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

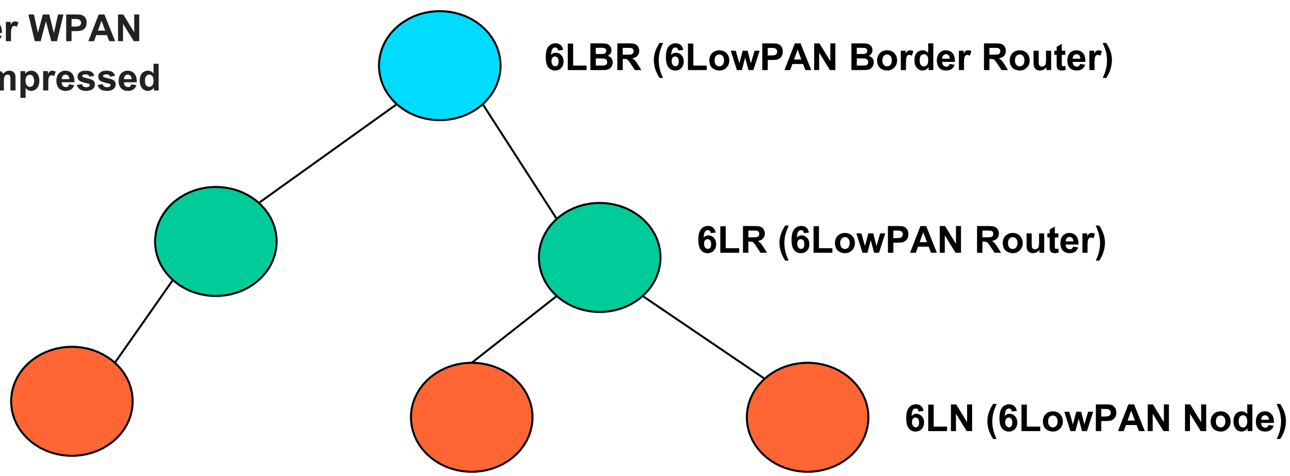
“ Constrained-Node Network: A network whose characteristics are influenced by being composed of a significant portion of constrained nodes.

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?

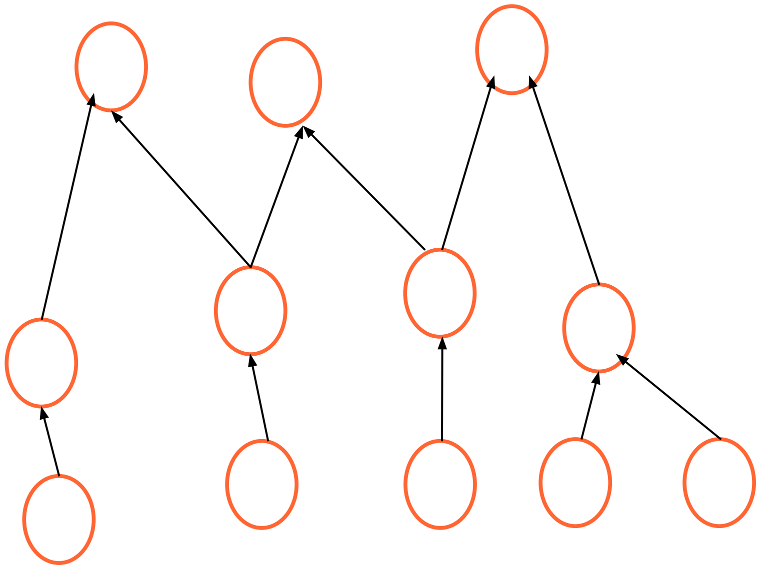
- Distance-vector protocols are based on calculating the Direction and Distance to any link in a network.
  - "Direction" usually means the next hop address and the exit interface.
  - "Distance" is a measure of the cost to reach a certain node.
- The least cost route between any two nodes is the route 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

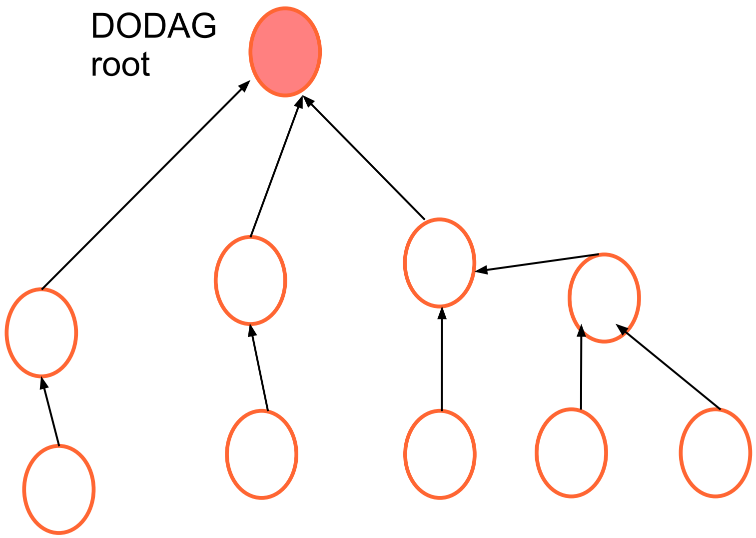
Acyclic

Graph

(DAG)

That is....

...Partitioned into one or more...



Destination

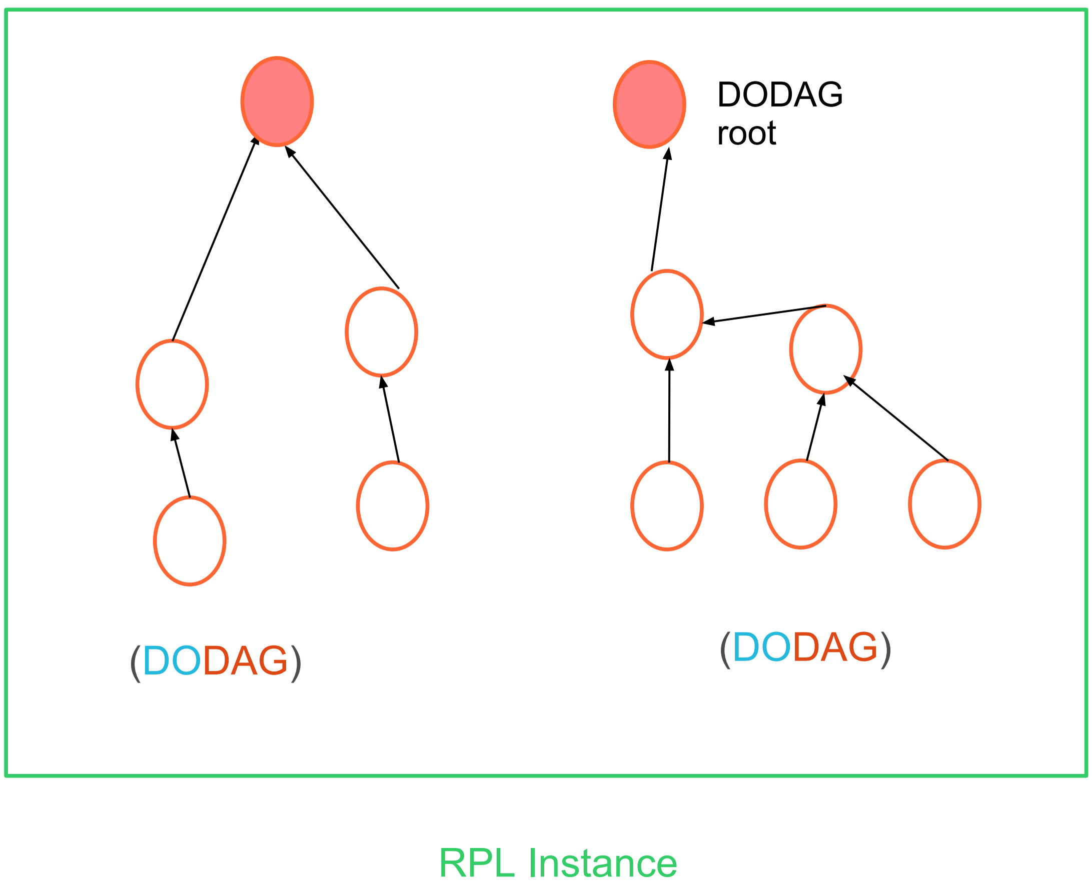
Oriented

DAGs

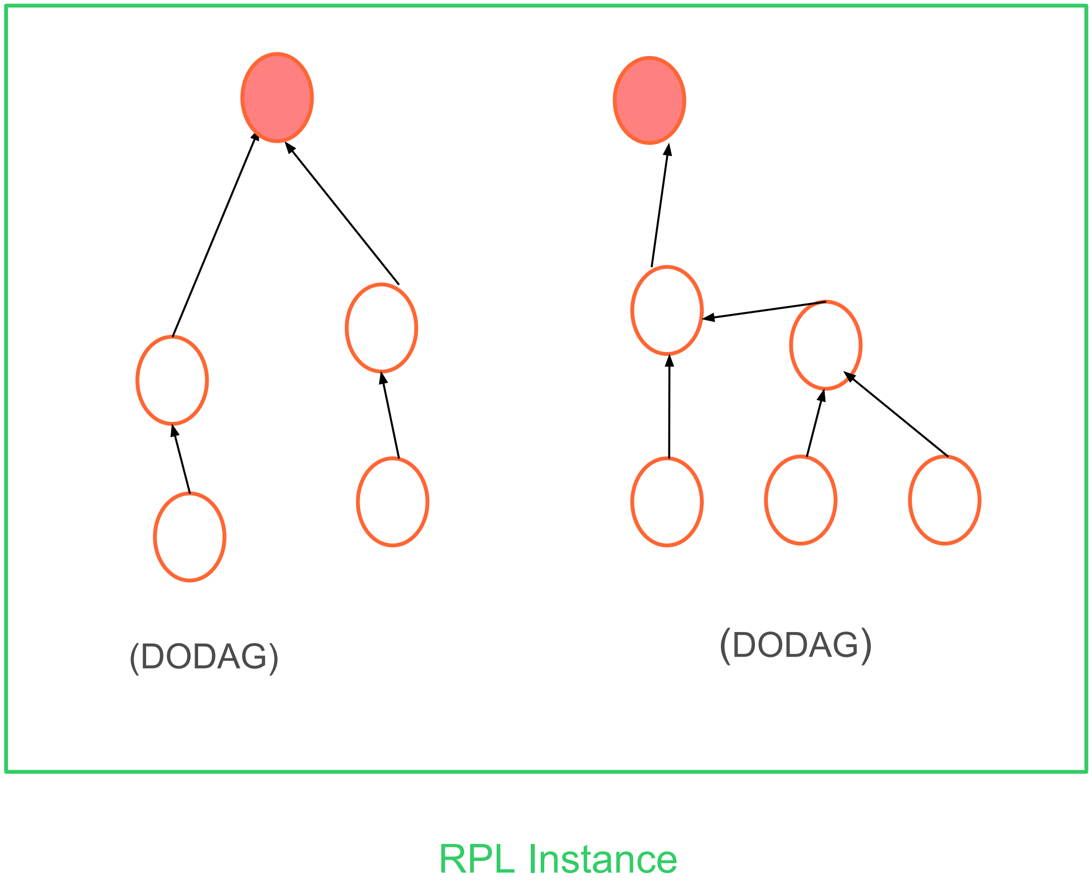
(DODAG)

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

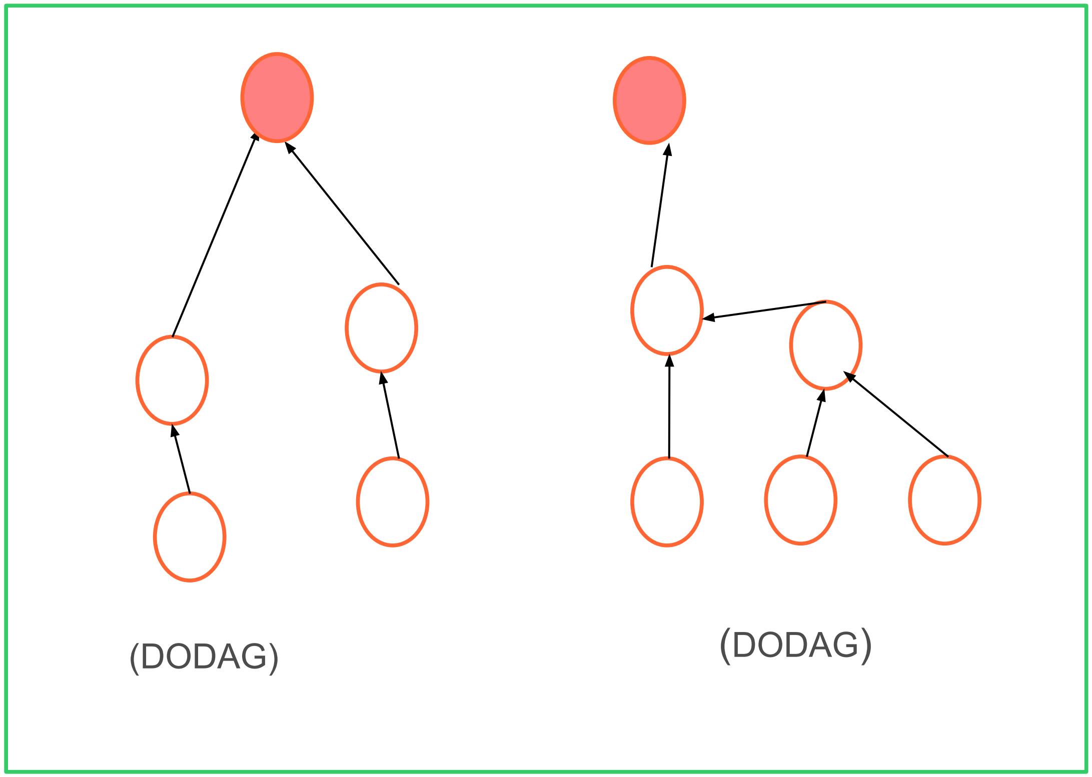
A RPL Instance is a set of one or more DODAGs that share a RPLInstanceID.



To Identify and maintain a topology RPL uses...

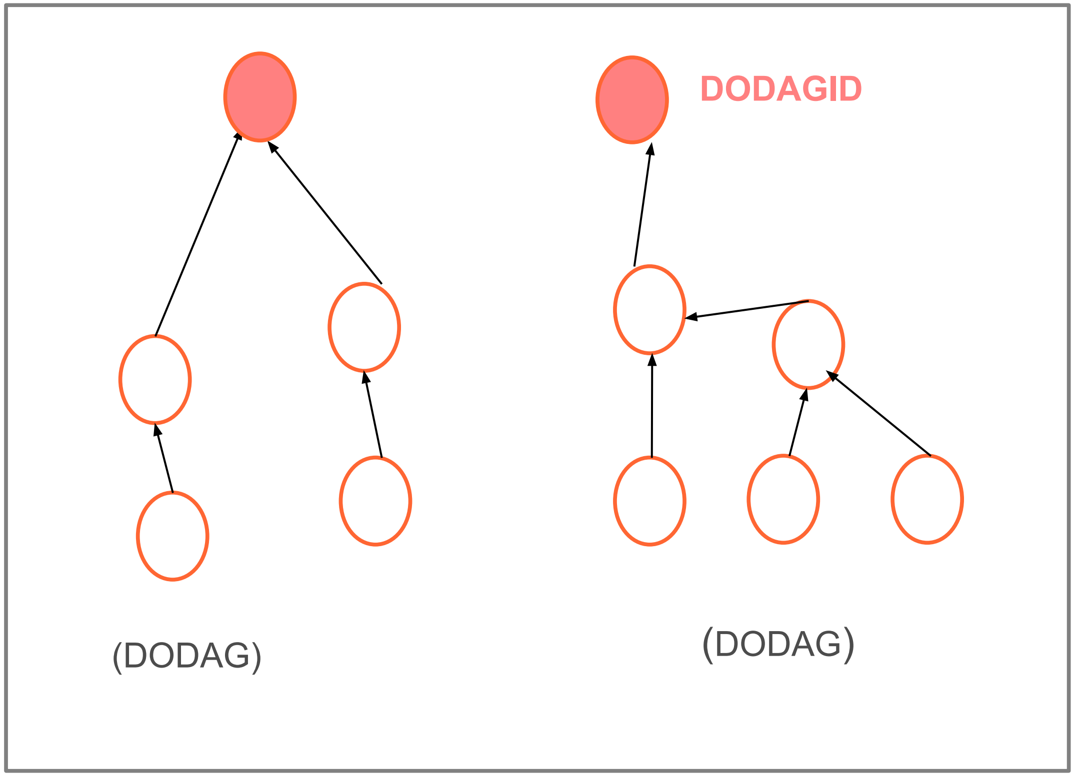


To Identify and maintain a topology RPL uses...



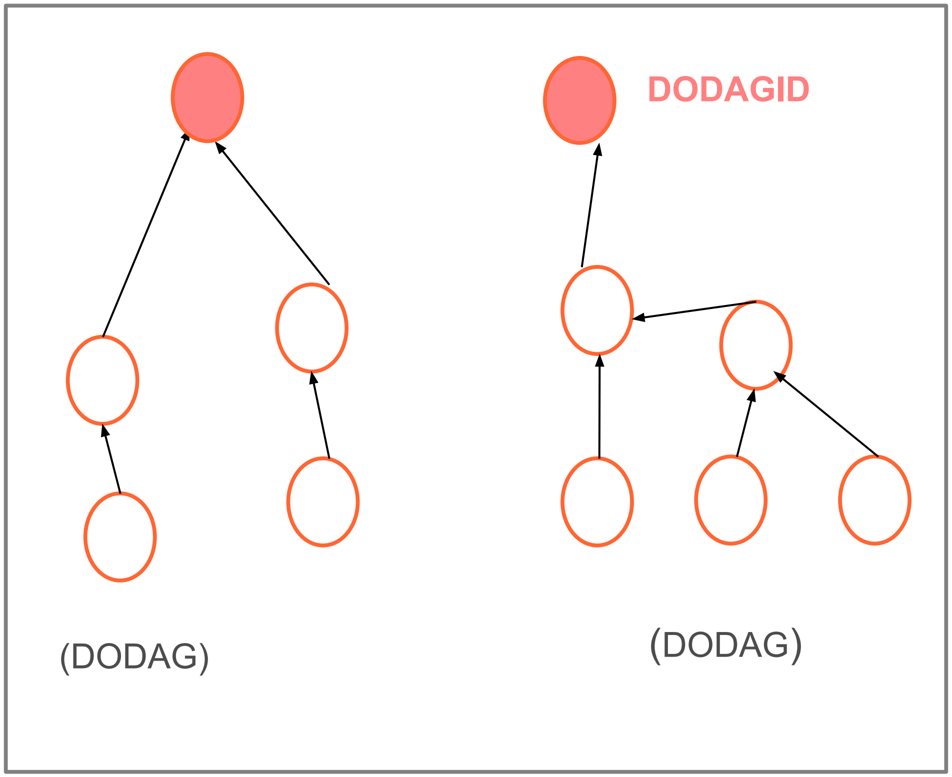
RPLInstanceID is a **unique** identifier within a network. DODAGs with the same RPLInstanceID share the same Function (OF) used to compute the position of node in the DODAG .

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

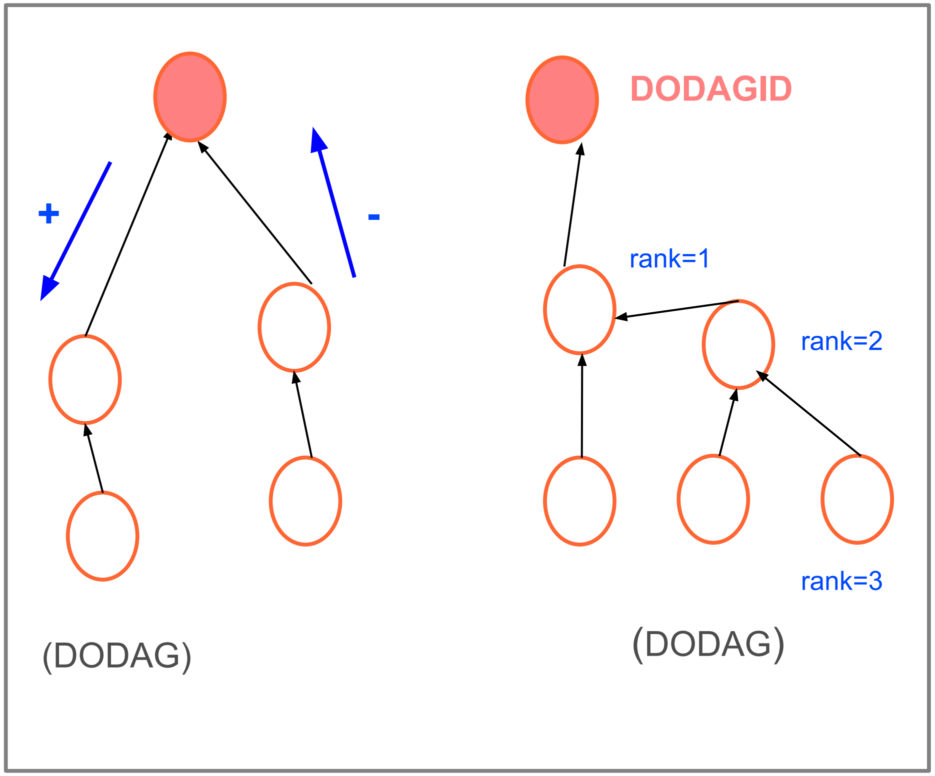
A DODAGVersionNumber Is a sequential counter that is incremented by the root to form a new version

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

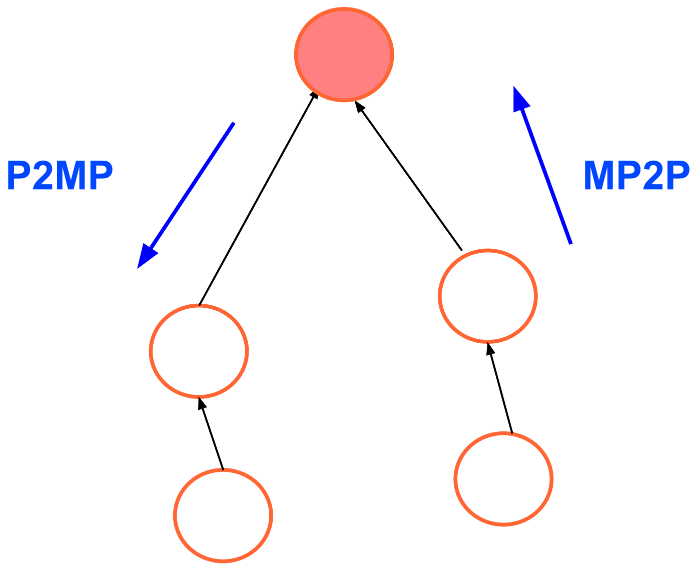
A DODAGVersionNumber Is a sequential counter that is incremented by the root to form a new version

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 is not expected to satisfy the goal, it only provides routes to nodes within the DODAG. e.g, provide interconnectivity during repair

Traffic Flows Supported by RPL



(DODAG)

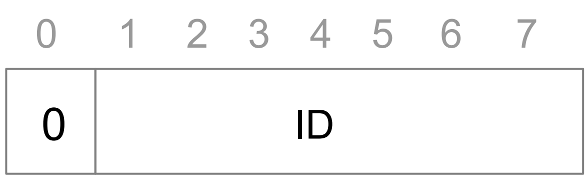
- MP2P
- P2MP
- P2P

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.



Global RPLInstanceID in 0...127

Local RPL Instance

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

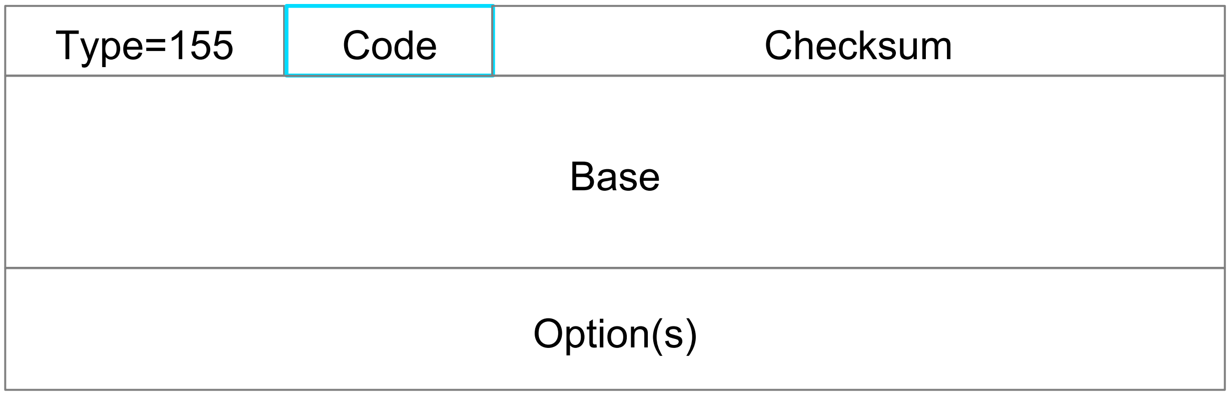
0	1	2	3	4	5	6	7
0	D	ID					

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 message is a ICMPv6 message



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

RPL Control message is a ICMPv6 message

Type=155	Code	Checksum
Security		
Base		
Option(s)		

Code: Identify the type of control message

0x80 → Secure DODAG Information Solicitation (SDIS)

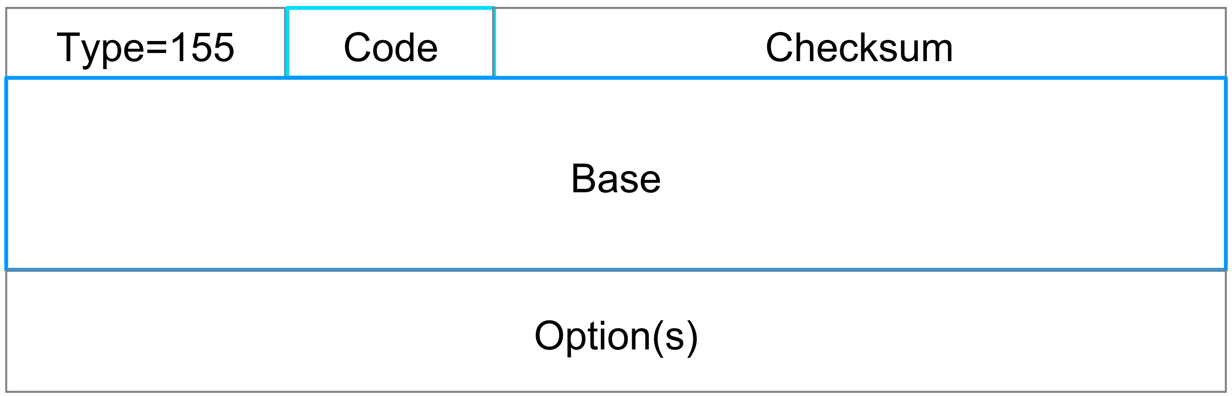
0x81 → Secure DODAG Information Object (SDIO)

0x82 → Secure Destination Advertisement Object (SDAO)

0x83 → SDAO-ACK

0X84 → Consitency Check

RPL Control message is a ICMPv6 message



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

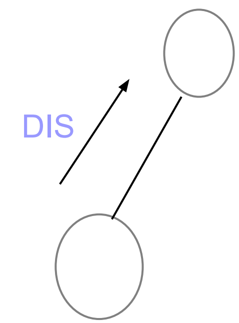
DODAG Information Solicitation (DIS)

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

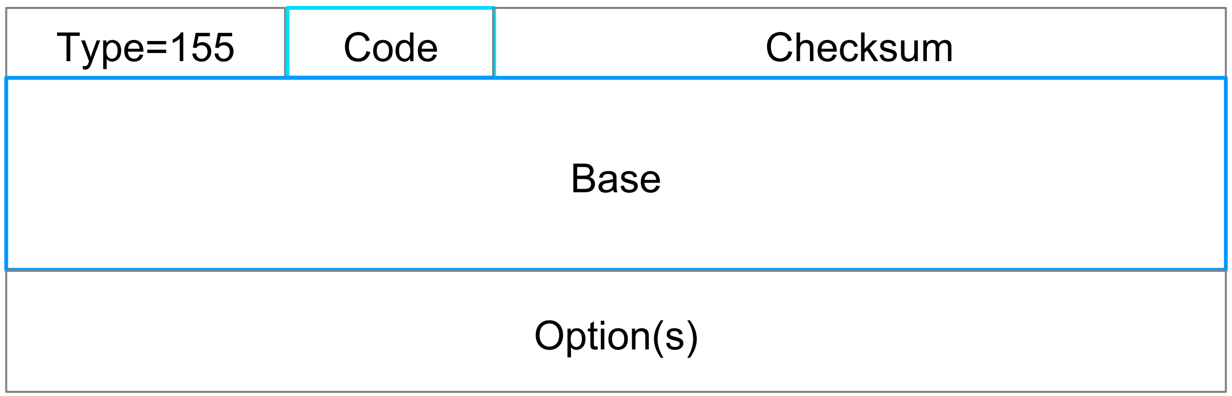
Its use is analogous to that of a Router Solicitation of IPv6 Neighbor Discovery

The DIS Base Object:

Flags	Reserved	Option(s)
unused	unused	Options: 0x00 Pad1 0x01 PadN 0x07 Solicited Information



RPL Control message is a ICMPv6 message



Code: Identify the type of control message

0x00 → DODAG Information Solicitation (DIS)

0x01 → DODAG Information Object (DIO)

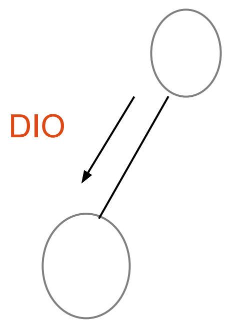
0x02 → Destination Advertisement Object (DAO)

0x03 → DAO-ACK

DODAG Information 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)

RPLInstanceID

8 bits field set by the DODAG root that indicates of which RPL Instance the DODAG is part.

DODAG Information Object (DIO)

RPLInstanceID	Version Number

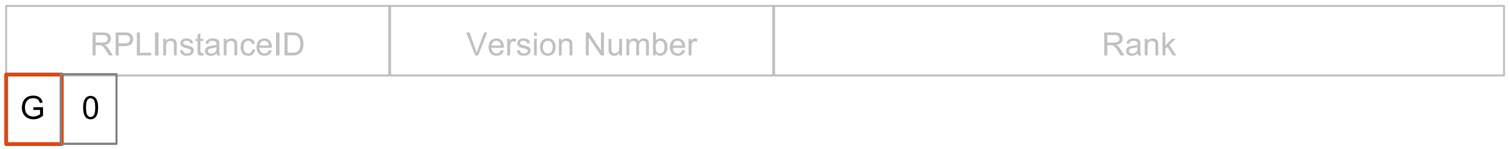
8-bit unsigned integer set by the DODAG root to the DODAGVersionNumber.

DODAG Information Object (DIO)

RPLInstanceID	Version Number	Rank

16 bit unsigned integer indicating the DODAG Rank of the node sending the DIO message.

DODAG Information Object (DIO)



Grounded (G): The Grounded flag indicates whether the DODAG advertised can satisfy the application defined goal. If the flag is set, the DODAG is grounded. If the flag is cleared, the DODAG is floating

DODAG Information Object (DIO)

RPLInstanceID	Version Number	Rank
G	0	MOP

Modo de Operation (MOP): identifies the mode of operation o fthe RPL Instance and distributed by DODAG root. All nodes who join the DODAG must be able to honor the MOP in order to fully participate as a router, or else they must only join as a leaf.

Values:
0: No Downward routes maintained by RPL.

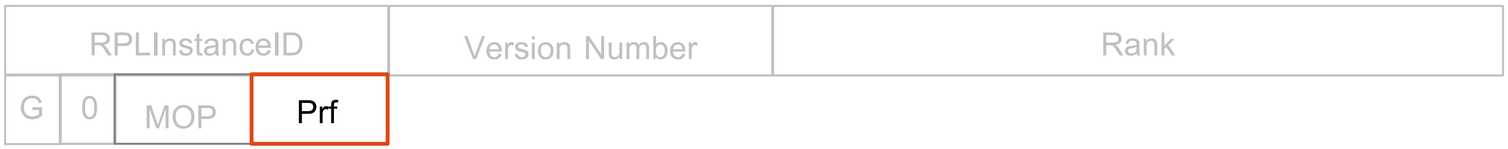
1: Non-Storing Mode of Operation

2: Storing Mode of Operation with no multicast support

3: Storing Mode of Operation with mulitcast support

4-7: unassigned

DODAG Information Object (DIO)

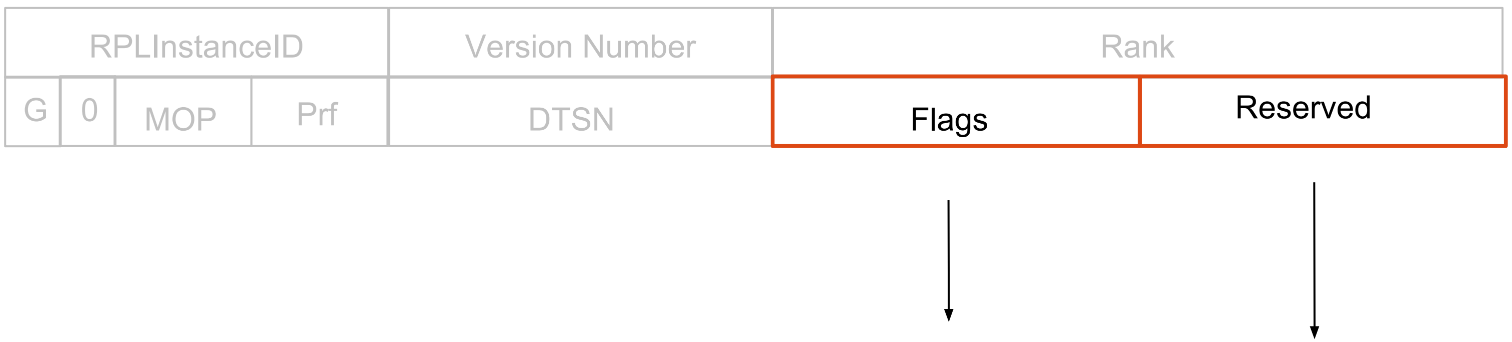


A 3-bit that defines how preferable the root of this DODAG is compared to other DODAG roots within the instance. DAGPreference ranges from 0x00 (least preferred) to 0x07 (most preferred).
The default is 0 (least preferred).

DODAG Information Object (DIO)

Destination Advertisement Trigger Sequence Number (DTSN): 8-bit set by the node issuing the DIO message. This flag is used as part of the procedure to maintain Downward routes.

DODAG Information Object (DIO)



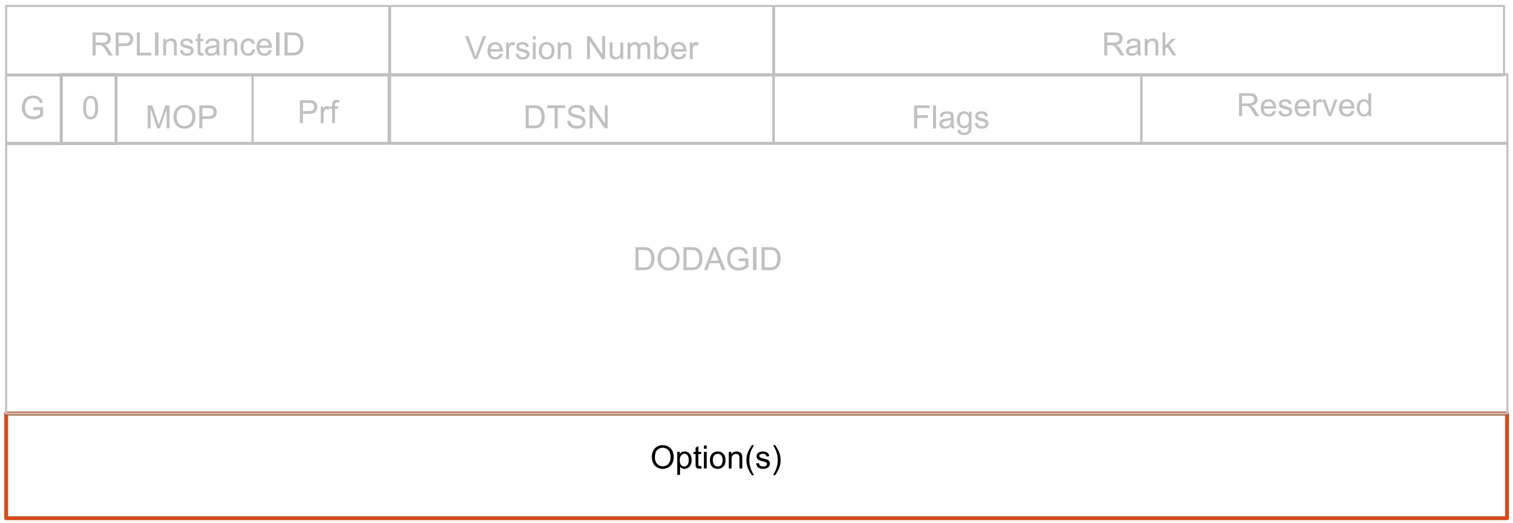
Unused. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

DODAG Information Object (DIO)

RPLInstanceID				Version Number	Rank	
G	0	MOP	Prf	DTSN	Flags	Reserved
DODAGID						

128-bit IPv6 address set by a DODAG root that uniquely identifies a DODAG. The DODAG must be a routable IPv6 address belonging to the DODAG root.

DODAG Information Object (DIO)



0x00: Pad 1
0x00: Pad N

0x00: DAG Metric Container

0x00: Routing Information

0x00: DODAG Configuration

0x00: Prefix Information

DODAG Information Object (DIO)

<table>
<tr><td colspan="4">RPLInstanceID</td><td>Version Number</td><td colspan="2">Rank</td></tr>
<tr><td>G</td><td>0</td><td>MOP</td><td>Prf</td><td>DTSN</td><td>Flags</td><td>Reserved</td></tr>
<tr><td colspan="7">DODAGID</td></tr>
<tr><td colspan="7">Option(s)</td></tr>
</table>

DIO Base Object

DIO Base Rules

1. For the following DIO Base fields, a node that is not a DODAG root MUST advertise the same values as its preferred DODAG parent. In this way, these values will propagate Down the DODAG unchanged and advertised by every node that has a route to that DODAG root. These fields are as follows:
    1. Grounded (G)
    2. Mode of Operation (MOP)
    3. DAGPreference (Prf)
    4. Version
    5. RPLInstanceID
    6. DODAGID

2. A node MAY update the following fields at each hop:
    1. Rank
    2. DTSN

3. The DODAGID field each root sets MUST be unique within the RPL Instance and MUST be a routable IPv6 address belonging to the root.

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.

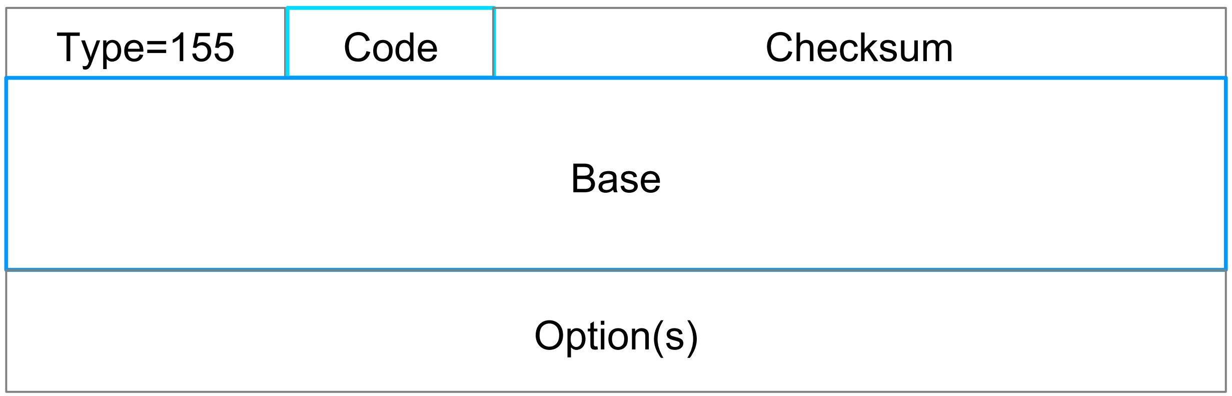
The configuration parameters of the Trickle timer are specified as follows:

Imin: learned from the DIO message as (2^DIOIntervalMin) ms. The default value of DIOIntervalMin is DEFAULT_DIO_INTERVAL_MIN.

Imax: learned from the DIO message as DIOIntervalDoublings. The default value of DIOIntervalDoublings is DEFAULT_DIO_INTERVAL_DOUBLINGS.

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.

RPL Control message is a ICMPv6 message

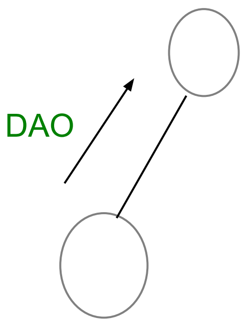


Code: Identify the type of control message

- 0x00 → DODAG Information Solicitation (DIS)
- 0x01 → DODAG Information Object (DIO)
- 0x02 → Destination Advertisement Object (DAO)
- 0x03 → DAO-ACK

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)

RPLInstanceID

8-bit field indicating the topology instance associated with the DODAG, as learned from the DIO.

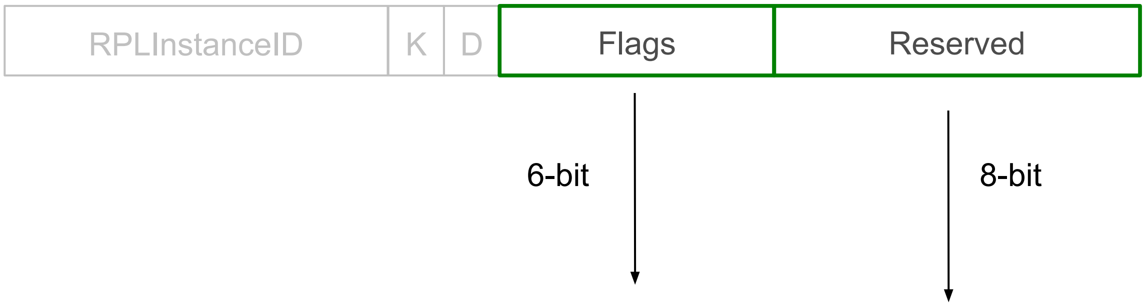
Destination Advertisement Object (DAO)

K: The 'K' flag indicates that the recipient is expected to send a DAO-ACK back.

Destination Advertisement Object (DAO)

D: The 'D' flag indicates that the DODAGID field is present.
This flag MUST be set when a local RPLInstanceID is used.

Destination Advertisement Object (DAO)



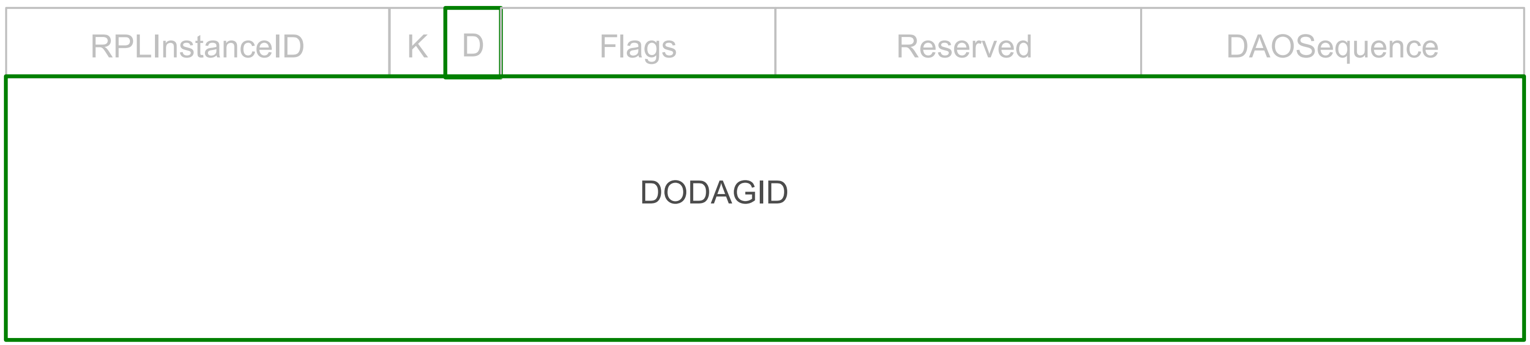
Unused. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Destination Advertisement Object (DAO)

RPLInstanceID	K	D	Flags	Reserved	DAOSequence

Incremented at each unique DAO message from a node
And echoed in the DAO-ACK message.

Destination Advertisement Object (DAO)



128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG.
This field is only present when the 'D' flag is set.

This field is typically only present when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID.

When a global RPLInstanceID is in use, this field need not be present.

Destination Advertisement Object (DAO)

RPLInstanceID	K	D	Flags	Reserved	DAOSequence
DODAGID					
Option(s)					

0x00: Pad 1
0x00: Pad N

0x05: RPL Target

0x06: Transit Information

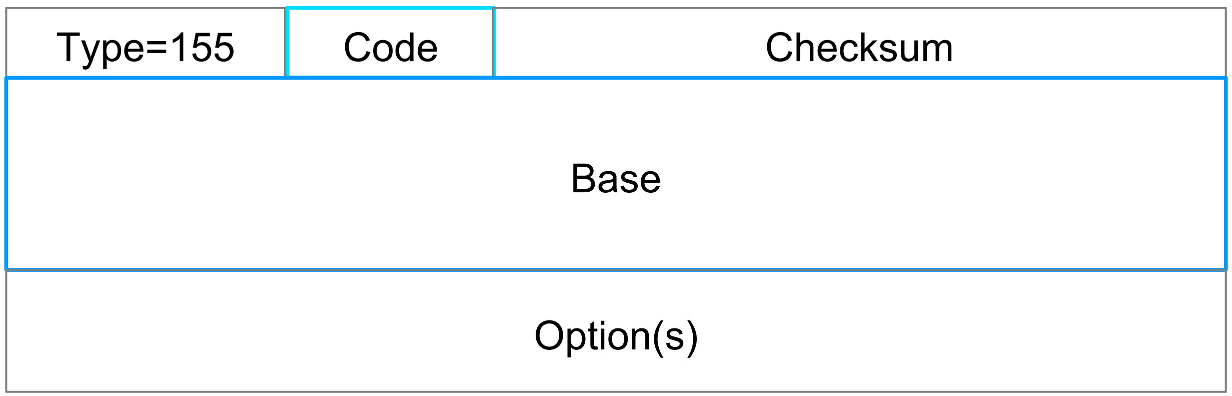
0x09: RPL Target Descriptor

Destination Advertisement Object (DAO)

RPLInstanceID	K	D	Flags	Reserved	DAOSequence
DODAGID					
Option(s)					

DAO Base Object

RPL Control message is a ICMPv6 message



Code: Identify the type of control message

0x00 → DODAG Information Solicitation (DIS)

0x01 → DODAG Information Object (DIO)

0x02 → Destination Advertisement Object (DAO)

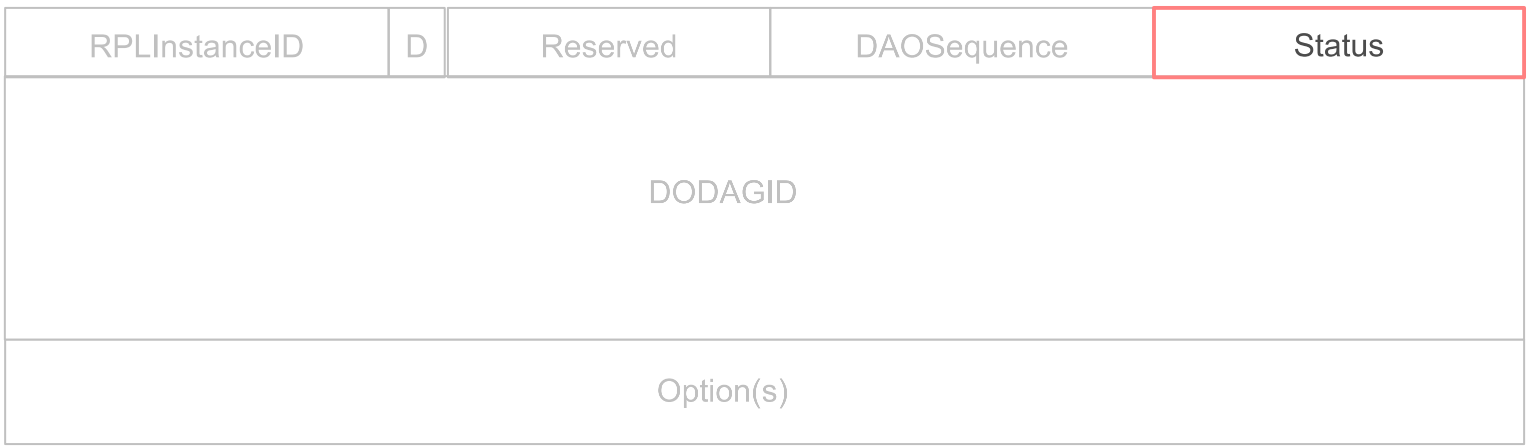
0x03 → DAO-ACK

Destination Advertisement Object Acknowledgement (DAO-ACK)

RPLInstanceID	D	Reserved	DAOSequence	Status
DODAGID				
Option(s)				

DAO-ACK Base Object

Destination Advertisement Object Acknowledgement (DAO-ACK)



0: Unqualified Acceptance
1-127: The node sending the DAO-ACK is willing to act as a parent, but the receiving node is suggested to find and use an alternate parent instead.

128-255: Rejection: The node sending the DAO-ACK is unwilling to act as a parent.

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Routing-MC-Type (Routing Metric/Constraint Type - 8 bits): the Routing Metric/Constraint Type field uniquely identifies each Routing Metric/Constraint object

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

'P' flag: the P field is only used for recorded metrics.
When cleared, all nodes along the path successfully recorded the corresponding metric. When set, this indicates that one or several nodes along the path could not record the metric of interest (either because of lack of knowledge or because this was prevented by policy).

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

'C' flag. When set, this indicates that the Routing Metric/ Constraint object refers to a routing constraint.

When cleared, the routing object refers to a routing metric.

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

'O' flag: The 'O' flag is used exclusively for routing constraints ('C' flag is set).
When set, this indicates that the constraint specified in the body of the object is optional. When cleared, the constraint is mandatory.

If the 'C' flag is zero, the 'O' flag MUST be set to zero on transmission and ignored on reception.

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

'R' flag: The 'R' flag is only relevant for a routing metric (C=0) and MUST be cleared for C=1. When set, this indicates that the routing metric is recorded along the path.
Conversely, when cleared, the routing metric is aggregated.

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A Field (3 bits): The A field is only relevant for metrics and is used to indicate whether the aggregated routing metric is additive, is multiplicative, reports a maximum, or reports a minimum.

- A=0: The routing metric is additive
- A=1: The routing metric reports a maximum
- A=2: The routing metric reports a minimum
- A=3: The routing metric is multiplicative

the total cost of a path is the sum of the costs of individual links along the path

the total cost of a path is the product of the costs of individual links along the path

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Prec field (4 bits): The Prec field indicates the precedence of this Routing Metric/Constraint object relative to other objects in the container. This is useful when a DAG Metric Container contains several Routing Metric objects. Its value ranges from 0 to 15. The value 0 means the highest precedence.

DAG Metric Container

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec |  Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Node Metric/Constraint Objects
- Link Metric/Constraint Objects

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

Objetive 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

Objetive Function (OF)

- Of0: Objective Function Zero
- Minimun Rank with Hysteresis OF.

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

Objective Function Zero (OF)

- R(N) = R(P) + Ri where:
  - Ri (rank_increase) = (Rf*Sp + Sr) * MinHopRankIncrease
  - Rf: Rank Factor (an implementation MAY recognize categories of peers and links, such as different link types, in which case it SHOULD be able to configure a more specific rank_factor to those categories)
  - Sp: Step_of_Rank (used to compute the amount by which to increase the rank along a particular link)
  - Sr: Strech_of_Rank (OF0 allows an implementation to stretch the step_of_rank in order to
  - enable the selection of at least one feasible successor and thus
  - maintain path diversity)
  - R(p): rank of preferred parent

MINIMUN_RANK_FACTOR <= Rf <= MAXIMUN_RANK_FACTOR

- MINIMUN_STEP_OF_RANK <= Sp <= MAXIMUN_STEP_OF_RANK
- 0 <= Sr <= MAXIMUN_RANK_STRECH

Objetive Function (OF)

- Of0: Objective Function Zero
- Minimun Rank with Hysteresis OF.

Minimum Rank with Hysteresis Objective Function (MRHOF)

- 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

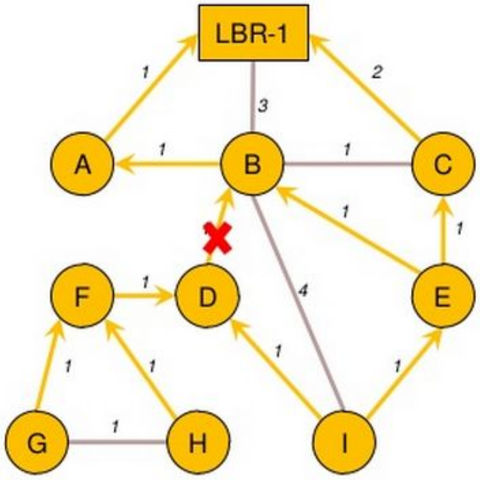
Minimum Rank with Hysteresis Objective Function (MRHOF)

- 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

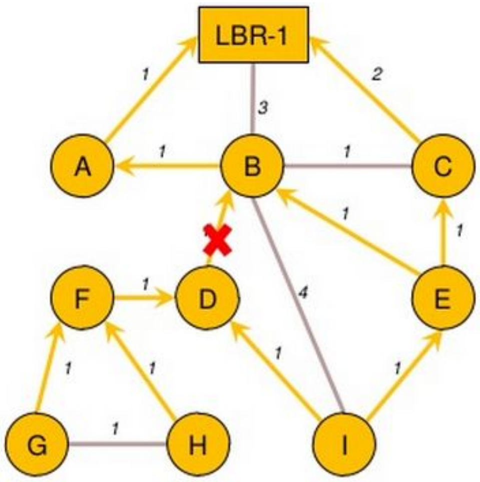
Loop Avoidance



Suppose link between nodes B and D is broken.

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

Loop Avoidance



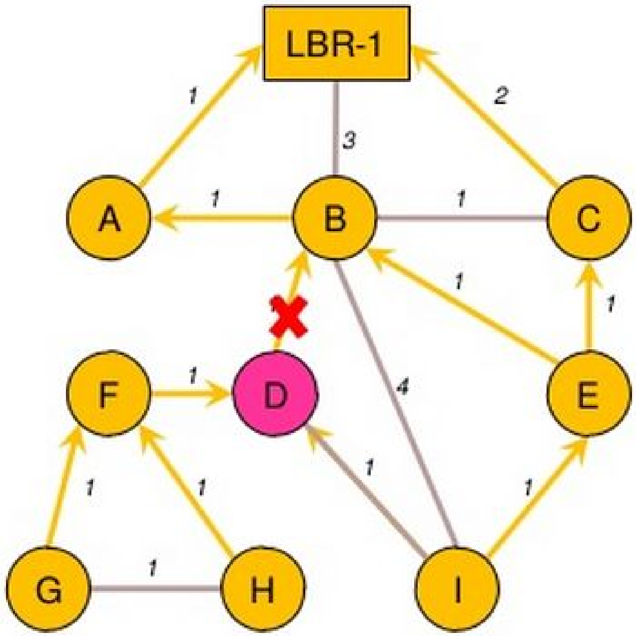
Suppose link between nodes B and D is broken.

 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

Loop Avoidance



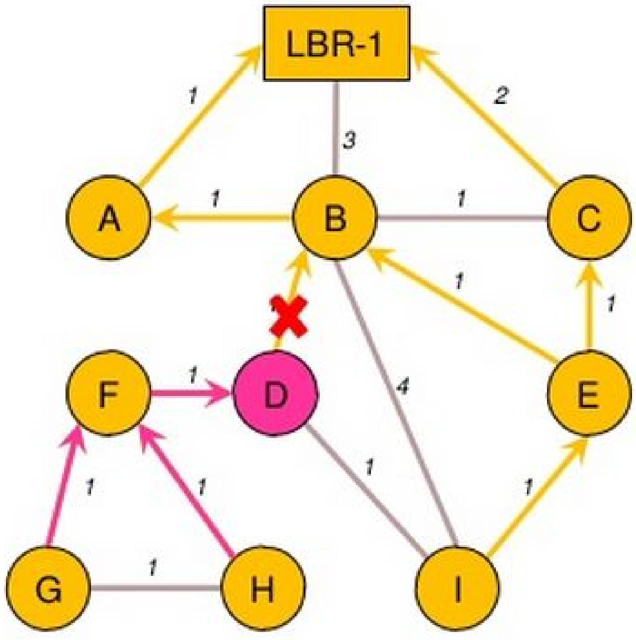
Node D play DIO to notify change of sub-DAG

 Parent Node alternative I have is E, so it does not leave the DAG of LBR-1

 I kind Node D Node from the Parent list

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

Loop Avoidance



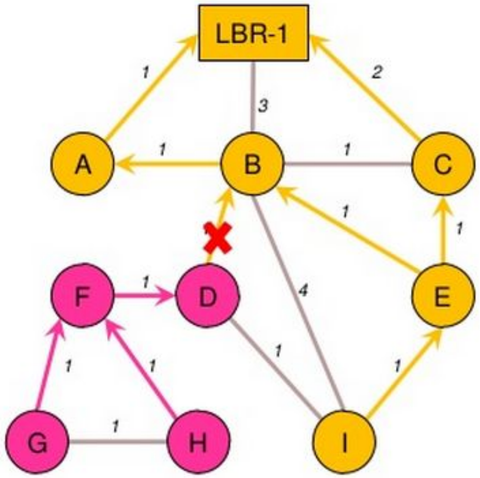
- Node F has no choice but to stay in the LBR-1 DAG should follow Node D Node F on DAG's floating node D

- Node F foundDIO

- Node   G and H according to the floating node F DAG

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

Loop Avoidance



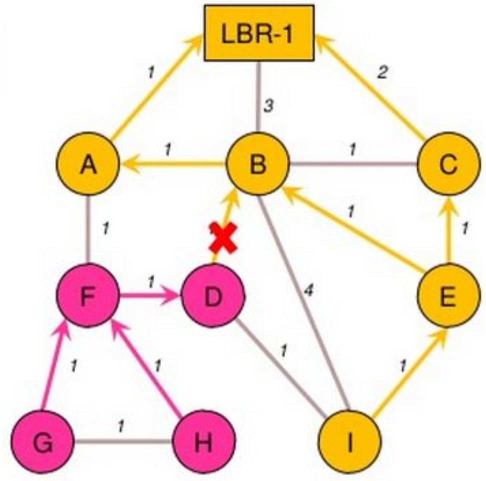
Node I found DIO

 Node D find opportunities to re-enter the last Grounded with depth 5 Node I

Node D started DAG Hop timer with 4 cycles of time associated with the node I.

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

Loop Avoidance



 Suppose link between A and F is set

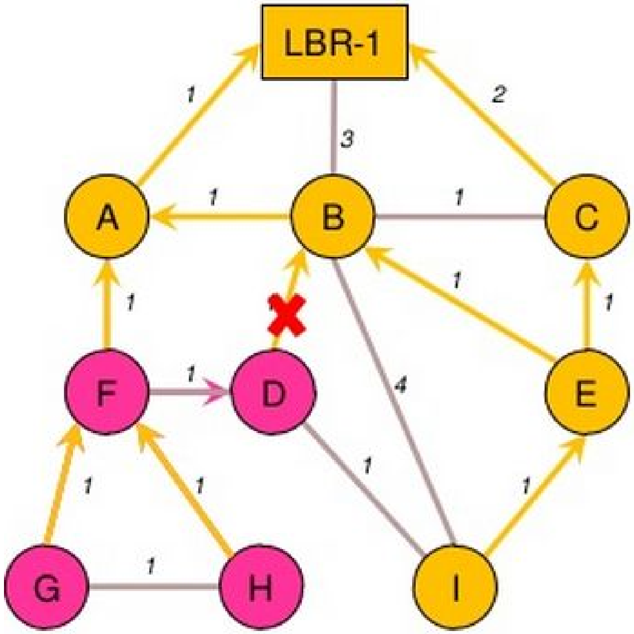
 Node A send DIO

- 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

Loop Avoidance



- DAG node F Timer Hop off.

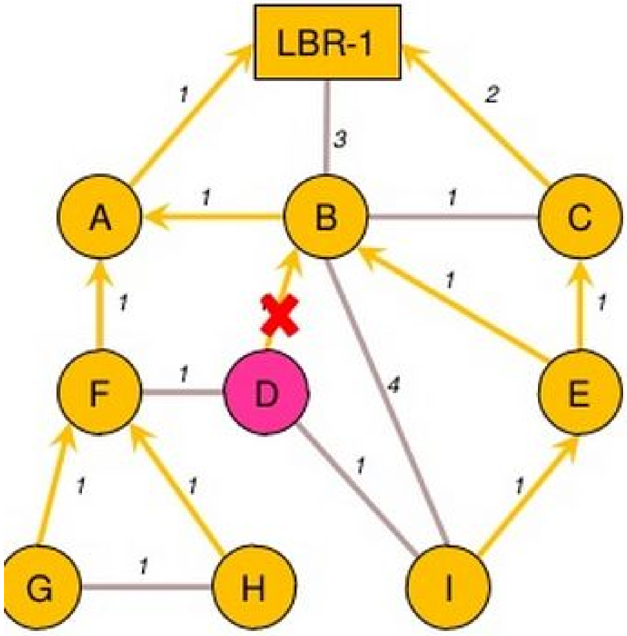
-Node F Grounded DAG with depth 2 by adding the Parent A

-Node F send DIO

-Node G and H join to the Grounded DODAG through F

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

Loop Avoidance

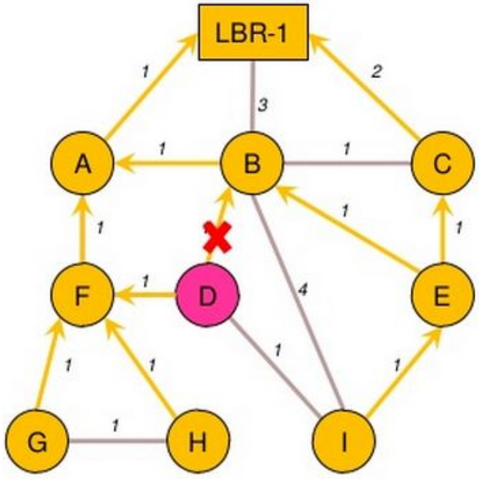


Node D see DAG's chances of re-entering LBR-1 with Depth 3 through Node F.

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

Loop Avoidance



- 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

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

- **ContikiRPL → https://github.com/contiki-os/contiki/tree/master/core/net/rpl**
- **TinyRPL → https://github.com/tinyos/tinyos-main/tree/master/tos/lib/net/rpl**
- **Unstrung → http://unstrung.sandelman.ca/**

- **https://tools.ietf.org/html/draft-hui-vasseur-roll-rpl-deployment-01**
- **A lot of Academia papers evaluating the performance of RPL**

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