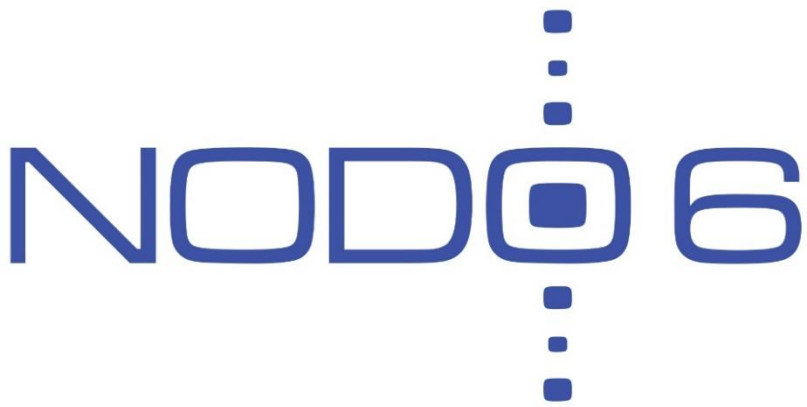


Introduction to ROLL

March 2016 – ICTP

Alvaro Vives (alvaro.vives@nodo6.com)
NODO6 (www.nodo6.com)

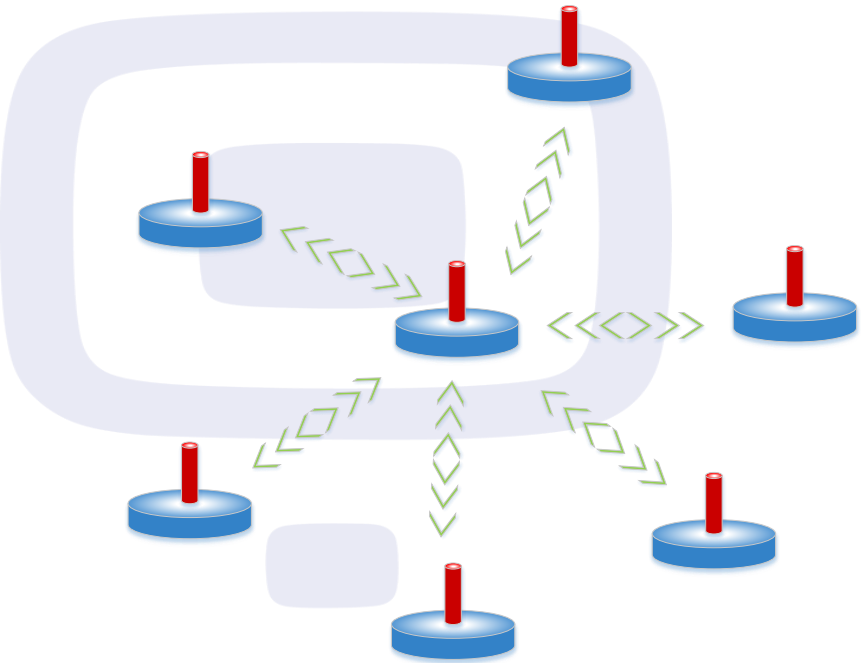
Contents

Objectives

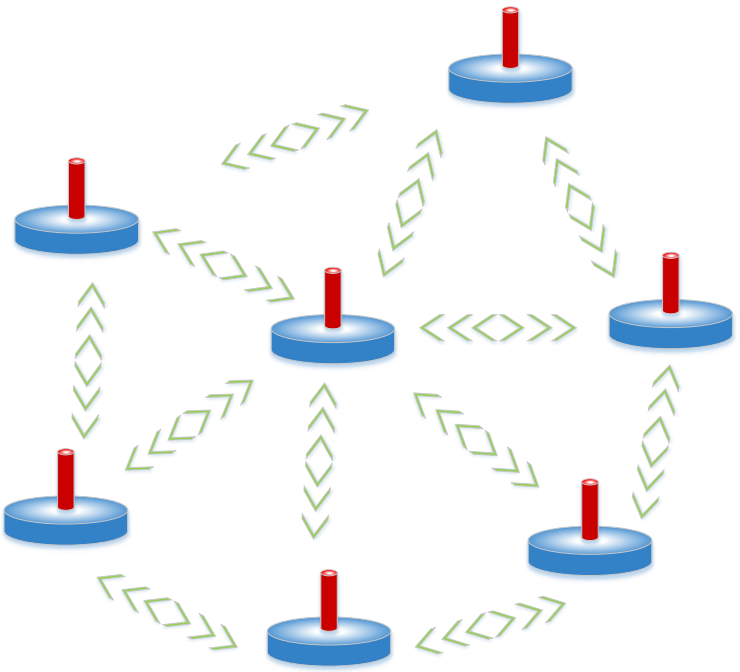
- Overview of frame delivery in Link-layer mesh
- Know what is ROLL
- Have an overview of ROLL standards
- Have an overview of RPL

Topologies (I)

- IEEE 802.15.4 networks support star and mesh topologies



Star Topology



Mesh Topology

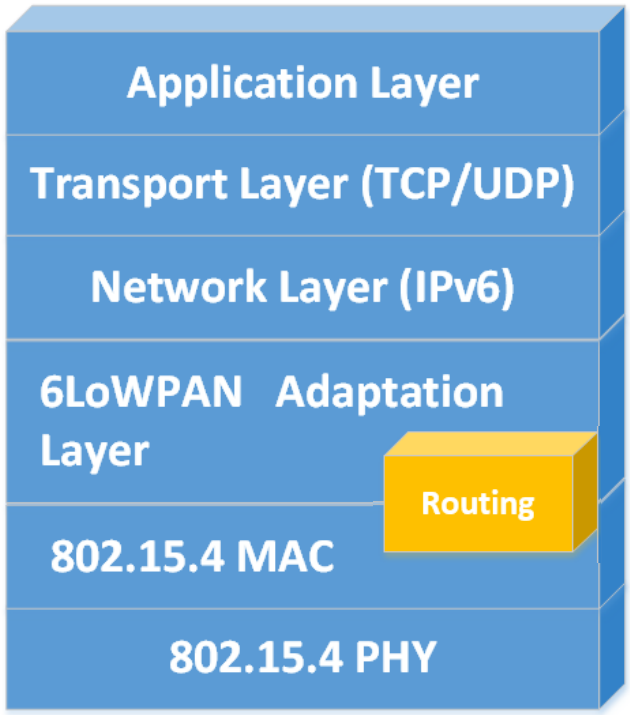
Topologies (II)

- Neither IEEE 802.15.4 nor 6LoWPAN [RFC4944] define how mesh topologies could be obtained and maintained
- 6LoWPAN formation and multi-hop routing can be supported either:

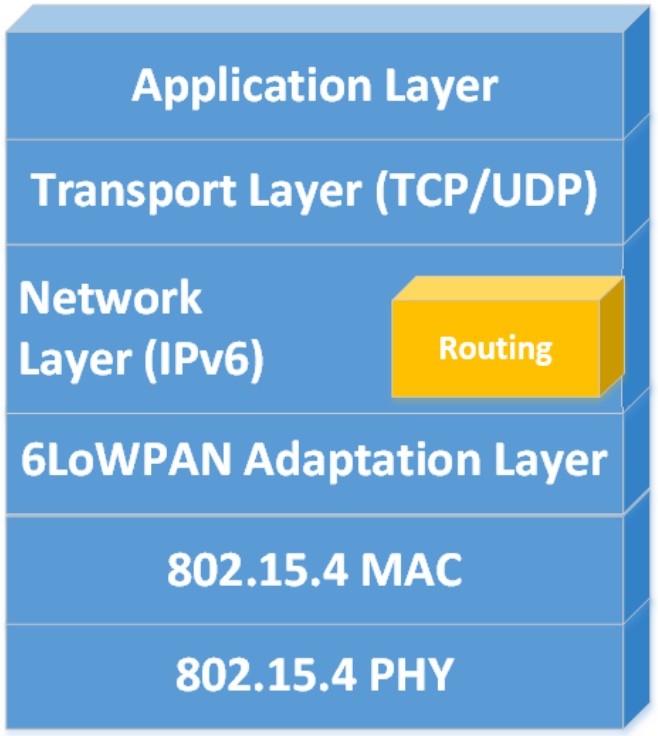
1. Below the IP layer (the adaptation layer or Logical Link Control (LLC) -> **mesh-under**
2. Or the IP layer -> **route-over**

Topologies (III)

- “Routing” in **mesh-under** vs. **route-over**: handles path computation and packet forwarding



Mesh-under



Route-over

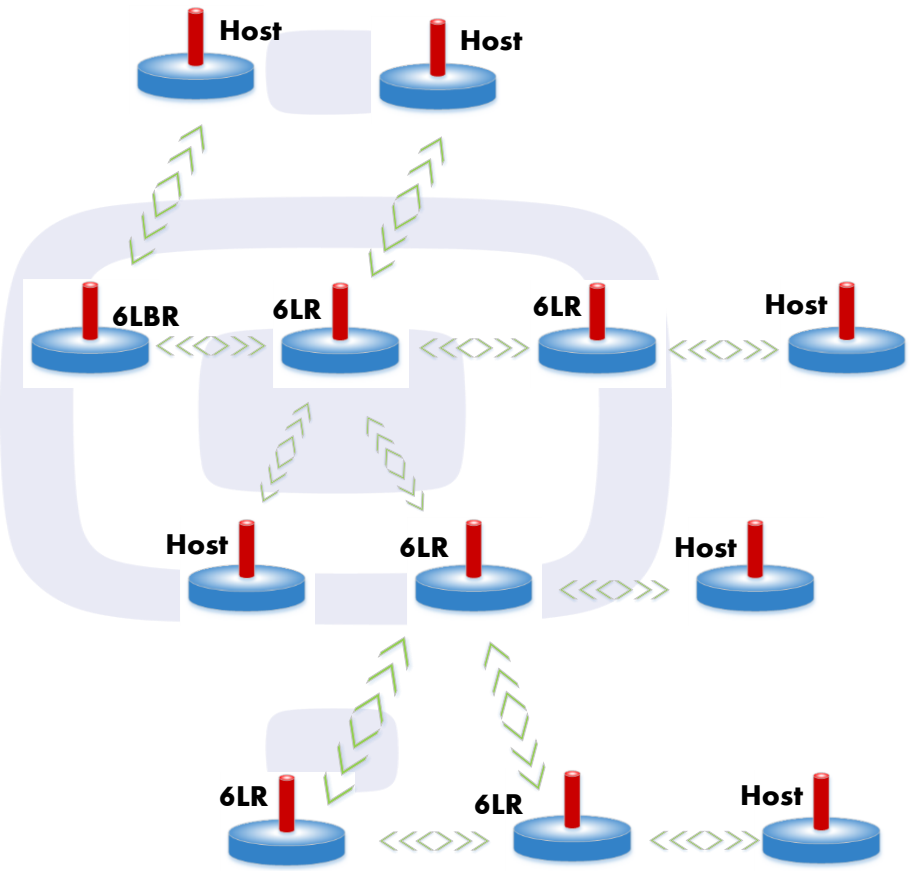
Topologies (IV)

- 6lowpan definitions:

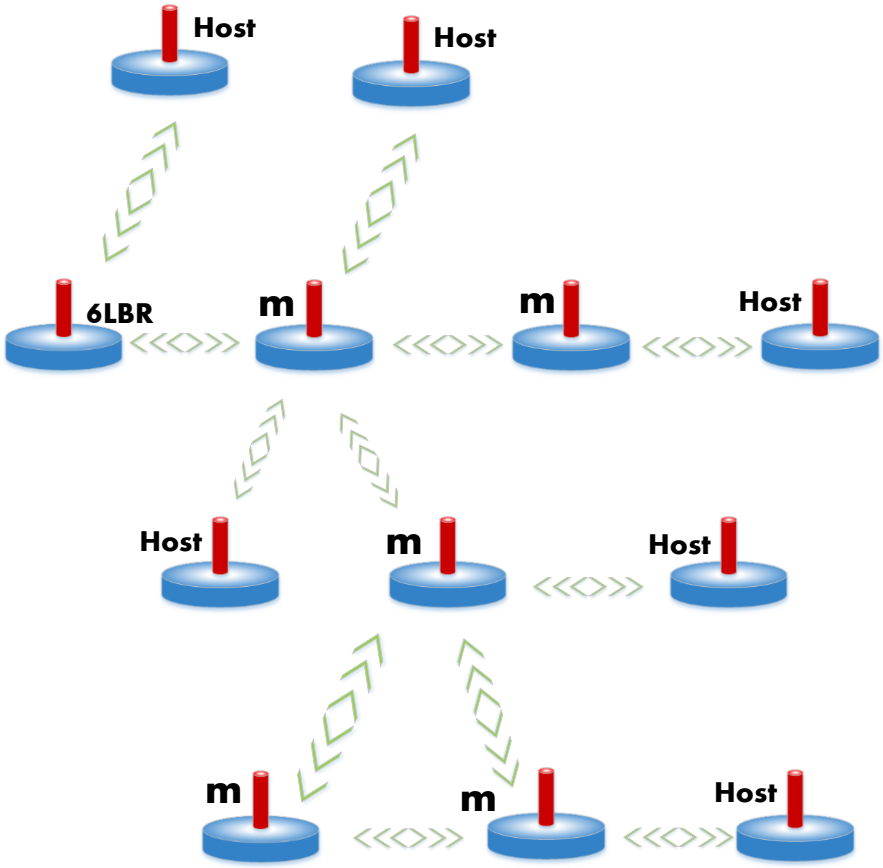
1. **6LBR** (6LoWPAN Border Router): perform route-over routing and connects to “outside”. Can send/receive its own information
2. **6LR** (6LoWPAN Router): perform route-over routing. Can send/receive its own information
3. **Mesh-under forwarder (m)**: perform mesh-under “routing”
4. **Host (h)**: Just send/receive its own information

- 802.15.4 definitions:
  1. **FFD** (Full Function Devices): Participate as routers in a mesh (802.15.4) -> 6LowPAN router
  2. **RFD** (Reduced Function Devices): Discover FFDs and send them their traffic (802.15.4) -> 6LowPAN host

Topologies (V)



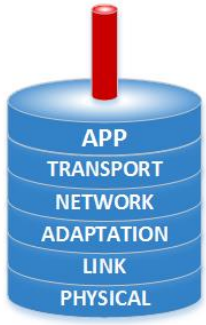
Route-over example



Mesh-under example

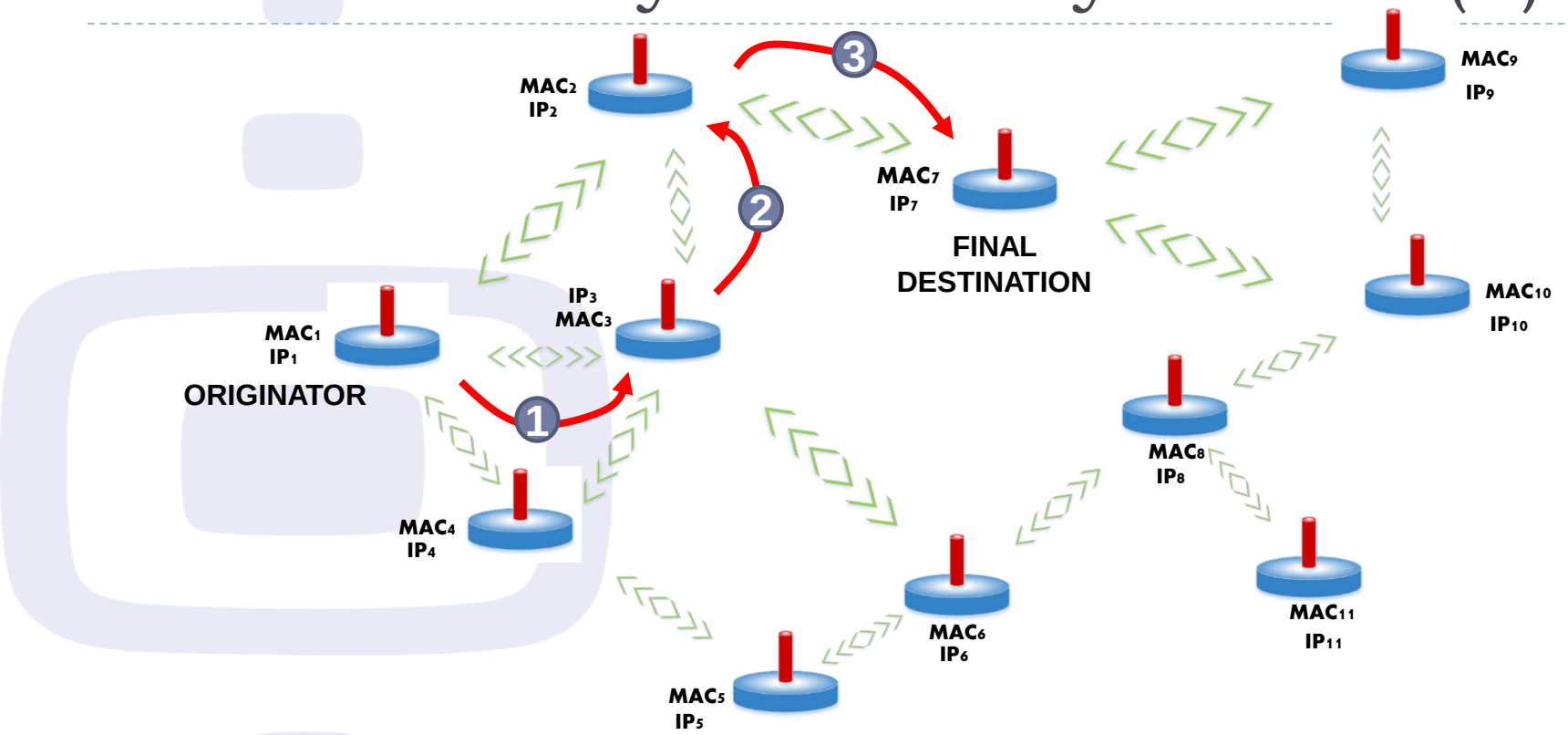
Frame Delivery in Link-layer Mesh (I)

- Mesh topology: 2 devices do not require direct reachability in order to communicate
- We are al **layer 2** (link-layer) -> **Mesh under**



- Mesh delivery is enabled by including a **mesh addressing header** prior to any other headers
- Mesh-under forwarders (m) use a link-layer “routing” table

Frame Delivery in Link-layer Mesh (II)



	802.15.4	6Lowpan: Mesh Addr. Header	IPv6	Upper Layer
1	Src: MAC1 | Dst: MAC3	Src: MAC1 | Final Dst: MAC7	Src: IP1 | Dst: IP7	
2	Src: MAC3 | Dst: MAC2	Src: MAC1 | Final Dst: MAC7	Src: IP1 | Dst: IP7	
3	Src: MAC2 | Dst: MAC7	Src: MAC1 | Final Dst: MAC7	Src: IP1 | Dst: IP7	

Frame Delivery in Link-layer Mesh (III)

- One LoWPAN may be built as one IPv6 link
- Using one IPv6 prefix (/64)

- In this case, mesh-under forwarding mechanisms must be supported

ROLL (I)

- ROLL (Routing Over Low power and Lossy networks)
- IETF standardization:
  - **roll**: LLNs have specific routing requirements that could not be satisfied with existing routing protocols.
  - Routing solutions for subset of application areas of LLNs: industrial, connected home, building and urban sensor networks
  - https://datatracker.ietf.org/wg/roll/charter/
- Among other things, defined:

1. RPL (IPv6 Routing Protocol for LLNs) [RFC6550]
2. MPL (Multicast Protocol for LLNs) [RFC7731]

ROLL (II)

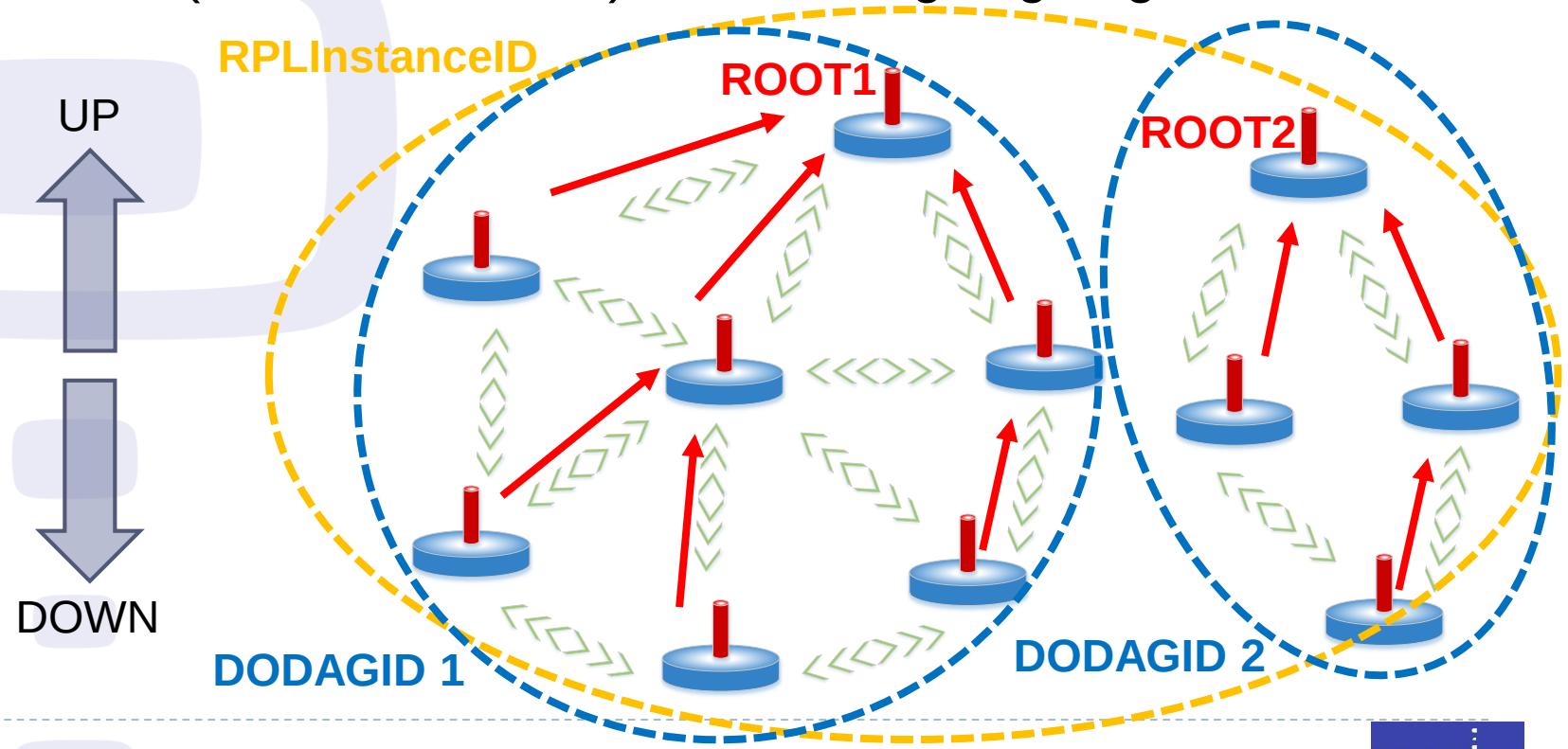
- Constraints of 6LoWPANs result in specific requirements for a routing protocol:

1. Minimal packet size of LoWPANs: routing protocol must impose **low (or no) overhead** on data packets
2. Low routing overhead (**low number of informative packets**) balanced with topology changes and power conservation
3. **Minimal computation and memory requirements** to satisfy the low cost and low power objectives. Avoiding storage and maintenance of large routing tables
4. Support for network topologies in which either FFDs or RFDs may be battery or mains-powered. So should work with sleeping nodes

- Existing routing protocols (OSPF, IS-IS, AODV, and OLSR) evaluated. Don’t satisfy all of these specific routing requirements

ROLL (III)

- **DAG** (Directed Acyclic Graph): Directed graph where all edges are oriented in a way that no cycles exist. All edges are contained in paths oriented toward and terminating at one or more root nodes
- **DODAG** (Destination-Oriented DAG): A DAG rooted at a single destination (the DODAG root) with no outgoing edges



ROLL (IV)

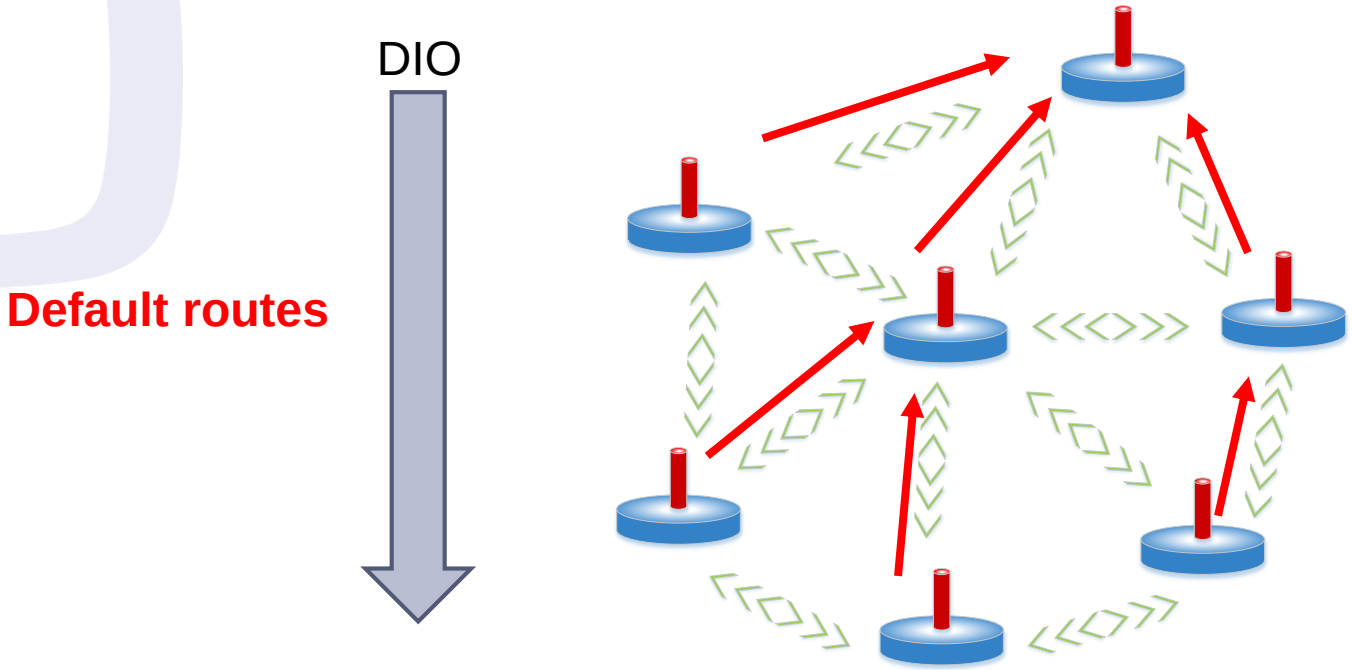
- RPL routes are optimized for traffic to/from one or more **roots** that act as sinks for the topology
- Resulting in DAG topologies, partitioned into one or more DODAGs, one DODAG per sink
- RPL provisions routes Up towards DODAG roots, forming a DODAG optimized according to an Objective Function (OF)
- RPL nodes construct and maintain these DODAGs through DODAG Information Object (DIO) messages

ROLL (V)

- New **ICMPv6** message: RPL control message. Type = 155
- Code: identifies the type of RPL control message
  1. 0x00: DODAG Information Solicitation (**DIS**)
  2. 0x01: DODAG Information Object (**DIO**)
  3. 0x02: Destination Advertisement Object (**DAO**)
  4. 0x03: Destination Advertisement Object Acknowledgment (**DAO-ACK**)
  5. 0x80: Secure DODAG Information Solicitation
  6. 0x81: Secure DODAG Information Object
  7. 0x82: Secure Destination Advertisement Object
  8. 0x83: Secure Destination Advertisement Object Acknowledgment
  9. 0x8A: Consistency Check
- Most have link-local scope
  - Src. Address link-local
  - Dst. Address link-local or all-RPL-nodes (ff02::1a)
- Exception DAO / DAO-ACK using GUA or ULA for src and dest

ROLL (VI)

- DIO (DODAG Information Object) is:
  - Sent “Downwards”
  - Used to discover and maintain “Upward” routes
  - Node discovers a parent, and can join that DODAG, setting that parent as default route
- DIS (DODAG Information Solicitation) is used to trigger DIO messages

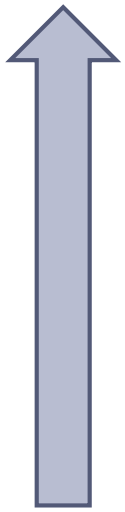


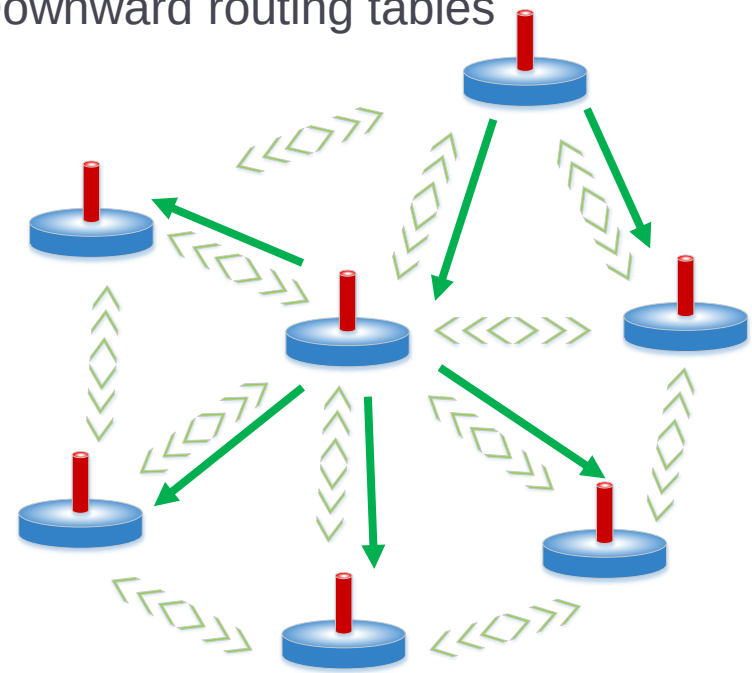
ROLL (VII)

- DAO (Destination Advertisement Object) is:
  - Sent “Upwards”
  - Used to discover and maintain “Downward” routes
  - Two modes:

1. "Storing“ Mode: nodes store Downward routing tables for their sub-DODAG. Each hop on a Downward route in a storing network examines its routing table to decide on the next hop.
2. "Non-Storing“ Mode: nodes do not store Downward routing tables

Each RPL Node configure routes to IPs/PREFIXES





Thanks!

- Questions?

- Contact: info@nodo6.com / training@nodo6.com
- http://www.nodo6.com
- https://www.linkedin.com/company/nodo6
- https://twitter.com/NODO6_RRSS