Internet of Things

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

Low-Power and Lossy Network

- A class of network in which both the routers and their interconnect are constrained
- LLN routers typically operate with constraints on processing power, memory, and energy (battery power)
- Their interconnects are characterized by high loss rates, low data rates, and instability.
- LLNs are comprised of anything from a few dozen to thousands of routers.
- Supported traffic flows include
    - point-to-point (between devices inside the LLN),
    - point-to-multipoint (from a central control point to a subset of devices inside the LLN), and
    - multipoint-to-point (from devices inside the LLN towards a central control point).

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.

Application areas for LLNs [RFC 7228]

- industrial monitoring,
- building automation (heating, ventilation, and air conditioning (HVAC), lighting, access control, fire),
- connected home, health care,
- environmental monitoring,
- urban sensor networks, energy management, assets tracking, and refrigeration.

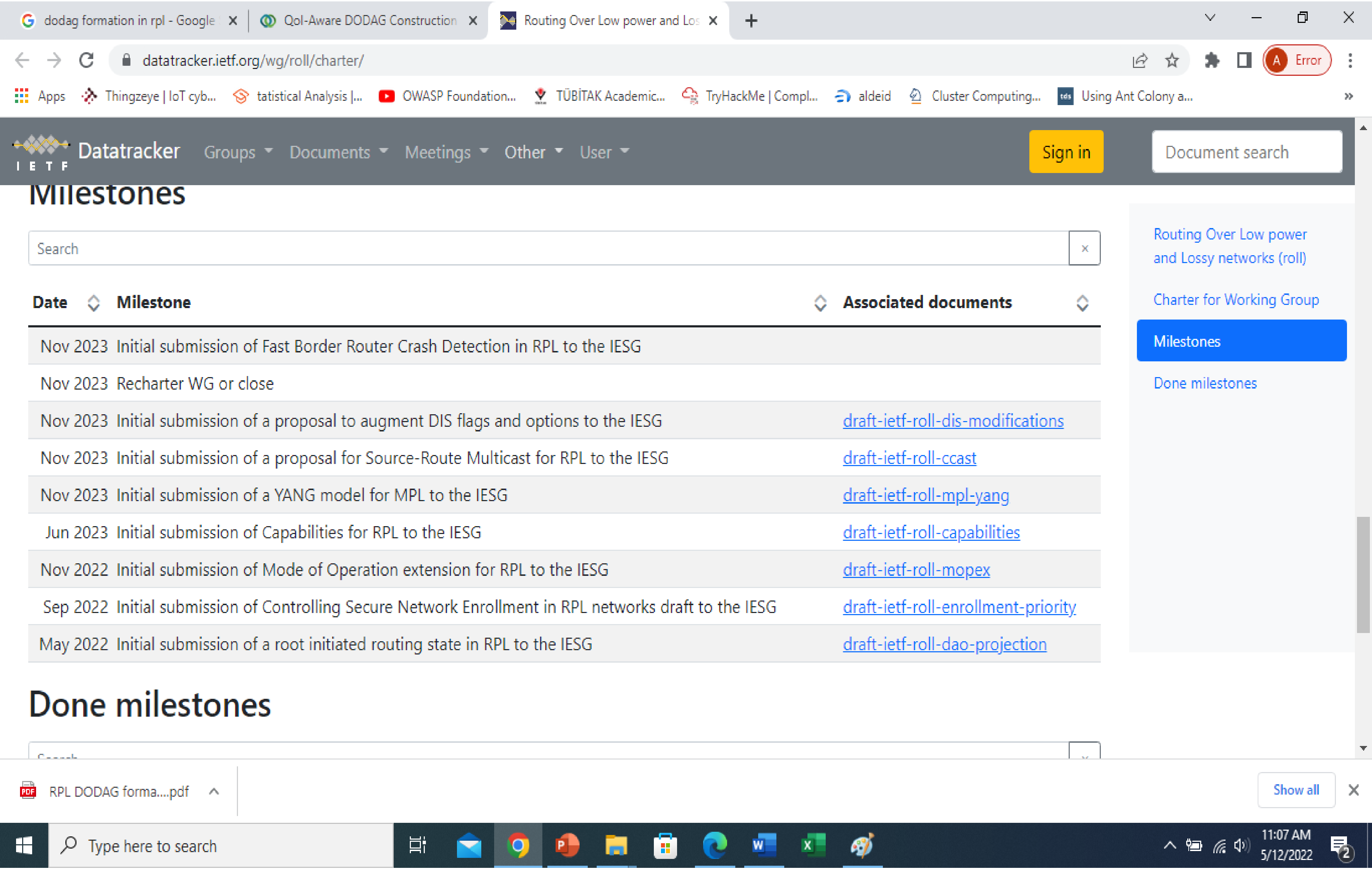
Routing over Low-Power and Lossy Networks (ROLL): Working Group

- IETF formed a working group: Routing over Low-Power and Lossy Networks (**ROLL**)
  - For specifying routing solutions for Low-Power and Lossy Networks (LLN)
  - Lossy links are normally wireless BUT
  - Can be wired such as Power Line Communication (PLC)

Objectives of ROLL

- to produce a set of **routing requirements**
- To determine whether or not **existing** IETF routing protocols would **satisfy the requirements**
- To define new **routing metrics** for routing in LLNs
- To establish a **routing security** framework

ROLL



Milestones

Date	Milestone	Associated documents
Nov 2023	Initial submission of Fast Border Router Crash Detection in RPL to the IESG	
Nov 2023	Recharter WG or close	
Nov 2023	Initial submission of a proposal to augment DIS flags and options to the IESG	draft-ietf-roll-dis-modifications
Nov 2023	Initial submission of a proposal for Source-Route Multicast for RPL to the IESG	draft-ietf-roll-ccast
Nov 2023	Initial submission of a YANG model for MPL to the IESG	draft-ietf-roll-mpl-yang
Jun 2023	Initial submission of Capabilities for RPL to the IESG	draft-ietf-roll-capabilities
Nov 2022	Initial submission of Mode of Operation extension for RPL to the IESG	draft-ietf-roll-mopex
Sep 2022	Initial submission of Controlling Secure Network Enrollment in RPL networks draft to the IESG	draft-ietf-roll-enrollment-priority
May 2022	Initial submission of a root initiated routing state in RPL to the IESG	draft-ietf-roll-dao-projection

Done milestones

Routing Requirements

- When designing a new protocol
  - Understanding of the requirements is necessary
- IETF working groups usually produce working requirement documents for follow up tracking of information
- A mobile Delay Tolerant Network used to study wildlife does not have much in common with a dense " always on "network used for industrial automation.
- **Application areas considered**
  - Urban networks (including Smart Grids)
  - building automation,
  - industrial automation, and
  - home automation

Routing Requirements

A list of routing requirements for networks made of LLNs

Unicast/anycast/multicast:

- support of unicast, anycast, and multicast traffic as mandatory

Adaptive routing:

- new paths are dynamically and automatically recomputed due to change of conditions in the network (e.g., link/node failure, mobility, etc.).
- the routing protocol must be able to compute routes optimized for different metrics (e.g., minimize latency, maximize reliability, etc.).
- the routing protocol must be able to find a path that satisfies specific constraints such as providing a path with a latency lower than a specified value.

Routing Requirements

- **Constraint-based routing**
  - The routing protocol has to support constraint-based routing

  Constraints:
  - **various node characteristics** such as energy, CPU, and memory
  - link attributes such as link latency.
- **Scalability:** LLNs are composed of larger number of nodes
  - For such networks, routing protocol may follow specific rules like **network partitioning**
- **Configuration and Management**
  - Be able to autoconfigureable with minimal or 0-configuration

Routing Requirements

Traffic characteristics: focused on data collection (telemetry)

- **multipoint-to-point (MP2P) traffic:** most of the traffic is from leaf nodes such as sensors to a data collection sink.
- **point-to-multipoint traffic:** when the sink sends a request to all nodes in the network, AKCs are necessary if reliability is needed,
- **point-to-point (P2P):** communication between devices in the network.

Routing Requirements

Node attribute: if there are sleeping nodes in the network,

- the routing protocol must discover the capability of a node to act as a proxy.
- A packet could be delivered to a proxy that could relay the packet to the destination once awakened.

Performance: Never design a protocol with hard numbers or bounds but with simple indications providing some order of magnitude.

- **For example,** routing protocol must find routes and report success or failure within several minutes.

Security

- Smart cities require minimal security
- **But in** Smart Grid, building automation, industrial automation, security is critical [Authentication and encryption are listed as absolute must]

IETF existing protocols for LLNs?

- **LLN Routers** interconnected by variety of links i.e., IEEE 802.15.4 or low WiFi
  - have constraints on processing, memory, and energy.
  - Can’t use OSPF, RIP, DSR, etc
- LLN links have high loss rate, low data rates, and instability due to dynamically formed topology

- **ROLL Group was of the view that** none of the existing protocols would satisfy the fairly unique set of routing requirements for LLNs

Thus ROLL was re-chartered **to design a new** routing protocol for Low-power and Lossy Networks called **RPL** with the objective to meet the requirements

Routing Metrics

A **metric** is a scalar used to determine the best path according to some **objective function**. For example, propagation delay.

A **constraint** is used to include or eliminate links or nodes that do not meet specific criteria

The objective function may combine link/node metrics and constraints such as “ find the path with the minimum delay that does not traverse any nonencrypted link. ”

Link quality level (LQL)

- LQL (an integer) characterizes the link quality (poor, fair, good).

OF may be like this: to find the path that provides the minimum number of links with poor quality (LQL is used as a metric).

Routing Metrics

- An OF defines how metrics are used to select routes and establish a node’s rank

Metrics include

- Hop count
- Latency
- Node energy
- Throughput
- Reliability: most common metric is Expected transmission (ETX) count: how **reliable** the link is

ETX – Expected Transmission Count

- ETX characterizes average number of packet transmissions required to successfully transmit a packet

- Shows the quality of a radio connection
  - How many times should I probably send a packet to be sure that it is received
- Value between 1 and infinity
  - Expected value based on past experience

RPL

Prime objective

- to design a highly modular protocol
  - where the core of the routing protocol would address the intersection of the application specific routing requirements,
  - and additional modules would be added as needed to address specific requirements
- RPL is designed for LLNs where constrained devices are interconnected by (wireless and wired) lossy links.
- In LLNs, the data traffic is usually limited
  - So the control traffic should be reduced whenever possible to save bandwidth and energy.

RPL

- **Due to its standardization**
  - It has contributed to the advancement of communication in the world of tiny, embedded networking devices by providing a baseline architecture for IoT
- **A distance vector routing protocol**
  - Routing is based on Destination Oriented Distance Acyclic Graph or DODAGs
  - where paths are constructed from each node in the network to the DODAG root (typically a sink or an LBR).
- **A source routing 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 Characteristics

- Ability to optimize and save energy

- Ability to support traffic patterns other than unicast communication

- Ability to run routing protocol over link layer with restricted frame sizes

- It builds a DODAG using an objective function that takes into account several metrics and/or constraints

- Several different OFs might be used in same time, each building its own DODAG

Host, router, node [as per RFC]

- **host** refers to an LLN device that can generate but does not forward RPL traffic;

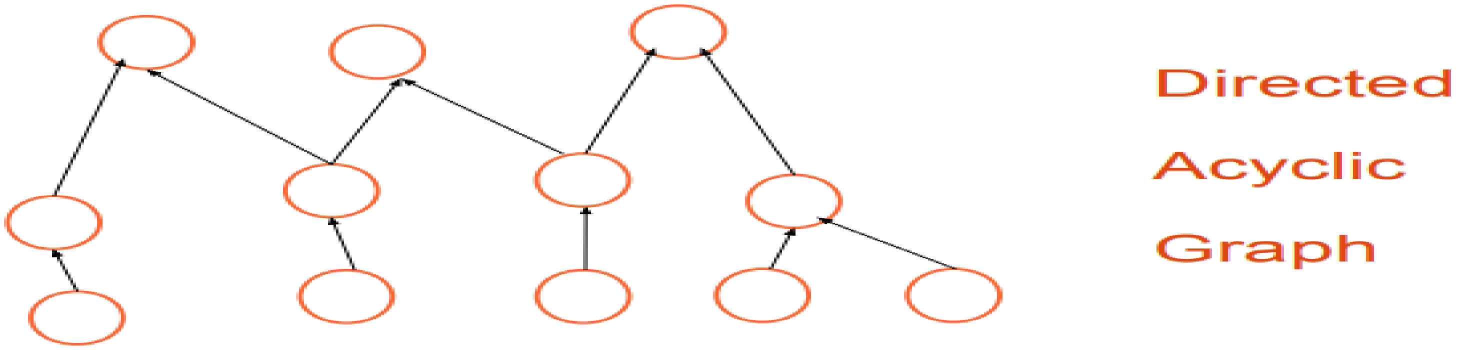
- **router** refers to an LLN device that can forward as well as generate RPL traffic; and

- **node** refers to any RPL device, either a host or a router.

RPL Related Terminology

RPL DAG

- **Directed Acyclic Graph:** A graph containing no cycles (like in spanning trees)
- **Root:** Destination of the node in DAG having no outgoing edge
- **Up:** any edge directed toward the root
- **Down:** any edge which is directed away from root

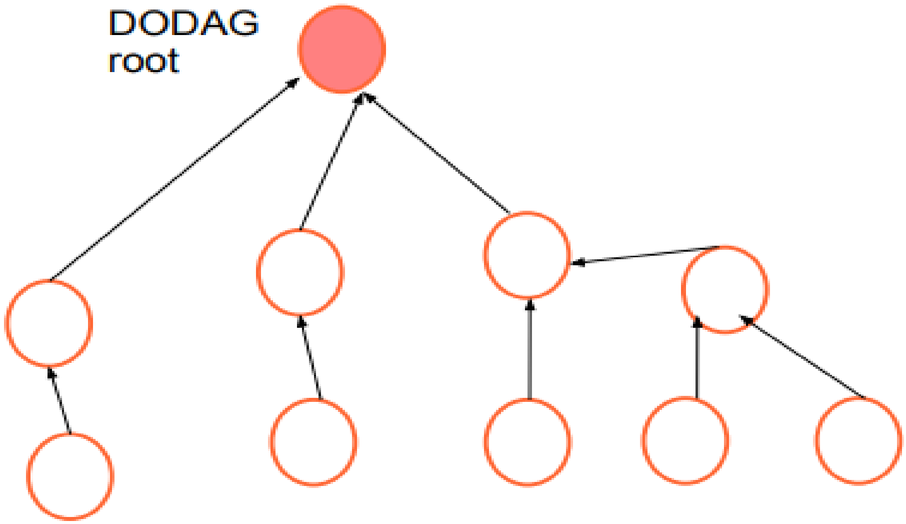


(DAG)

RPL DODAG

Destination Oriented DAG (DODAG)

- Special type of DAG where each node wants to reach a single destination at a single DAG root (DODAG root) with no outgoing edges

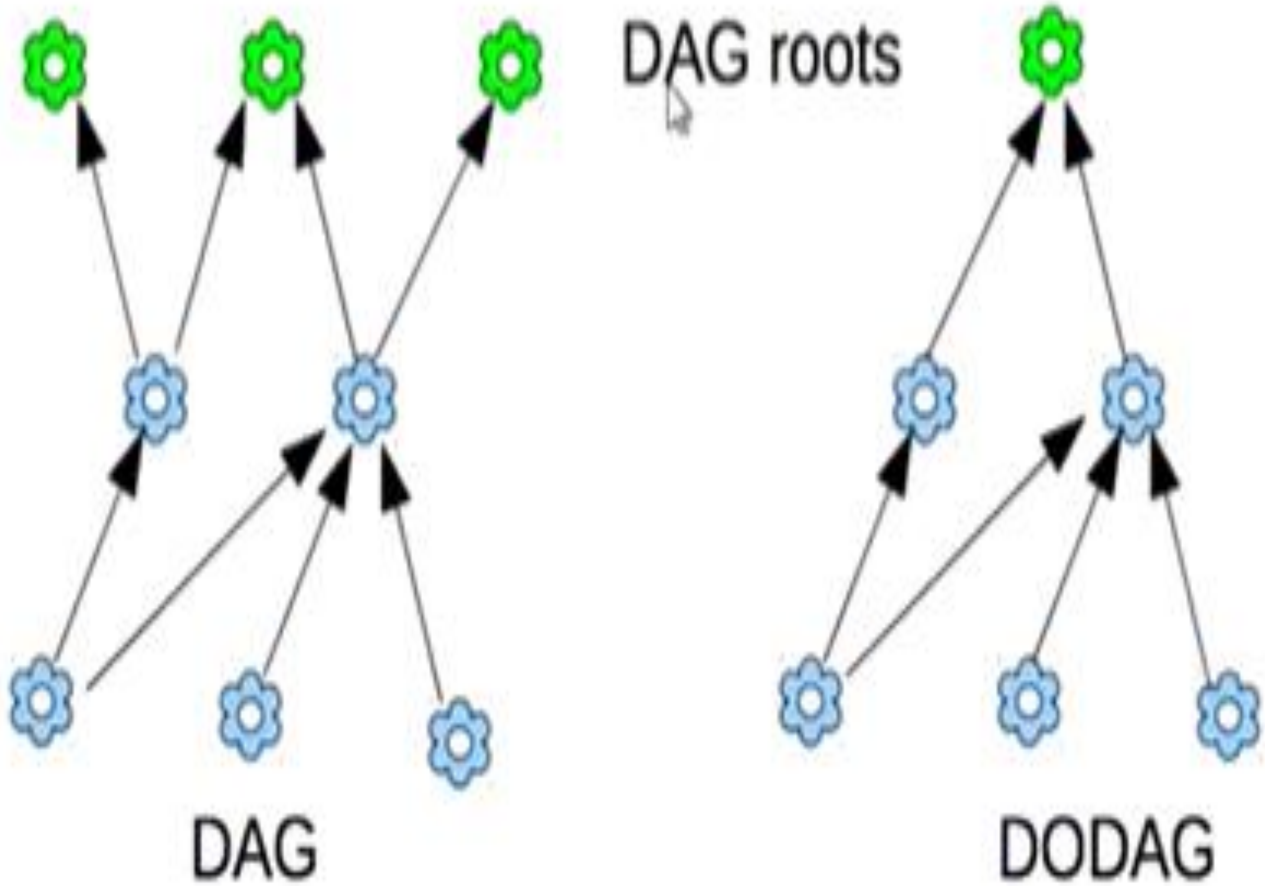


Destination

Oriented

DAGs

(DODAG)



DAG

DODAG

RPL Topology

- do not typically have predefined topologies
- RPL routes are optimized for traffic to or from one or more roots that act as sinks for the topology.
- RPL organizes a topology as a Directed Acyclic Graph (DAG) that is partitioned into one or more Destination Oriented DAGs (DODAGs), one DODAG per sink.
- If the DAG has multiple roots, then it is expected that the roots are joined by a common backbone, such as a transit link.

RPL Objective Function

Objective Function:

- supports to know whether we are near to / away from the root

- Decided by the programmer or designer

- Something to minimize
  - For example: reliability, energy, latency
- Once decided, assign a number

RPL Identifiers

RPL uses four values to identify and maintain a topology:

- RPLInstanceID
- DODAGID
- DODAGVersionNumber
- Rank

RPLInstanceID

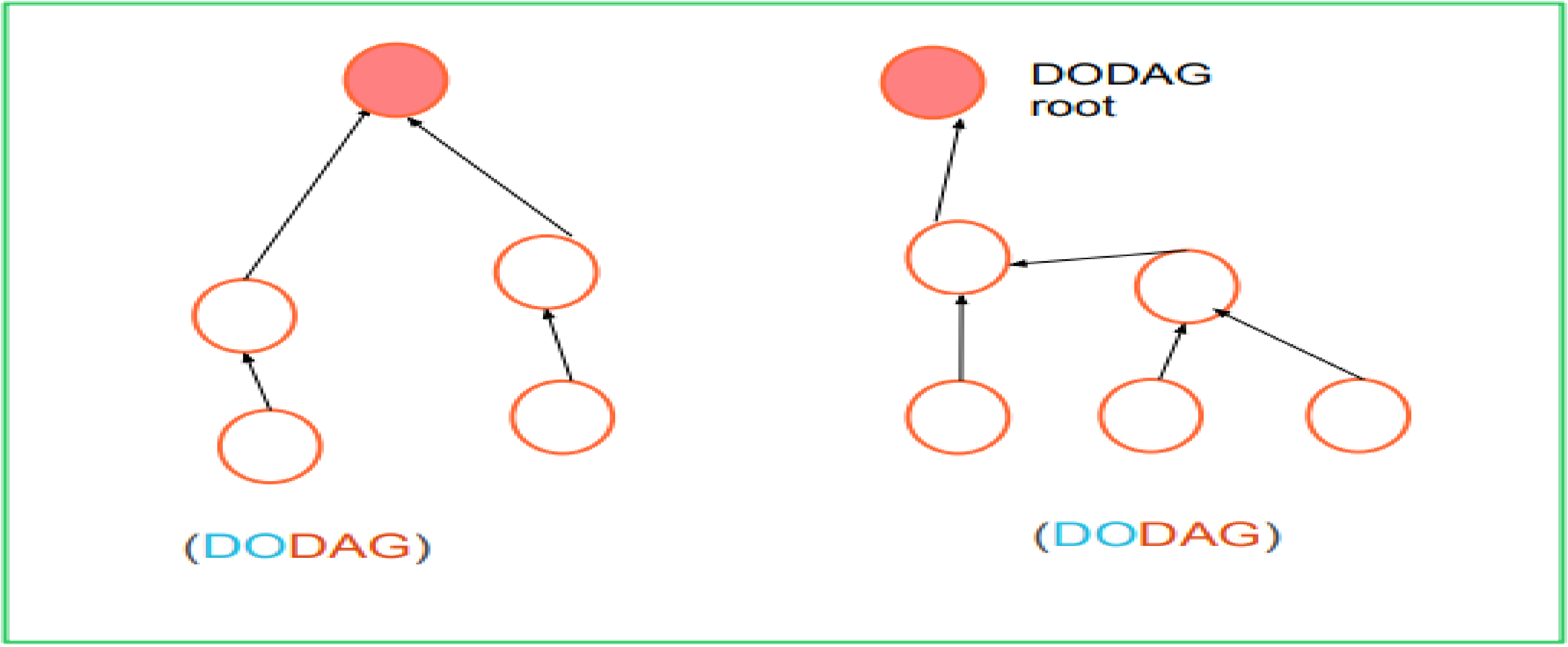
- A RPLInstanceID identifies a set of one or more Destination Oriented DAGs (DODAGs).

- A network may have multiple RPLInstanceIDs,
  - each of which defines an independent set of DODAGs,
  - which may be optimized for different Objective Functions (OFs) and/or applications.
- The set of oen or more DODAGs identified by a RPLInstanceID is called a RPL Instance.
- At most, a RPL node can belong to one DODAG in a RPL Instance. Each RPL Instance operates independently of other RPL Instances.

- All DODAGs in the same RPL Instance use the same OF.

RPL Instance

- **RPLInstanceID:** 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 .



RPL Instance

RPL **DODAG ID**

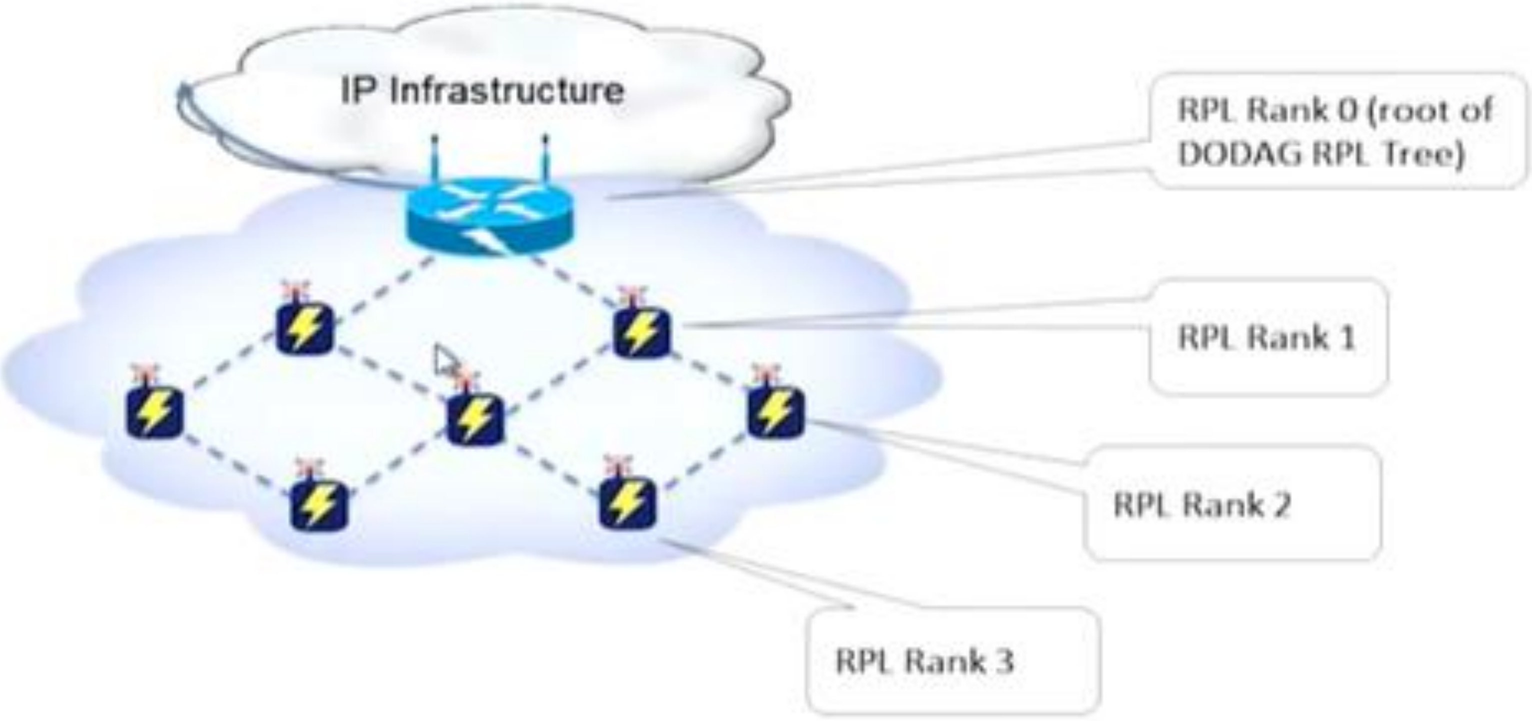
- **DODAG ID**: 128 bits IPv6
  - Until root does not change, root ID remains the same
  - The scope of a DODAGID is a RPL Instance.
  - The combination of RPLInstanceID and DODAGID uniquely identifies a single DODAG in the network.
  - A RPL Instance may have multiple DODAGs, each of which has an unique DODAGID.

RPL **DODAG Version**

- **DODAG version**: each new shape of DODAG refers to new version
  - The scope of a DODAGVersionNumber is a DODAG.
  - A DODAG is sometimes reconstructed from the DODAG root, by incrementing the DODAGVersionNumber.
  - The combination of RPLInstanceID, DODAGID, and DODAGVersionNumber uniquely identifies a DODAG Version.

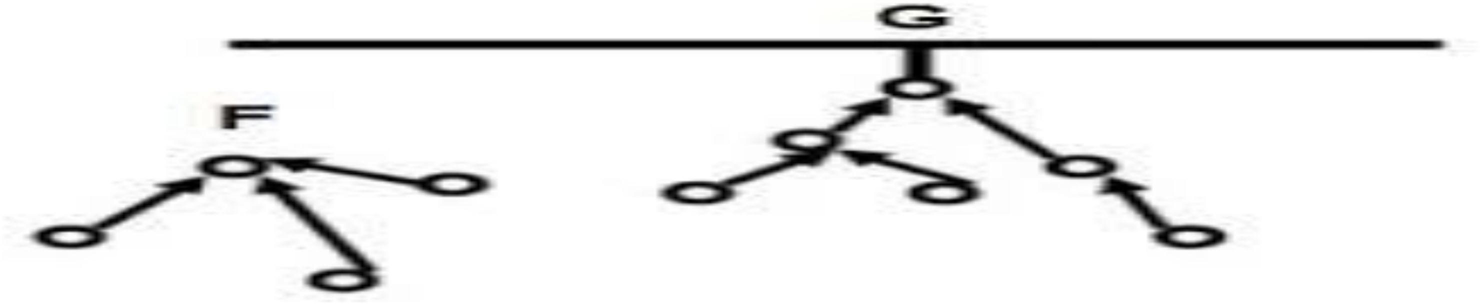
RPL Rank

- **Rank:** node’s individual position relative to other nodes w.r.t. DODAG root.



RPL: more terms

- **Goal:** where a DODAG wants to reach (can be a wired network)
- **Grounded:** when a DODAG reaches its goal
- **Floating:** when a DODAG is not connected or yet to reach the goal



RPL: more terms

- **Parent**: where arrow is pointing towards (multiple child)
- **Child:** where the arrow comes from (multiple parents)
- **Sub-DODAG**: any sub-tree of given DODAG
- **Storing:** Storing nodes keep the whole routing table they know how to go from one node to another (**downward**)
- **Non-storing:** simple, do not restore entire routing table, only know about their parents

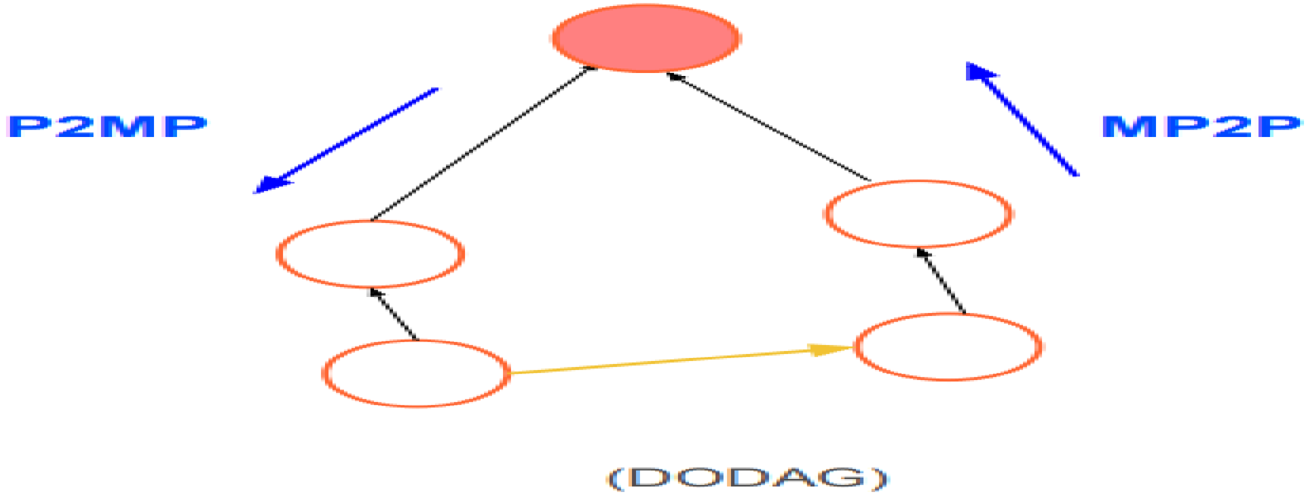
The whole DODAG (except Root which is always storing) has to maintain the uniformity: to be either storing or non-storing

RPL Traffic flows

Traffic moves either toward the DODAG root or down towards the leaf

DODAG properties

- Many to one communication: upwards
- One to many communication: downwards
- One to one communication: upwards-downwards

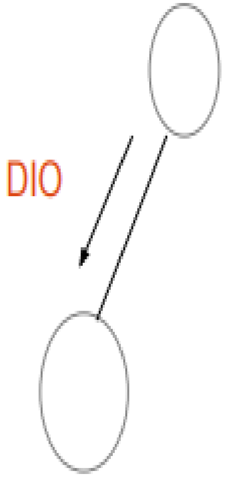


RPL Control Messages: ICMPv6 messages

The RPL specification defines four types of control messages for topology maintenance and information exchange

1. DODAG Information Object (DIO):

- ✓ main source of routing control information
- ✓ Multicast downwards to let other nodes to know about it (if they are interested to join)
- ✓ It may store information like
  - ✓the current Rank of a node,
  - ✓the current RPL Instance,
  - ✓the IPv6 address of the root,
  - ✓Grounded or not
  - ✓Storing or not

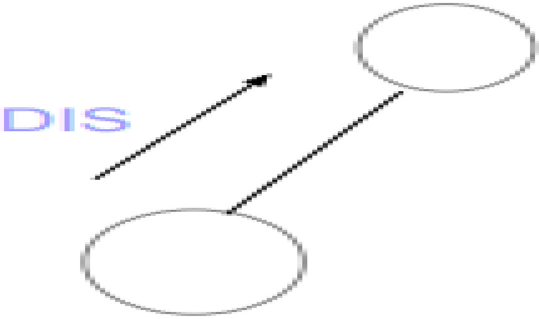


RPL Control Messages

2. Destination Information Solicitation (DIS):

- ✓ When no announcement is heard and a node wants to join a DODAG
  - ✓ It sends a control message to know if any DODAG exists (i.e., is there any DODAG)
  - ✓ Link local multicast request (neighbor discovery)

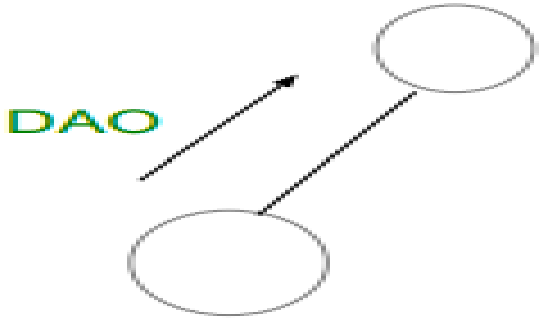
Solicit a DODAG Information Object (DIO) from a RPL node



RPL Control Messages

3. Destination Advertisement Object(DAO):

- ✓ A request send by a child to parent or root 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.
- ✓ This messages requests to allow the child to join to a DODAG



RPL Control Messages

4. DAO-ACK:

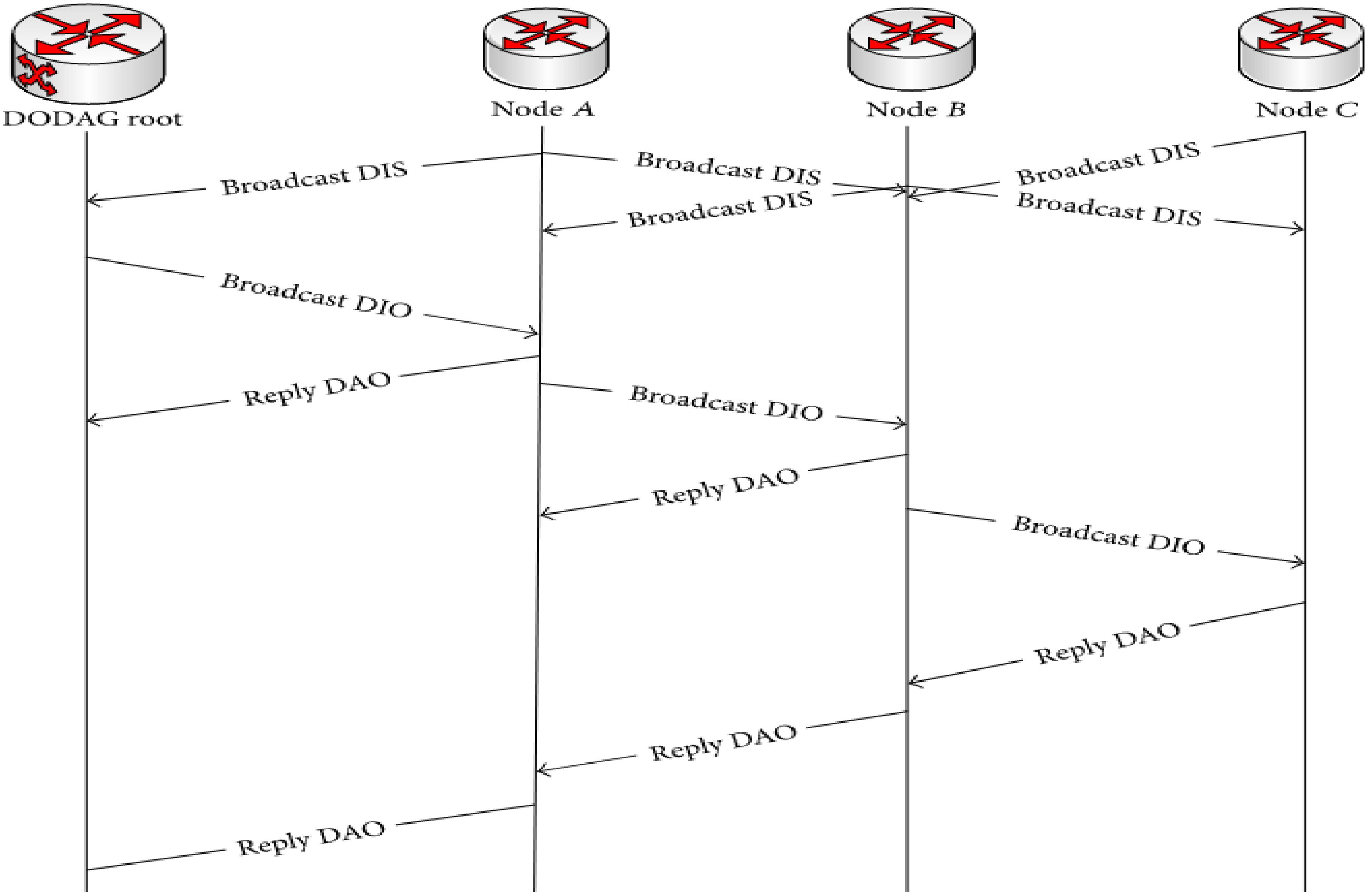
- ✓ Reponses sent by a root or parent to the child (either Yes or No) to acknowledged DAO message

Consistency check:

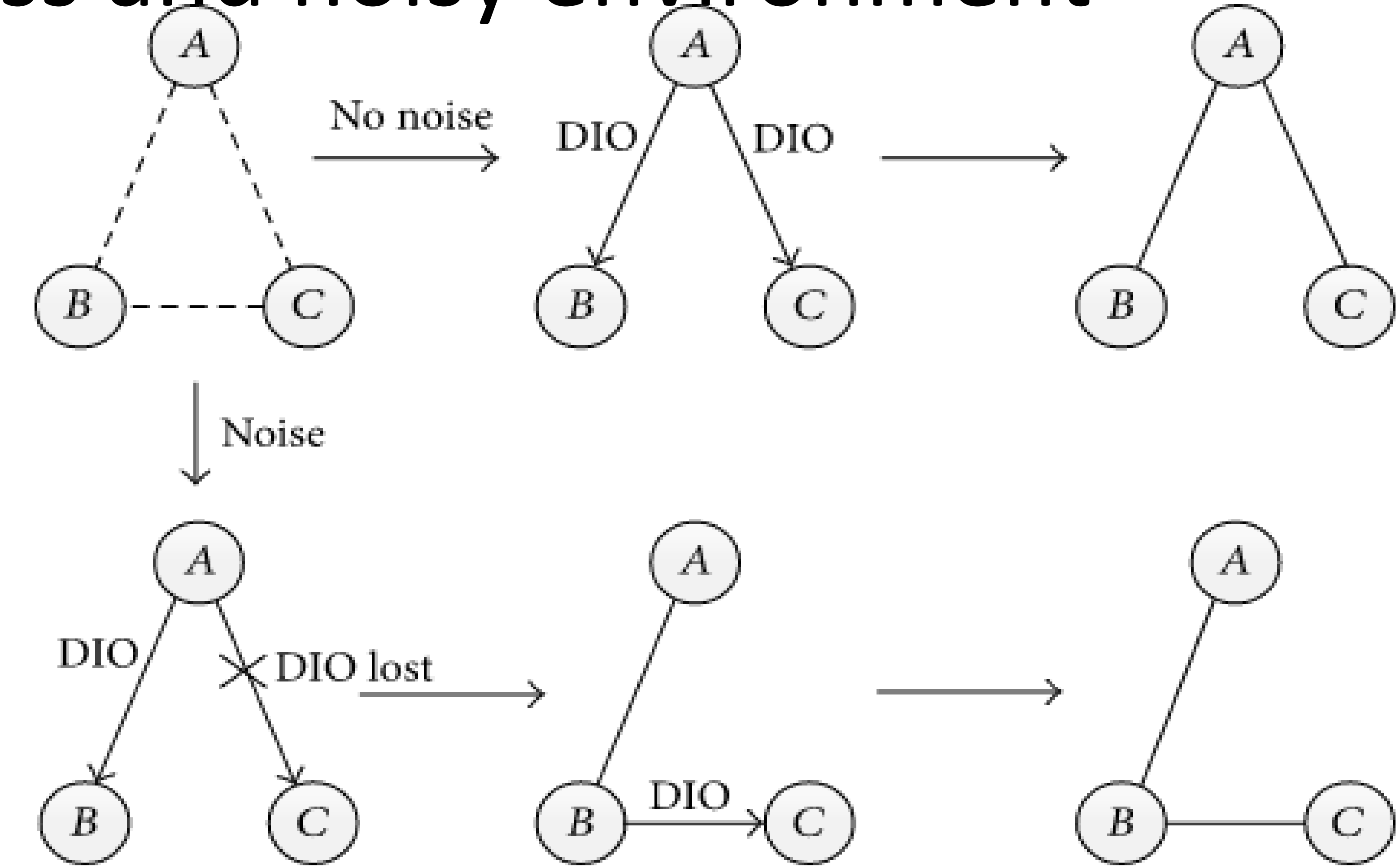
- ✓ Deals with security

DODAG Formation

Simplified DODAG formation Process



DODAG discovery processes in noise-less and noisy environment



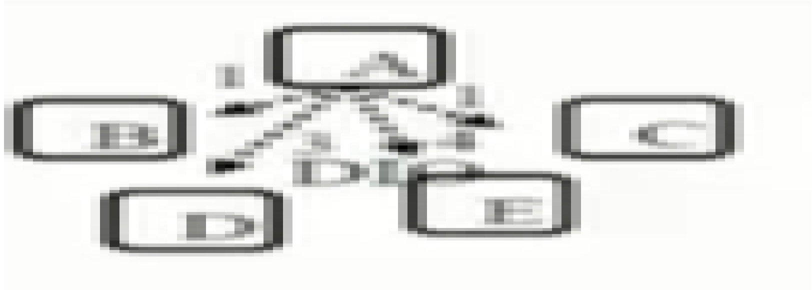
DODAG formation

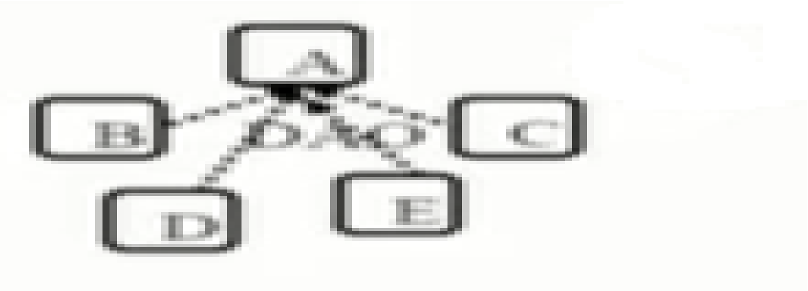
- ✓ Root in a DODAG is a special node.
  - ✓ All nodes don’t have capability to be root
  - ✓ A root is programmed that way

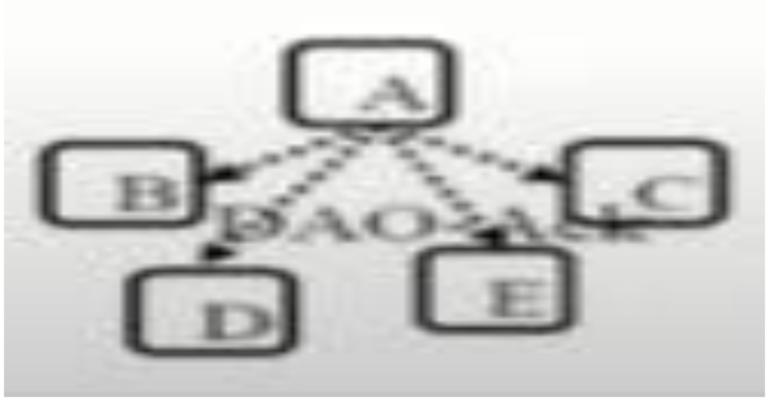
Consider 5 nodes A, B, C, D and E. In order to form a DODAG, the following steps will involve

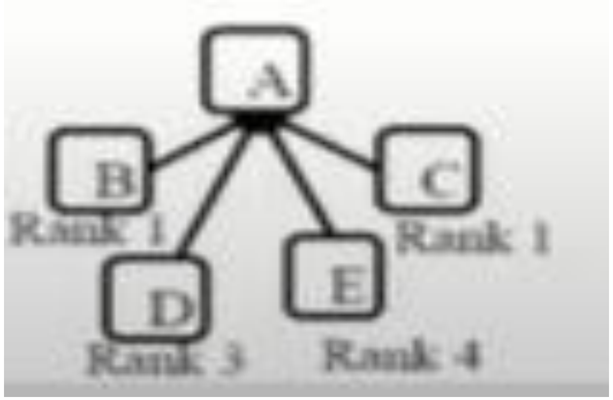
- ✓ A multicast DIOs
- ✓ Remaining nodes (B,C,D and E) upon receiving DIOs will try to join regardless of their distance.
  - ✓ Upon receiving DIOs they also come to know their distances from A as 1, 1, 3, 4 respectively
- ✓ B, C and D send DAOs to A
- ✓ A accepts them by replying with DAO-ACK

DODAG formation









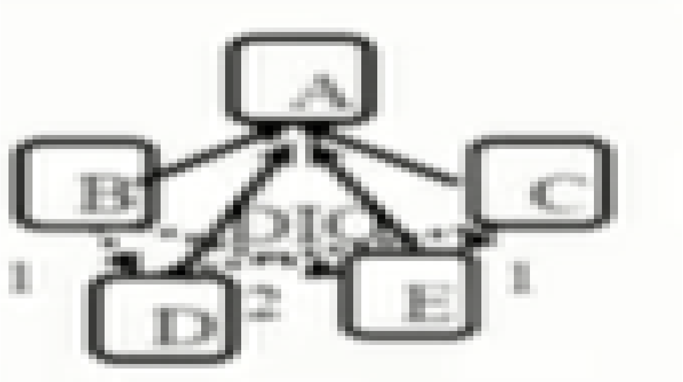
DODAG formation

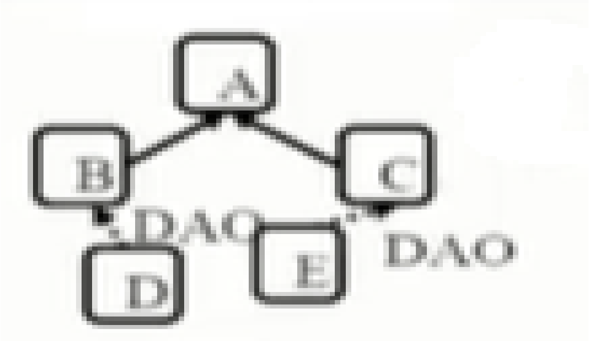
- ✓ Now next round begins with the nodes nearest to A (i.e., B and C)
- ✓ B & C start sending DIOs

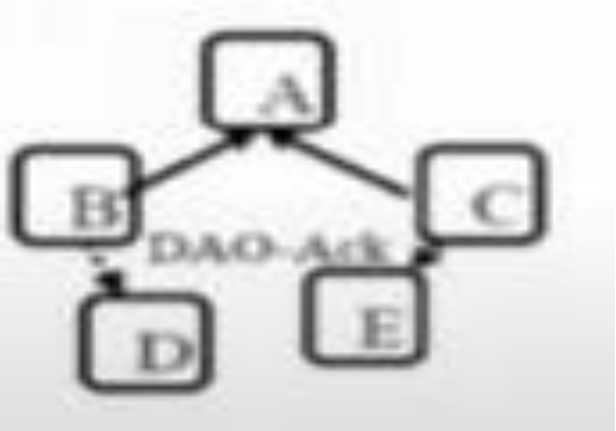
Upon receiving DIOs

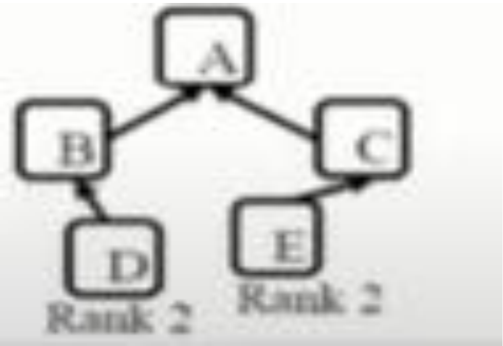
- ✓ D identifies that its distance from B & C are a 1 & 2 respectively
  - ✓So D being closer to B, sends DAO to B
- ✓ E identifies that its distance from B & C are a 2 & 1 respectively
  - ✓E being closer to C, sends DAO to C
- ✓ B sends a DAO-ACK to D while C to E in order to finalize the DODAG formation

DODAG formation

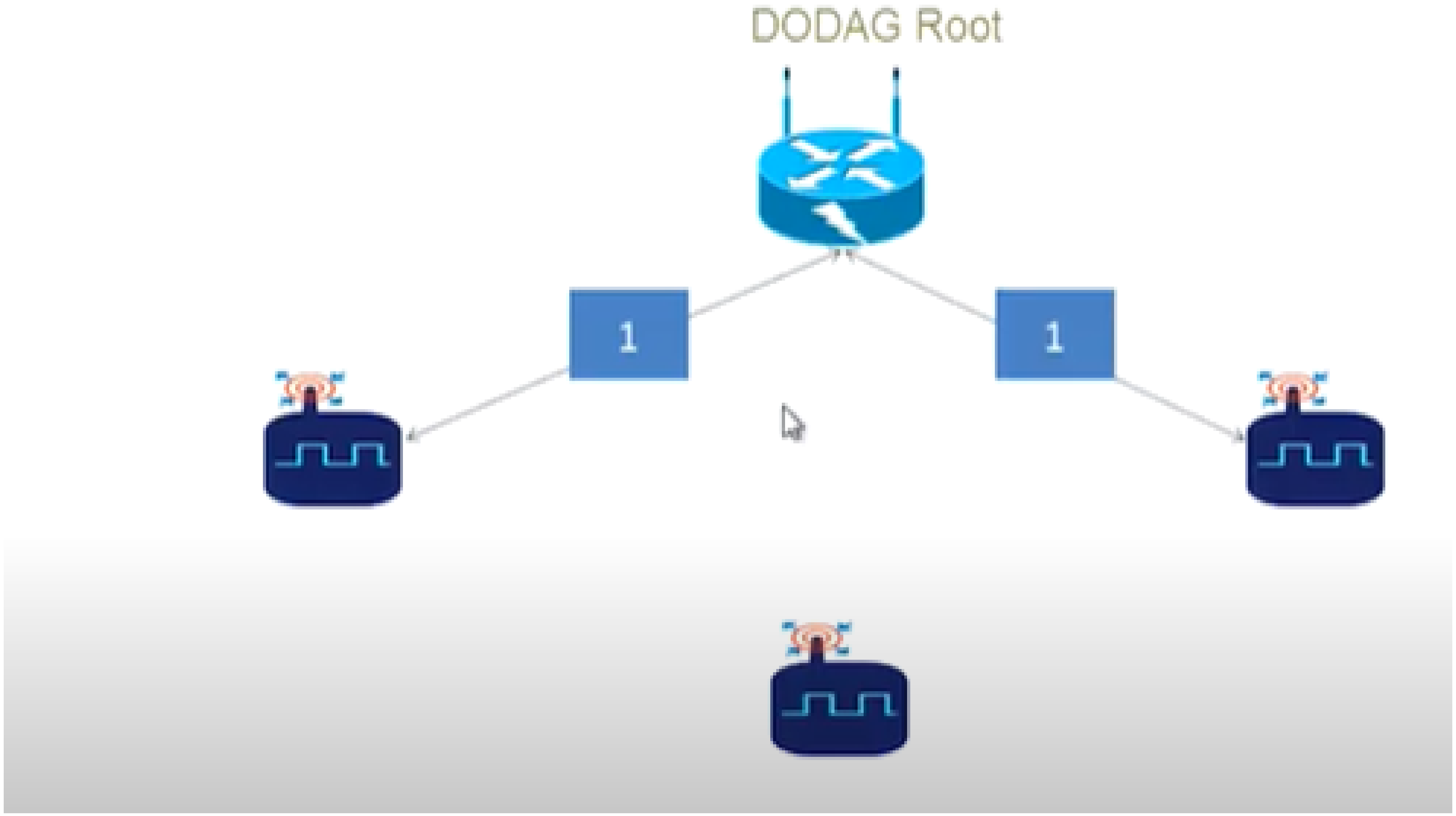




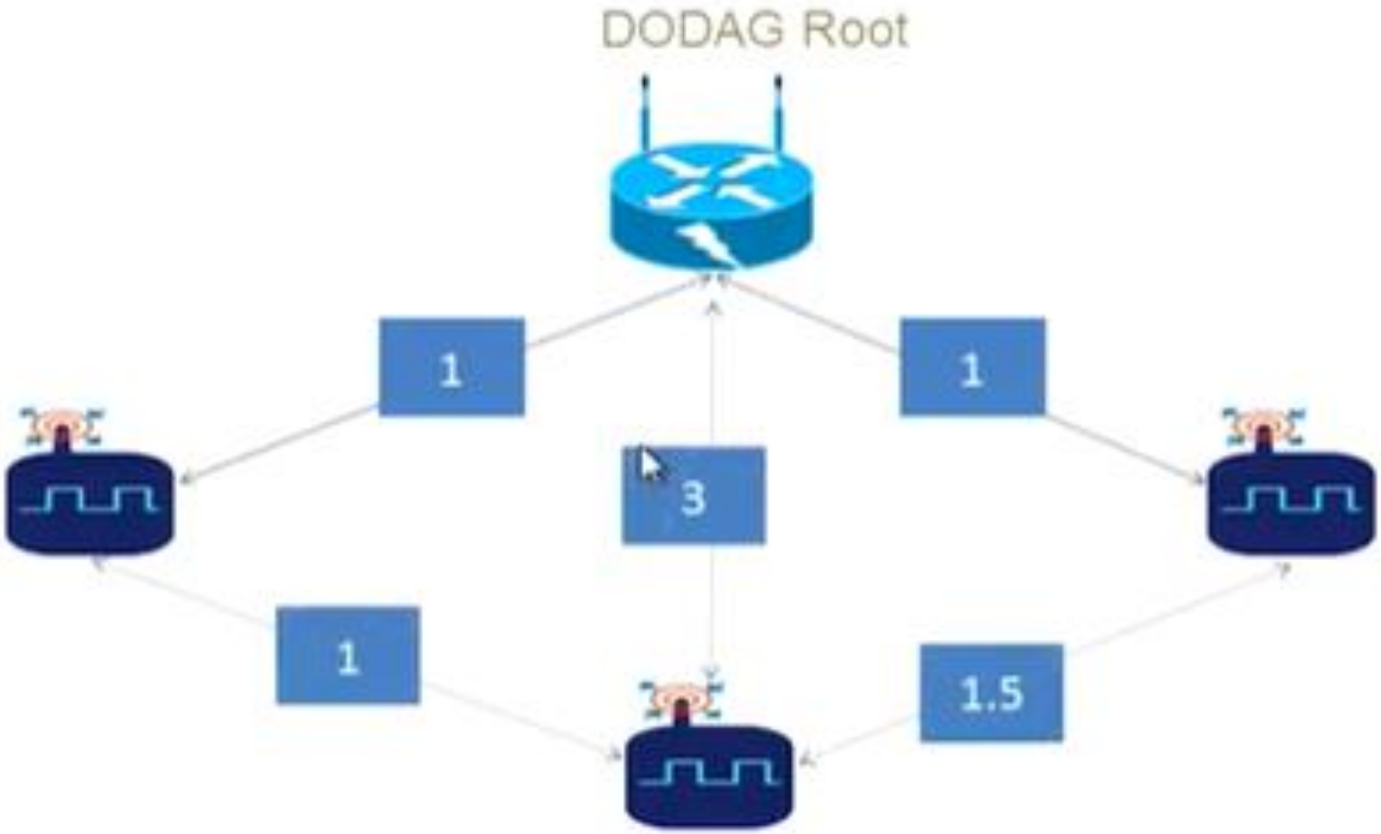




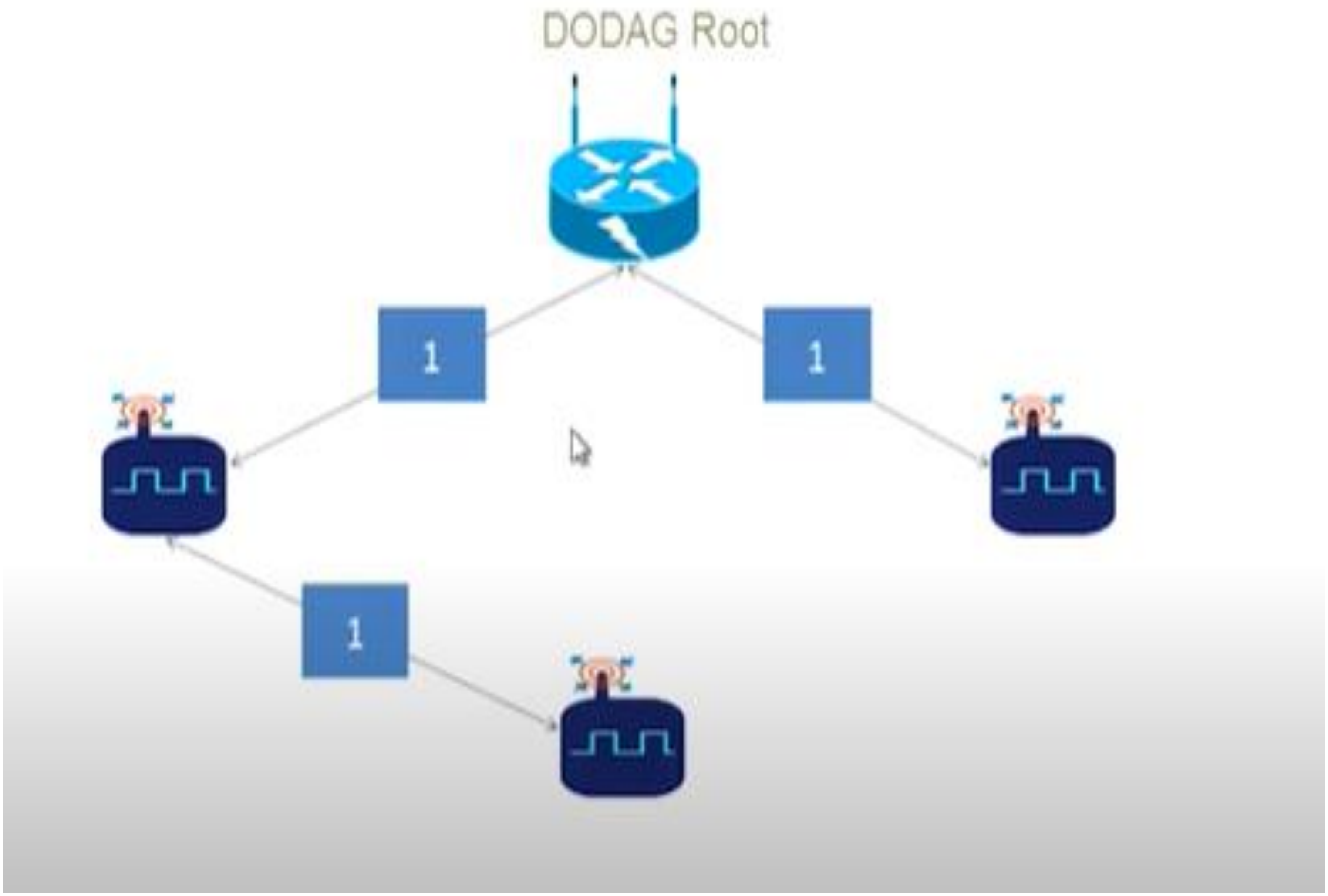
Routing with ETX as OF



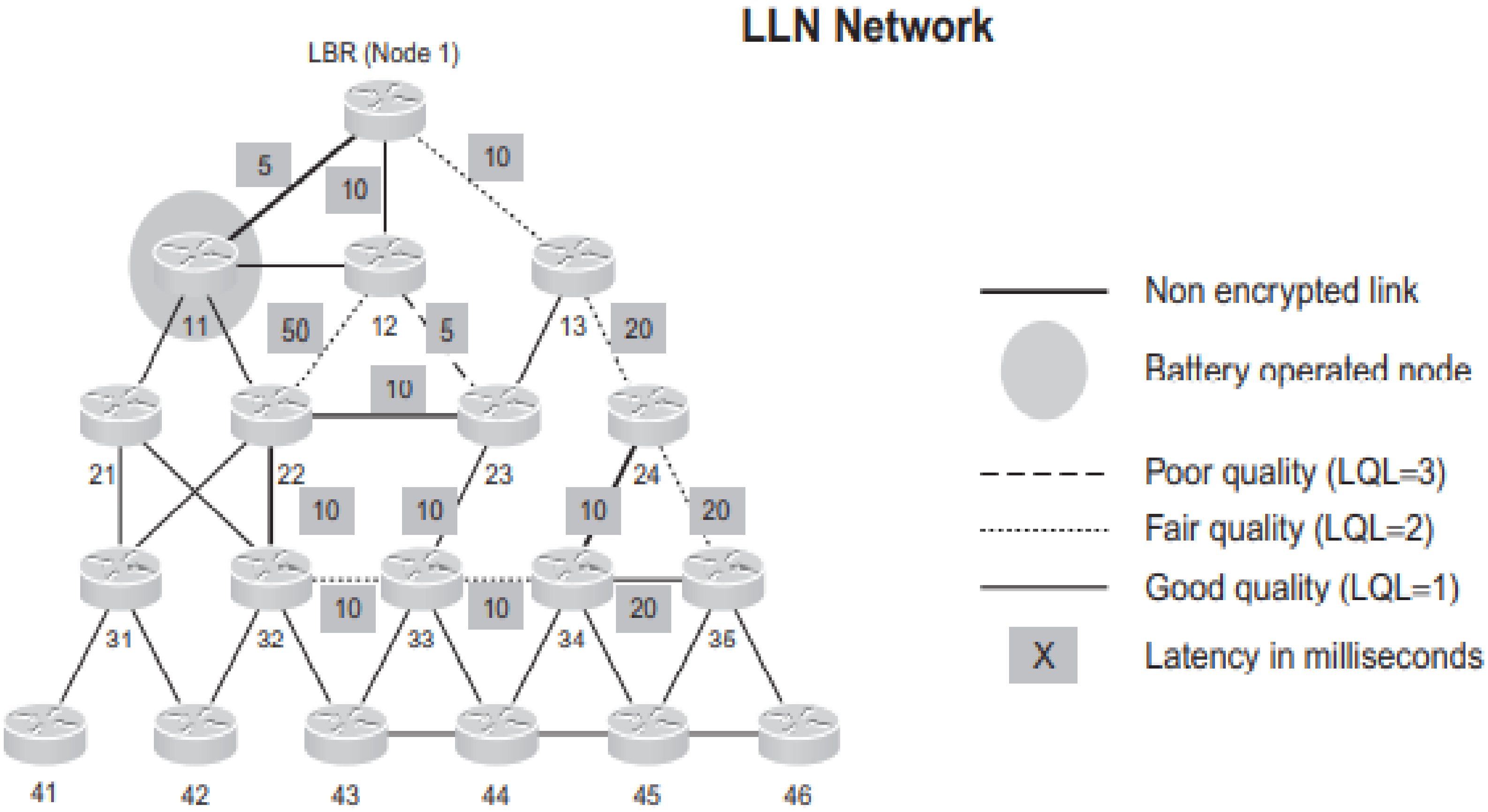
Routing with ETX as OF



Routing with ETX as OF



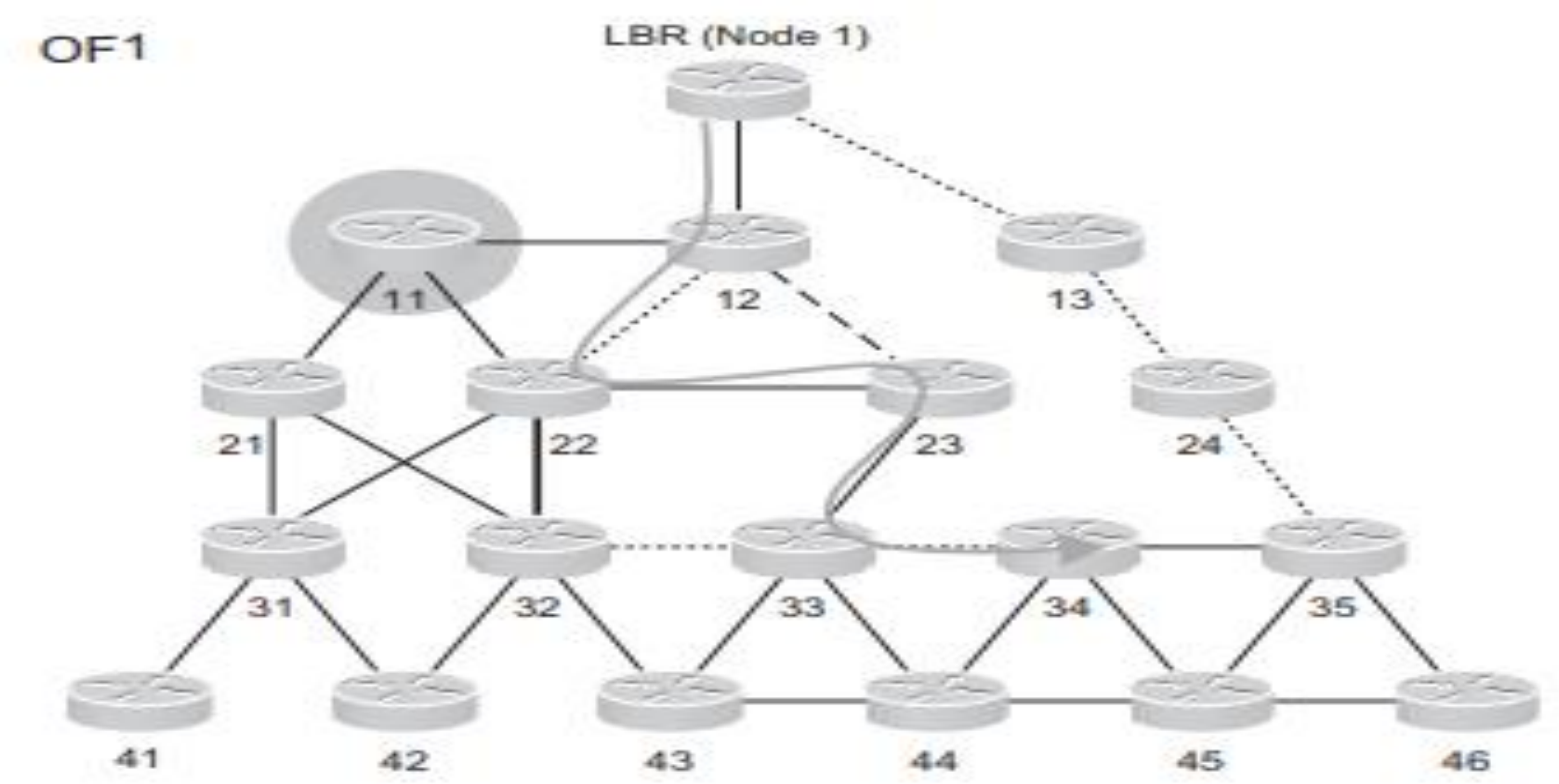
Objective Function



best computed path from LBR to node 34?

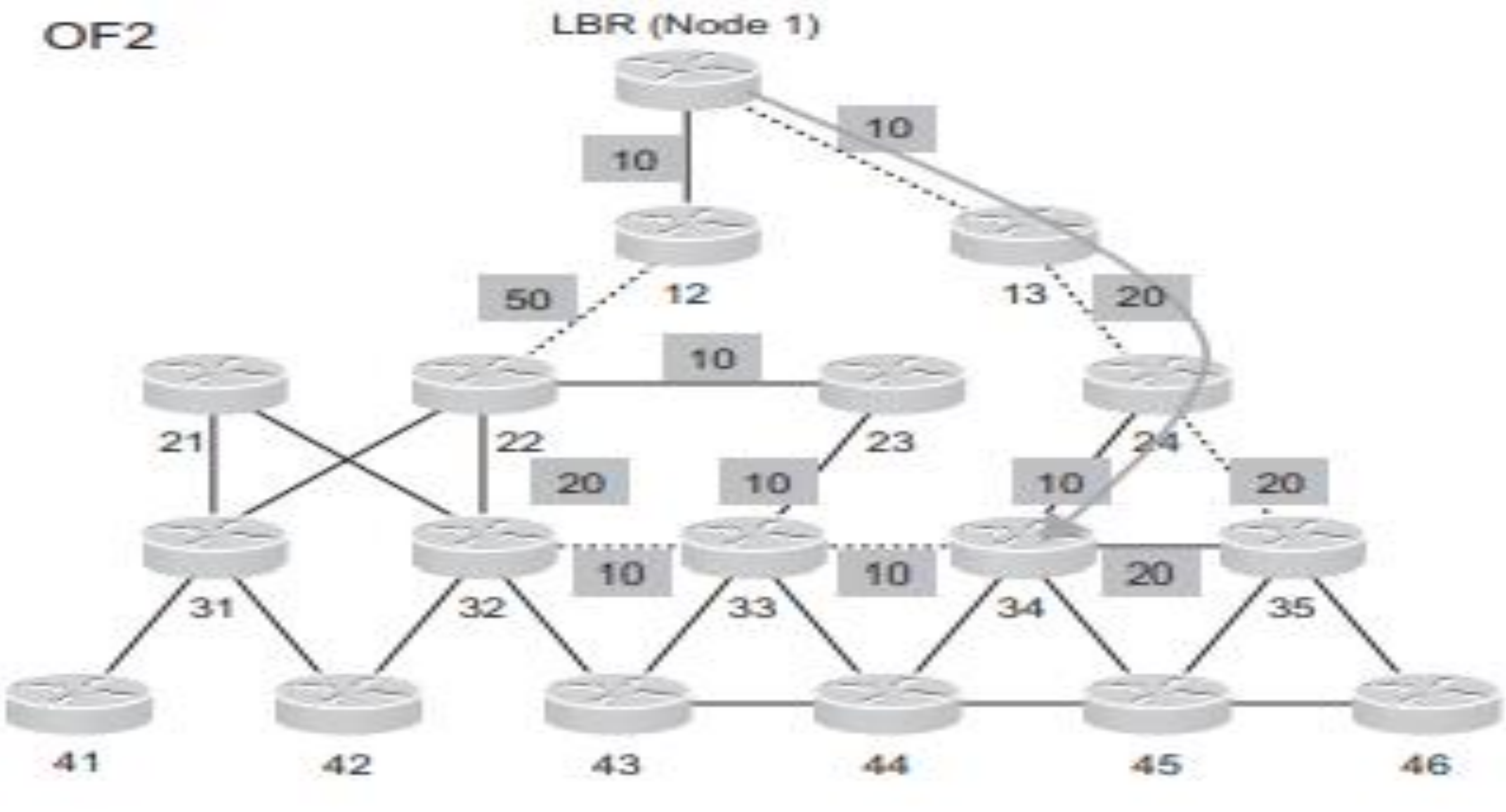
Objective Function

- OF1:“ Use the **LQL** as a global recorded metric and favor paths with the **minimum number of low and fair quality links**, use the link color as a **link constraint to avoid non-encrypted links**. ”



Objective Function

- **OF2:**“Find the best path in terms of latency while avoiding poor quality links and battery-operated nodes. ”



Extra Supporting Slides