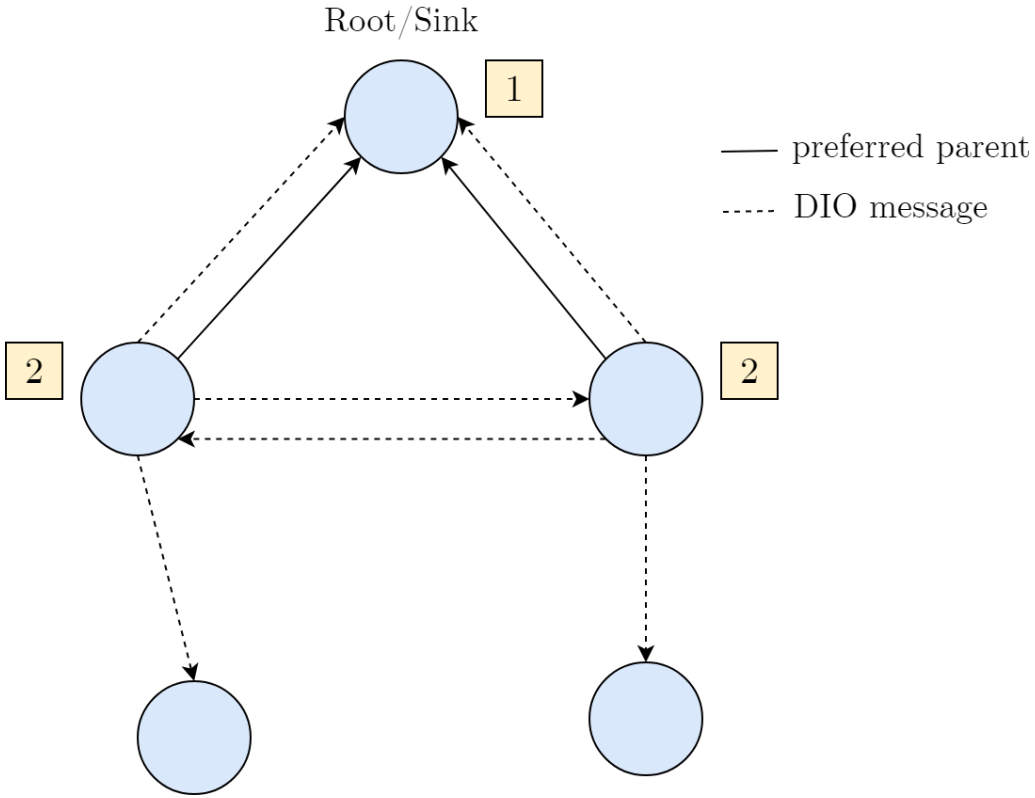

Figure 4: DODAG construction process

DODAG Construction

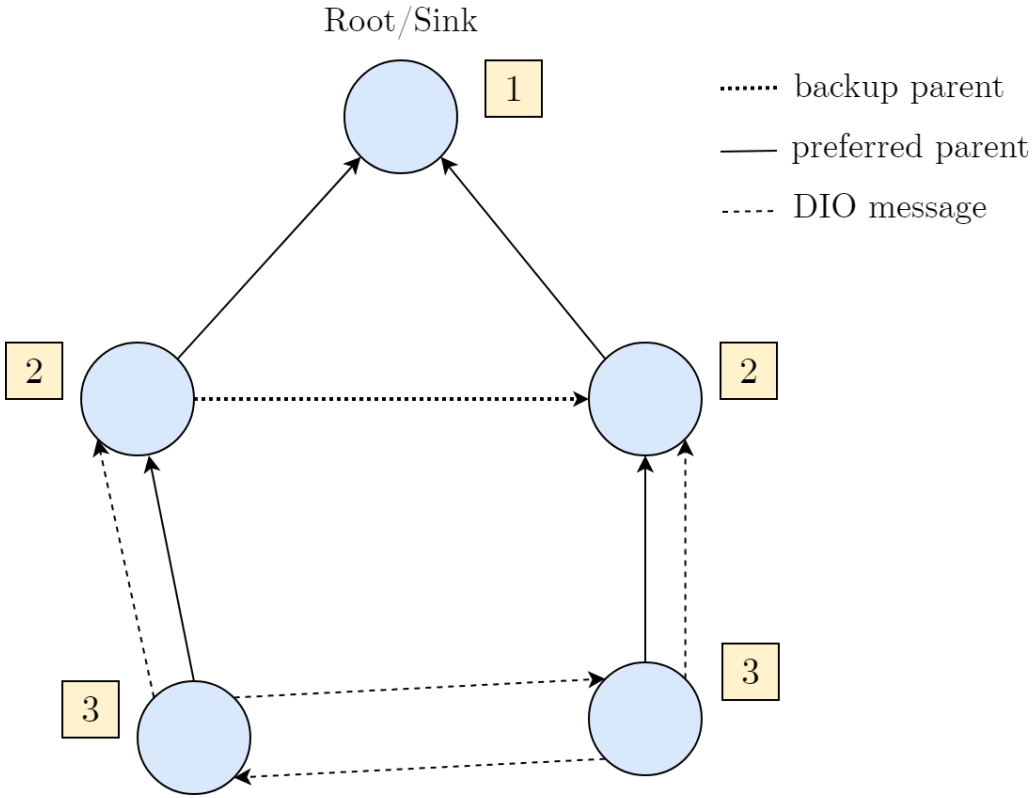

Figure 4: DODAG construction process

DODAG Construction

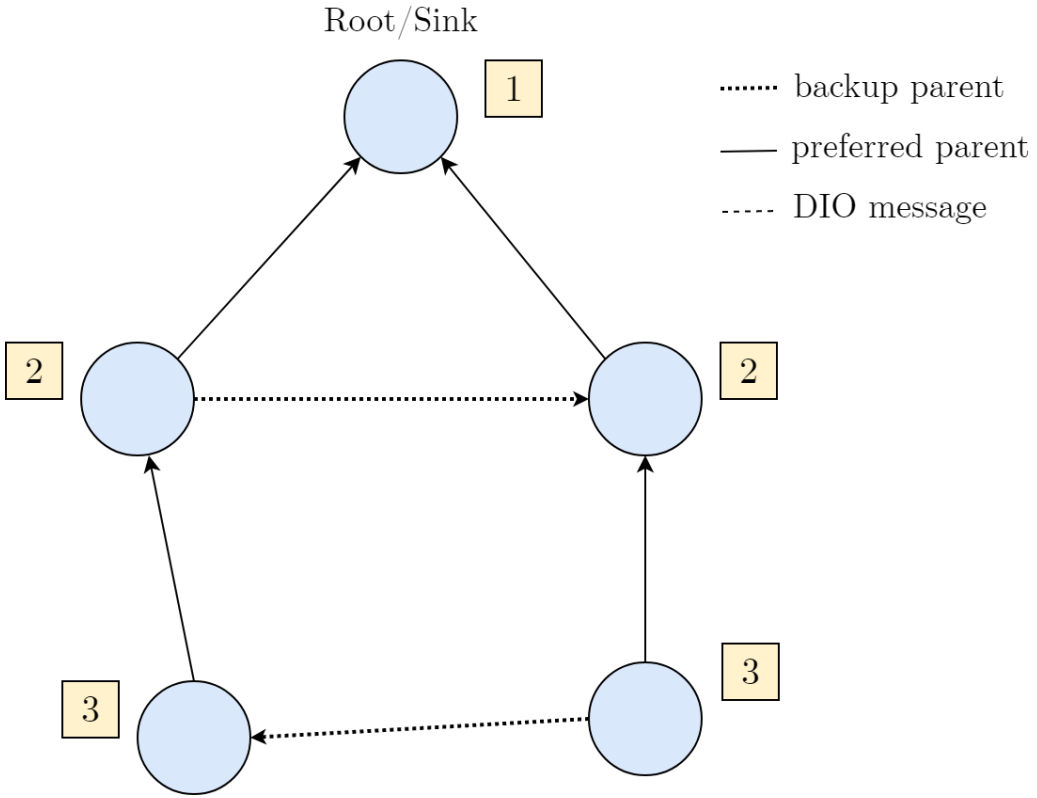

Figure 4: DODAG construction process

DAO Message

- Triggered by reception of DIO message
- Informs parent about node's reachability
- Propagates *upwards* with a DAODelay per node
- Transfers data for MP2P, P2P routing

DAO Message Propagation

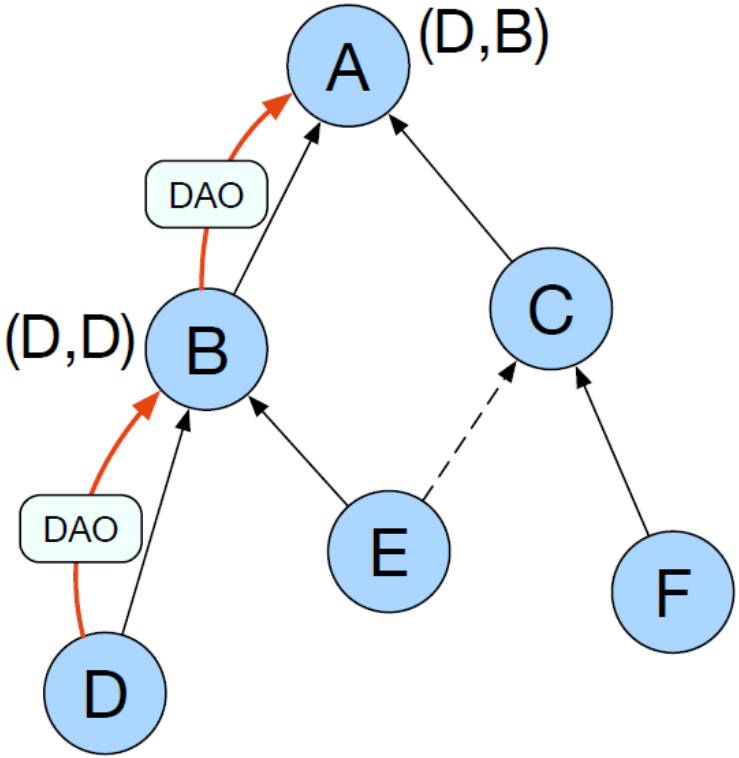

Figure 5: DAO message propagation in DODAG[1]

Operating Modes

- **Storing** — each node stores routing table based on information from DAO message
- **Non-storing** — only root performs source routing based on data from all DAO messages

MP2P Routing

For MP2P communication, each node forwards the message to the preferred parent, ultimately reaching the root–sink.

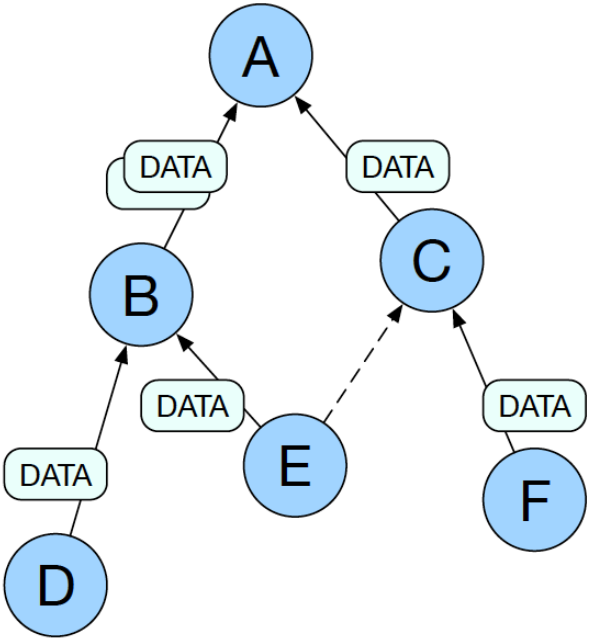

Figure 6: Multipoint-to-point routing[1]

P2P Routing in Storing Mode

Packet is forwarded *upwards* until a common ancestor is reached.

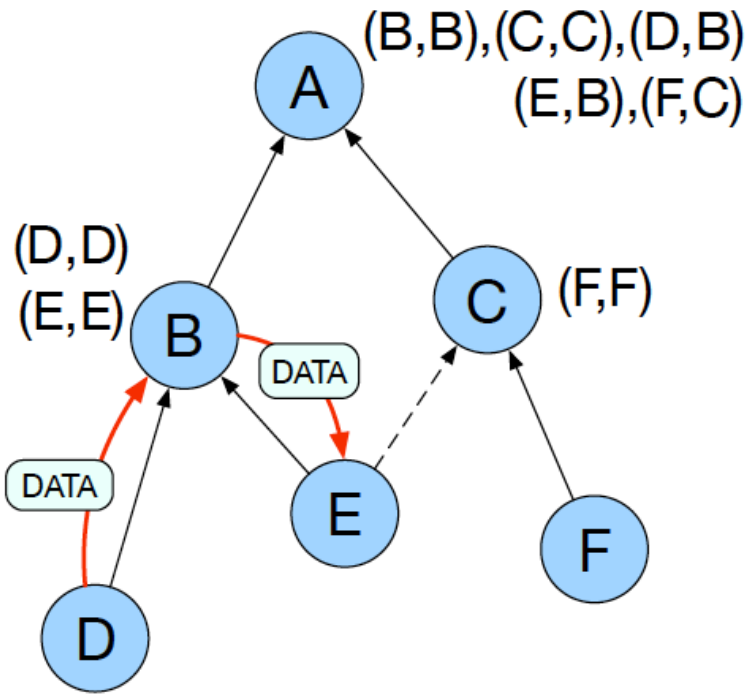

Figure 7: Point-to-point routing in storing mode[1]

P2P Routing in Non-Storing Mode

Packet is forwarded to the root, then source-routed to target.

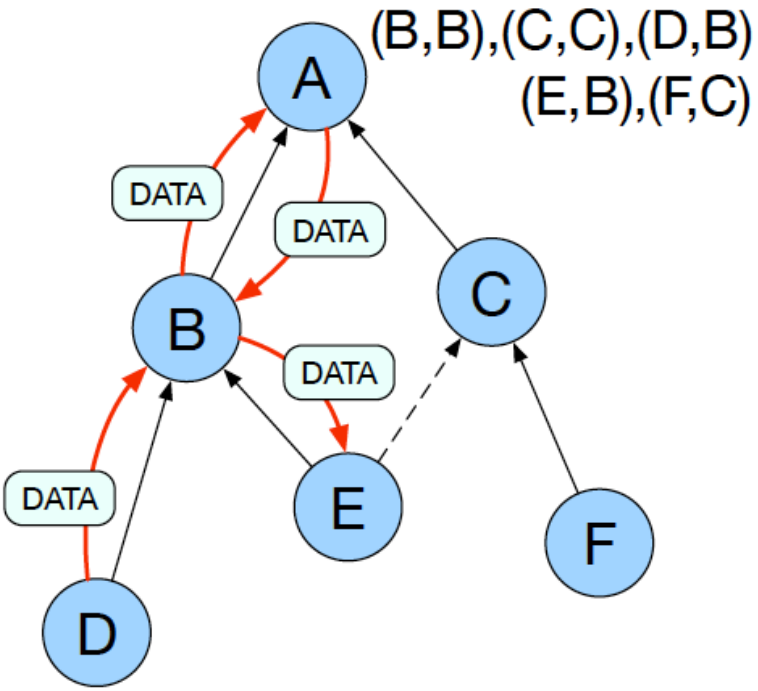

Figure 8: Point-to-point routing (D → E) in non-storing mode[1]

Trickle Algorithm

Defines the intervals between DIO transmissions in a way to:

- Minimize messaging frequency in stable network
- Ensure fast DODAG (re)construction

Trickle Algorithm Time Intervals

Timeline is divided into dynamic intervals of size I:

$$I_{max} = I_{min} * 2^{M} \tag{1}$$

where I_{min} — least possible time between two DIO transmissions, M — upper limit on number of times I is doubled.

Trickle Algorithm DIO Transmissions

If $< \sigma_{rc}$ messages heard from other nodes, DIO can be multicast, σ_{rc} – redundancy constant.

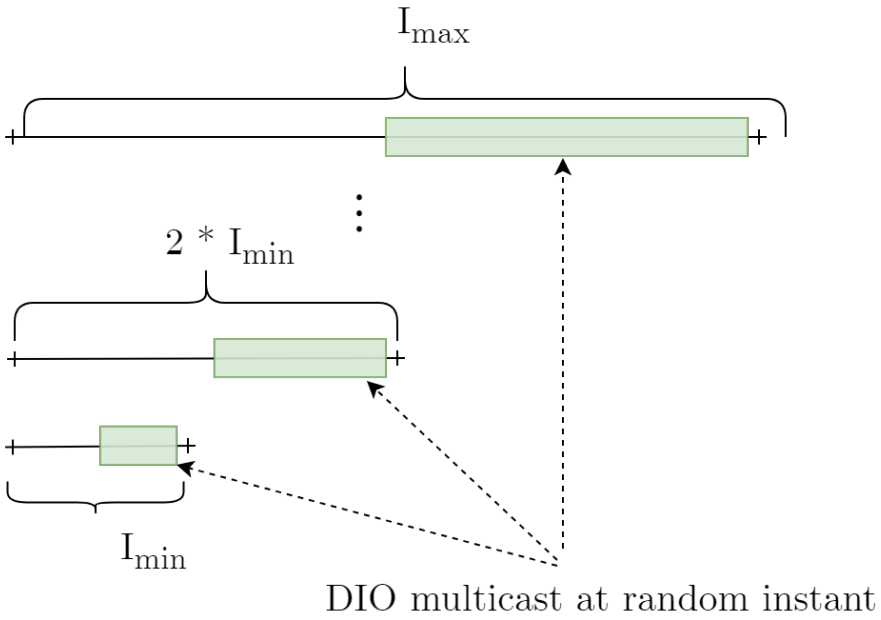

Figure 9: DIO transmission intervals

Trickle Timer

Trickle timer (interval I) reset to I_{min}, if inconsistency is detected:

- Node joins a new DODAG
- Incremented DAG sequence number (rebuild triggered)
- Possiblity of a loop

RPL Limitations

- Highly sensitive to trickle timer configuration, which is implementation-dependent
- MP2P and P2P traffic remain very costly in terms of memory and reliability[2]
- Poor performance in mobility scenarios[3]
- Limited support for heterogeneity of devices' resource constraints

[3] Kamgueu, Nataf, and Ndie, "Survey on RPL enhancements: a focus on topology, security and mobility".

RPL Improvements

- P2P route discovery mode[4] optimizes P2P communication by creating temporary DODAG with root at the sender
- DualMOP-RPL[5] allows to combine storing and non-storing modes in single instance
- RPLca+[6] and Trickle-L^2 mechanisms improve link quality estimation and reduce corresponding overhead[7]

[4] Goyal et al., *Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks.*

[5] Ko et al., "DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network".

[6] E. Ancillotti, Bruno, and Conti, "Reliable Data Delivery With the IETF Routing Protocol for Low-Power and Lossy Networks".

[7] Emilio Ancillotti et al., "Trickle-L²: Lightweight link quality estimation through Trickle in RPL networks".