RPL- Routing over Low Power and Lossy Networks

Michael Richardson
Ines Robles

IETF 94

Questions to answers today

1. What is a low power/lossy network? How does that relate to IoT?

2. What is RPL and how does it work?

3. Why couldn't we do this with other (IETF) routing protocols?

4. What are some applicability examples/real life deployments?

Questions to answers today

1. What is a low power/lossy network? How does that relate to IoT?

2. What is RPL and how does it work (high level)?

3. Why couldn't we do this with other (IETF) routing protocols?

4. What are some applicability examples/real life deployments?

Constrained Node

“ Constrained Node: A node where some of the characteristics that are otherwise pretty much taken for granted for Internet nodes at the time of writing are not attainable, often due to cost constraints and/or physical constraints on characteristics such as size, weight, and available power and energy. The tight limits on **power, memory, and processing resources** lead to hard upper bounds on state, code space, and processing cycles, making optimization of energy and network bandwidth usage a dominating consideration in all design requirements. Also, some layer-2 services such as full connectivity and broadcast/multicast may be lacking.” RFC 7228

Constrained Network

“ Constrained Network: A network where some of the characteristics pretty much taken for granted with link layers in common use in the Internet at the time of writing are not attainable.

Constraints may include:

o **low achievable bitrate/throughput** (including limits on duty cycle),

o **high packet loss and high variability of packet loss** (delivery rate),

o **highly asymmetric link characteristics**,

o **severe penalties for using larger packets** (e.g., high packet loss due to link-layer fragmentation),

o **limits on reachability over time** (a substantial number of devices may power off at any point in time but periodically "wake up" and can communicate for brief periods of time), and

o lack of (or severe constraints on) advanced services such as IP multicast.” RFC 7228

Constrained-Node Network

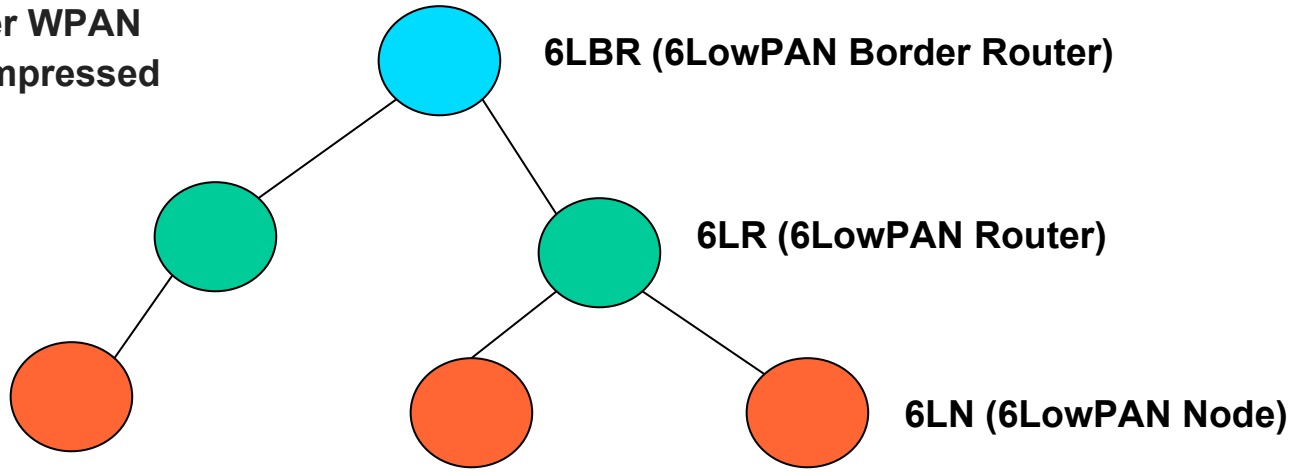
“ Constrained-Node Network: A network whose characteristics are influenced by being composed of a significant portion of constrained nodes.

A constrained-node network always is a constrained network because of the network constraints stemming from the node constraints, but it may also have other constraints that already make it a constrained network.” - RFC 7228

LLN: Low-Power and Lossy Network

“ LLN: Low-Power and Lossy Network. Typically composed of many embedded devices with **limited power, memory, and processing resources** interconnected by a variety of links, such as IEEE 802.15.4 or low-power Wi-Fi. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (heating, ventilation, and air conditioning (HVAC), llighting, access control, fire), connected home, health care, environmental monitoring, urban sensor networks, energy management, assets tracking, and refrigeration.” RFC 7228

IPv6 over Low power WPAN (6lowpan) - IPv6 compressed



Questions to answers today

1. What is a low power/lossy network? How does that relate to IoT?

2. What is RPL and how does it work ?

3. Why couldn't we do this with other (IETF) routing protocols?

4. What are some applicability examples/real life deployments?

RPL is a ...

- Distance Vector (DV) protocol
- Source Routing Protocol

What is a Distance Vector (DV) protocol?

- The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network
- Intra-domain routing protocol
- Requires that a router inform its neighbors of topology changes periodically
- Have less computational complexity and message overhead

What is a Distance Vector (DV) protocol?

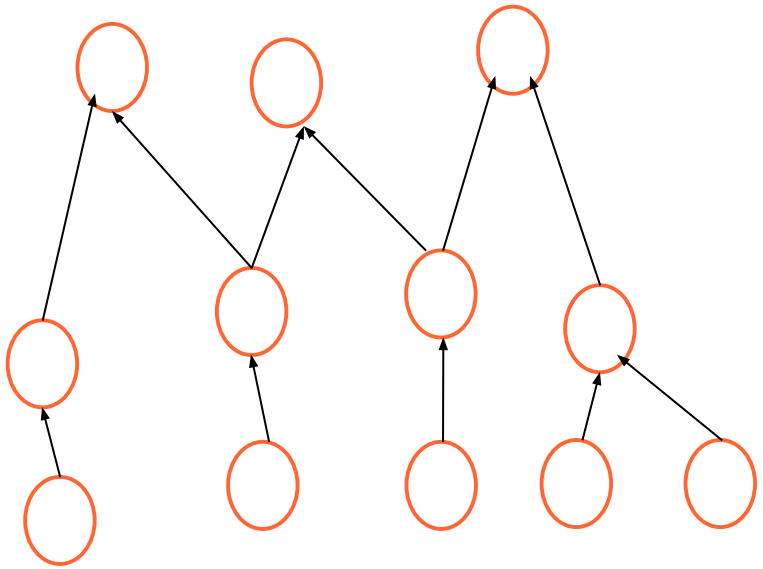
- Distance-vector protocols are based on calculating the Direction and Distance to any link in a network.
  - "Direction" usually means the next hop address and the exit interface.
  - "Distance" is a measure of the cost to reach a certain node.
- The least cost route between any two nodes is the route with minimum distance.
- Each node maintains a vector (table) of minimum distance to every node.
- The cost of reaching a destination is calculated using various route metrics

What is a Source Routing (path addressing) protocol?

Allows a sender of a packet to partially or completely specify the route the packet takes through the network.

Enables a node to discover all the possible routes to a host.

RPL organizes a topology as a...



Directed

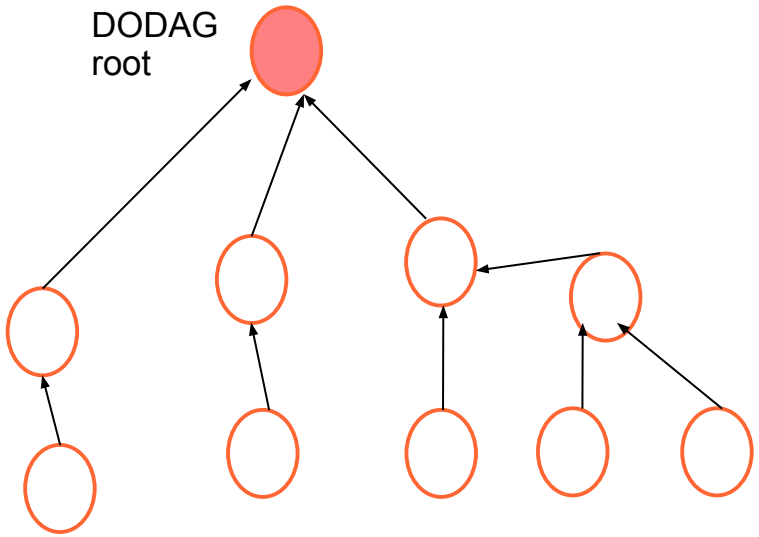
Acyclic

Graph

(DAG)

That is....

...Partitioned into one or more...



Destination

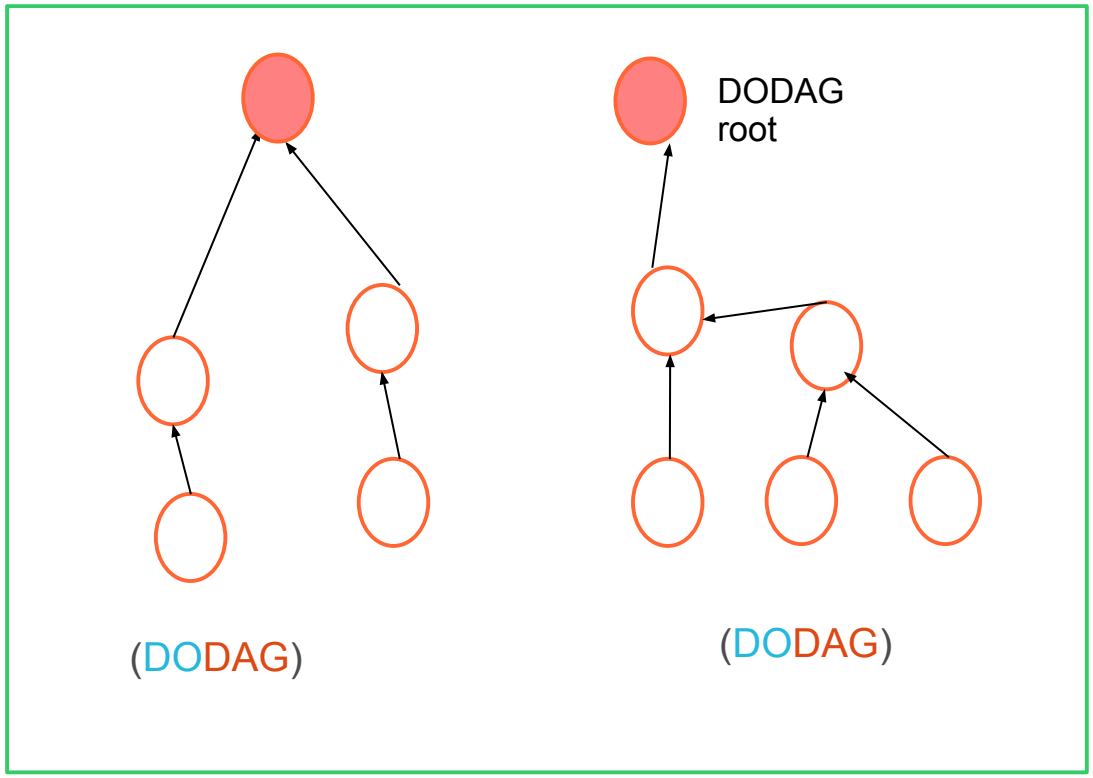
Oriented

DAGs

(DODAG)

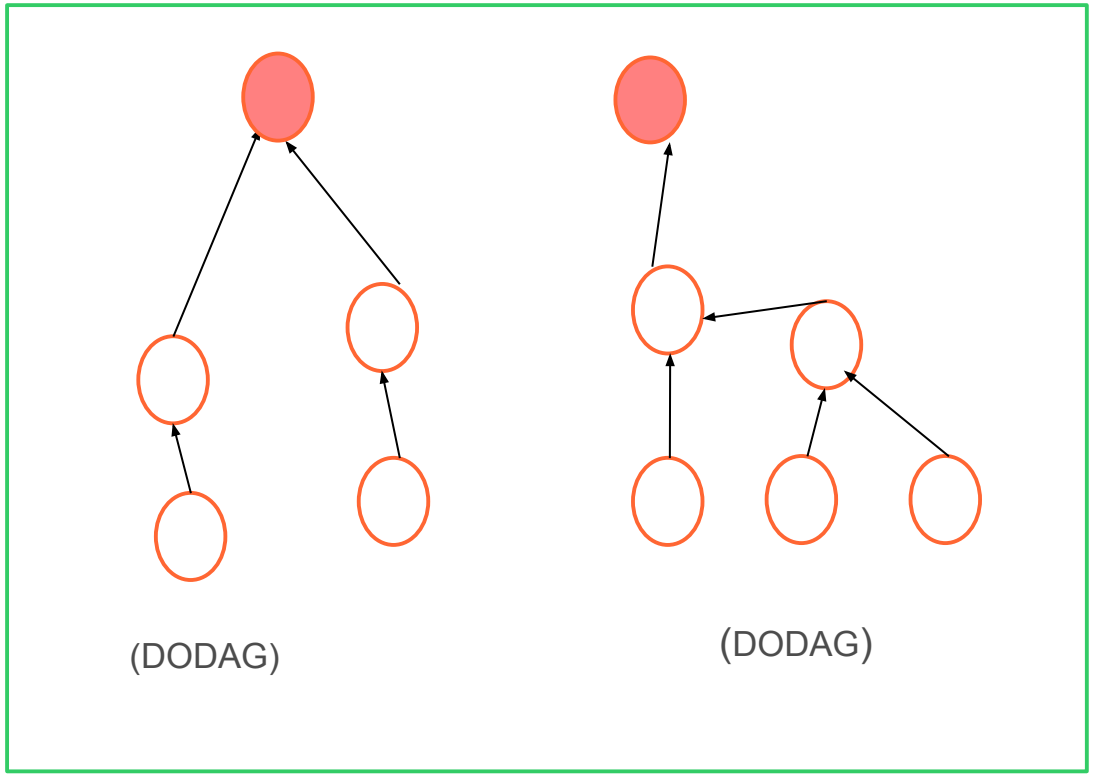
A DAG rooted at a single destination at a single DAG root (DODAG root) with no outgoing edges

A RPL Instance is a set of one or more DODAGs that share a RPLInstanceID.



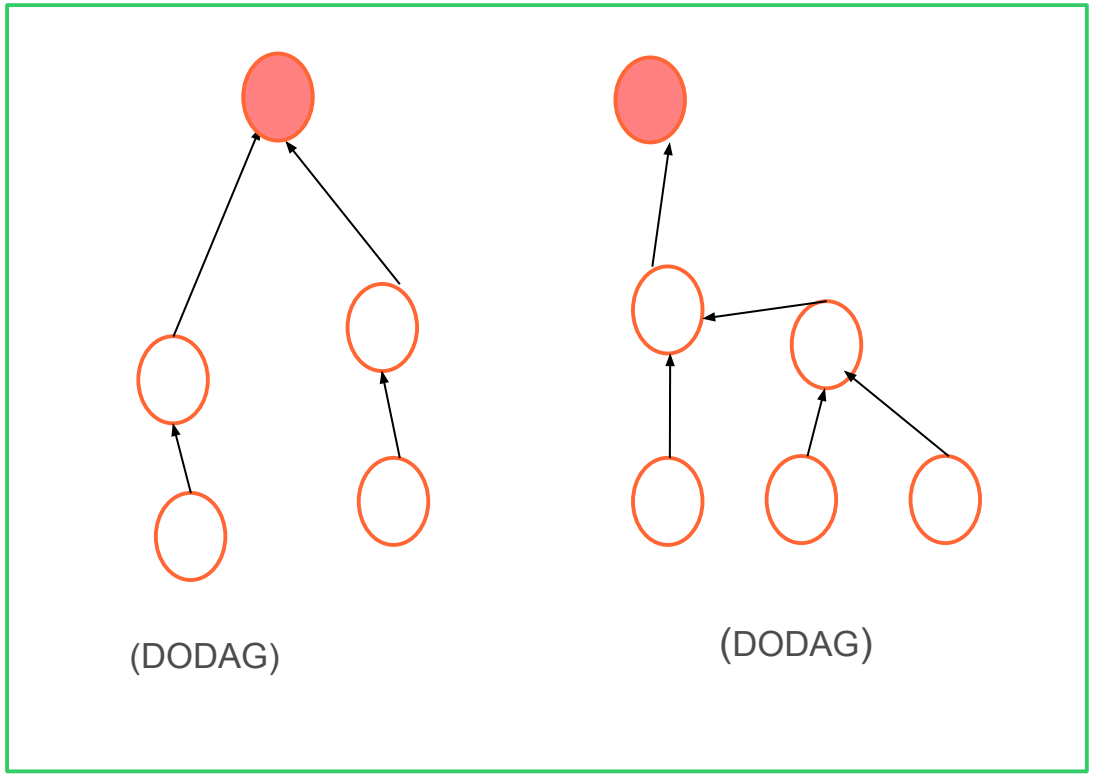
RPL Instance

To Identify and maintain a topology RPL uses...



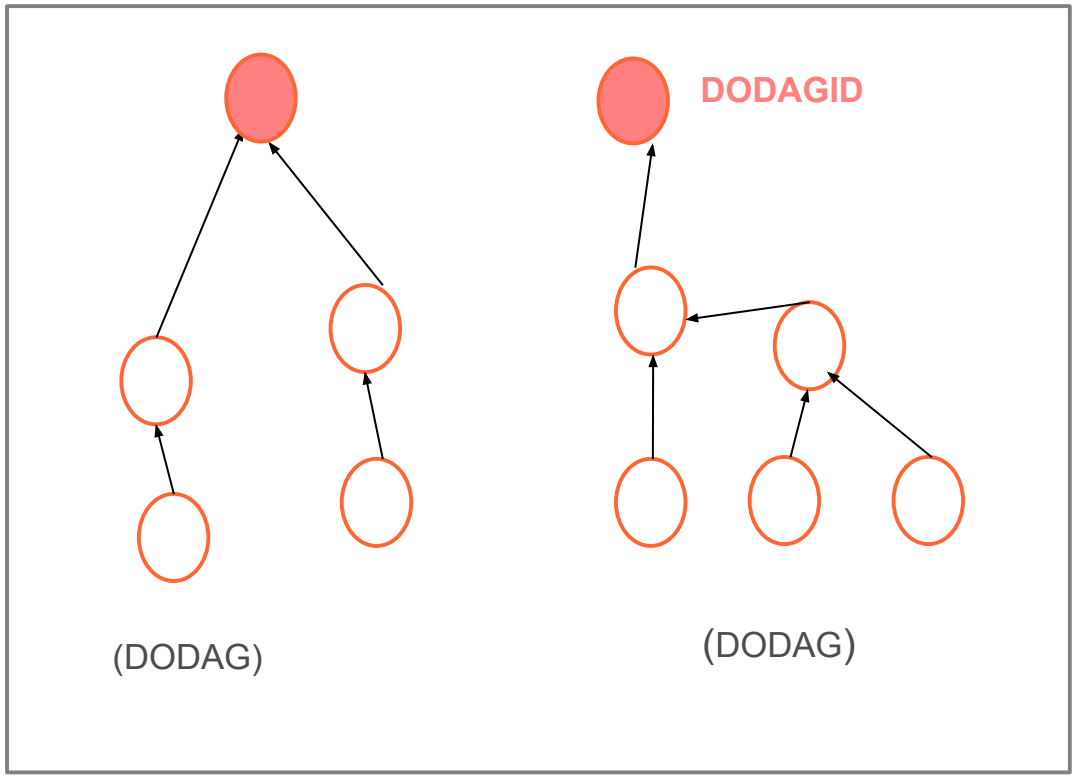
RPL Instance

To Identify and maintain a topology RPL uses...



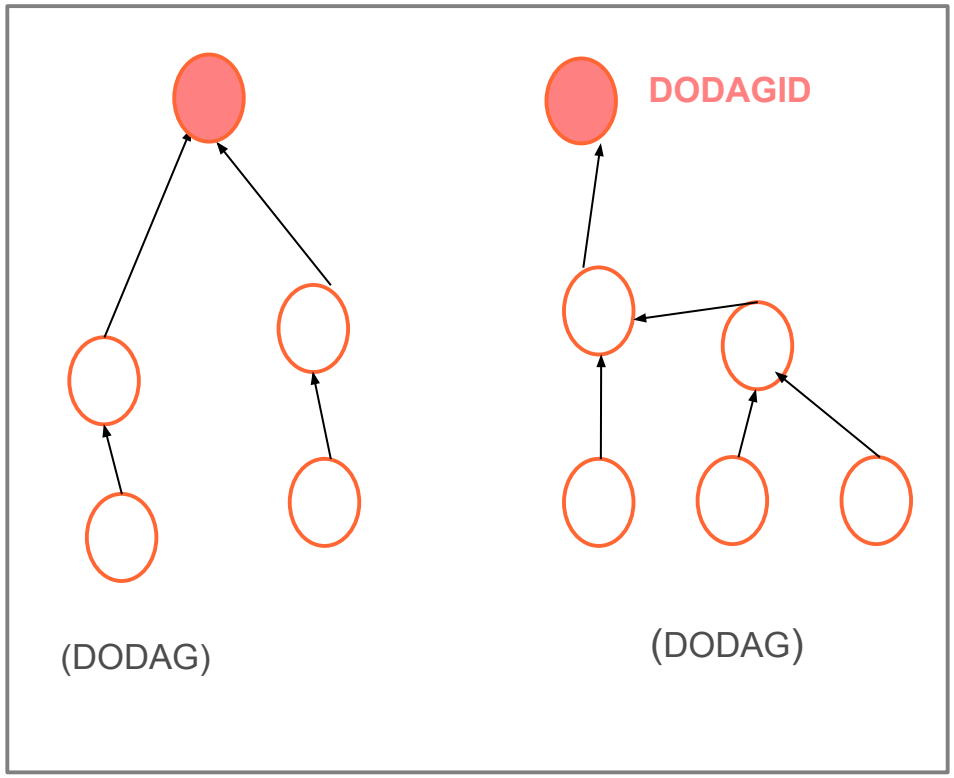
RPLInstanceID is a **unique** identifier within a network. DODAGs with the same RPLInstanceID share the same Function (OF) used to compute the position of node in the DODAG .

To Identify and maintain a topology RPL uses...



RPLInstanceID is a unique identifier within a network.

To Identify and maintain a topology RPL uses...



DODAGVersionNumber

A DODAGVersion is a specific iteration of a DODAG with a given DODAGID

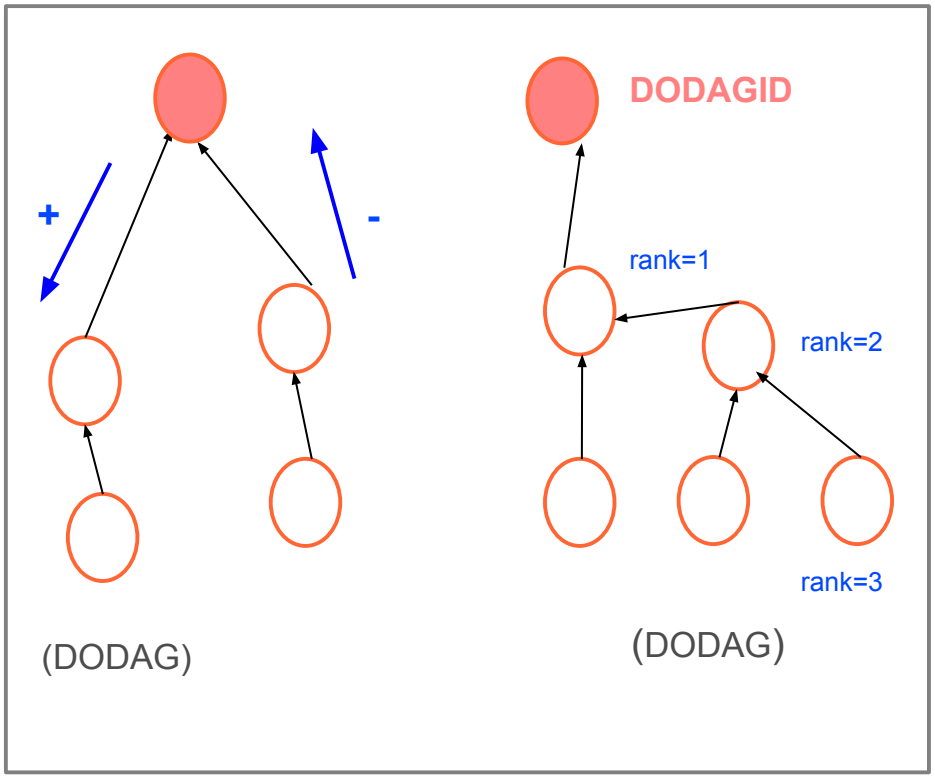
A DODAGVersionNumber Is a sequential counter that is incremented by the root to form a new version

RPLInstanceID is a unique identifier within a network.

To Identify and maintain a topology RPL uses...

Rank

Defines the node's Individual position Relative to other nodes with respect to DODAG root



DODAGVersionNumber

A DODAGVersion is a specific iteration of a DODAG with a given DODAGID

A DODAGVersionNumber Is a sequential counter that is incremented by the root to form a new version

RPLInstanceID is a unique identifier within a network.

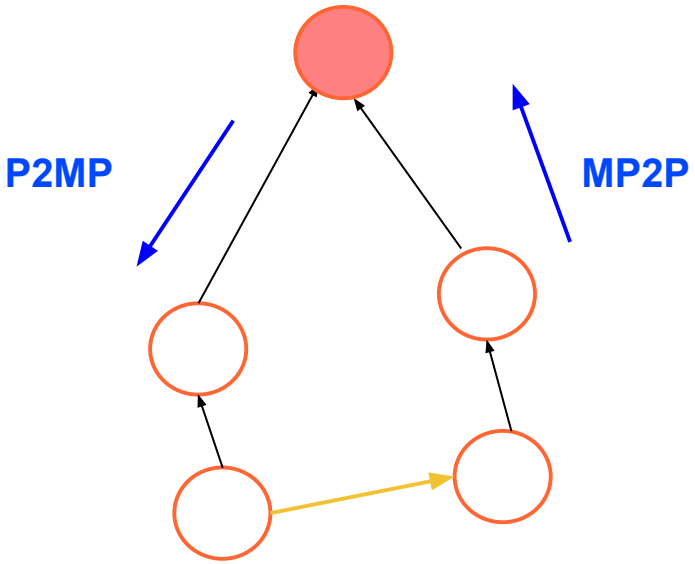
Grounded and Floating DODAG

- A grounded DODAG offers connectivity to hosts that are required for satisfying the application goal
- A floating DODAG is not expected to satisfy the goal, it only provides routes to nodes within the DODAG. e.g, provide interconnectivity during repair

Storing and Non-Storing Mode-of-Operation

- A storing LLN keeps a downward routing table at each node.
  - traffic travels only as far as common parent.
  - storing mode limited by size of routing table
    - nodes with lower rank, have bigger tables!
    - protocol fails when any table is full.
- A non-storing LLN sends **all** traffic to root. Root uses source routes to send traffic to leafs.
  - limited by ram of DODAG root/6LBR, but usually non-constrained device
  - requires more bits on wire, which often is more expensive (energy-wise) than more ram, or compute cycles.
- new work (“dao-projection”) tries to add some routes to non-storing mode.
- original protocol (pre-2011) thought to mix and match, but this proved unworkable.

Traffic Flows Supported by RPL



(DODAG)

- MP2P
- P2MP
- P2P (special DODAG, always non-storing)

RPL Instance

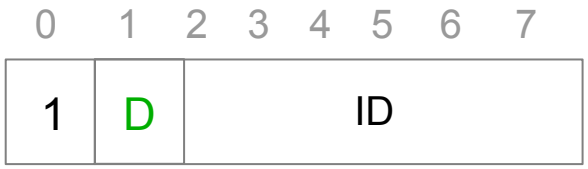
- A RPL Node may belong to multiple RPL Instances, and it may act as router in some and as a leaf in others.
- Type: Local and Global
- Control and Data packets has a RPLInstance field.

Global RPL Instance

- Are coordinated, have one or more DODAGs, and are typically long-lived.
- A global RPLInstanceID must be unique to the whole LLN.

Local RPL Instance

- Are always a single DODAG whose singular root owns the corresponding DODAGID and allocates the local RPLInstanceID

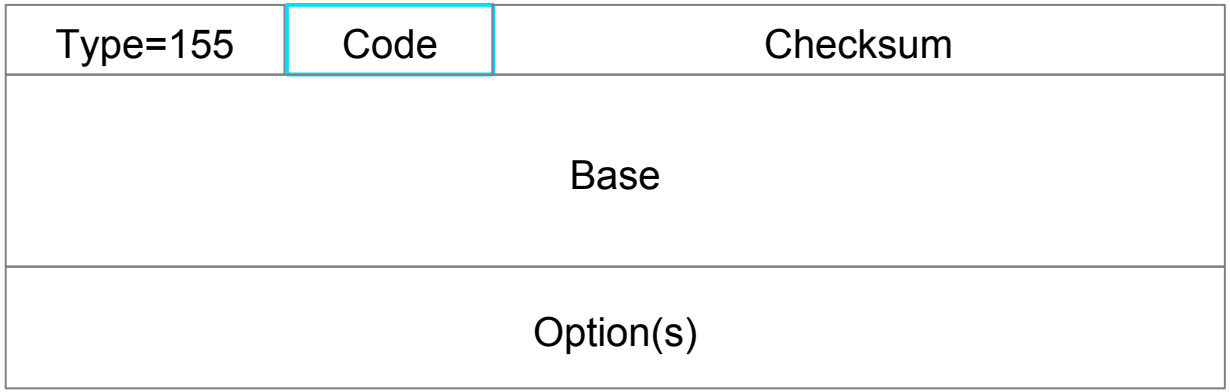


Local RPLInstanceID in 0...63

D=0 in control messages

D is used in data packets to indicate whether the DODAGID is the source or Destination of the packet. D=1 the dest. Address of the packet must be the DODAGID.

RPL Control messages are ICMPv6 messages



Code: Identify the type of control message

0x00 → DODAG Information Solicitation (DIS)

0x01 → DODAG Information Object (DIO)

0x02 → Destination Advertisement Object (DAO)

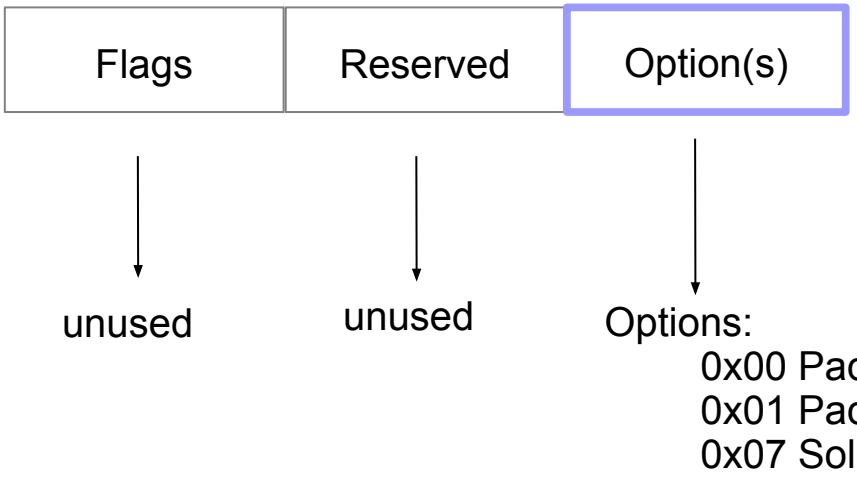
0x03 → DAO-ACK

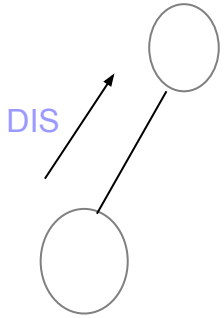
DODAG Information Solicitation (DIS)

Solicit a DODAG Information Object (DIO) from a RPL node

Its use is analogous to that of a Router Solicitation of IPv6 Neighbor Discovery

The DIS Base Object:

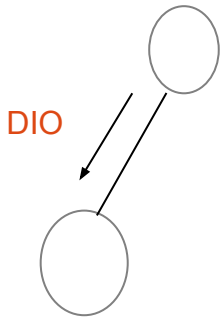




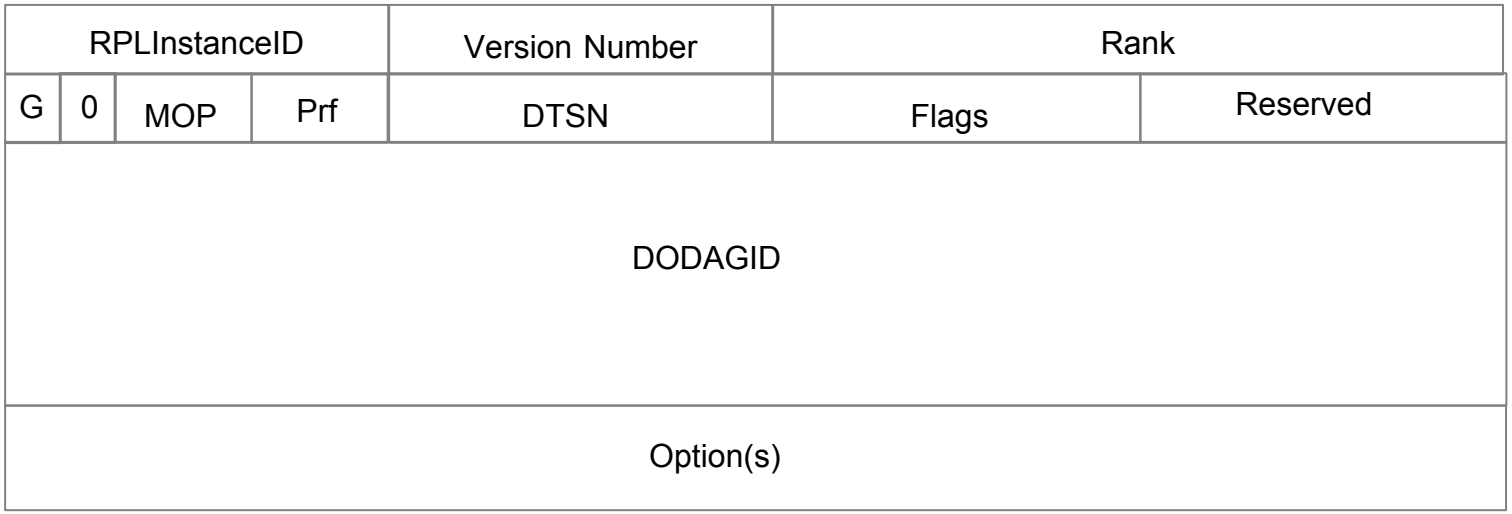
DODAG Information Object (DIO)

Carries information that allows a node to:

- Discover a RPL instance
- Learn its configuration parameters
- Select a DODAG parent set
- Maintain the DODAG



DODAG Information Object (DIO)



DIO Base Object

DIO Transmission

RPL nodes transmit DIOs using a Trickle Timer.

Trickle's basic primitive is simple: every so often,

> a node transmits data unless it hears a few other transmissions whose data suggest its own transmission is redundant.

rfc6206 - was published seperately and describes the trickle algorithm

- used by Homenet configuration protocols:
  - draft-ietf-homenet-dncp-12 (Distributed Node Consensus Protocol)
  - draft-ietf-homenet-hncp-09 (Home Networking Control Protocol)
- Babel uses similar/equivalent mechanism, btw.

The configuration parameters of the Trickle timer are specified as follows:

Imin: learned from the DIO message as (2^DIOIntervalMin) ms.

Imax: learned from the DIO message as DIOIntervalDoublings.

k: learned from the DIO message as DIORedundancyConstant. The default value of DIORedundancyConstant is DEFAULT_DIO_REDUNDANCY_CONSTANT. In RPL, when k has the value of 0x00, this is to be treated as a redundancy constant of infinity in RPL, i.e., Trickle never suppresses messages.

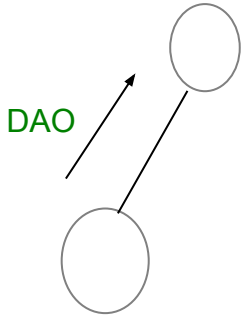
Destination Advertisement Object (DAO)

- Used to propagate destination information Upward along the DODAG.

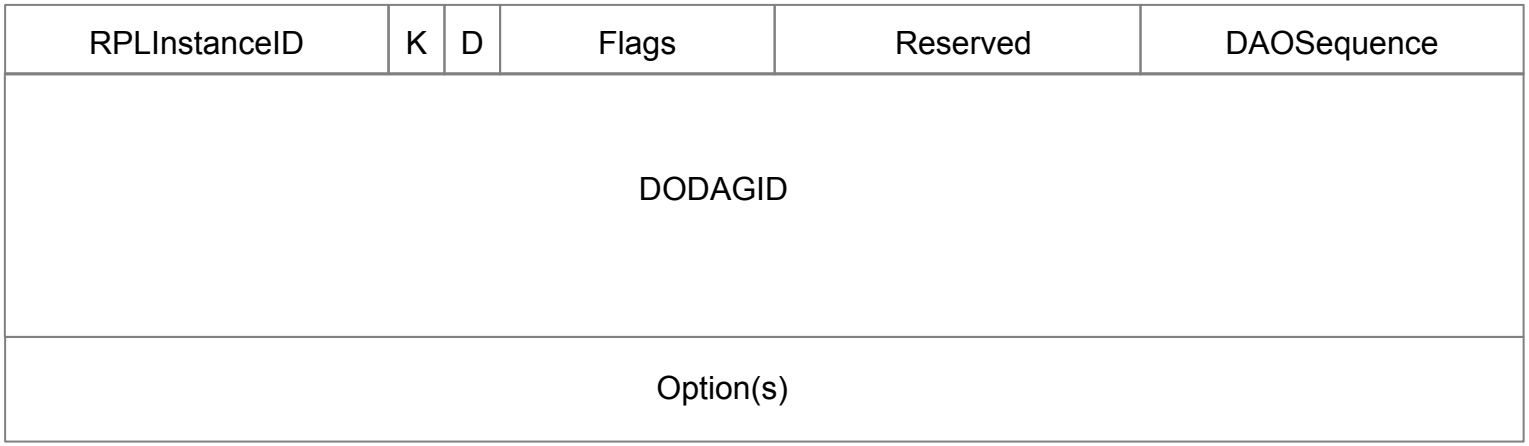
- In **Storing** mode, the DAO message is unicast by the child to the selected parent (s).

- In **Non-Storing mode**, the DAO message is unicast to the DODAG root.

- The DAO message may optionally, upon explicit request or error, be acknowledged by its destination with a Destination Advertisement Acknowledgement (DAO-ACK) message back to the sender of the DAO.

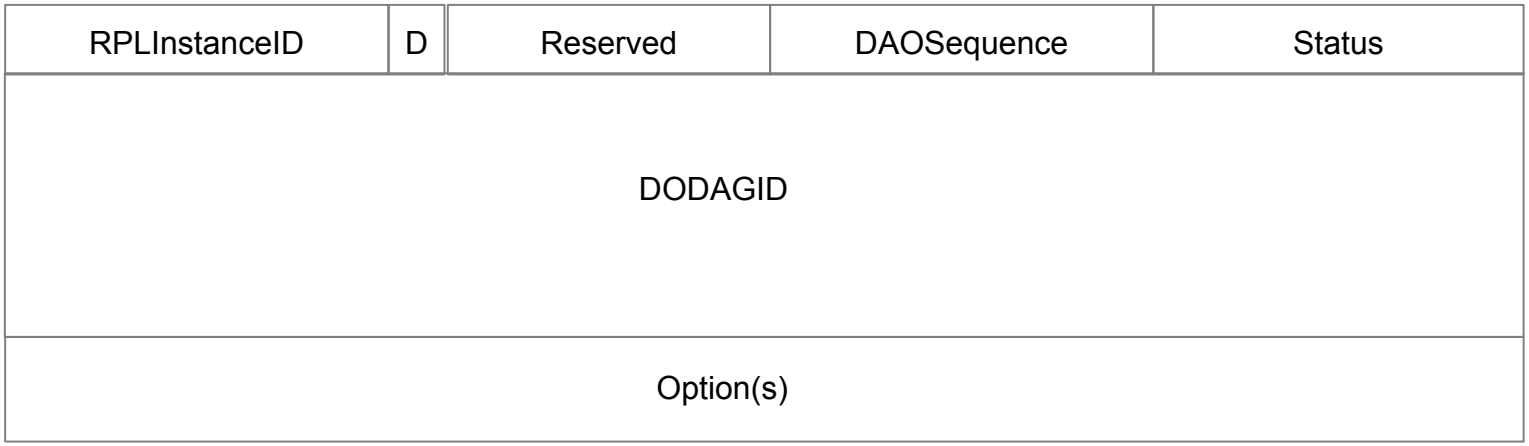


Destination Advertisement Object (DAO)



DAO Base Object

Destination Advertisement Object Acknowledgement (DAO-ACK)



DAO-ACK Base Object

Operation as Leaf Node

A RPL node may attach to a DODAG as a leaf node only.
One example of such a case is when a node does not understand or does not support (policy) the RPL Instance's OF or advertised metric/constraint,the node may either join the DODAG as a leaf node or may not join the DODAG.

A node operating as a leaf node must obey the following rules:

1. It MUST NOT transmit DIOs containing the DAG Metric Container.

2. Its DIOs MUST advertise a DAGRank of INFINITE_RANK.

3. It MAY suppress DIO transmission, unless the DIO transmission has been triggered due to detection of inconsistency when a packet is being forwarded or in response to a unicast DIS message, in which case the DIO transmission MUST NOT be suppressed.
4. It MAY transmit unicast DAOs

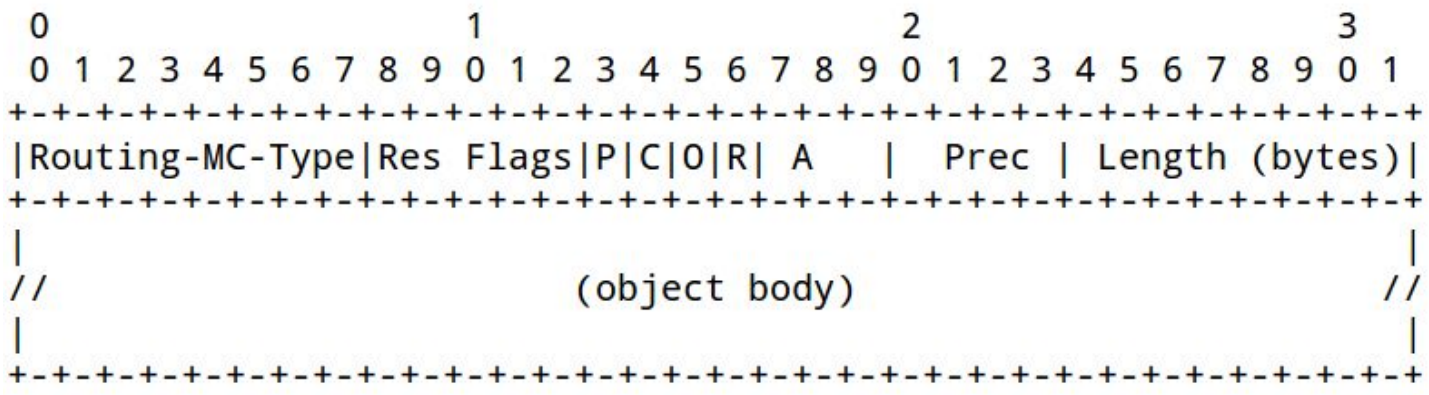
5. It MAY transmit multicast DAOs to the '1 hop' neighborhood

DAG Metric Container

The DAG Metric Container option MAY be present in DIO or DAO messages

The DAG Metric Container is used to report metrics along the DODAG.

Type = 0x02	Option Length	Metric Data



Node Metric/Constraint Objects

- Node State and Attribute Object
  - Propose to reflect Node workload (CPU, Memory, etc)
- Node Energy Object
  - Constraint
  - three types of power sources: "powered", "battery", and "scavenger"
- Hop Count Object
  - Can be used as metric or constraint
  - Constraint: max number of hops can be traversed
  - Metric: total number of hops traversed

Link Metric/Constraint Objects

- Throughput Object:
  - Currently available throughput (Bytes per second)
- Latency:
  - Can be used as a metric or constraint
  - Constraint: Max latency allowable on path
  - Metric: aditive metric updated along path
- Link Reability:
  - Link Quality Level Reliability (LQL): 0=Unknown, 1=High, 2=Medium, 3=Low
  - Expected Transmission Count (ETX) (Average number of TX to deliver a packet)
- Link Colour:
  - Metric or constraint, arbitrary admin value

Objective Function (OF)

- Define how RPL nodes select and optimize routes within a RPL Instance.
- Define how nodes translate one or more metrics into a rank.
- Define how nodes select parents

Objective Function (OF)

- OF0: Objective Function Zero
- Minimum Rank with Hysteresis OF.

We have only begun to scratch the surface of Objective Functions.

Objective Function Zero (OF)

- OF0 is designed as a default OF that will allow interoperation between implementations in a wide spectrum of use cases
- Objective Function Zero is designed to find the nearest Grounded root
- OF0 selects a preferred parent and a backup feasible successor if one is available. All the upward traffic is normally routed via the preferred parent with no attempt to perform any load balancing

Minimum Rank with Hysteresis Objective Function (MRHOF) (RFC 6719)

- Objective Function that selects routes that minimize a metric, while using hysteresis to reduce churn in response to small metric changes.
- MRHOF works with additive metrics along a route, and the metrics it uses are determined by the metrics that the DIO messages advertise.
  - For example, the use of MRHOF with the latency metric allows RPL to find stable minimum-latency paths from the nodes to a root in the Directed Acyclic Graph (DAG) instance
- pronounced “Mister Hoff”

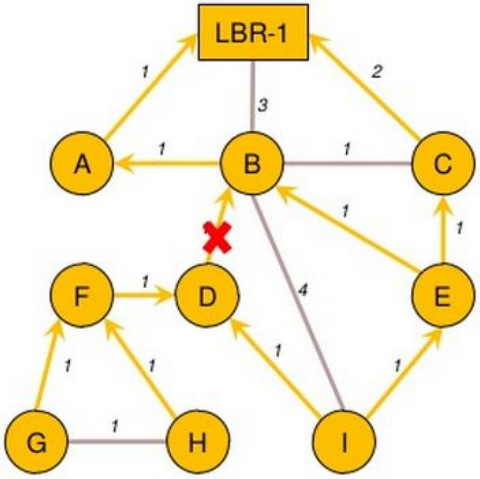
Minimum Rank with Hysteresis Objective Function (MRHOF)

- The Minimum Rank with Hysteresis Objective Function, MRHOF, is designed to find the paths with the smallest path cost while preventing excessive churn in the network. It does so by using two mechanisms.
  - First, it finds the minimum cost path, i.e., path with the minimum Rank.
  - Second, it switches to that minimum Rank path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. This second mechanism is called "hysteresis".
- MRHOF may be used with any additive metric as long as the routing objective is to minimize the given routing metric.
- Nodes MUST support at least one of these metrics: hop count, latency, or ETX.

Node/link Metric	Rank
Hop-Count	Cost
Latency	Cost/65536
ETX	Cost

Conversion Metric to Rank

RPL has mechanism for loop detection and DODAG Repair



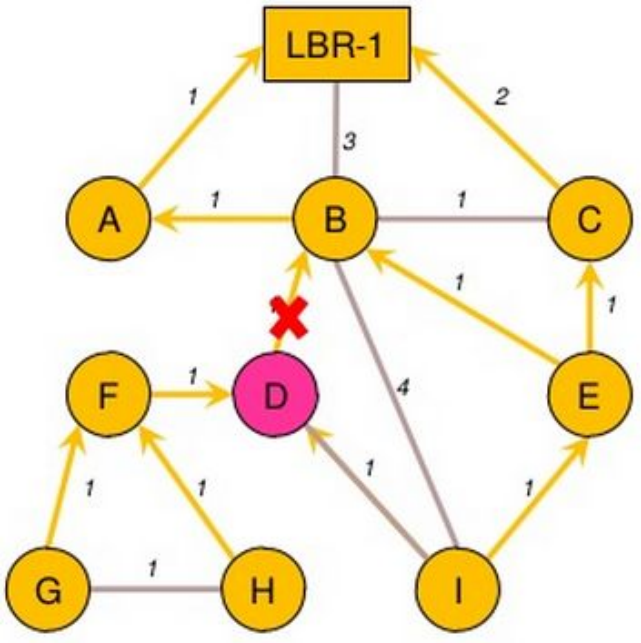
Suppose link between nodes B and D is broken: (for instance, a metal door closes!)

- Node D type node B in its list

- Parent Node D is no longer any time in grounded DODAG Parent, so it will be the root of floating DAG itself

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair



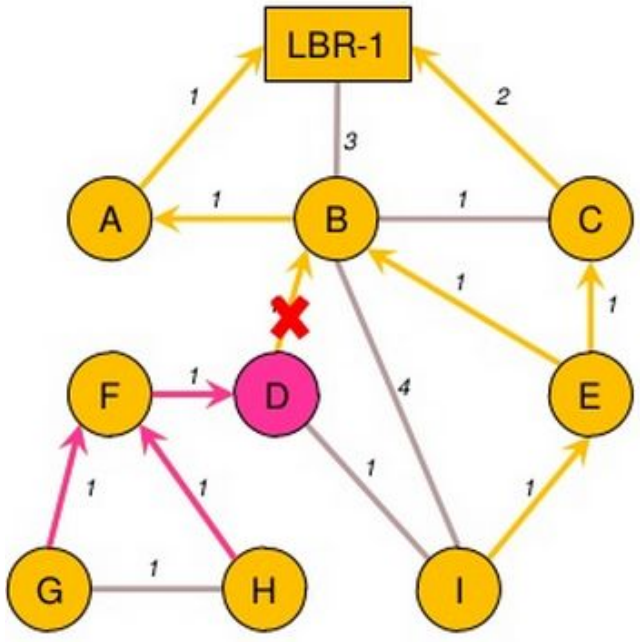
Node D play DIO to notify change of sub-DAG

- Node I has alternate parent E, so it does not leave the DAG of LBR-1

- I eliminates Node D Node from possible Parents list

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair



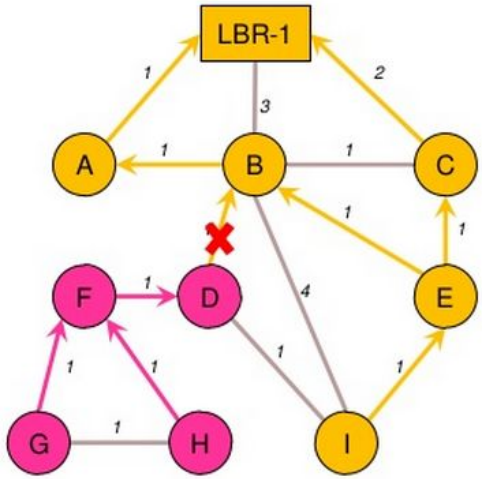
- Node F follows D node, as D leaves LBR-1 DAG: it has no choice

- Node F hears DIO from D.

- Node G and H also follow floating node F DAG

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair

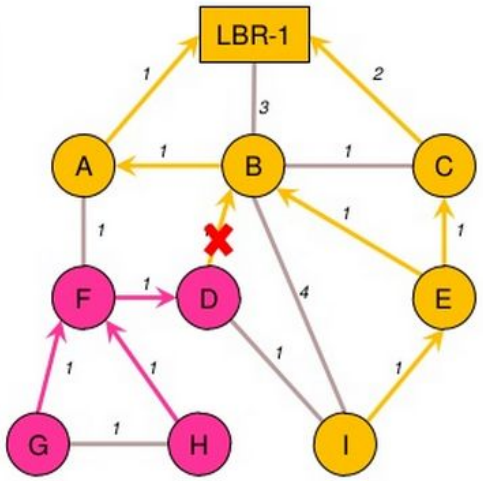


Node I found DIO

 Node D listens to DIOs for opportunities to re-enter the last Grounded with depth 5 Node I

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair



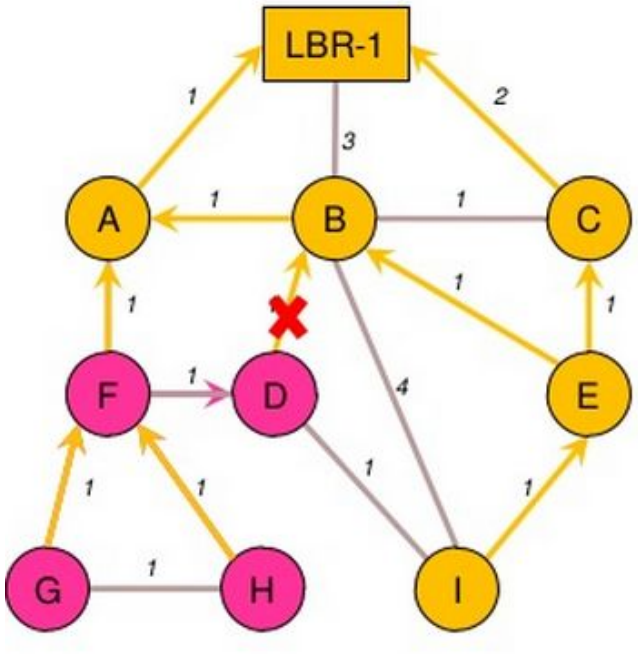
- Suppose link between A and F is set
- Node A send DIO

- Node F release notice Grounded DAG re-entry opportunities with depth 2 through node A DAG

- Hop Node F started with 1 cycle timer associated with the node A

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair

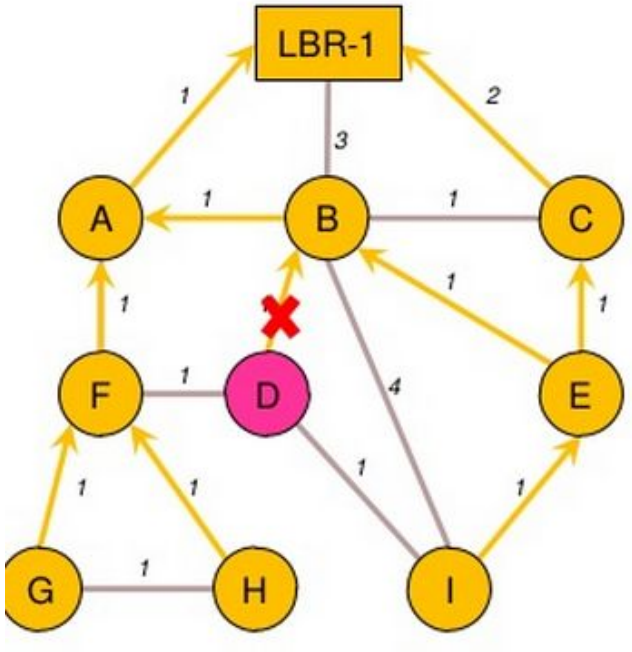


- DAG node F Timer goes off, issues DIS, receives DIO from A!

- Node F Grounded DAG with depth 2 by adding the Parent A
- Node F send DIO with new rank/etc.
- Node G and H join to the Grounded DODAG through F

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair

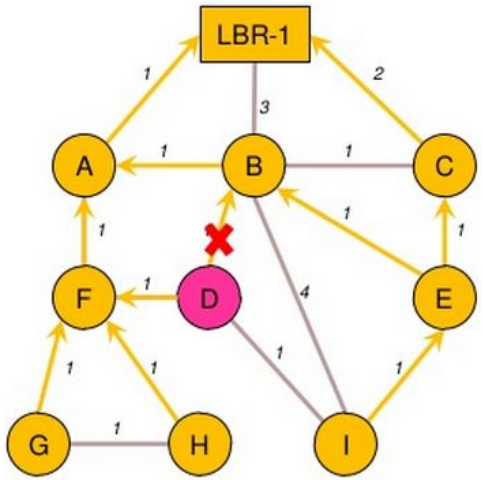


Node D hears new DIO from F.

Node D start DAG Hop cycle timer with 2 attached to node F, while other timer is running DAG Hop with 4 cycles associated with the first node

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

RPL has mechanism for loop detection and DODAG Repair



- DAG node D Hop timer with 2 cycles tend to end first.
- Node D engaged with depth 3 Grounded

DAG by adding Node F's parent.

- End

Source: http://www.slideshare.net/vinatech/slide-tt?related=1

Questions to answers today

1. What is a low power/lossy network? How does that relate to IoT?

2. What is RPL and how does it work?

3. Why couldn't we do this with other (IETF) routing protocols?

4. What are some applicability examples/real life deployments?

draft-ietf-roll-protocols-survey (not published)

https://datatracker.ietf.org/doc/draft-ietf-roll-protocols-survey/ (from 2009)

9. Conclusion

Figure 1 shows that no existing IETF protocol specification meets the criteria described in Section 4. Therefore, having a routing protocol for LLNs requires new protocol specification documents. Whether such documents describe modifications to existing protocols or new protocols it outside the scope of this document and warrants further discussion. However, the results in Figure 1 may provide some insight or guidance in such a discusssion, indicating what protocol mechanisms may be better suited to LLNs than others.

Questions to answers today

1. What is a low power/lossy network? How does that relate to IoT?

2. What is RPL and how does it work?

3. Why couldn't we do this with other (IETF) routing protocols?

4. What are some applicability examples/real life deployments?

RPL Implementations

- Open Source
  - **ContikiRPL → https://github.com/contiki-os/contiki/tree/master/core/net/rpl**
    - **used in multiple companies, with some variations among them**
  - **TinyRPL → https://github.com/tinyos/tinyos-main/tree/master/tos/lib/net/rpl**
  - **Unstrung → http://unstrung.sandelman.ca/**
    - **intended for gateways and other non-constrained (class >2) systems**
- **https://tools.ietf.org/html/draft-hui-vasseur-roll-rpl-deployment-01**
- **A lot of Academia papers evaluating the performance of RPL**
- **Known to be a number of proprietary implementations:**
  - **Cisco (more than one?), Huawei, Itron*, Landisgyr*, Sigma Design,**

"*" companies you haven't heard of at the IETF before

RPL adapted to Mobility

- RPL was designed for static sensor networks
- But, there are implementations that modify RPL and adapt it to mobility environments, such as:
  - mRPL - smart-HOP RPL, a hand-off mechanism within RPL
  - MT-RPL - Mobility-Triggered RPL, a cross-layer protocol operating at layers 2 and 3.
  - RPL-Vanet - RPL for vehicular environments.

Conclusion

- RPL is the routing protocol for Low Power and Lossy Networks developed in ROLL IETF Working Group
- RPL Control Messages are used to build a topology
- Implementations were developed and help to identify features to improve the protocol

Arigatou!

;-)

Q & A

Back up Slides

A bit more from ROLL…. ;-)

ROLL Documents

Requirements

- Routing Requirements for Urban Low-Power and Lossy Networks - RFC 5548
- Industrial Routing Requirements in Low-Power and Lossy Networks - RFC 5673
- Home Automation Routing Requirements in Low-Power and Lossy Networks - RFC 5826
- Building Automation Routing Requirements in Low-Power and Lossy Networks - RFC 5867

Terminology: Terms Used in Routing for Low-Power and Lossy Networks - RFC 7102

Methods/Algorithms used by RPL

- The Trickle Algorithm - RFC 6202
- Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks - RFC 6551
- Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL) - RFC 6552
- The Minimum Rank with Hysteresis Objective Function - RFC 6719

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks - RFC 6550

RPL-P2P

- Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks - RFC 6997
- A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network - RFC 6998

Security:

- A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs) - RFC 7416

Active I-D

draft-ietf-roll-admin-local-policy-03: Forwarder policy for multicast with admin-local scope in the Multicast Protocol for Low power and Lossy Networks (MPL)

draft-ietf-roll-applicability-ami-11 : Applicability Statement for the Routing Protocol for Low Power and Lossy Networks (RPL) in AMI Networks

draft-ietf-roll-applicability-home-building-12 : Applicability Statement: The use of the RPL protocol suite in Home Automation and Building Control

draft-ietf-roll-applicability-template-07: ROLL Applicability Statement Template

draft-ietf-roll-mpl-parameter-configuration-07 : MPL Parameter Configuration Option for DHCPv6

draft-ietf-roll-trickle-mcast-12 : Multicast Protocol for Low power and Lossy Networks (MPL)

Related Internet-Drafts

	Document	Date	Status	IPR
	Related Internet-Drafts			
	draft-robles-roll-useofrplinfo-02 **When to use RFC 6553, 6554 and IPv6-in-IPv6**	2015-10-19 29 pages New	I-D Exists	
	draft-tan-roll-clustering-00 **RPL-based Clustering Routing Protocol**	2015-06-25 10 pages	I-D Exists	
	draft-thubert-roll-dao-projection-01 **Root initiated routing state in RPL**	2015-10-19 13 pages New	I-D Exists	1
	draft-turner-roll-dio-ctx-00 **RPL DIO Option for Specifying Compression Contexts**	2015-09-28 5 pages	I-D Exists	
	draft-wang-roll-adaptive-data-aggregation-00 **Design of Adaptive Data Aggregation Schemes**	2015-10-18 8 pages New	I-D Exists	

And still a bit more… :-)

http://www.ietf.org/mail-archive/web/roll/current/maillist.html#01252

draft-ietf-roll-protocols-survey: Criteria to evaluate existing protocols

- routing state
- loss response
- control cost
- link cost
- node cost

Results of draft-ietf-roll-protocols-survey

Protocol	Routing State	Loss Response	Control Cost	Link Cost	Node Cost
OSPF/IS-IS	Fail	Fail	Fail	Pass	Fail
OLSRv2	Fail	?	?	Pass	Pass
TBRPF	Fail	Pass	Fail	Pass	?
RIP	Pass	Fail	Pass	?	Fail
AODV	Pass	Fail	Pass	Fail	Fail
DYMO	Pass	?	Pass	?	?
DSR	Fail	Pass	Pass	Fail	Fail