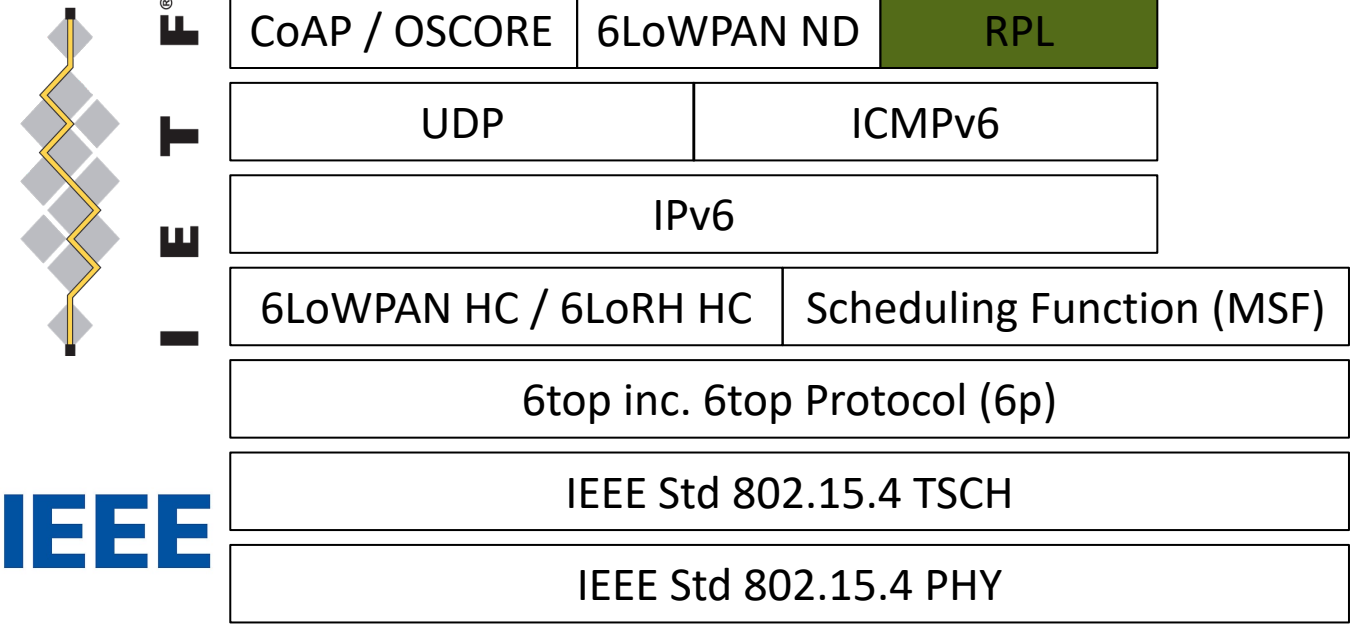
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Routing Layer in LLNs

Georgios Z. **PAPADOPOULOS**
e-mail: georgios.papadopoulos@imt-atlantique.fr
web: www.georgiospapadopoulos.com
twitter: @gzpapadopoulos
youtube: www.youtube.com/c/gzpapadopoulos

Context: The 6TiSCH Protocol Stack



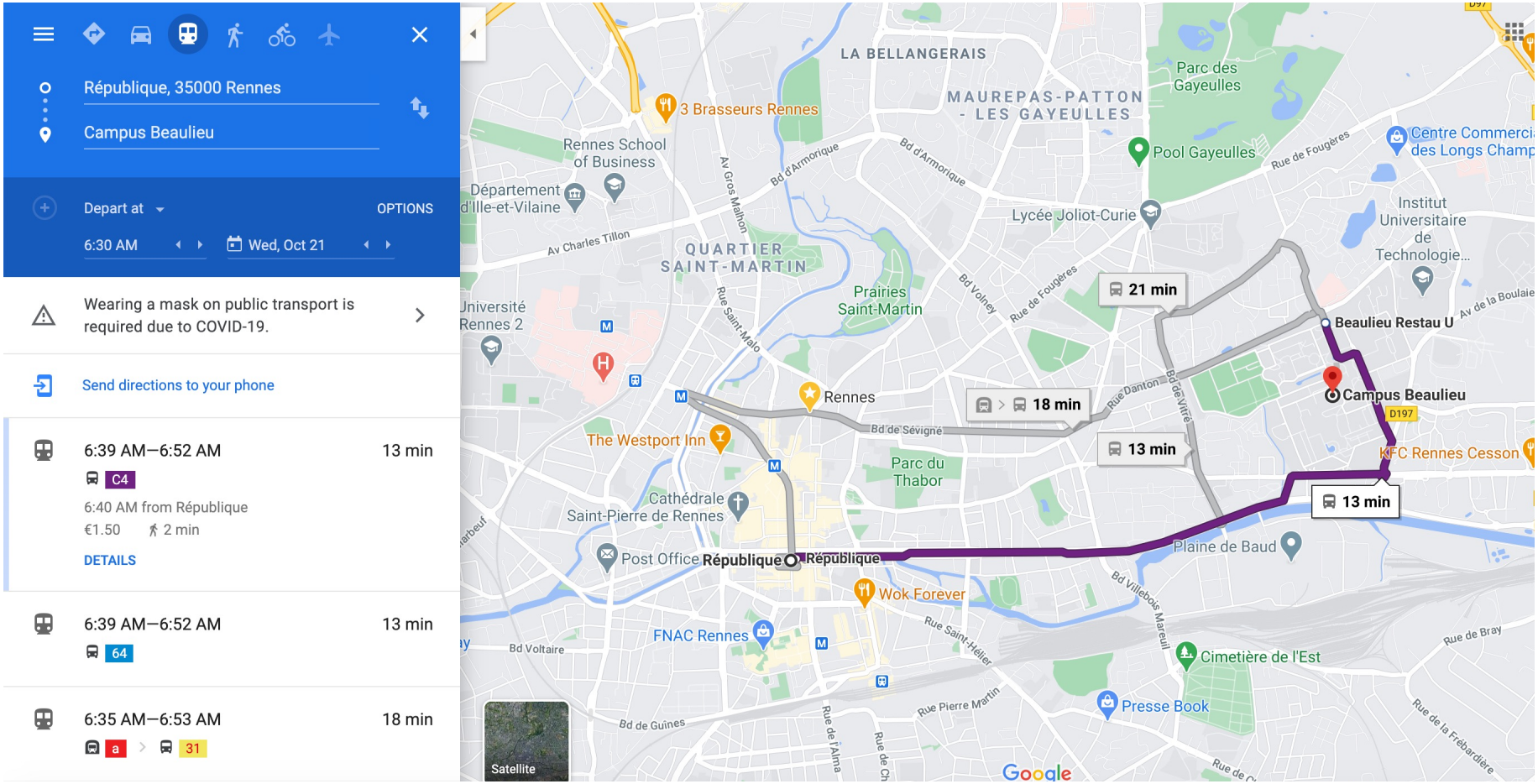
CHAPTER OUTLINE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CHAPTER OUTLINE

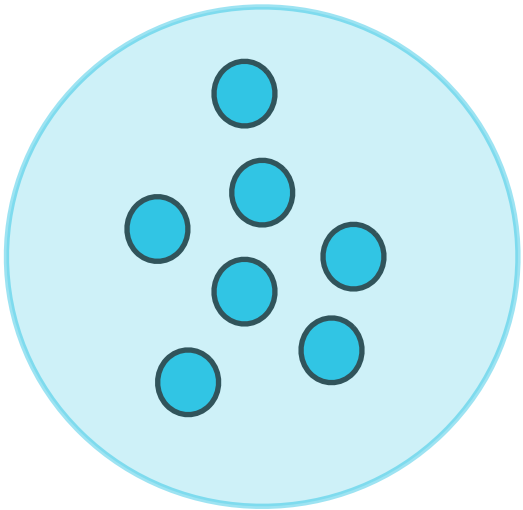
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Routing

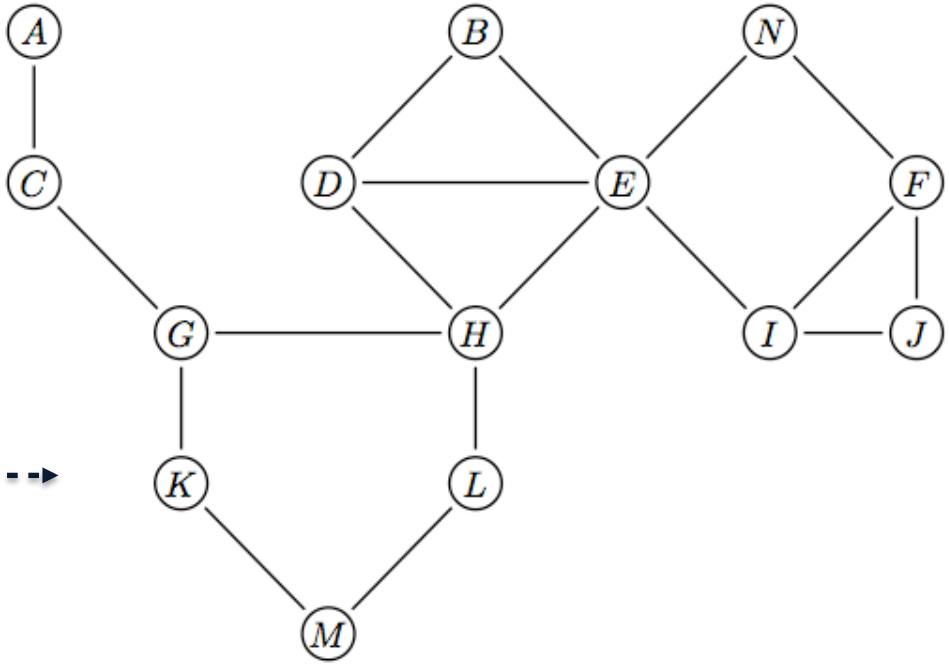


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Routing Protocol



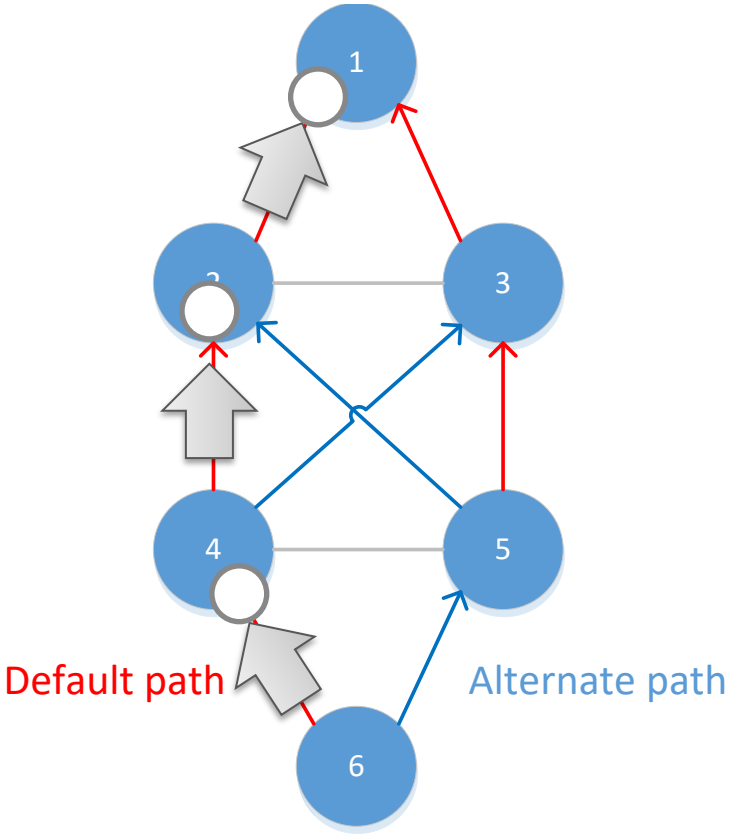
All nodes hear each others



Extended network area

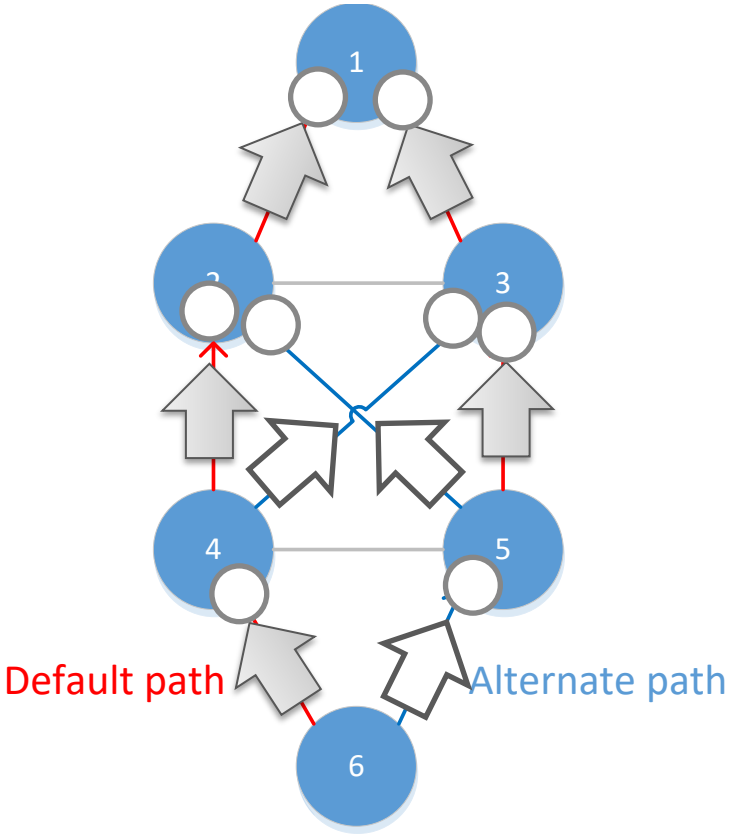
Routing Protocol: the Two Principles

- Routing
  - Create and maintain the routes
  - Goals
    - Minimum signalling
    - No loops
    - Quick convergence
    - Take into account the link variability
    - Energy consumption
    - …



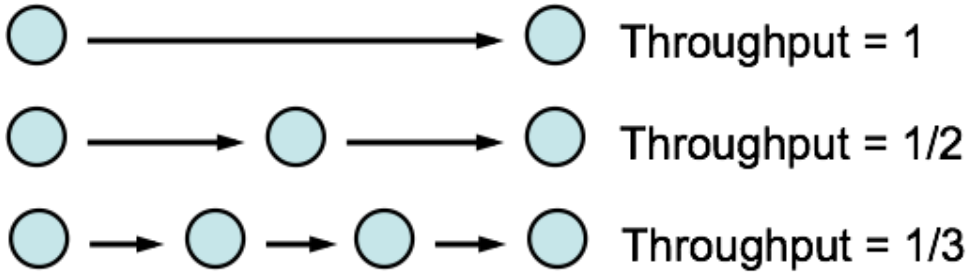
Routing Protocol: the Two Principles

- Routing
  - Create and maintain the routes
  - Goals
    - Minimum signalling
    - No loops
    - Quick convergence
    - Take into account the link variability
    - Energy consumption
    - …
- Forwarding process
  - ARQ and HARQ (ARQ + FEC)
  - Replication, Elimination, Overhearing



Metrics : *How do you compare two routes?*

- How do you select a route / a neighbour
  - End-to-end
    - Hop count
    - Packet Delivery ratio
    - Delay
    - Throughput
    - Energy consumption



  - Hop-by-hop
    - Expected Transmission Time (ETT)
    - Expected Transmission Count (ETX)
    - Energy consumption

CHAPTER OUTLINE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. Centralized vs. Distributed

- Centralized: A controller sets up the routes
- Distributed: The nodes set up the routes

	centralized/SDN OpenFlow [2], PCE [3], SR [4]	*distributed/traditional* IGP [5, 6], RSVP-TE [1]	***hybrid*** **Fibbing**
forwarding paths:			
- *configuration*	simple (declarative & global)	complex (indirect & per-device)	**simple** (declarative & global)
- *manageability*	high (direct control)	low [7, 8] (need for coordination)	**high** (direct control)
- *path installation*	slow (by controller, per-device)	fast (by device, distributed)	**fast** (by device, distributed)
robustness:			
- *network failures*	slow (by controller)	fast (local)	**fast** (local)
- *controller failures*	hard (ad-hoc synch)	native (distributed)	**easy** (synch via IGP)
- *partitions*	hard (uncontrollable devices)	best (distributed)	**best** (fallback on distributed)
routing policies:	highest (any path)	- low for IGP (shortest paths) - highest for RSVP (any path)	**high** (any non-loopy paths)

Table 1: Fibbing combines the advantages of existing control planes, avoiding the main drawbacks.

Stefano Vissicchio, Olivier Tilmans, Laurent Vanbever, Jennifer Rexford, “Central Control Over Distributed Routing”, in Proc. of the 2015 ACM SIGCOMM.

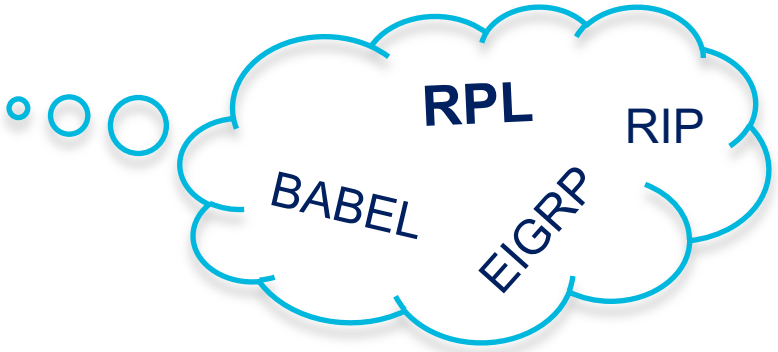
2. Link State vs. Distance Vector

- Link State:
  - All nodes have complete map of the topology.
  - Thus, all nodes know about the path to reach any node in the network.

2. Link State vs. Distance Vector

- Link State:
  - All nodes have complete map of the topology.
  - Thus, all nodes know about the path to reach any node in the network.

- Distance Vector:
  - Each node has information only about the next hop.
  - Each node advertise its “distance” (cost) from the destination.
  - Thus, the other nodes, discover the best relay node.

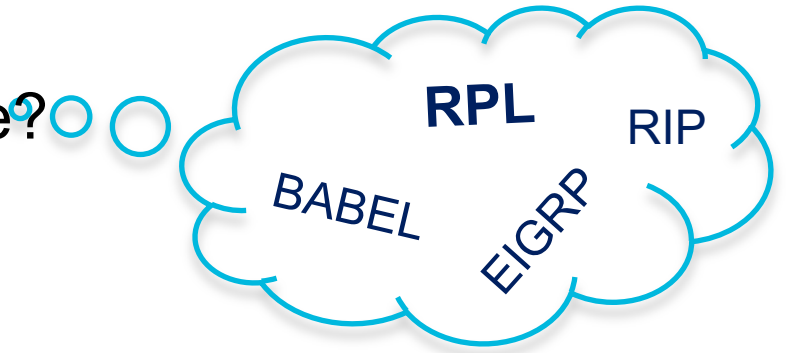


2. Link State vs. Distance Vector

- Link State:
  - All nodes have complete map of the topology.
  - Thus, all nodes know about the path to reach any node in the network.

- Distance Vector:
  - Each node has information only about the next hop.
  - Each node advertise its “distance” (cost) from the destination.
  - Thus, the other nodes, discover the best relay node.

- Do you see the Pros & Cons here?



3. Proactive vs. Reactive

- Proactive (Stateful):
  - The routes are maintained based on periodic updates.
  - High routing overhead.
  - Traditional distributed shortest-path protocols.

3. Proactive vs. Reactive

- Proactive (Stateful):
  - The routes are maintained based on periodic updates.
  - High routing overhead.
  - Traditional distributed shortest-path protocols.

RPL

Vs.

- Reactive (On-demand):
  - Discover routes when needed.
  - Source-initiated route discovery.

LOADng

3. Proactive vs. Reactive

- Proactive (Stateful):
  - The routes are maintained based on periodic updates.
  - High routing overhead.
  - Traditional distributed shortest-path protocols.

RPL

Vs.

- Reactive (On-demand):
  - Discover routes when needed.
  - Source-initiated route discovery.

LOADng

- Tradeoff:

3. Proactive vs. Reactive

- Proactive (Stateful):
  - The routes are maintained based on periodic updates.
  - High routing overhead.
  - Traditional distributed shortest-path protocols.

RPL

Vs.

- Reactive (On-demand):
  - Discover routes when needed.
  - Source-initiated route discovery.

LOADng

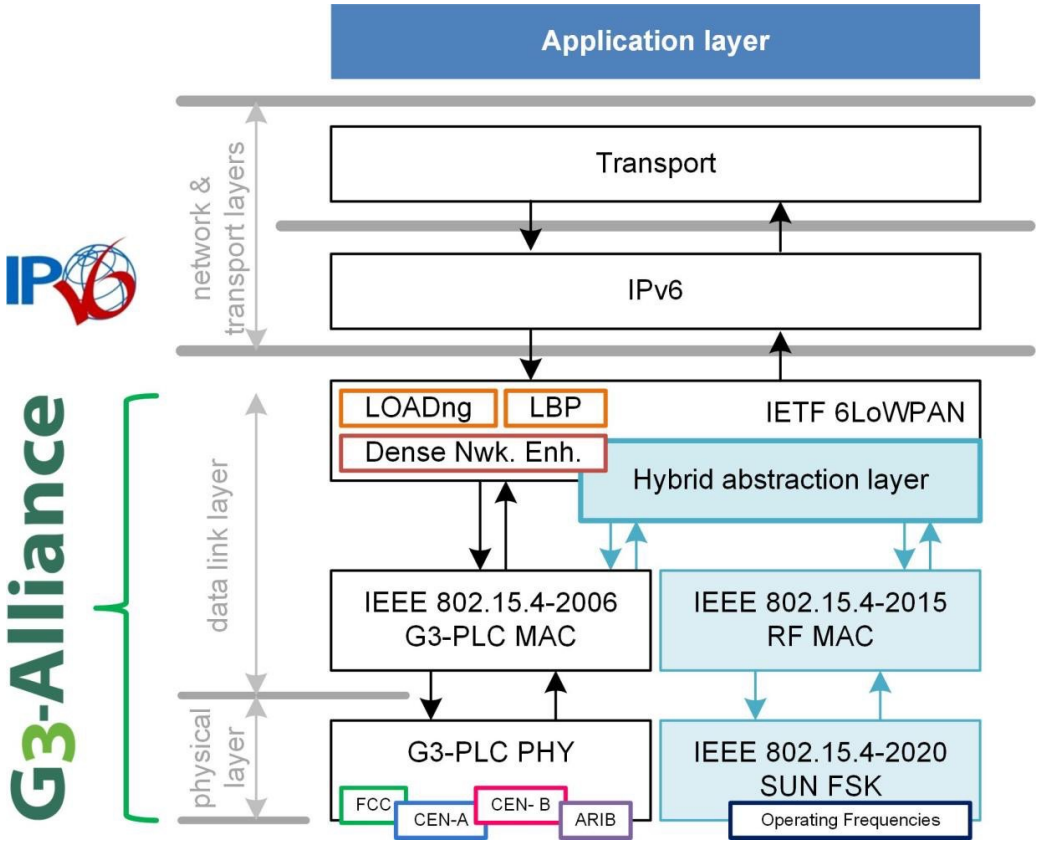
- Tradeoff:
  - State maintenance traffic vs. route discovery traffic.
  - Delay via maintained route vs. delay for route discovery.

CHAPTER OUTLINE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LOADng

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom


Application layer
Transport
IPv6
network & transport layers
LOADng
LBP
Dense Nwk. Enh.
IETF 6LoWPAN
Hybrid abstraction layer
IEEE 802.15.4-2006
G3-PLC MAC
IEEE 802.15.4-2015
RF MAC
G3-PLC PHY
IEEE 802.15.4-2020
SUN FSK
FCC
CEN-A
CEN- B
ARIB
Operating Frequencies
data link layer
physical layer
G3-Alliance

LOADng (Reactive Protocol)

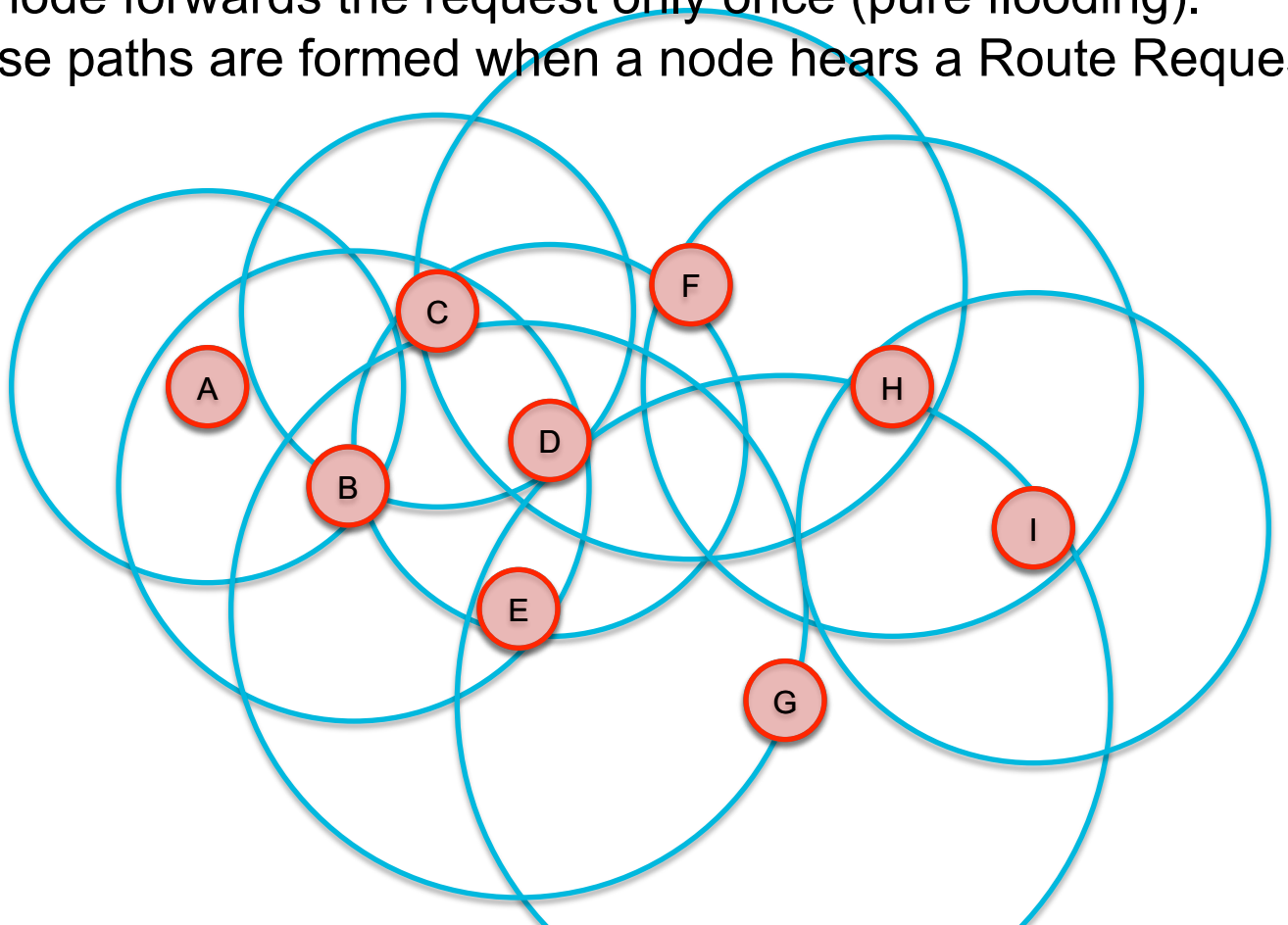
- The extended version of the AODV protocol.
- It is the routing protocol in the Linky devices of ENEDIS

LOADng: Route Request

- Flood a Route Request:
  - Source floods Route Request in the network.
  - Each node forwards the request only once (pure flooding).
  - Reverse paths are formed when a node hears a Route Request.

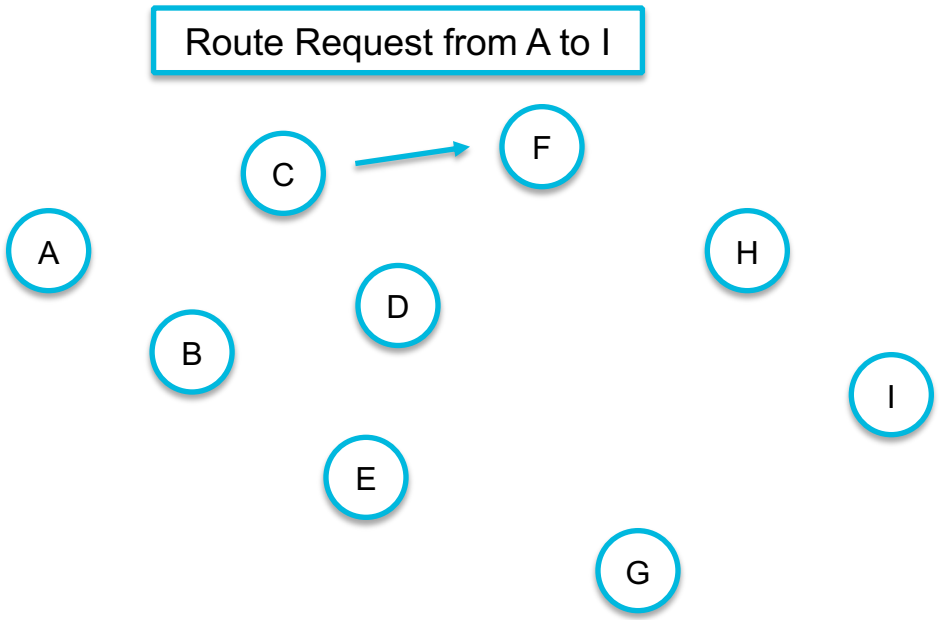
LOADng: Route Request

- Flood a Route Request:
  - Source floods Route Request in the network.
  - Each node forwards the request only once (pure flooding).
  - Reverse paths are formed when a node hears a Route Request.



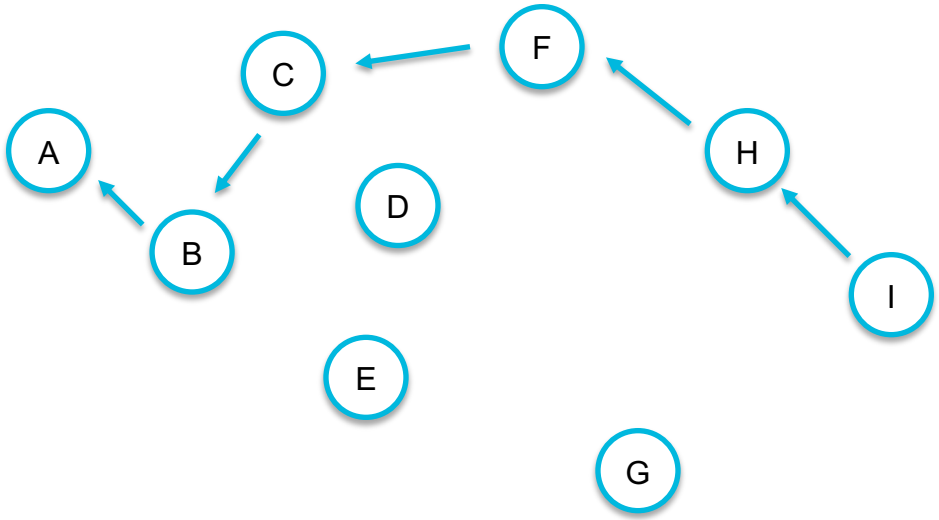
F Routing Table

Dest.	Next Hop	Dest Seq N.	Hops	Lifetime
A	C	4	3	100
C	*	10	1	100



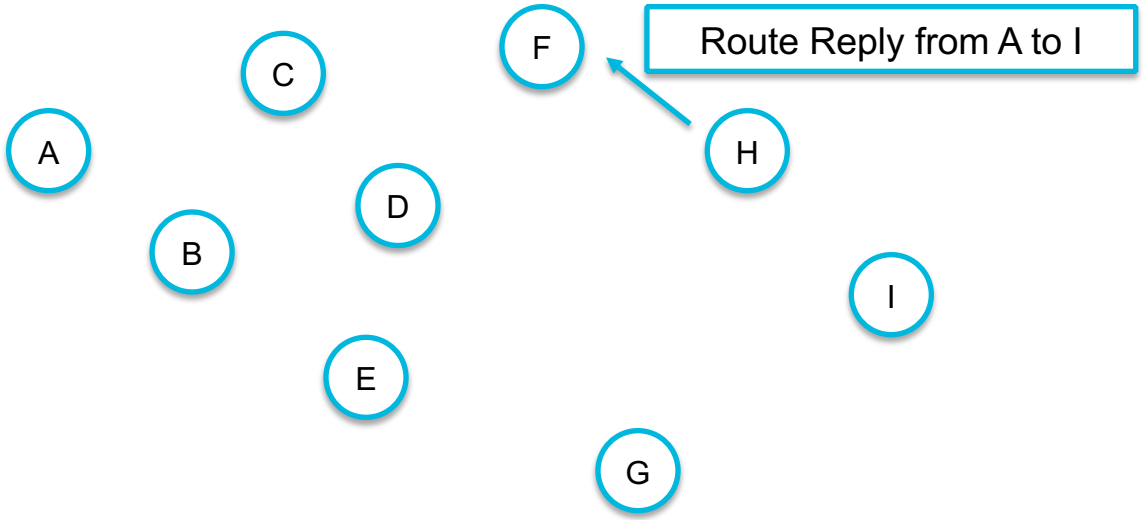
LOADng: Route Reply

- Flood a *Route Request*
- *Route Reply* is forwarded via the reverse path:
  - Forming thus the forward path.



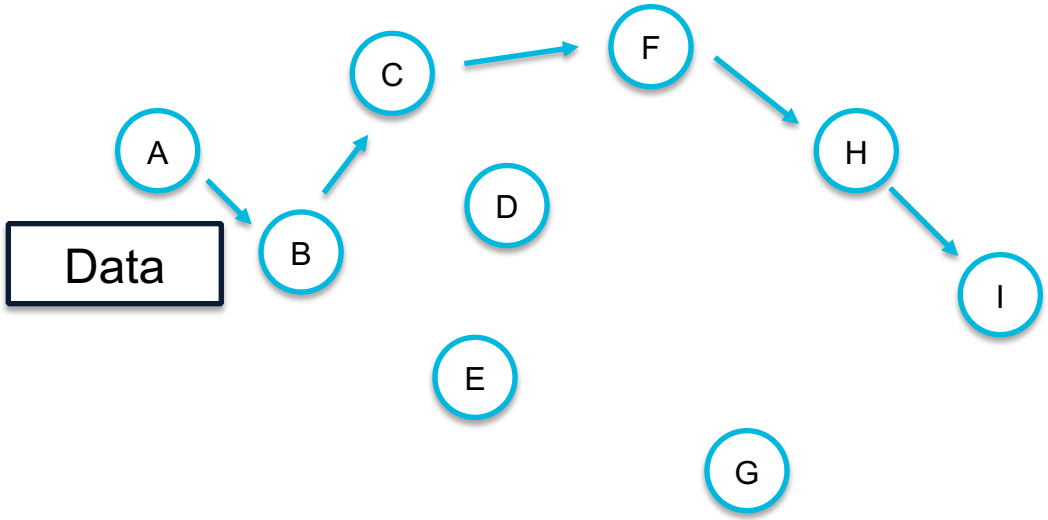
F Routing Table

Dest.	Next Hop	Dest Seq N.	Hops	Lifetime
A	C	4	3	99
C	*	10	1	99
I	H	6	2	100
H	*	10	1	100



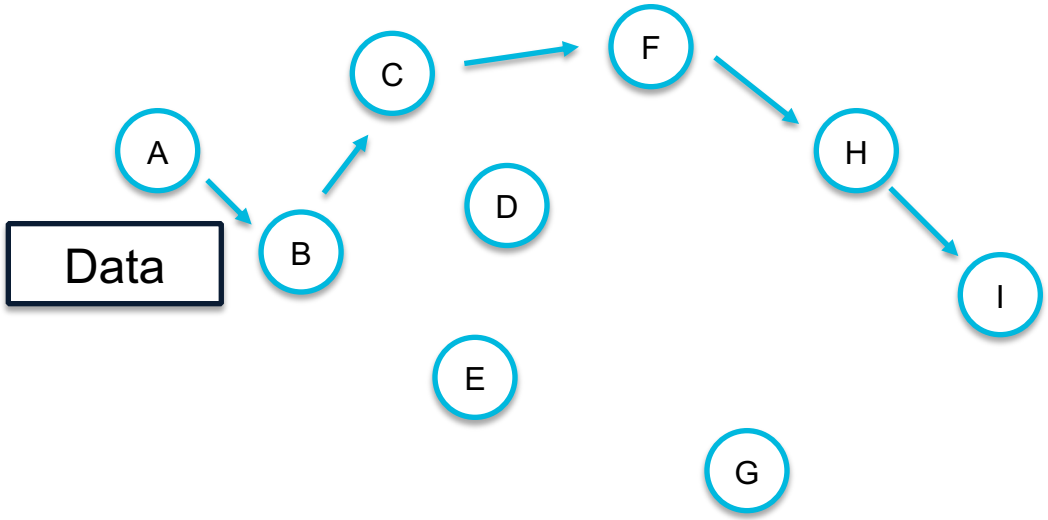
LOADng: Forward Path

- Flood a *Route Request*
- *Route Reply* is forwarded via the reverse path
- The forward path is used to route data packets.

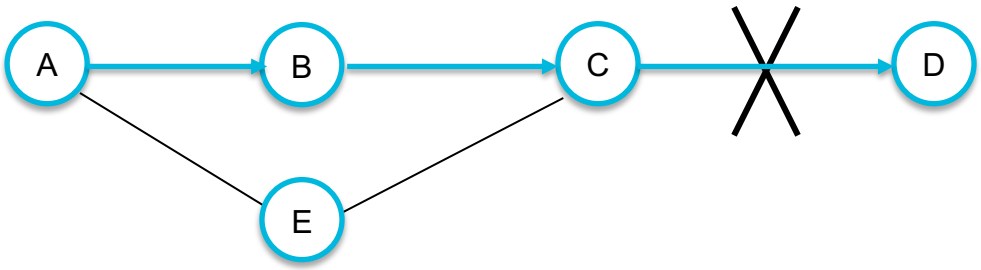


LOADng: Route Expiry

- Flood a *Route Request*
- *Route Reply* is forwarded via the reverse path
- The forward path is used to route data packets
- Unused paths expire based on a timer.

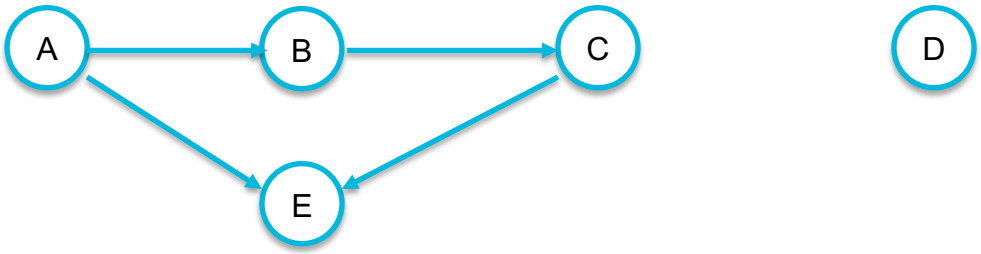


LOADng: Routing Loops



- Assume, link C-D fails, and node A does not know about it (route error packet from C is lost).
- C performs a Route Discovery for D.
- Node A receives the Route Request (via path C-E-A)
- Node A replies, since A knows a route to D via node B

LOADng: Routing Loops

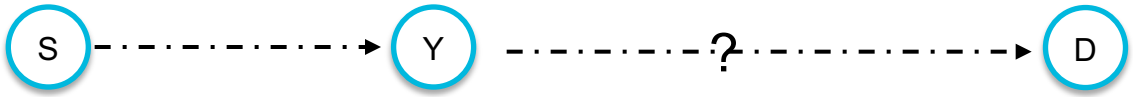


- Assume, link C-D fails, and node A does not know about it (route error packet from C is lost).
- C performs a Route Discovery for D.
- Node A receives the Route Request (via path C-E-A)
- Node A replies, since A knows a route to D via node B
- Results in a loop: C-E-A-B-C

Destination Sequence Number (DSN)

- Each node maintains a Destination Sequence Number (DSN) per destination which represents the "freshness" of the route (acts as a time stamp).
- The higher is the DSN number, more up to date is the path.
- DSN increments every time a node sends a message.

Use of Sequence Numbers in LOADng



Dest seq. no. = 10

Has a route to D with seq. no = 7

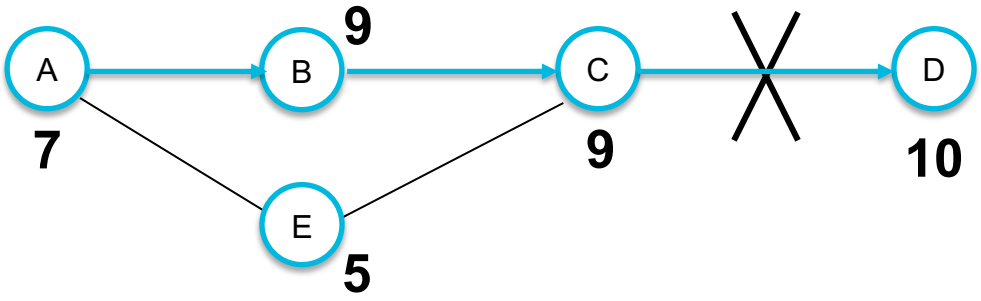
Seq. no. = 15

RREQ carries 10

Y does not reply, but forwards the RREQ

Loop freedom: Intermediate node replies with a Route Reply (instead of forwarding request) **only if it has a route with a higher associated sequence number**.

LOADng: Loop Avoidance



All DSN values are for D

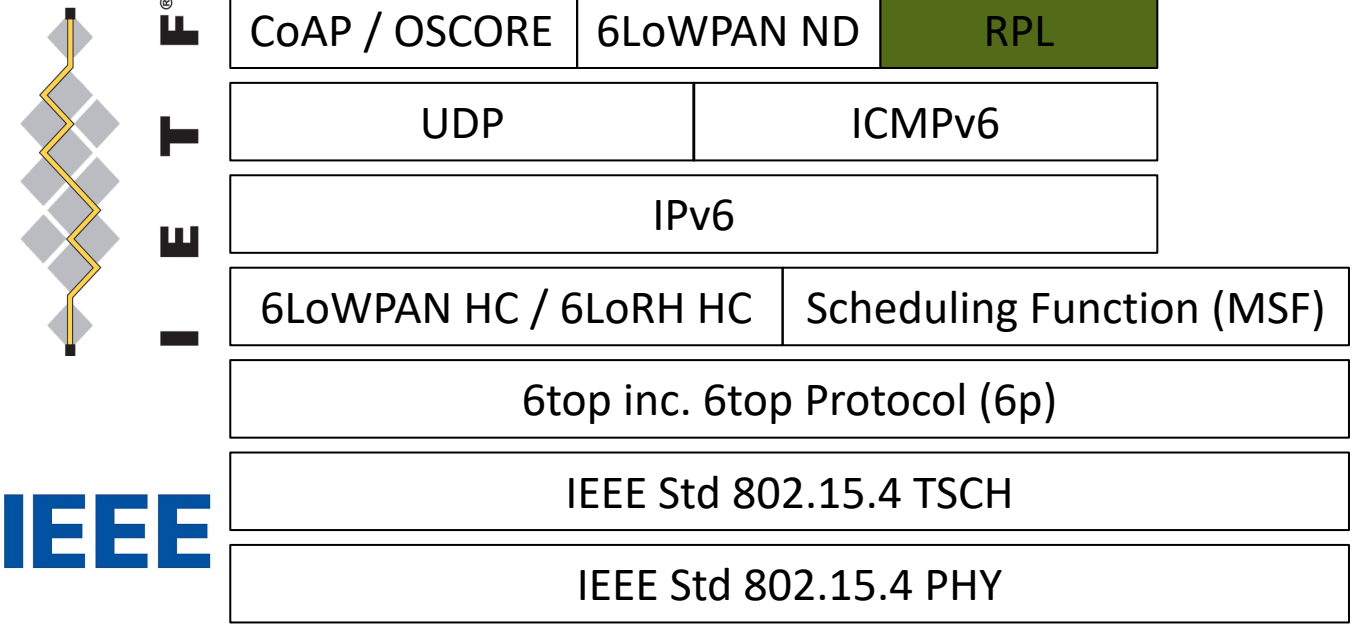
- The C-D link failure increments the DSN at C (now is 10).
- If C needs route to D, RREQ transports the DSN value (10).
- A does not reply as its own DSN is less than 10.

RPL

Feel free to watch the related video on YouTube:
RPL (RFC 6550): the IPv6 Routing Protocol for Low-power and Lossy Networks (LLNs)

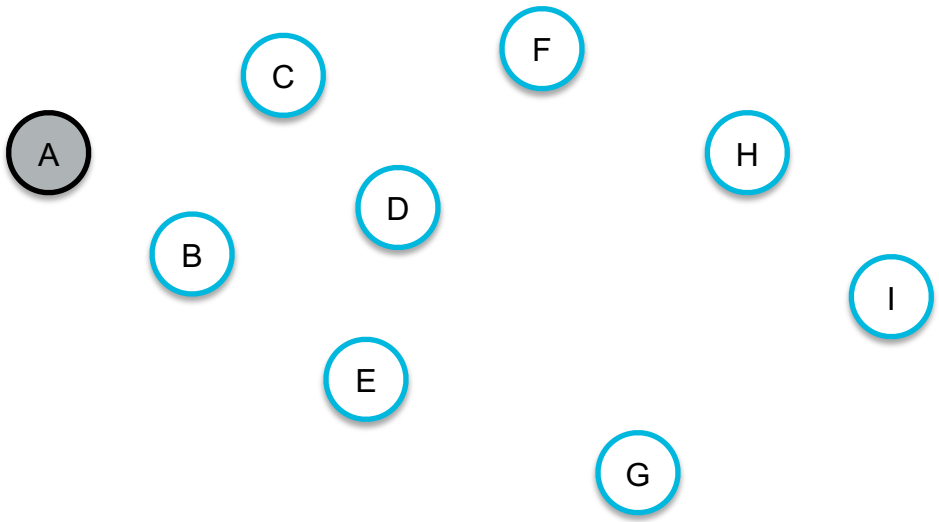
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Context: The 6TiSCH Protocol Stack



RPL

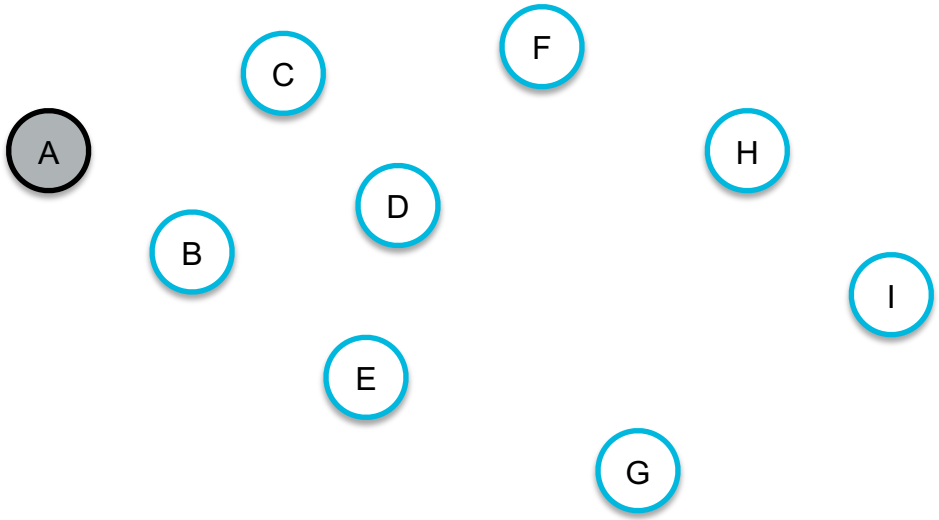
A is the root node.



RPL

A is the root node.

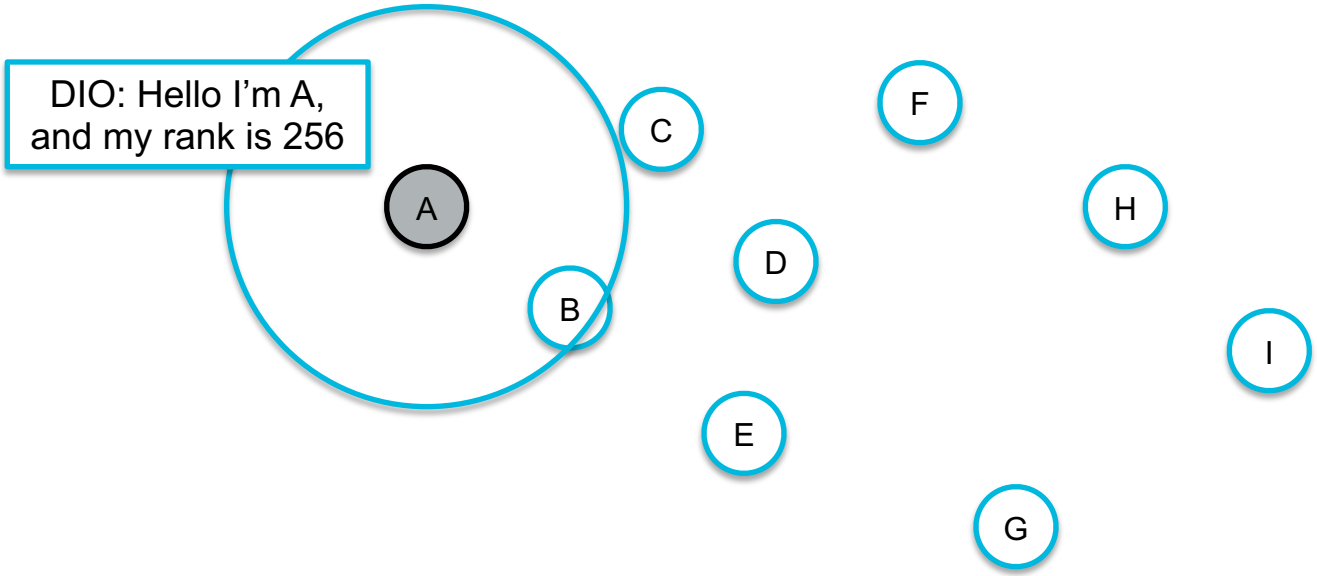
- Periodic transmission of DIO control packets:
  - In RPL network, each node transmit control packets periodically that are called DIO.



RPL

A is the root node.

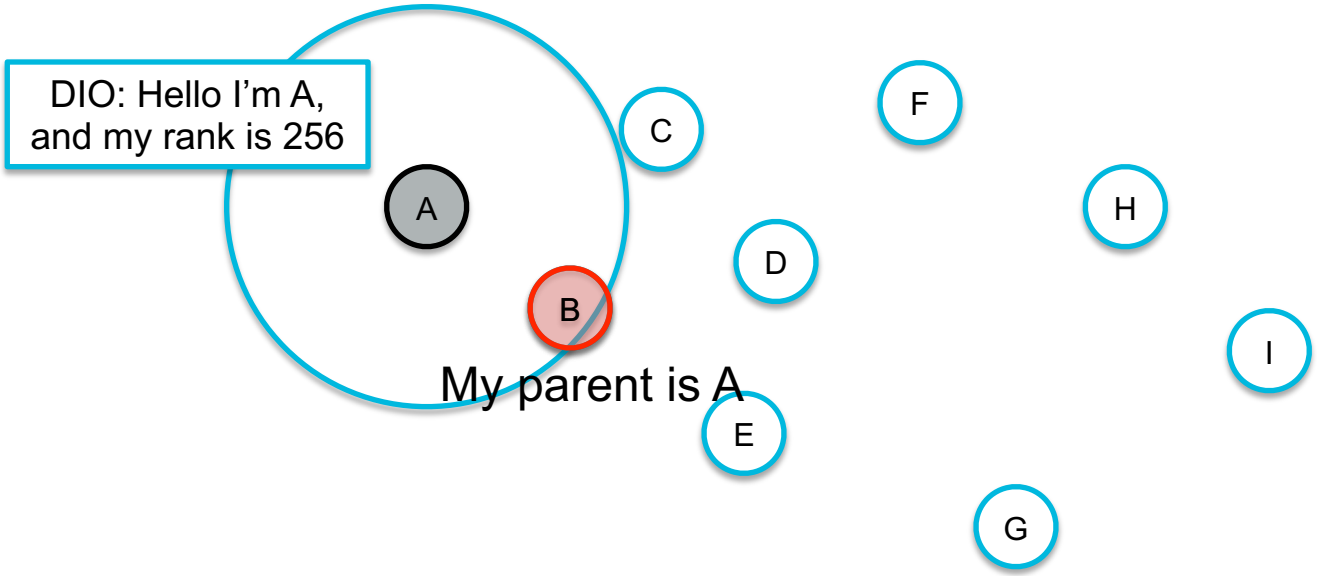
- Periodic transmission of DIO control packets:
  - It starts from the root node.
  - By transmitting the DIO, the nodes advertise the RPL network.
  - Moreover, there is the information about who is the Root, and what is the cost toward the Root.



RPL

A is the root node.

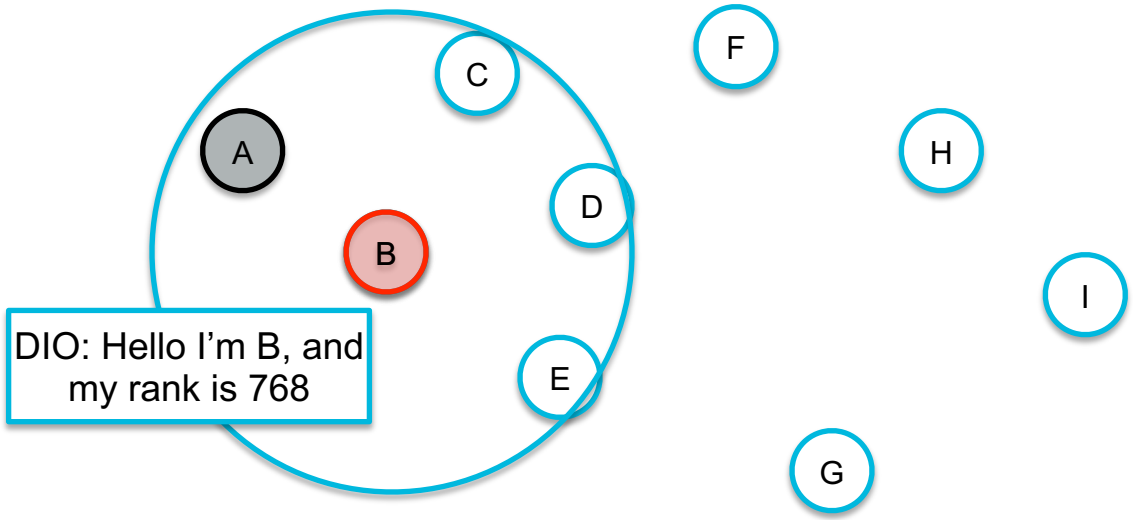
- Periodic transmission of DIO control packets:
  - In this case, only the node B receives this control packet.
  - B will calculate its own Rank, will join the network and will select the root node as its Parent.
  - Since, it is the only DIO packet that it received.



RPL

A is the root node.

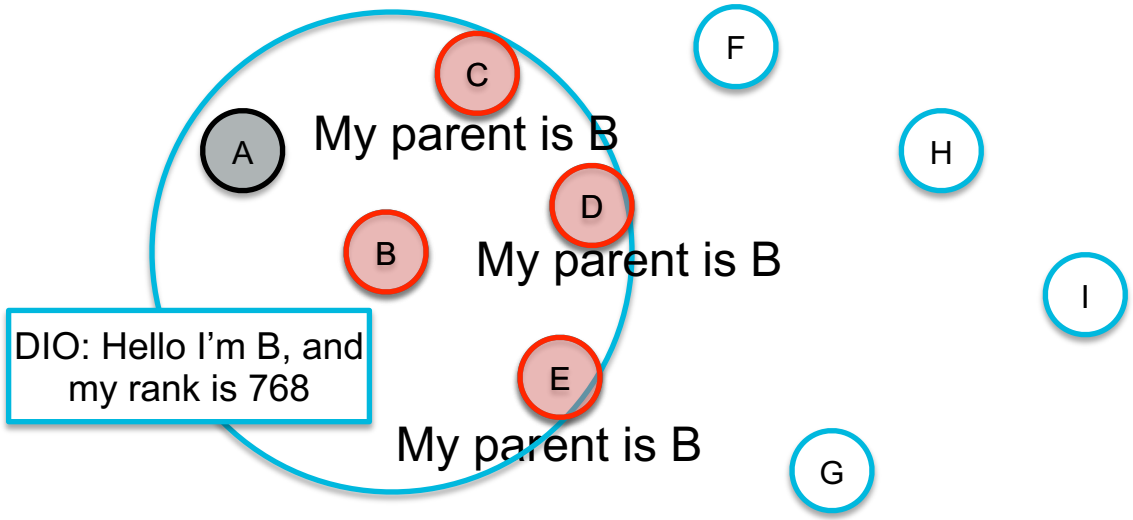
- Periodic transmission of DIO control packets:
  - Then node B will start periodically transmitting DIO control packets indicating its own rank.



RPL

A is the root node.

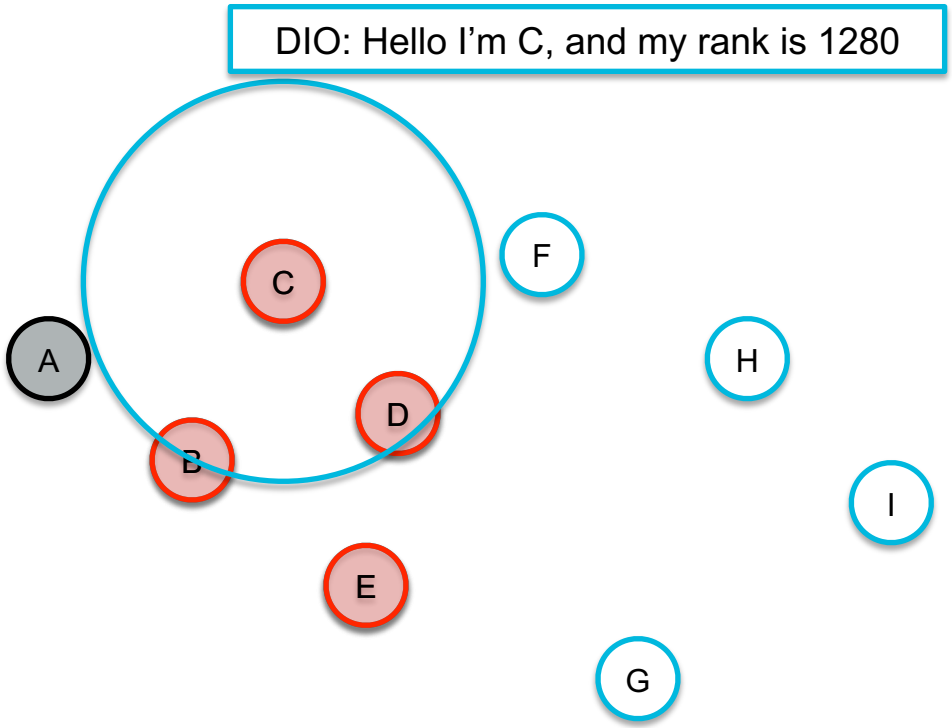
- Periodic transmission of DIO control packets:
  - Nodes C, D and E will receive this DIO control packet sent by the node B, and they will perform the same operation as node B.
  - They will compute their own rank based on the rank of the node B, and will choose node B as their Parent.



RPL

A is the root node.

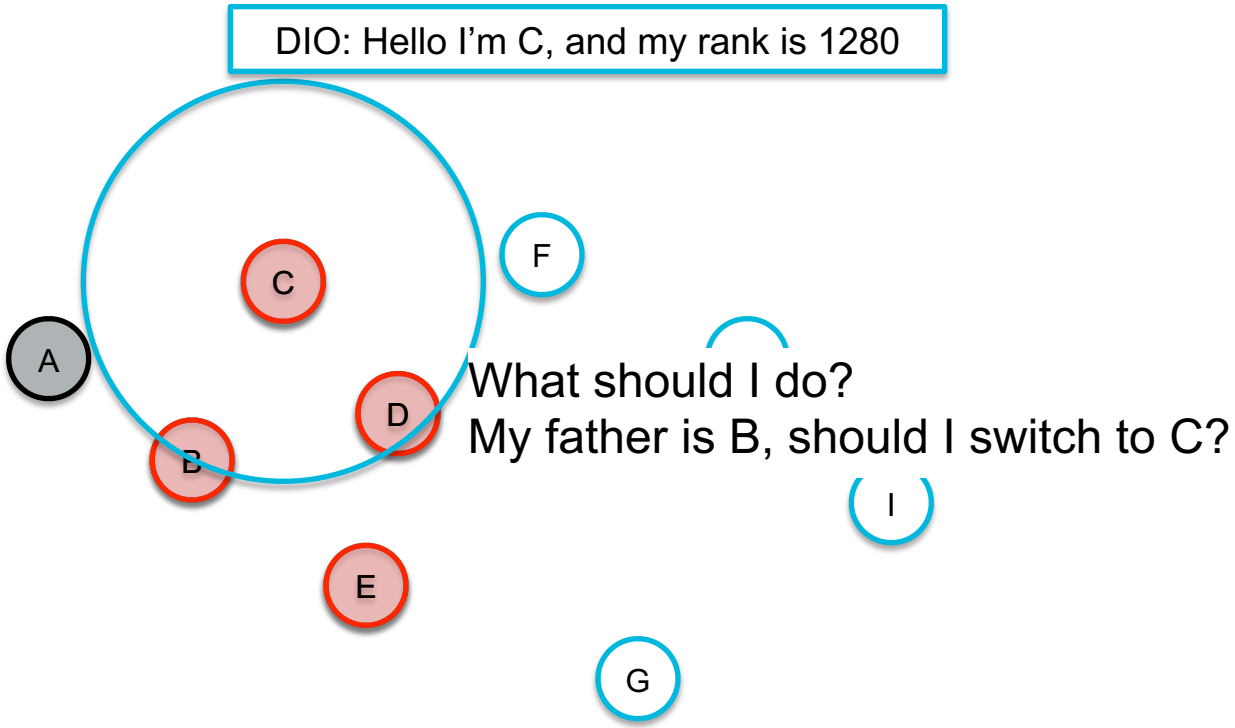
- Periodic transmission of DIO control packets:
  - Then they will also start sending DIO control packets periodically.



RPL

A is the root node.

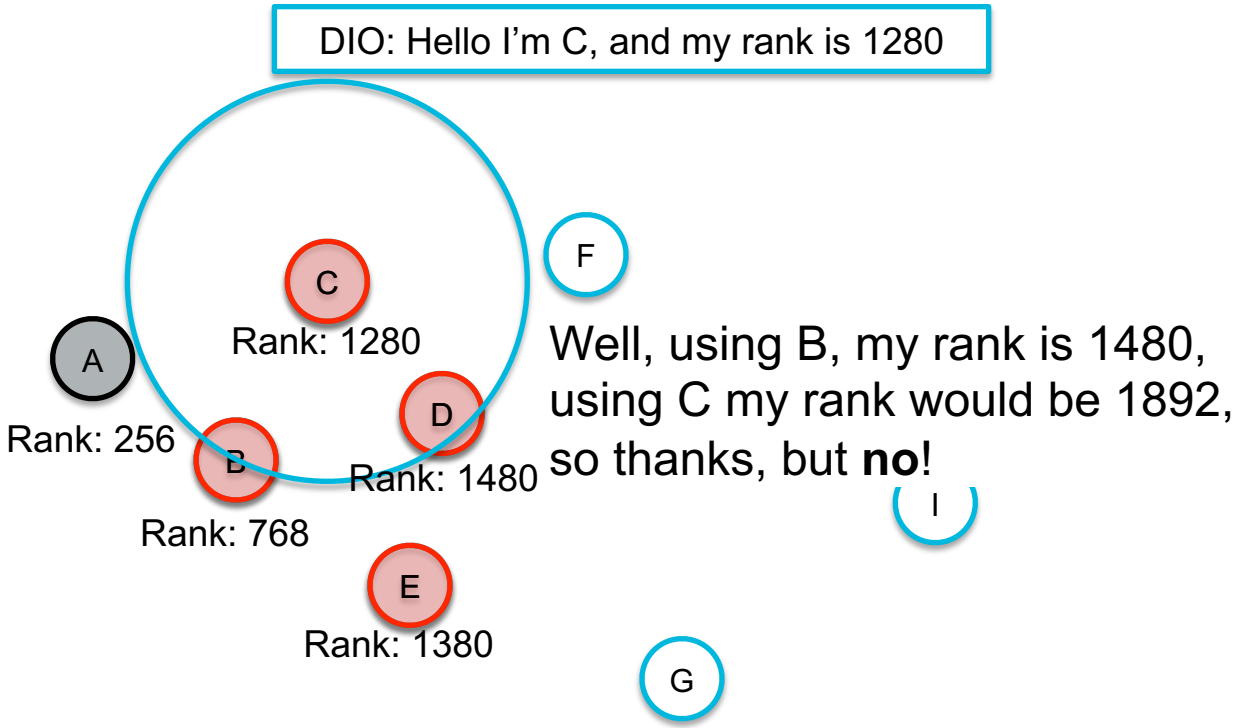
- Periodic transmission of DIO control packets:
  - When the node D receives the DIO from the C, it needs to choose which will be its preferred parent (stay with B or switch to C).



RPL

A is the root node.

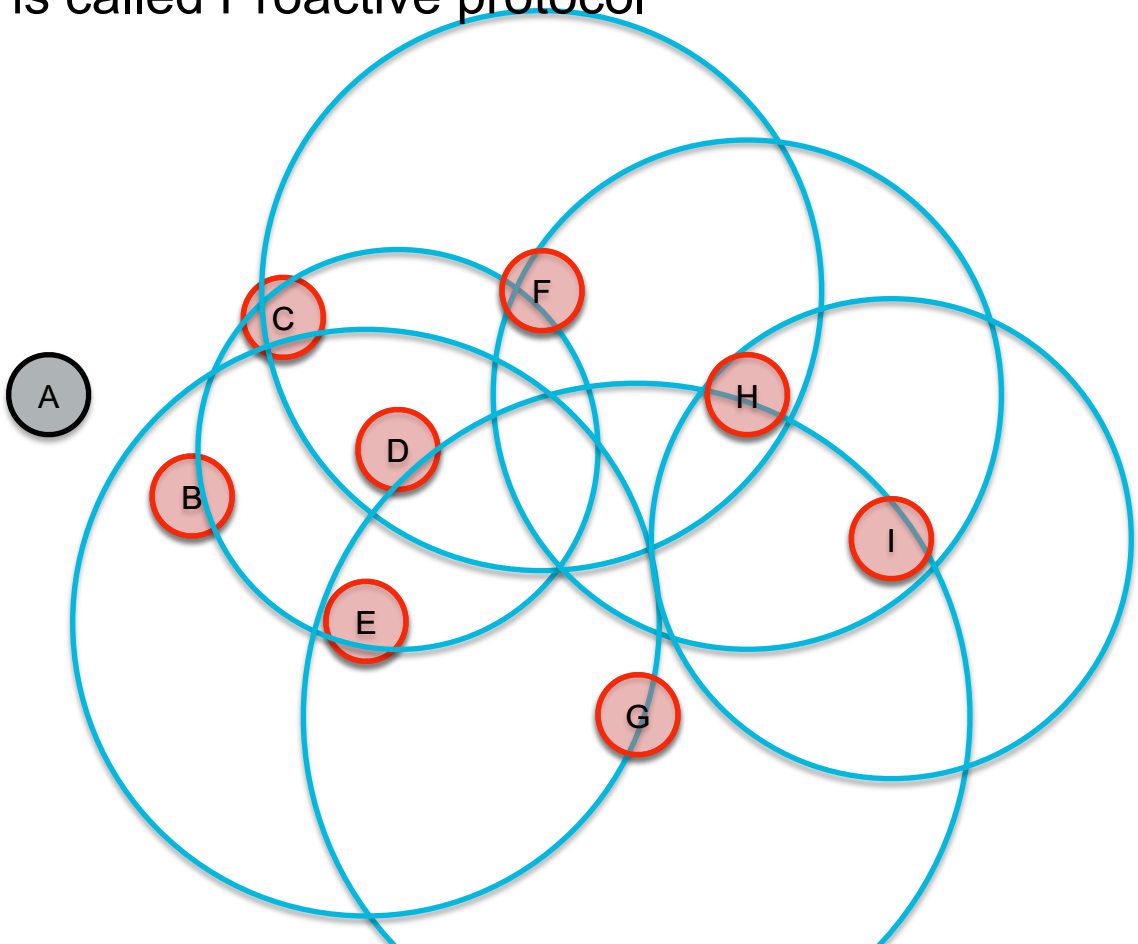
- Periodic transmission of DIO control packets:
  - D will compute the rank that it would obtain through the C; and if this rank is smaller than its own rank, then it will switch to node C.



RPL

A is the root node.

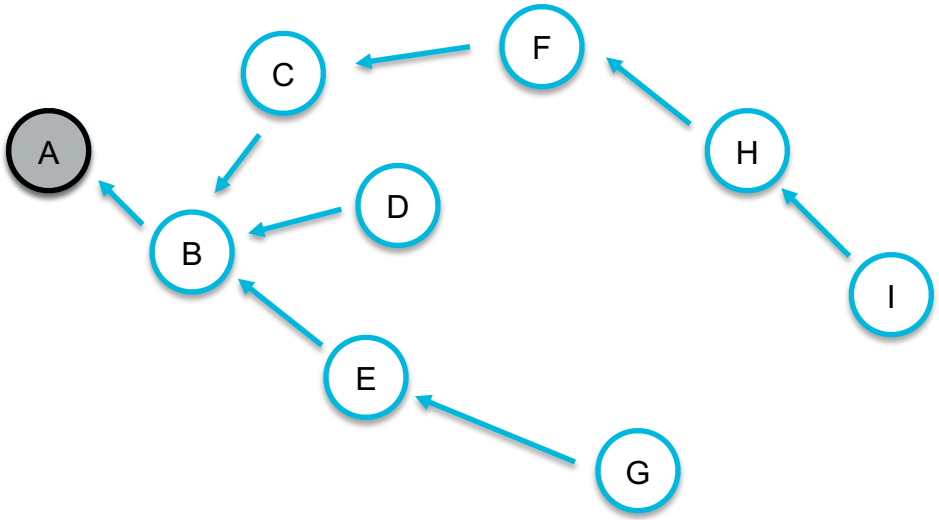
- Periodic transmission of DIO control packets:
  - This process will continue throughout the lifetime of the network.
  - This is why it is called Proactive protocol



RPL

A is the root node.

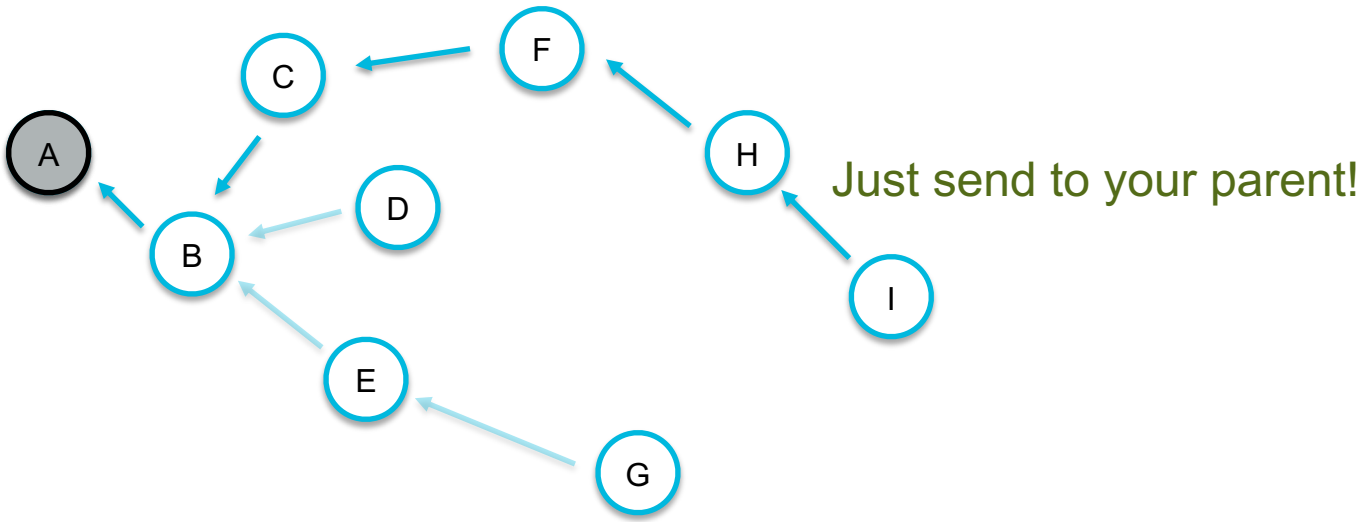
- Periodic transmission of DIO control packets.
- Build a Destination Oriented Directed Acyclic Graph:
  - At the end, all nodes have their preferred parent.
  - And the DODAG is built.



RPL

A is the root node.

- Periodic transmission of DIO control packets.
- Build a Destination Oriented Directed Acyclic Graph.
- Now, the nodes can send their data packets!



CHAPTER OUTLINE

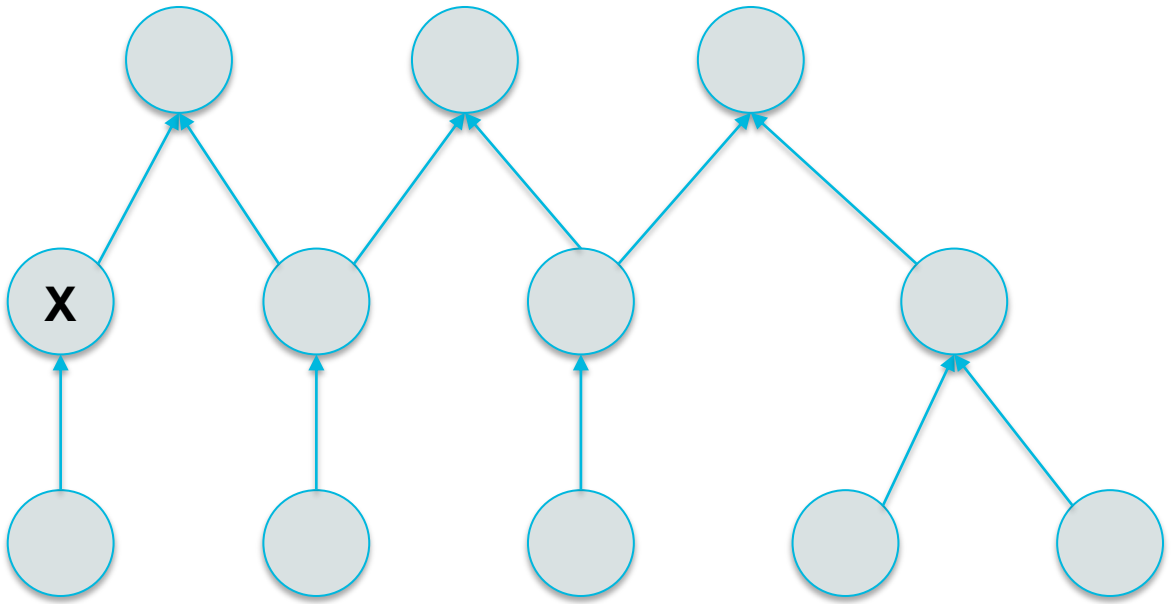
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RPL: Key Concepts

- RPL specifically designed for LLNs
  - Agnostic to underlying link layer technologies (802.15.4, PLC, Low Power WiFi)
- A Decentralized routing protocol
- An extensive IPv6 Distance Vector (DV) protocol
- Source Routing (path addressing) protocol
  - Allows a sender of a packet to partially or completely specify the route the packet takes through the network
- Proactive as opposed to Reactive

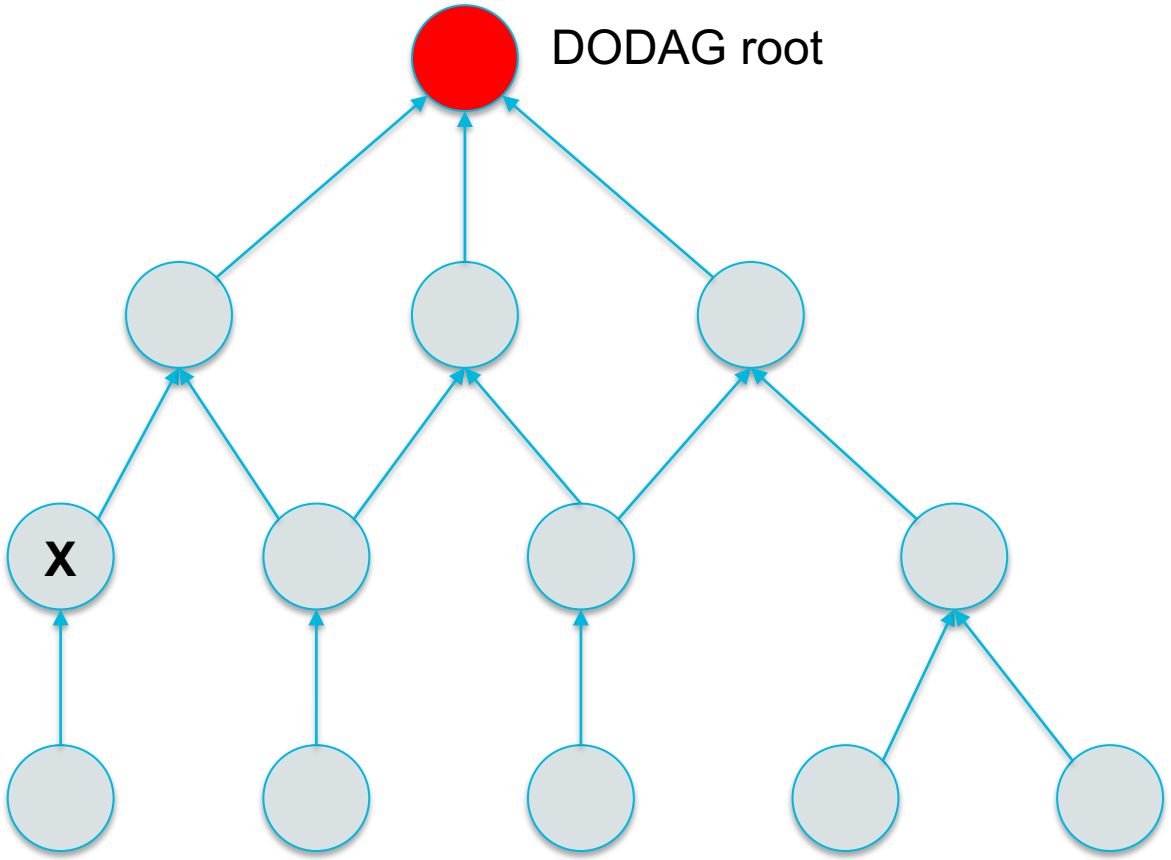
Directed Acyclic Graph (DAG)

In the context of routing, a DAG consists of vertices/sommets (nodes) and edges/d'arêtes (links), each *edge/link* connecting one node to another (directed) in such a way that it is not possible to start at *Node X* and follow a directed path that cycles back to *Node X (**acyclic**)*.



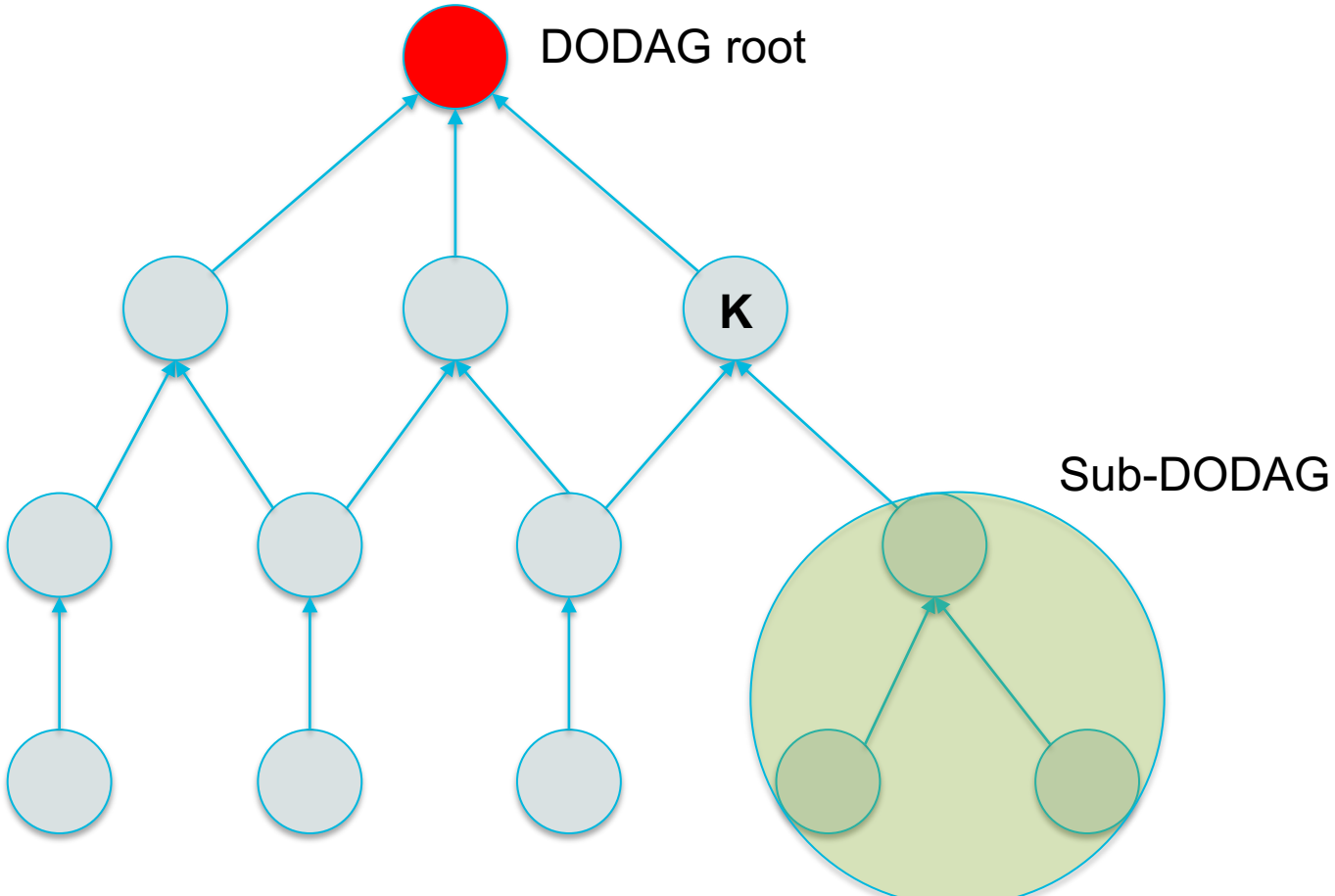
Destination Oriented DAG (DODAG)

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



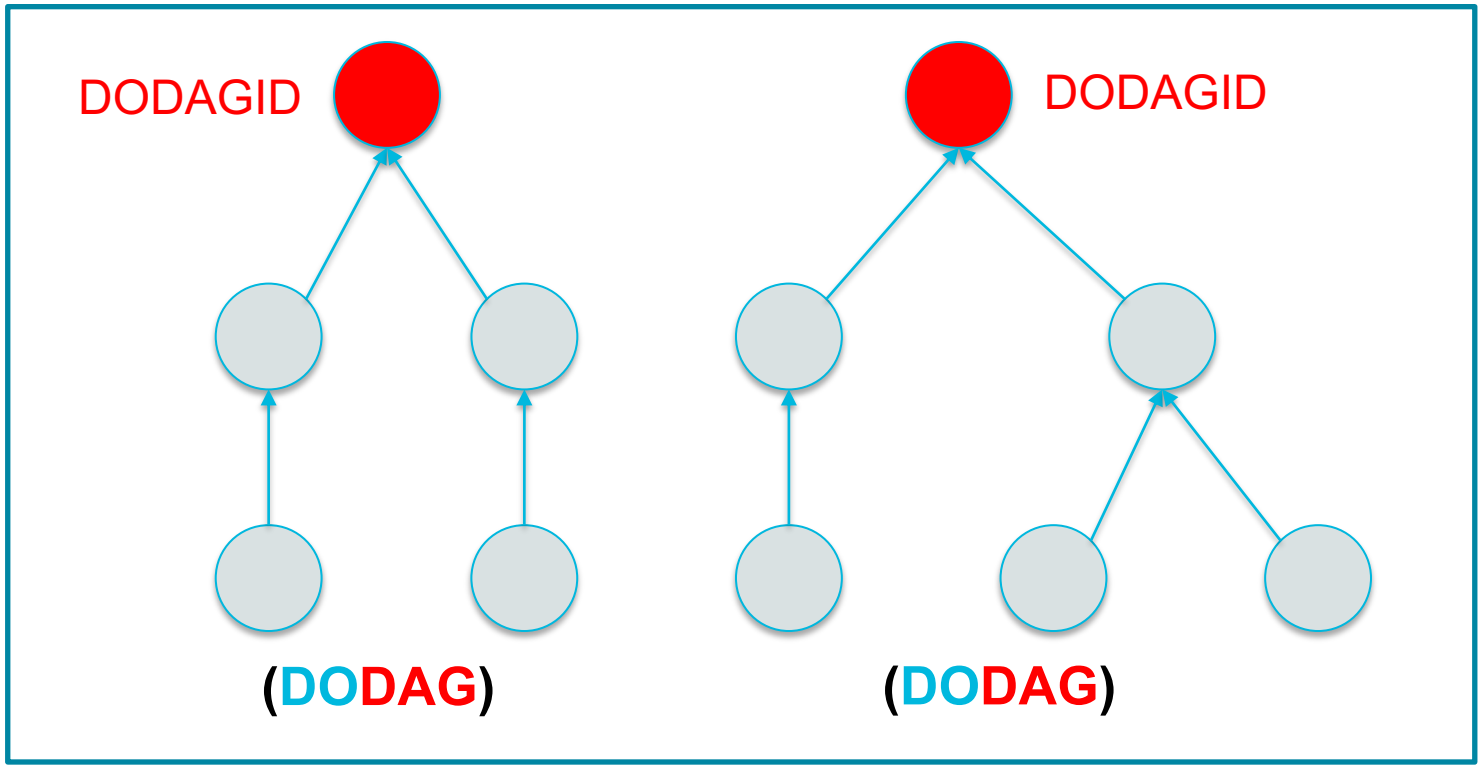
Sub-DODAG

The **sub-DODAG** of a node is the set of other nodes whose paths to the DODAG root pass through that node. Nodes in the sub-DODAG of a node have a greater Rank than that node.



RPL Instance

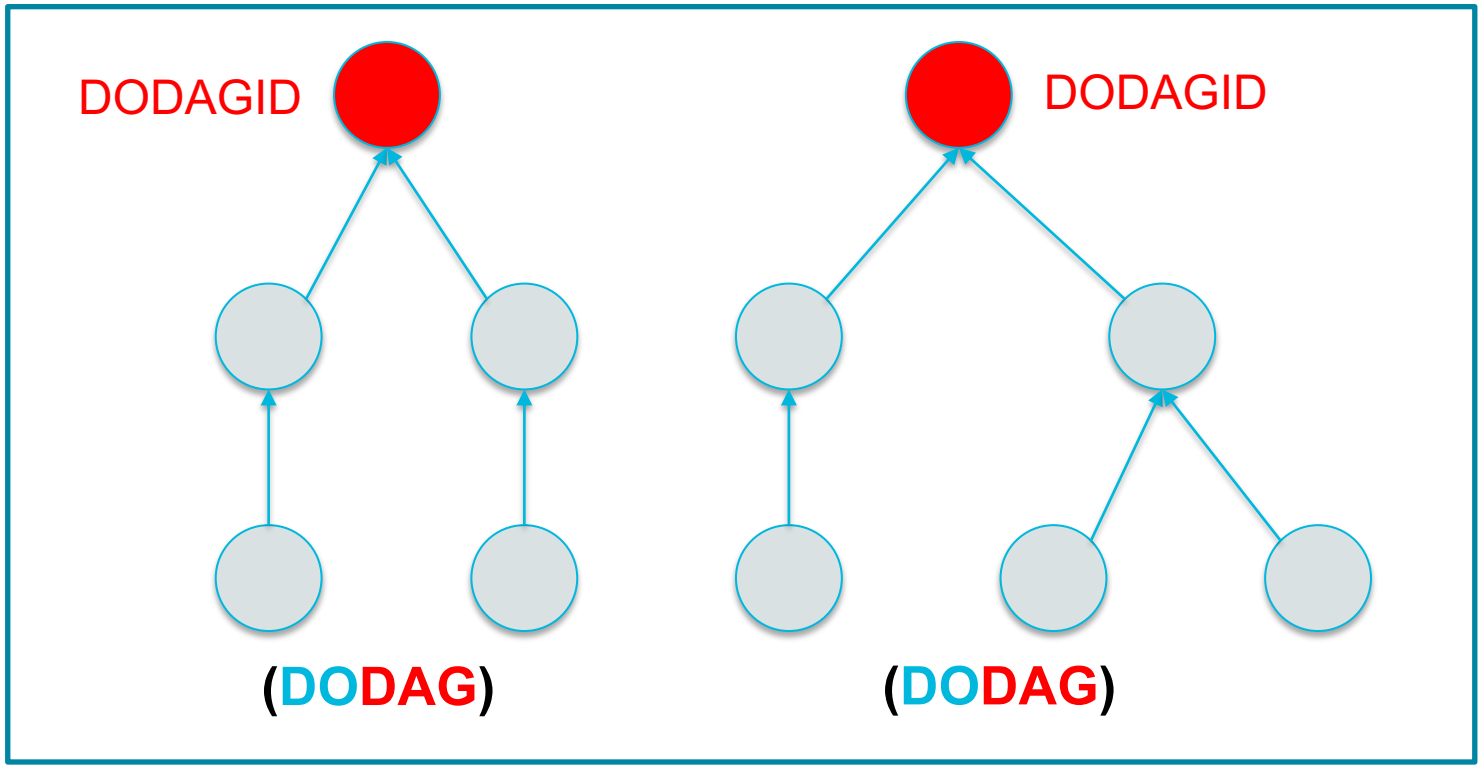
A **RPL Instance** consists of one or more DODAGs that share a RPLInstanceID.



RPL Instance

RPLInstanceID

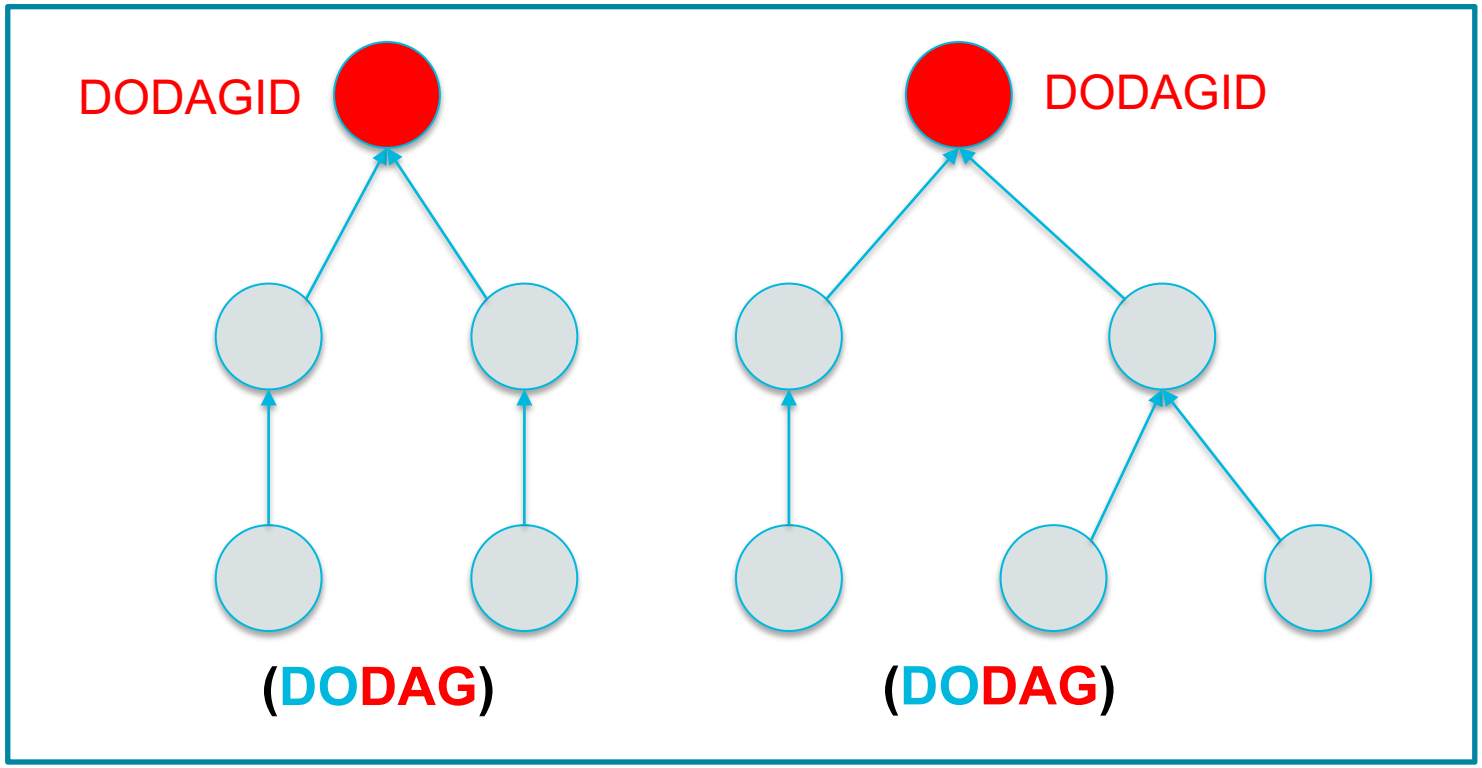
Each RPL Instance is identified by RPLInstanceID, a unique identifier within a network. DODAGs with the same RPLInstanceID share the same Objective Function (OF) used to compute the position of node in the DODAG.



RPL Instance

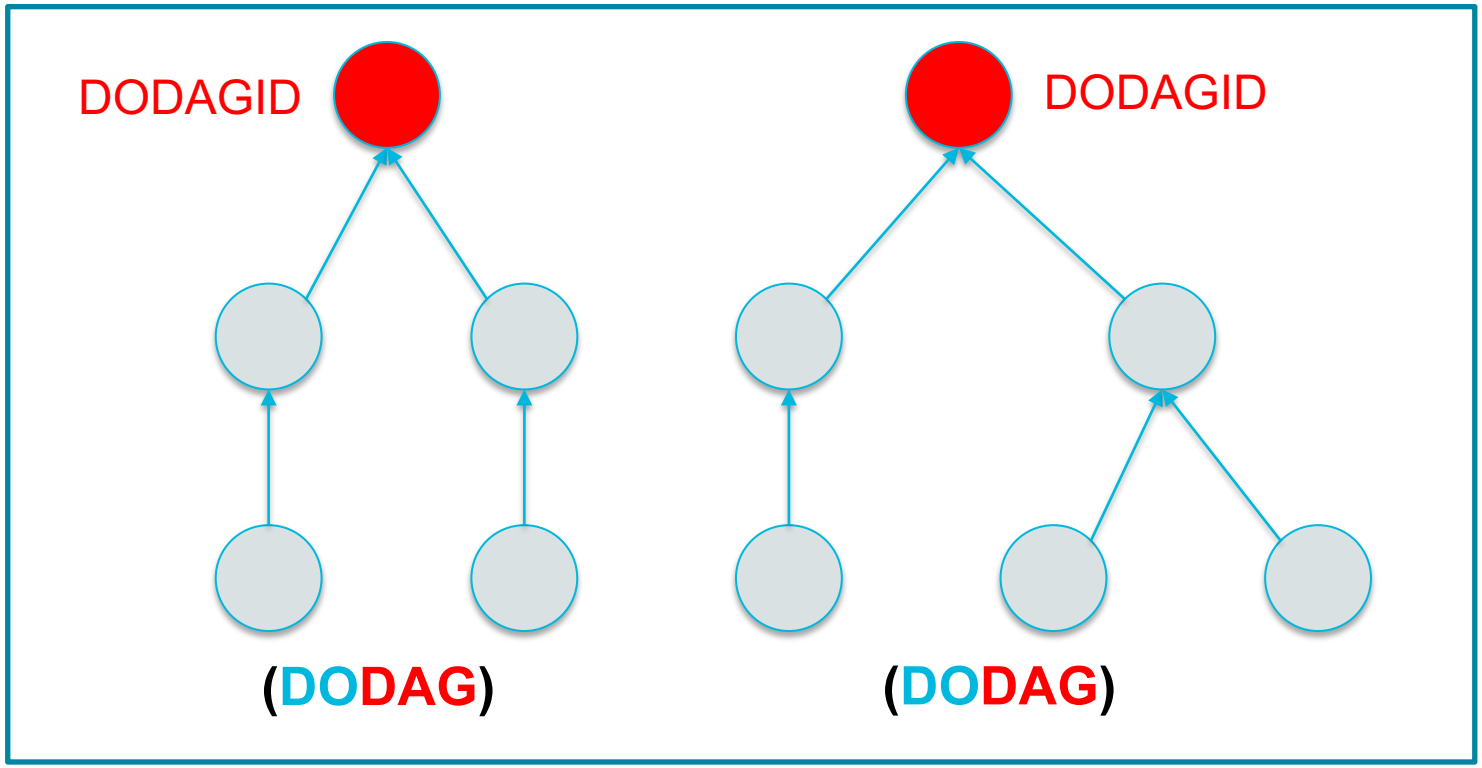
DODAGID

DODAG Root is identified by **DODAGID** (node IPv6 address)



DODAG Version

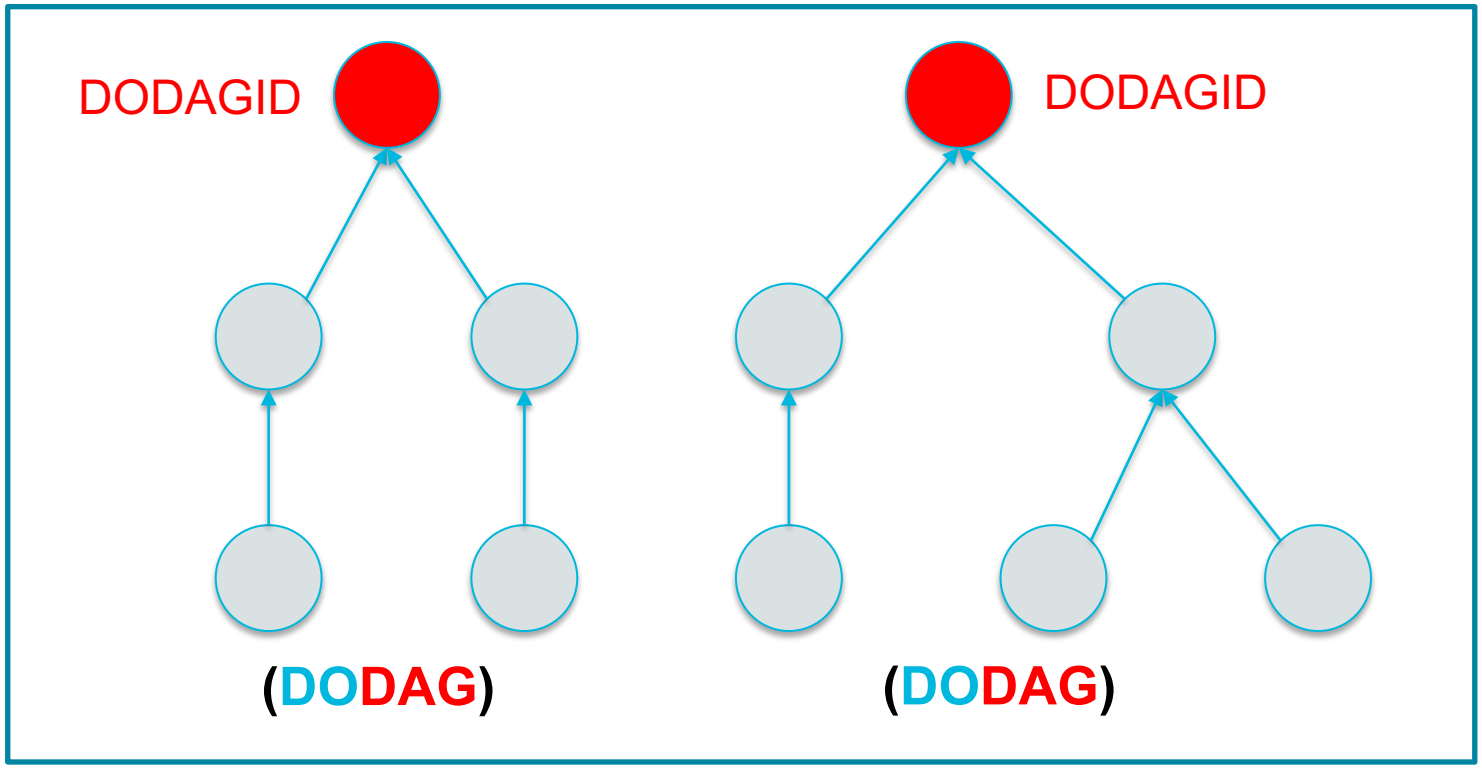
DODAG Version is a specific iteration ("Version") of a DODAG with a given DODAGID.



DODAGVersionNumber

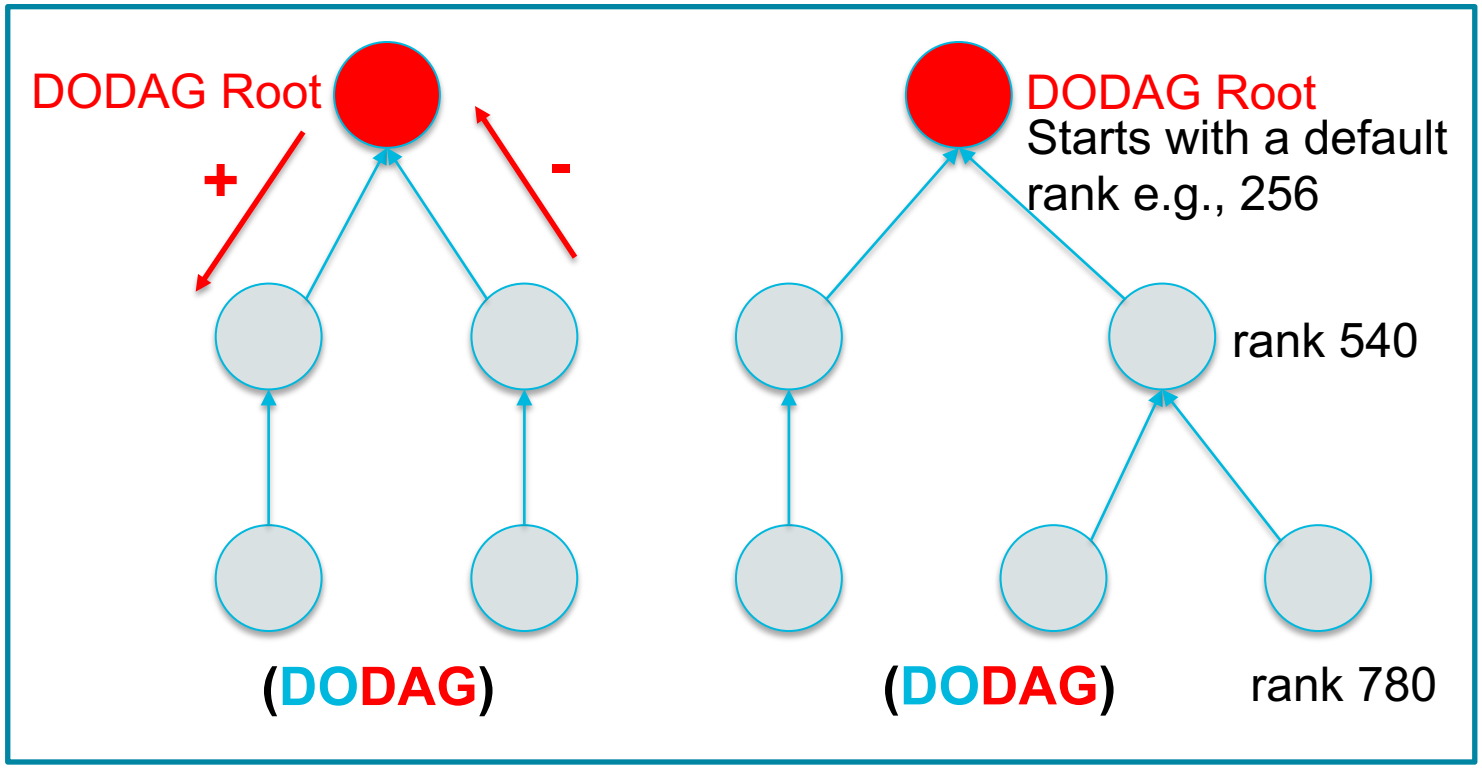
- A DODAG is sometimes reconstructed from the DODAG root.

DODAGVersionNumber is a sequential counter that is incremented by the root to form a new Version of a DODAG. A DODAG Version is identified uniquely by the (*RPLInstanceID*, *DODAGID*, *DODAGVersionNumber*) tuple.

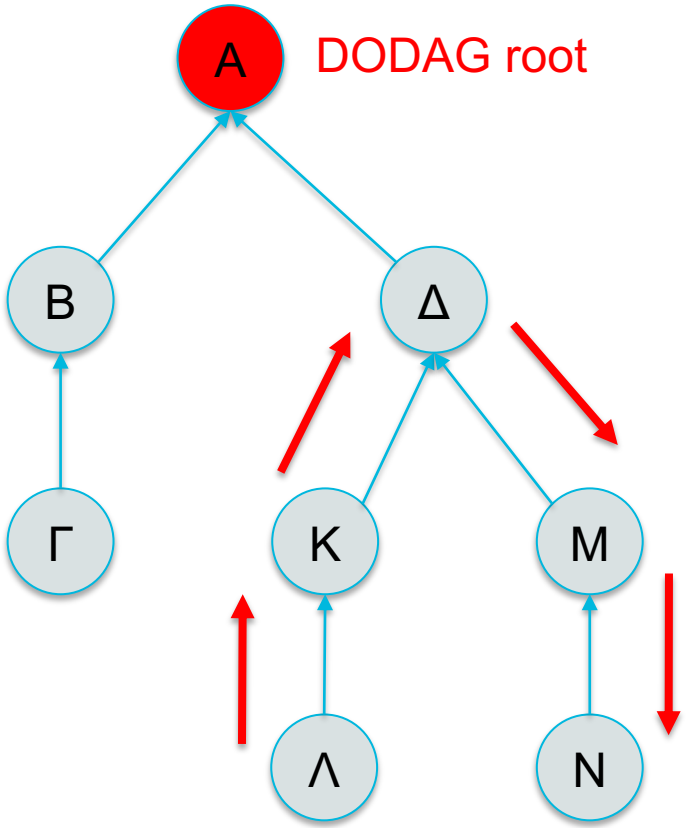


Rank

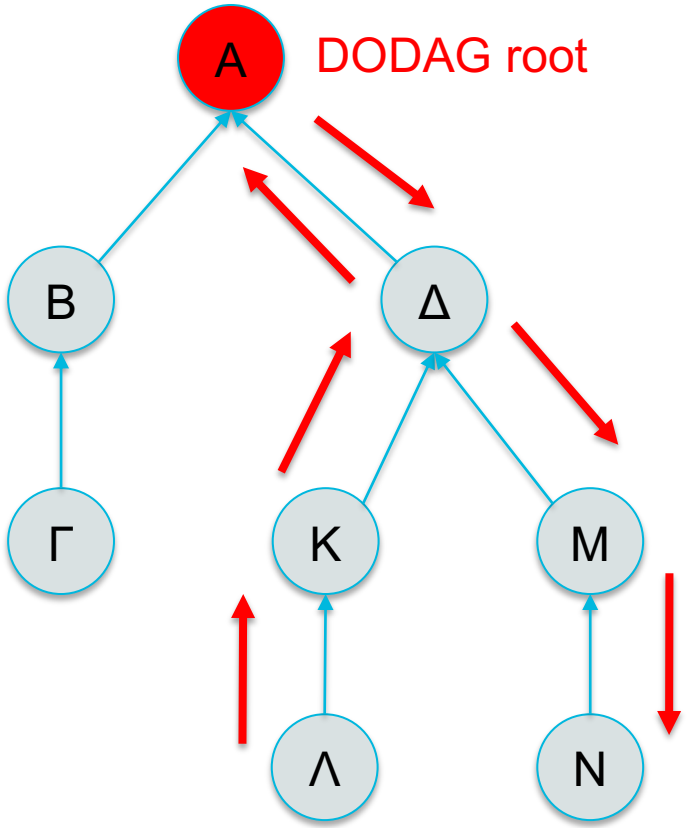
RANK defines the node's individual position with respect to DODAG root, and strictly increases in the "Down" direction and strictly decreases in the "Up" direction.



Storing and Non-Storing mode-of-operation



Storing



Non-storing

Storing and Non-Storing mode-of-operation

- A storing the nodes keep 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.

Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

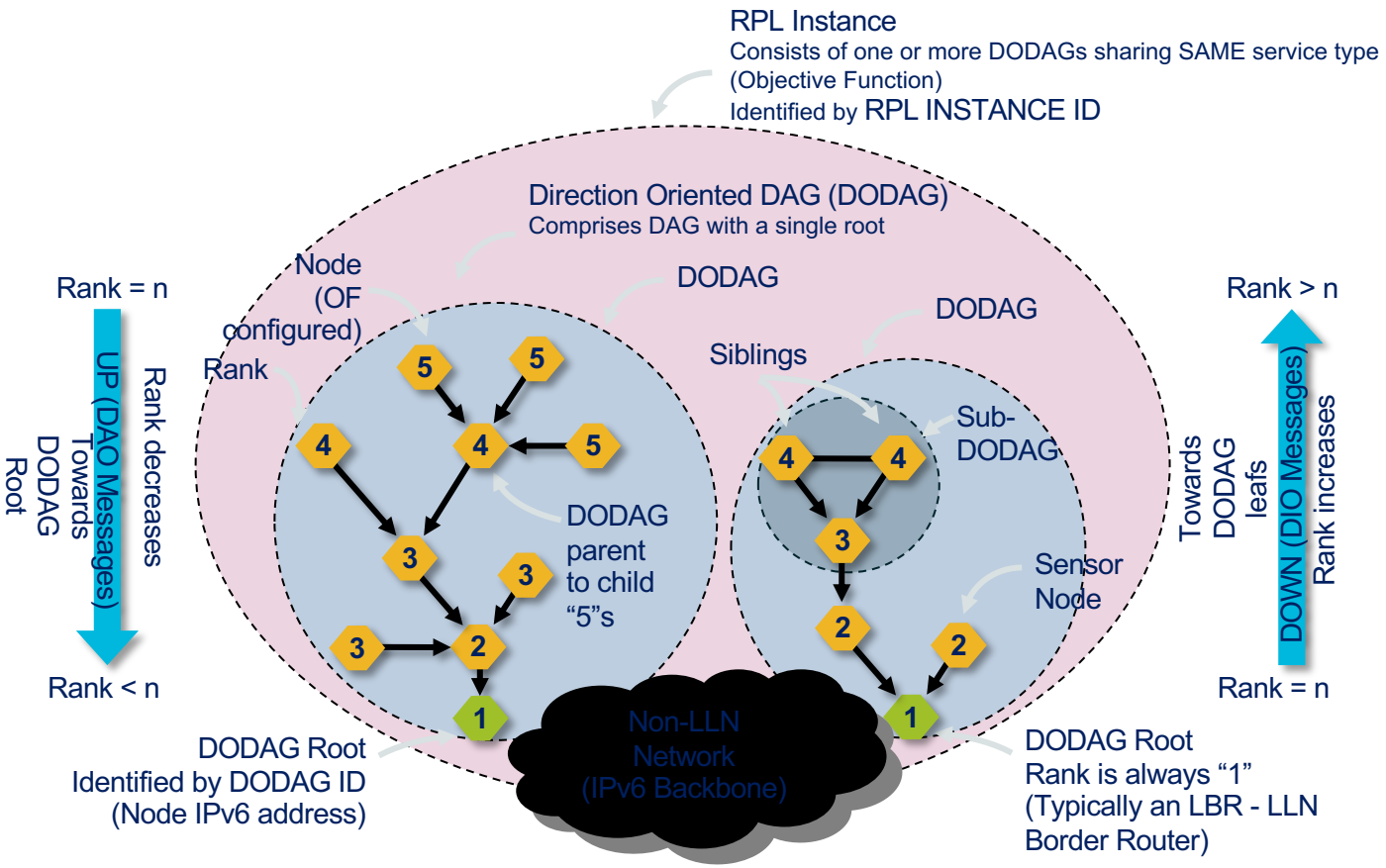
Storing and Non-Storing mode-of-operation

- A storing the nodes keep 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 the nodes send all traffic to root. Root uses source routes to send traffic to leafs.
  - limited by ram of DODAG root, but usually non-constrained device
  - requires more bits on wire, which often is more expensive (energy-wise) than more ram, or compute cycles.

Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

RPL Terminology

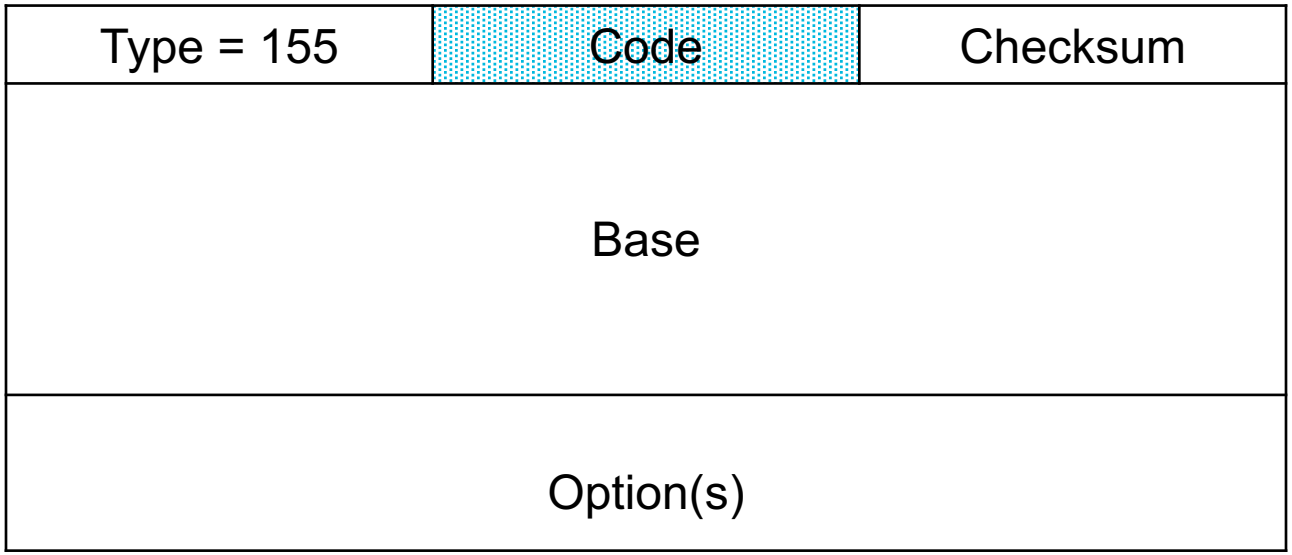


From Pascal Thubert "RPL 2017" slides

RPL Control Messages

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RPL Control Messages are ICMPv6 Messages



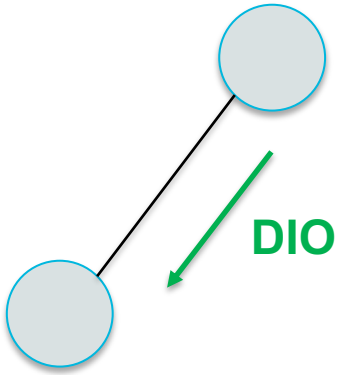
Code: Identify the type of control message

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

DODAG Information Object (DIO)

Carries information that allows a node to:

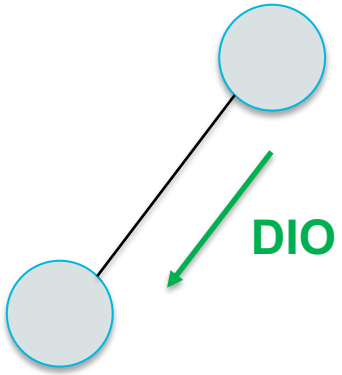
- Discover a RPL instance
- Learn its configuration parameters
- Select a DODAG parent set (ranks)
- Maintain the DODAG



DODAG Information Object (DIO)

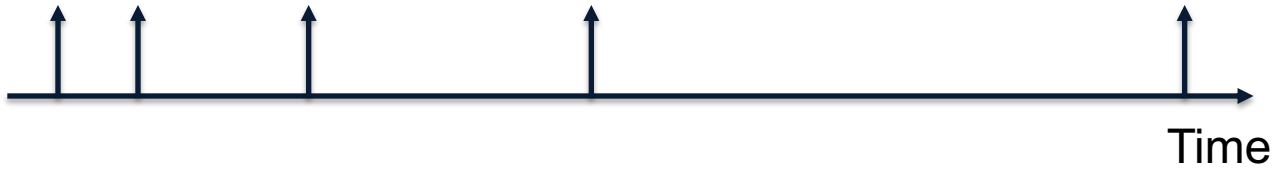
Carries information that allows a node to:

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



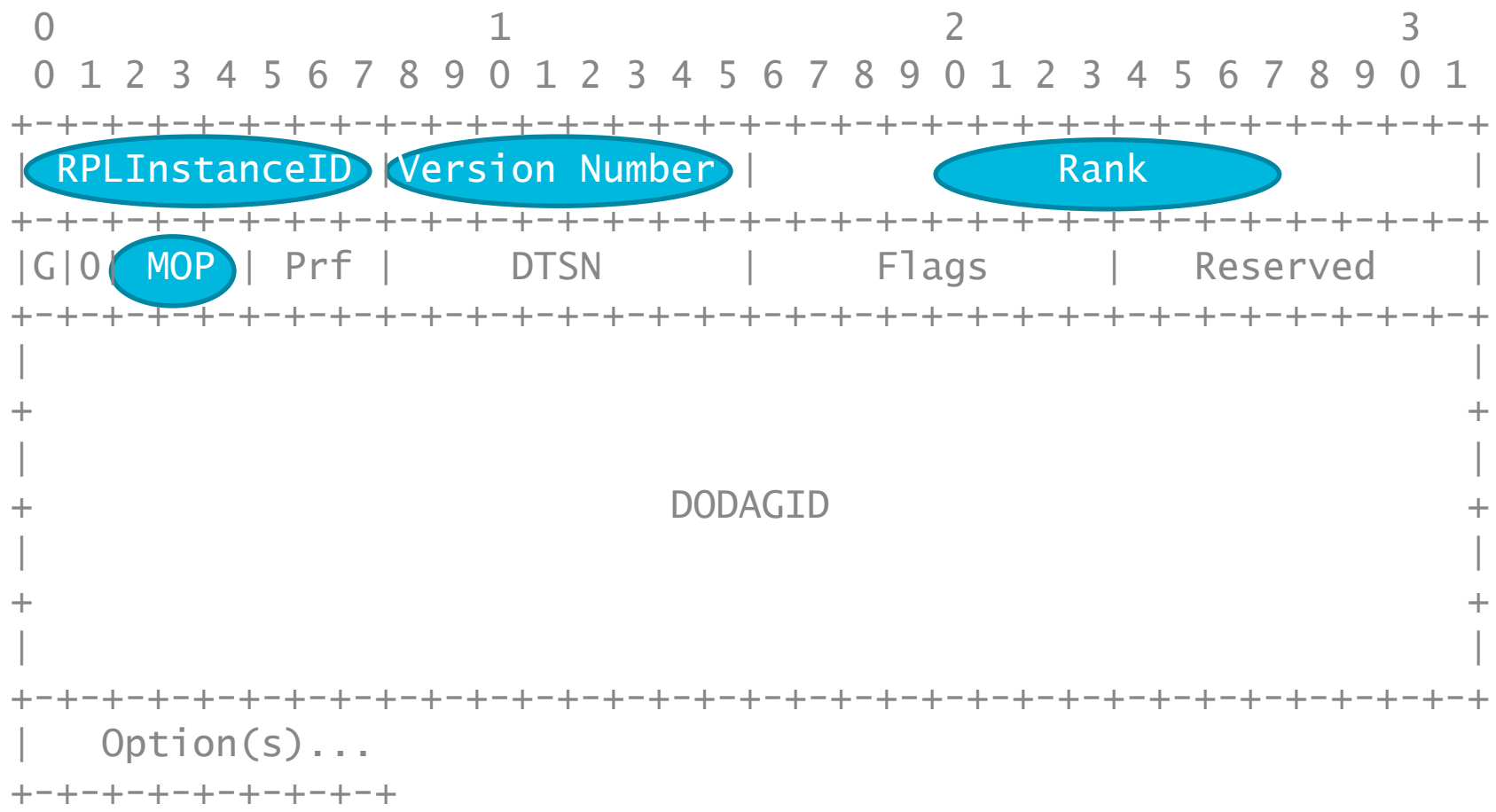
RPL uses Trickle timer to control the unnecessary transmissions of DIO messages i.e., overhead.

- 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.



Double the time if the network is stable

DODAG Information Object (DIO)



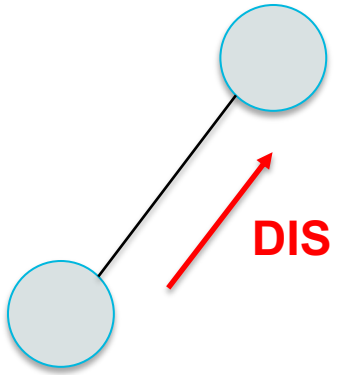
The DIO Base Object

DODAG Information Solicitation (DIS)

- Solicit a DIO message from a RPL node, in other words, ask routers around to generate DIO message.
- It is similar to that of a Router Solicitation of IPv6 Neighbor Discovery.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |   Reserved    |   Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                The DIS Base Object
```

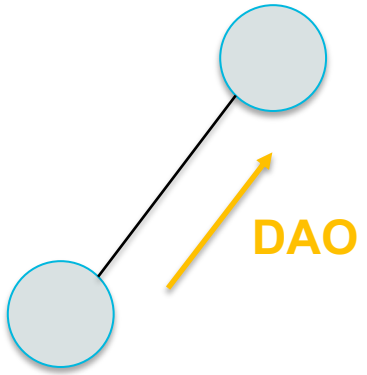
```
Options:
      0x00 Pad1
      0x01 PadN
      0x07 Solicited Information
```



Destination Advertisement Object (DAO)

Propagates destination information Upward along the DODAG

DAO messages are used to construct and maintain downward (descendante) routes (from Root to other nodes).



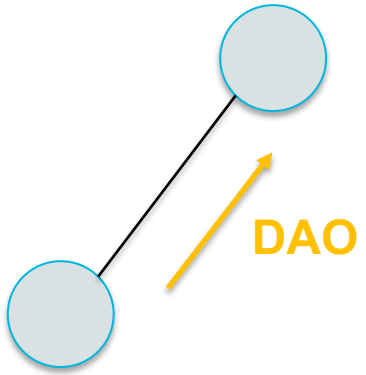
Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

Destination Advertisement Object (DAO)

Propagates destination information Upward along the DODAG

DAO messages are used to construct and maintain downward (descendante) routes (from Root to other nodes).

- 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.



Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

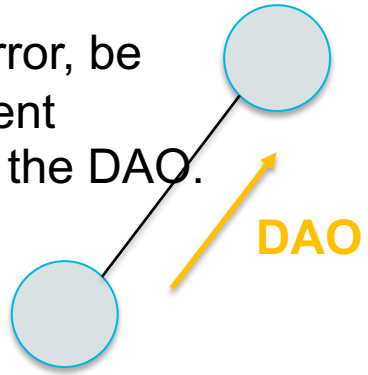
Destination Advertisement Object (DAO)

Propagates destination information Upward along the DODAG

DAO messages are used to construct and maintain downward (descendante) routes (from Root to other nodes).

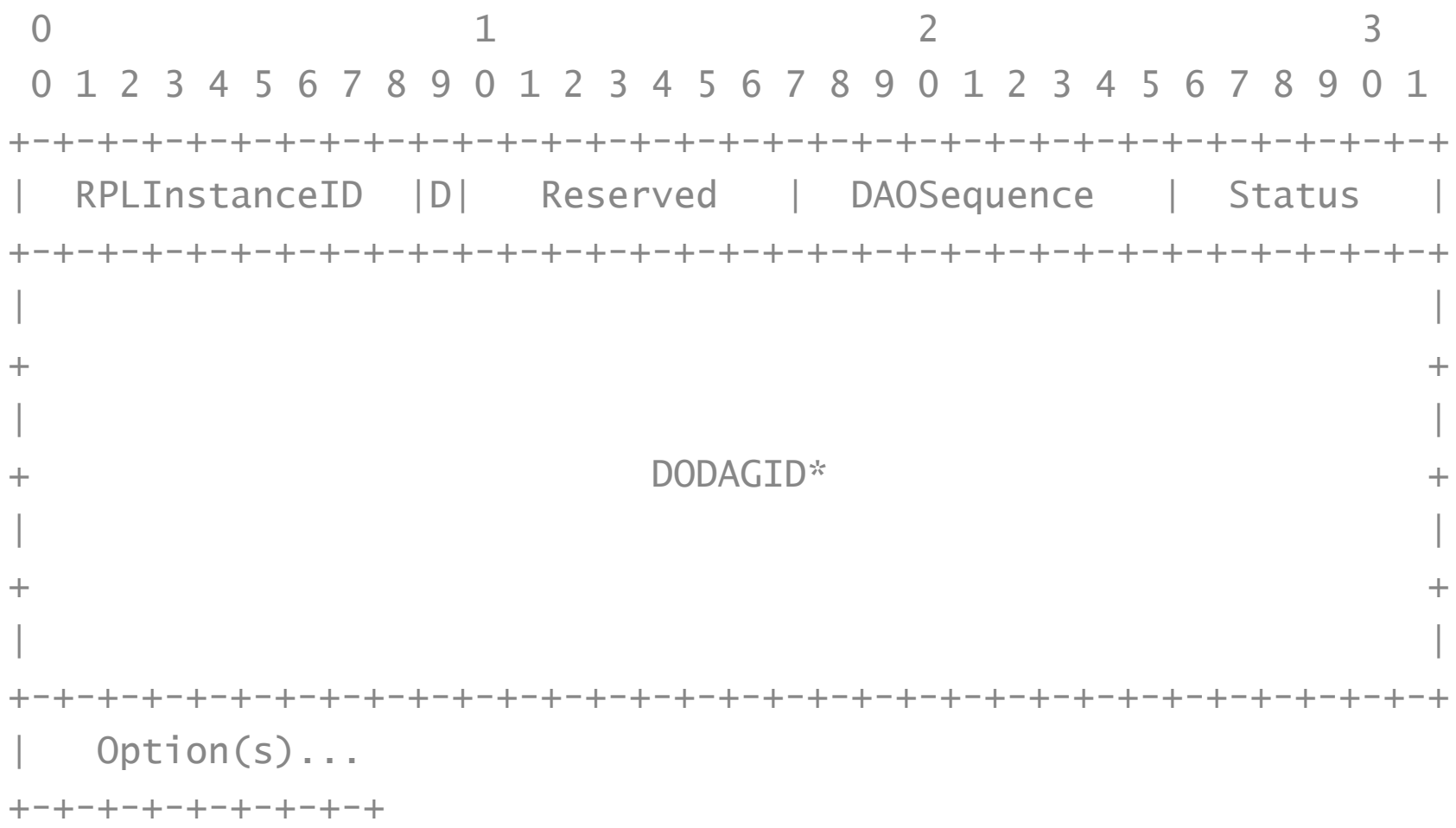
- In Storing mode, the DAO message is unicast by the child to the selected parent(s).
- In Non-Storing mode, the DAO message is unicast to the DODAG root.

- 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.



Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

Destination Advertisement Object Acknowledgement (DAO-ACK)



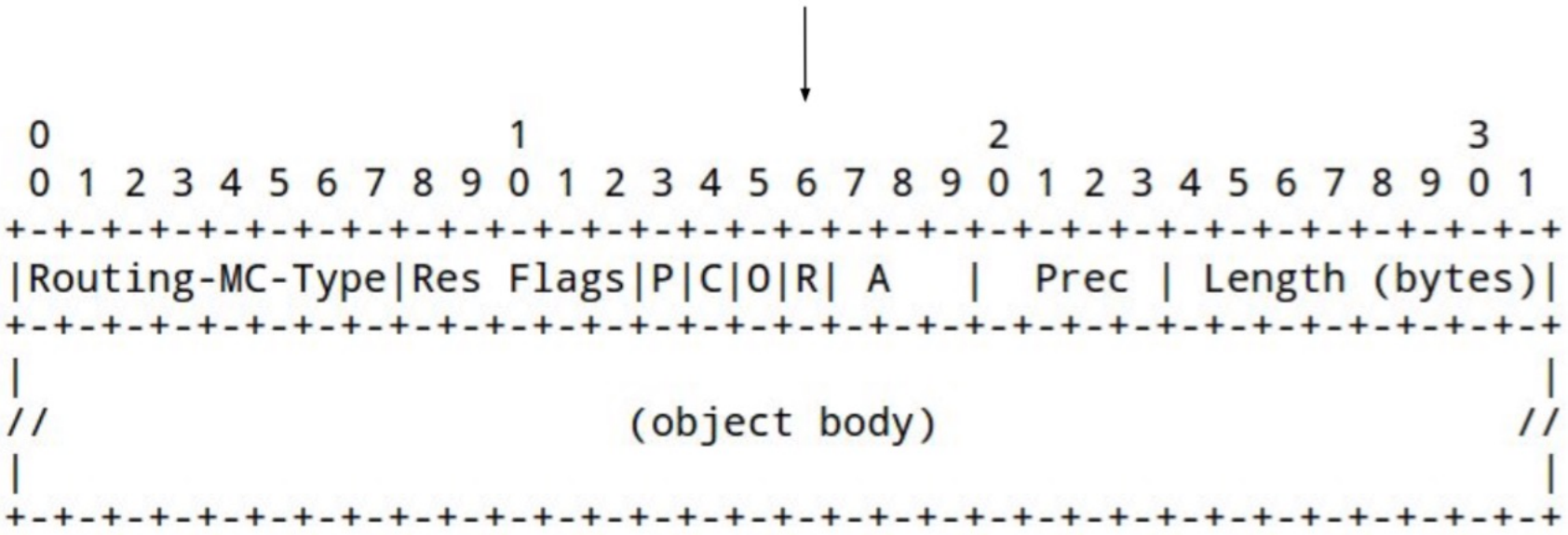
The DAO-ACK Base Object

DAG Metric Container

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

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.
- The metrics can be either Node-based or link-based.



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 (bits per second).

Latency:

- Can be used as a metric or constraint.
- Constraint: Max latency allowable on path.
- Metric: additive metric updated along path.

Link Reliability:

- 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).

Objective Functions

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RFC 6550: Objective Function (OF)

- Defines how RPL nodes select and optimize routes within a RPL Instance.
- Translates one or more metrics into a Rank, and its purpose is to avoid loops.
- Defines how RPL nodes select parents.

The Objective Function (OF) defines *how RPL nodes select and optimize routes within a RPL Instance*. The OF is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how nodes translate one or more metrics and constraints, which are themselves defined in [RFC6551], into a value called Rank, which approximates the node's distance from a DODAG root. An OF also defines how nodes select parents.

Objective Functions (OFs)

- OF0: Objective Function Zero:
  - Based on the parent rank + link property.
- MRHOF: Minimum Rank with Hysteresis OF:
  - Based on additive metrics.
  - For example, it calculates the end-to-end metric, e.g., of the reliability, the throughput or the latency.
- CAOF: Common Ancestor OF:
  - A multi-path routing, based on common ancestor + additive metric.

Objective Function Zero (OF0): RFC 6552

- OF0 is designed as a default OF.
- OF0 is designed to find the root, by selecting a preferred parent and a backup successor if one is available.
- Then, all the upward (*ascendant*) traffic is routed via the preferred parent.

OF0 works by computing the rank based on the addition of a scalar representing the link properties to the rank of the preferred parent.

Objective Function Zero (OF0): RFC 6552

my rank

my parent rank

$$R(N) = R(P) + rank\_increase$$

$$rank\_increase = ((Rf * Sp + Sr) * MinHopRankIncrease)$$

rank_factor (Rf), (strictly positive integer):
A configurable factor that is used to multiply the effect of the link properties in the rank_increase computation.
If none is configured, then a rank_factor of 1 is used.

Arbitrary number (eg. 256)

stretch_of_rank (Sr), (unsigned integer):
The maximum augmentation to the step_of_rank of a preferred parent to allow the selection of an additional feasible successor.
If none is configured to the device, then the step_of_rank is not stretched.

step_of_rank (Sp) : Depends on the link quality, and uses the Expected Transmission Count (ETX) [RFC 6551], values between [1; 9]

RFC 8180, Section 5.1.1 : OF0 Parameter Values

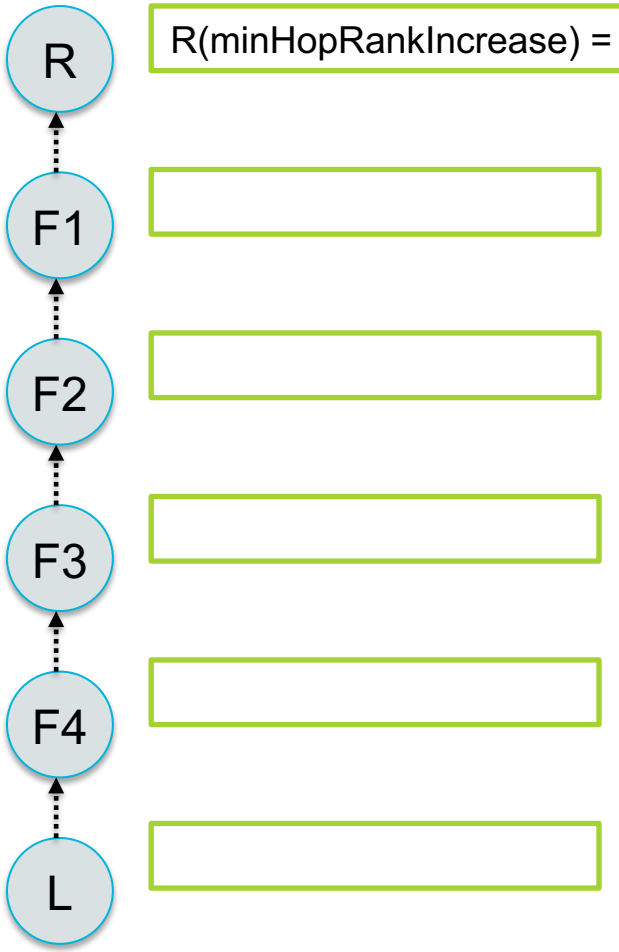
OF0 Parameters	Value
Rf	1
Sp	(3 * ETX) - 2
Sr	0
MinHopRankIncrease	DEFAULT_MIN_HOP_RANK_INCREASE (256)
MINIMUM_STEP_OF_RANK	1
MAXIMUM_STEP_OF_RANK	9
ETX limit to select 3 a parent	3

An implementation MUST follow OF0's normalization guidance as discussed in Sections 1 and 4.1 of [RFC6552].

- Sp SHOULD be calculated as (3*ETX)-2.
- The minimum value of Sp (MINIMUM_STEP_OF_RANK) indicates a good quality link.
- The maximum value of Sp (MAXIMUM_STEP_OF_RANK) indicates a poor quality link.
- The default value of Sp (DEFAULT_STEP_OF_RANK) indicates an average quality link.
- Candidate parents with ETX greater than 3 SHOULD NOT be selected.

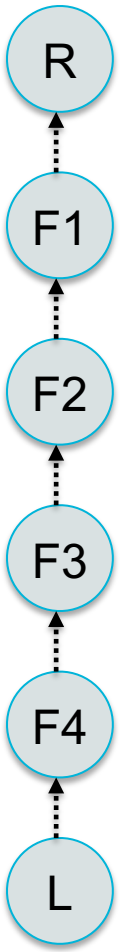
This avoids having ETX values on used links that are larger that the maximum allowed transmission attempts.

OF0 Example



Rank computation example for a 5-hop network, where numTx=100 and numTxAck=75 for all links.

RFC 8180, Section 5.1.2 : OF0 Example



R(minHopRankIncrease) = 256

R(F1)=R(R0) + 512 = 768

R(F2)=R(F1) + 512 = 1280

R(F3)=R(F2) + 512 = 1792

R(F4)=R(F3) + 512 = 2304

R(L)=R(F4) + 512 = 2816

R(N) = R(P) + rank_increment

rank_increment = (Rf*Sp + Sr) * minHopRankIncrease =>

rank_increment = (1*((3*ETX) -2) + 0) * 256 =>

rank_increment = ((3*ETX) -2) * 256 =>

rank_increment = ((3*numTx/numTxAck) - 2) * 256 =>

rank_increment = ((3*100/75) - 2) * 256 = 512

Rank computation example for a 5-hop network, where numTx=100 and numTxAck=75 for all links.

Local vs Global Repair

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Global versus Local Repair

- Global repair: A new DODAG iteration:
  - This initiates a new DODAG Version.
  - A new Sequence number (DODAGVersionNumber) generated by the root.
  - Nodes in the new DODAG Version can choose a new position whose Rank is not constrained by their Rank within the old DODAG Version.

Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

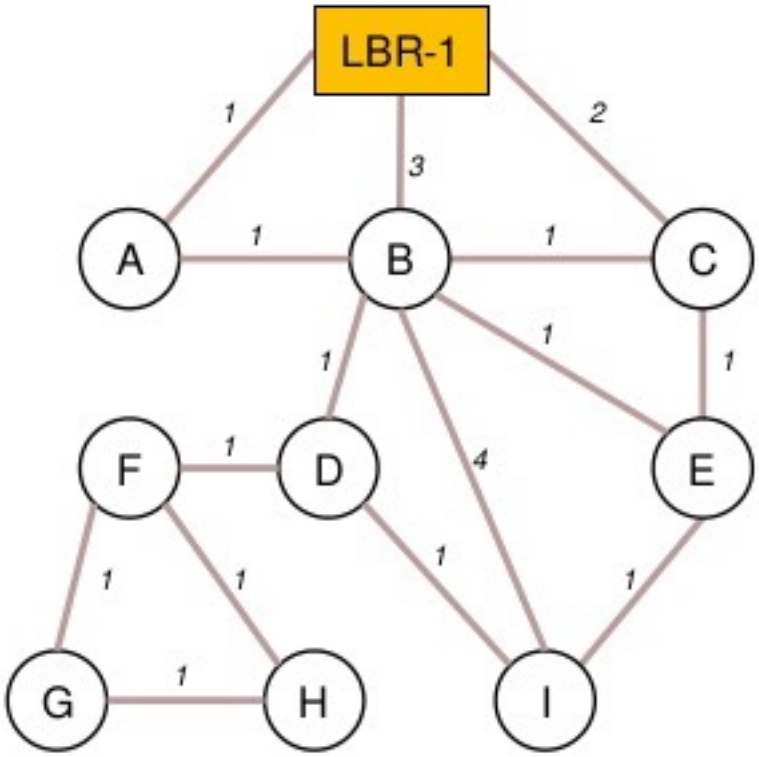
Global versus Local Repair

- Global repair: A new DODAG iteration:
  - This initiates a new DODAG Version.
  - A new Sequence number (DODAGVersionNumber) generated by the root.
  - Nodes in the new DODAG Version can choose a new position whose Rank is not constrained by their Rank within the old DODAG Version.

- Local Repair: find a “quick” local repair (*réparation*) path:
  - Only requiring local changes!
  - May not be optimal according to the OF.
  - Moving UP and Jumping are cool.
  - Moving Down is risky: Count to Infinity Control.

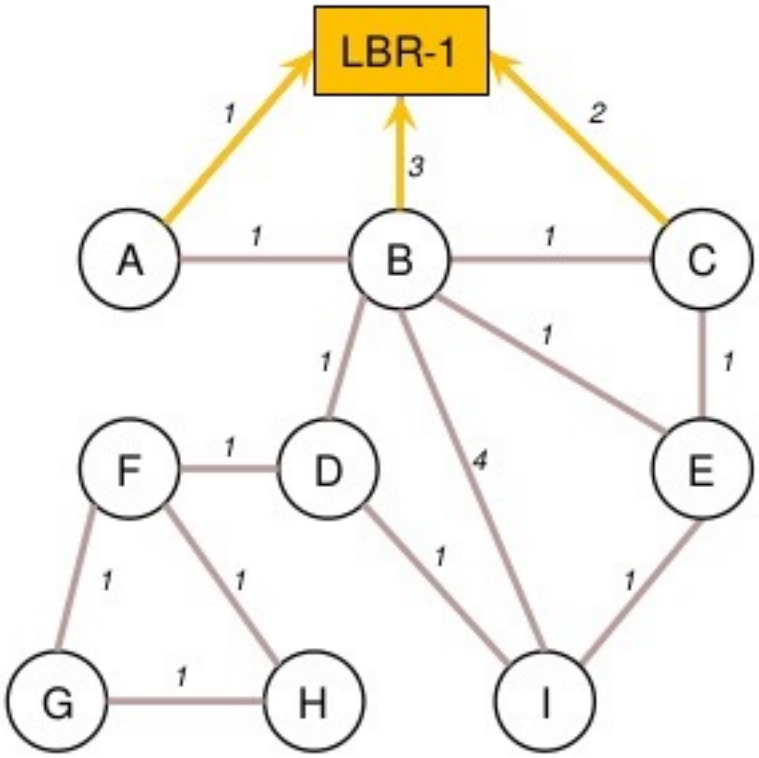
Source from Michael Richardson and Ines Robles “RPL- Routing over Low Power and Lossy Networks”, IETF 94

Local Repair: Example



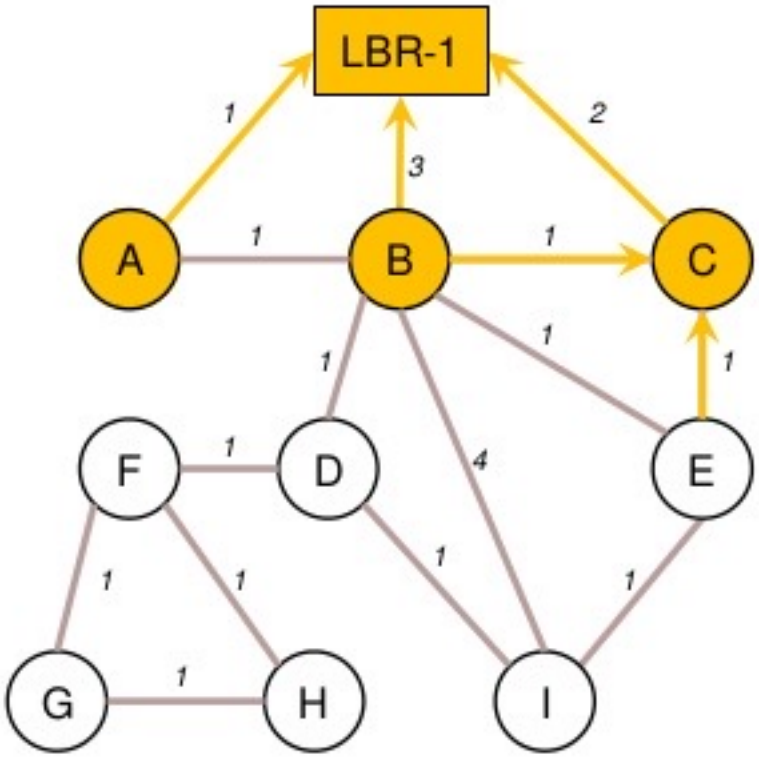
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



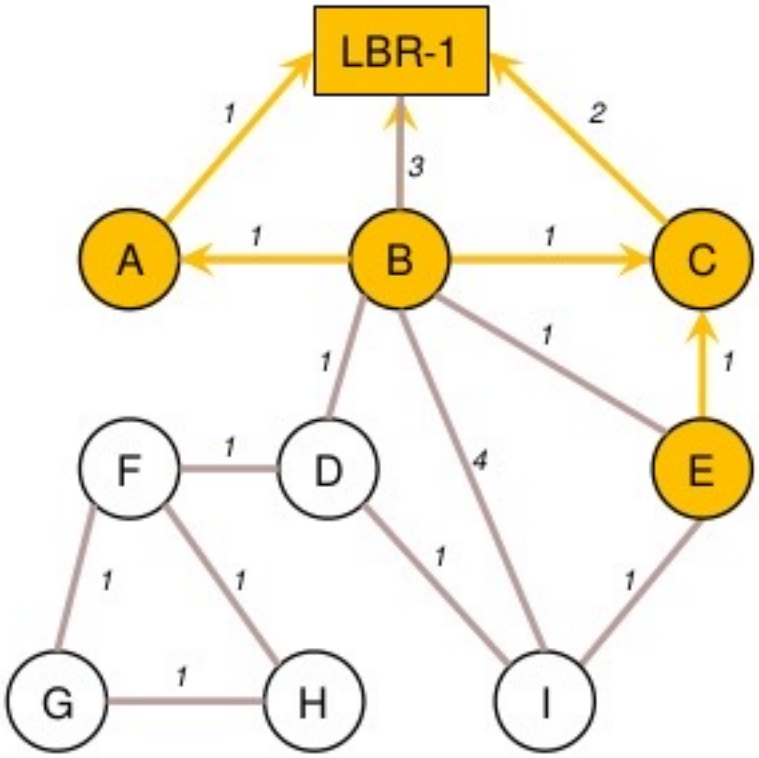
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



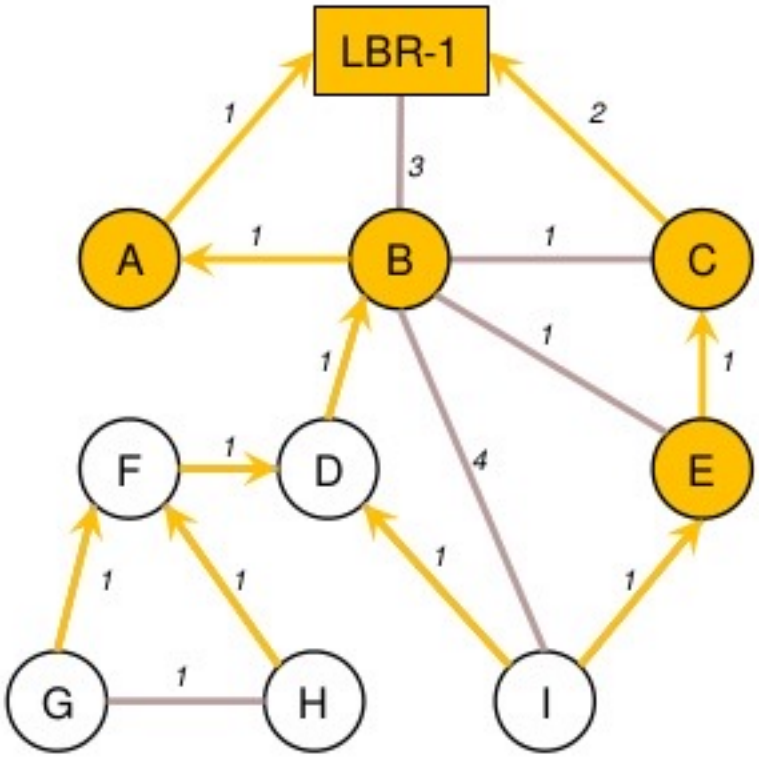
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



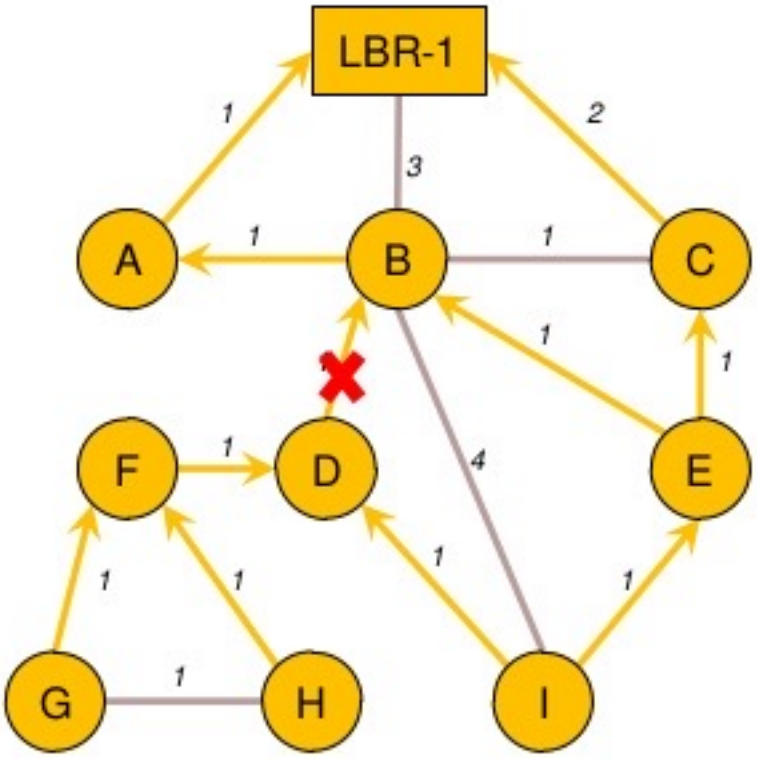
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



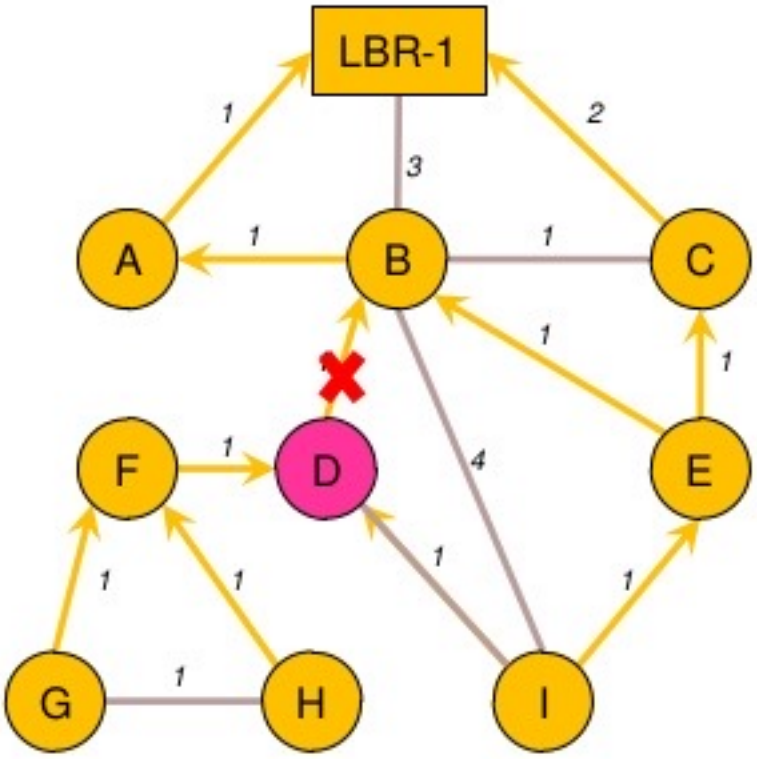
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



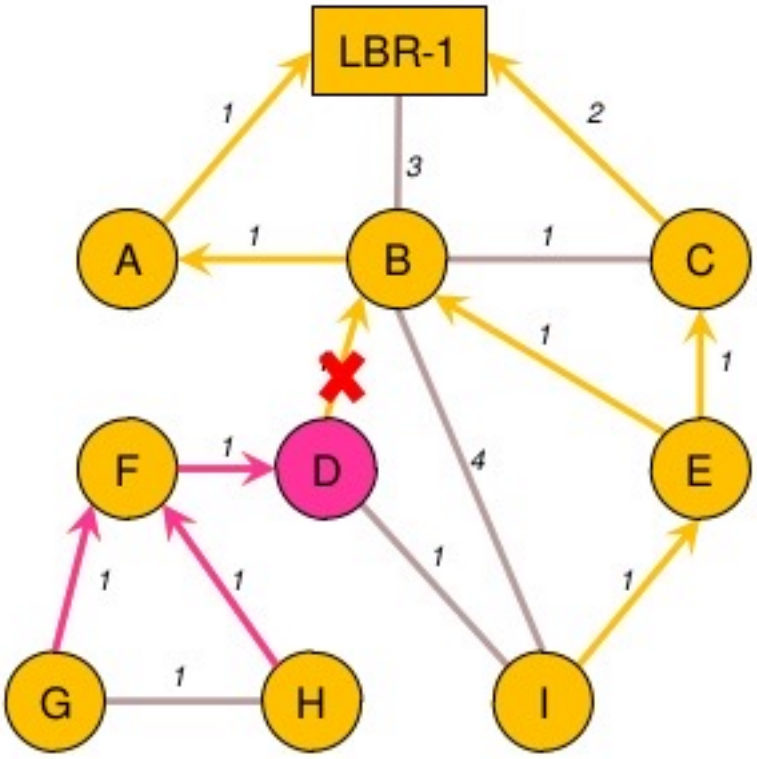
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



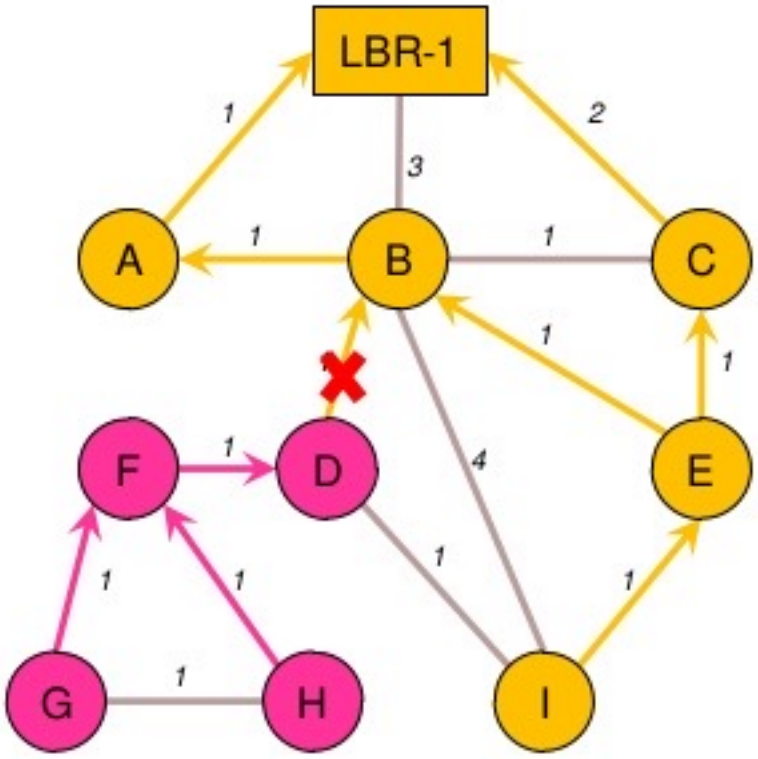
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



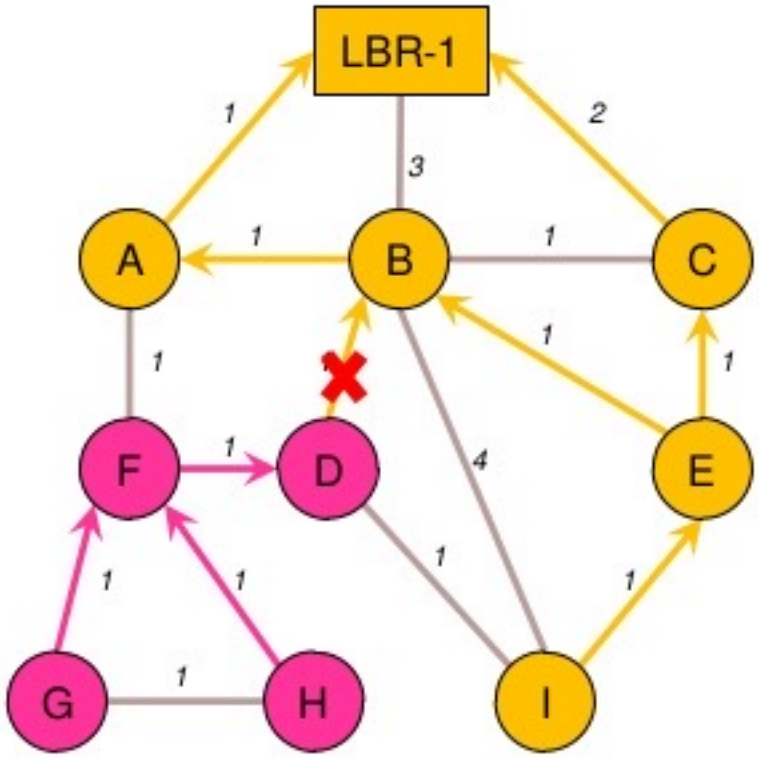
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



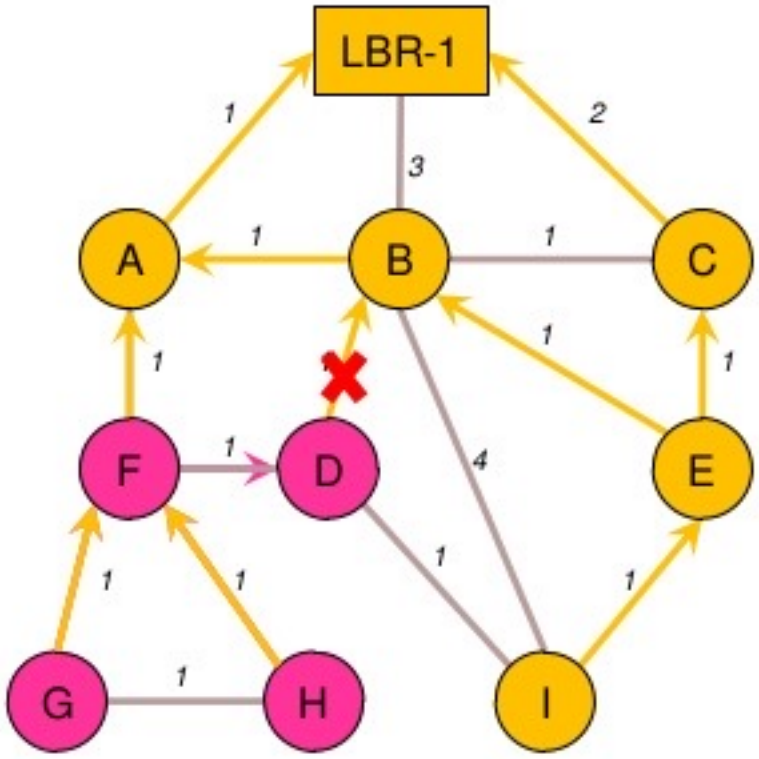
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



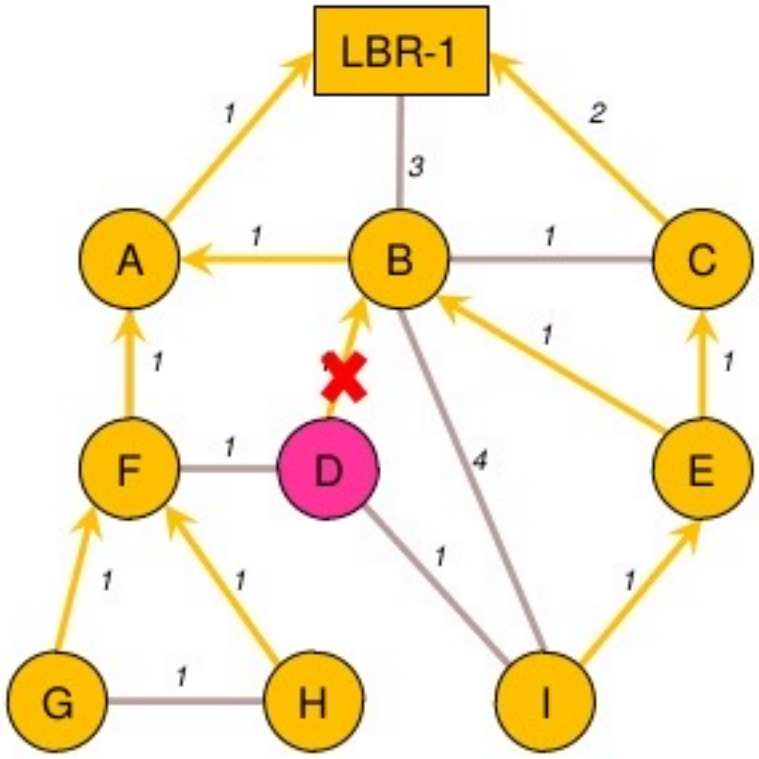
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



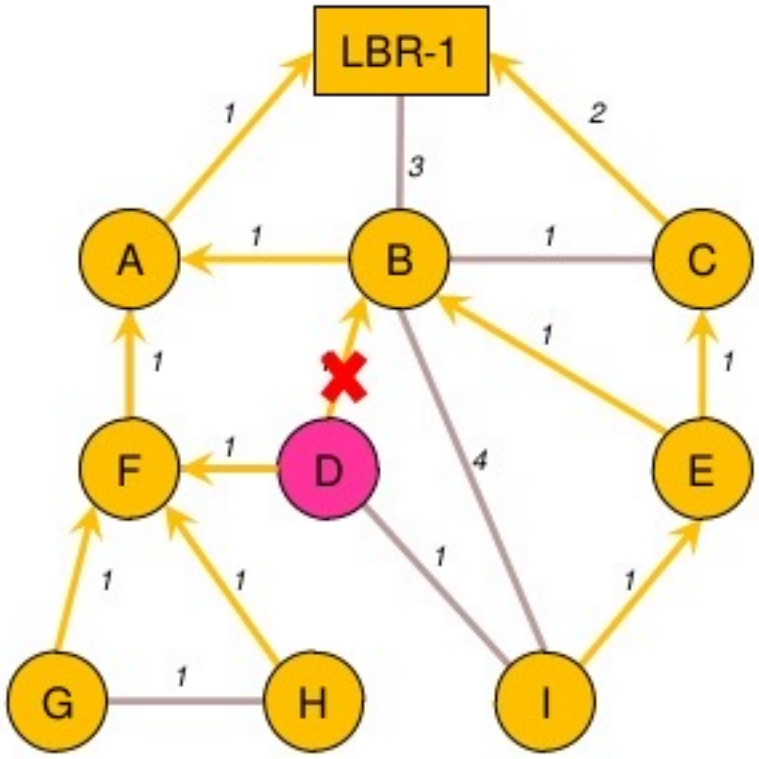
Source: https://www.slideshare.net/vinatech/slide-tt?related=1

Local Repair: Example



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

Local Repair: Example



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

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Routing Layer in LLNs

Georgios Z. **PAPADOPOULOS**
e-mail: georgios.papadopoulos@imt-atlantique.fr
web: www.georgiospapadopoulos.com
twitter: @gzpapadopoulos
youtube: www.youtube.com/c/gzpapadopoulos