



Tuesday 18/7 - 9h30-12

IETF 102 ROLL

Routing over Low-Power And Lossy Networks

Chairs:

Peter van der Stok

Ines Robles

Jiye Park - (BIG Thanks for helping with IETF 102 :)

Secretary:

Michael Richardson



Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

BCP 9 (Internet Standards Process)

BCP 25 (Working Group processes)

BCP 25 (Anti-Harassment Procedures)

BCP 54 (Code of Conduct)

BCP 78 (Copyright)

BCP 79 (Patents, Participation)

<https://www.ietf.org/privacy-policy/> (Privacy Policy)

Source: <https://www.ietf.org/about/note-well/>

Meeting Materials

- 9:30-12:00 Tuesday Morning session I
- Remote Participation
 - Jabber Room: <xmpp:roll@jabber.ietf.org?join>
 - Meetecho: <http://www.meetecho.com/ietf102/roll/>
 - Etherpad: <https://etherpad.tools.ietf.org/p/notes-ietf-102-roll?useMonospaceFont=true>
- Minutes taker:
- Jabber Scribe: Rahul Jadhav
- **Please sign blue sheets :-)**

AGENDA ROLL IETF 102

Date: Tuesday, July 17, 2018 (EDT)

Time: 9:30 - 12:00 - 150 minutes - Morning session I

Place: Duluth - 2nd Floor/Convention Floor

Time	Topic	Presenter
9:30 - 9:40 (10 min)	WG Status - Introduction	Peter
9:30 - 10:10 (30 min)	BIER-ROLL Design team	Toerless
10:10 - 10:20 (10 min)	Efficient Route Invalidation draft-ietf-roll-efficient-npdao-03	Rahul
10:20 - 10:30 (10 min)	Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs) draft-ietf-roll-aodv-rpl-04	Charlie
10:30 - 10:45 (15 min)	Root initiated routing state in RPL draft-ietf-roll-dao-projection-04	Pascal
10:45 - 11:00 (15 min)	RPL Observations draft-rahul-roll-rpl-observations-01	Rahul
11:00 - 11:10 (10 min)	RPL DAG Metric Container Node State and Attribute object type extension draft-koutsiamanis-roll-nsa-extension-02	Aris (Remotely)
11:10 - 11:20 (10 min)	Traffic-aware Objective Function draft-ji-roll-traffic-aware-objective-function-01	Aris (Remotely)
11:20 - 11:35 (15 min)	A YANG model for Multicast Protocol for Low power and lossy Networks (MPL) draft-ietf-roll-mpl-yang-01	Peter
11:35 - 11:50 (15 min)	Routing for RPL Leaves draft-thubert-roll-unaware-leaves-05	Pascal
11:50 - 12:00 (10 min)	Open Floor	Everyone

Milestones

Date	Milestone
Apr 2018	Initial Submission of a proposal with uses cases for RPI, RH3 and IPv6-in-IPv6 encapsulation to the IESG
Aug 2018	Initial submission of a root initiated routing state in RPL to the IESG
Dec 2018	Initial submission of a proposal to augment DIS flags and options to the IESG
Jan 2019	Initial submission of a proposal for Source-Route Multicast for RPL to the IESG
Jul 2018	Initial submission of a solution to the problems due to the use of No-Path DAO Messages to the IESG
Jul 2018	Initial submission of a reactive P2P route discovery mechanism based on AODV-RPL protocol to the IESG
Jul 2019	Initial submission of a Forwarder Selection Protocol for MPL to the IESG
Mar 2019	Initial submission of a YANG model for MPL to the IESG
Sep 2019	Recharter WG or close

State of Active Internet-Drafts

Draft	Status
draft-ietf-roll-aodv-rpl-04	WGLC - Discussion today
draft-ietf-roll-dao-projection-04	Discussion today
draft-ietf-roll-forw-select-00	On hold
draft-ietf-roll-useofrplinfo-23	AD Evaluation, waiting for IESG Telechat
draft-ietf-roll-dis-modifications-00	To be continued
draft-ietf-roll-mpl-yang-02	Discussion today
draft-ietf-roll-bier-ccast-01	Bier-roll design team takes over
draft-ietf-roll-efficient-npdao-03	WGLC - Discussion today

Related Internet-Drafts

Draft	Status
draft-thubert-roll-unaware-leaves-05	Discussion today
draft-rahul-roll-rpl-observations-01	Call for adoption - Discussion today
draft-koutsiamanis-roll-nsa-extension-02	Discussion today
draft-ji-roll-traffic-aware-objective-function-01	Discussion today
draft-thubert-roll-bier-01	Bier-roll design team takes over
draft-baraq-roll-drizzle-00	Not email in ML about this

Open tickets

Ticket	Status
#179- Security considerations for dao projection	Being addressed
#180 - issues to address in dao projection draft (lifetime, MOP, retransmissions, route cleanup)	Being addressed

ROLL and BIER

IETF 102 Montreal 2018

ERROR – no draft found

Toerless Eckert tte+ietf@cs.fau.de (Huawei USA),

v1.0

Overview

- draft-thubert-roll-bier, draft-ietf-roll-ccast
 - Describe options how to use bitstrings to address unicast, multicast destinations in ROLL/RPL networks
 - draft-ietf-roll-ccast
 - Uses Bloom filters to compress bitstrings
 - draft-ietf-roll-ccast
 - Attempts to create an overall solutions
 - (BIER, BIER-TE) x (Bloom-filter, full-bitmask)
 - BIER: bitstring identifies receivers, not hop-by-hop nodes
 - BIER-TE: bitstring identifies receivers and hop-by-hop nodes
 - Allows to further reduce state on RPL nodes: Only needs to have state for direct downstream neighbors, not all receivers reachable via them.
 - Discussed via presentations at IETF101
 - See presentation slide deck (slides from Pascal)
 - Conclusion:
 - Enough interest in room to attempt forming a design team

Since then

- Little progress
- Design team bureaucratic work was done:
 - Non-WG mailing list: roll-bier-dt@ietf.org
 - Free to subscribe, open, has normal public archive
 - AD approved charter of design-team (required for getting an email address) on Wiki
 - <https://trac.ietf.org/trac/roll/wiki/roll-bier-dt>
 - Intended Charter:

The intended first phase charter of this design team is to generate guidance for next steps in ROLL to scope and adopt work to support efficiently encoded "bitstring" style addresses. This addressing is meant to support more lightweight unicast and multicast forwarding options - both in terms of packet header size and/or per-node state. One starting point is stateless BIER-TE bitstring style addresses allowing to specify paths just as a sequence of bit addresses. The other starting point is bloom filters to compress large, stateful BIER bitstring style addressing.

The design team will consult with BIER WG, but because ROLL is focusing on low bitrate, low-power, software-forwarding, its solution space can explore options that may not necessarily be considered appropriate for general purpose BIER solutions that desire to support ASIC/NPU forwarding.
- No activity on design team yet
 - Call for offsite conf-call time selection unanswered.
- Can represent only my own analysis/recommendations on following slides

High level view

- I find it easier to resolve high level points first before concluding on encoding details.
- Rest of slides are about such high level points:
- 1. Bloom filter
 - Protocol should support it as an option, but:
 - Think operators can/will only adopt when they have more statistical data
 - Also two main challenges I see

1.a BIER as underlay vs. BIER end-to-end

- BIER-WG so far focused on BIER as underlay for IP multicast in SP networks
 - End-to-end BIER still not official BIER charter item (but may happen some time).
 - Example use case benefit of no-IP-multicast end-to-end BIER
 - For content/video distribution:
 - <https://datatracker.ietf.org/doc/draft-purkayastha-bier-multicast-http-response/>
- Problem summary (IP Multicast over BIER):
 - Additional IP Multicast header that may be unnecessary (What's its value ?)
 - Signaling from receiver to sender to join Group/Channel.
 - If sending application can directly send BIER packets:
 - Every packet can be sent individually to a different subset of receivers. Fast/Flexible!
 - Think about sending control actions to switch a subset of Lightbulbs to go on/off (every bier-bit one light bulb)
- What has that got to do with Bloom filters ?
 - A bloom filter does not identify the set of receivers but a random superset of them
 - Examples: More lights will go on/off than you wanted
 - Unless you duplicate the actual, non-compressed BIER bitstring at the IP multicast or application level
 - How do you dynamically make just the subset of lightbulbs join the multicast group that you want to send the on/off command to ? You don't. All IP multicast apps doing this enumerate the target nodes (unnecessarily) in the data payload.

1.b More bloom filter ?problems?

- BIER/BIER-TE avoid duplicate/loops by usually resetting bits when a node figures the bits are “unnecessary”
 - BIER-TE when a bit has been “transited” (bier indicating node/link)
 - BIER when egress interface does not lead to a particular bit (keep only relevant bits)
- Not 100% sure, but: Bloom with BIER logic (draft-ietf-roll-ccast):
 - Should work “fine” when the effect of the underlying RPL “loop” prevention effectively limits forwarding to a tree – no two paths leading to the same receiver
 - Otherwise you would have duplicates to that receiver.
- Similar BIER-TE: Do we expect that we run BIER-TE only across an underlay tree or an arbitrary underlay topology ?
 - With Tree it would work with bloom filters
 - With arbitrary topology not (loops, duplicates)
- Not 100% sure... for discussion/analysis.

2. Using BIER-TE efficiently

- Using BIER-TE bitstring seems like a good way to save state on RPL routers (just direct adjacencies)
 - But size of bitstring may be an issue.
 - Can always use more bitstrings. Shorter bit-string length, more groups (sets)
 - More state required: Bits for midpoints need to be replicated into every set, or more network wide optimization necessary to avoid this.
 - #groups packets necessary to send to every receiver.
- BIER-TE/BIER never optimized for IP unicast so far!
 - But this seems to be a key use case target in ROLL/RPL
- Two bit string lengths and unicast vs. multicast model:
 - Use bitstring only for routers, not endpoint.
 - Bitstring is only used to hop-by-hop route packet to last-hop router. Last-hop-router then uses endpoint-ID to deliver (eg: from packets destinations IP unicast address).
 - If 90% of nodes are endpoints: only need 10% bitstring size (for unicast only!).
 - Multicast: use 10 x longer bitstring. Its software forwarding, so it can easily be done. 10 times larger bitstring still cheaper than sending multiple packets. We just do not want this in 1Tbps ASICs because its really expensive there.

3. Path engineering via BIER-TE

- Assume we want to allow root to express various alternative paths to the same destination (unicast/multicast) via bitstring.
- Signaling to assign those bits:
- Feel very confident if assignments are done through signaling from root, we have easier global control than other signaling options.
 - Reassigning/modifying bits / addresses can be done coordinated by root
 - Eg: Seamless readdressing: Assign second new bit to nodes first, then remove old bits to have “make-before-break” logic of possible paths. Much harder to do without central control.
 - If you already have a necessary point of central functions (root), the solution should maximize benefits from it.
 - Always good to avoid introducing unnecessary centralization, but equally good to leverage the benefits one could get from it.

4. Encapsulation / compression

- Clueless on how RPL minimizes encapsulation overhead
- As mentioned: IP Multicast + BIER headers are IMHO useless overhead if the solutions could just use BIER end-to-end.
 - Don't know if/where RPL/ROLL networks MUST use IP multicast
 - If apps would be written new for the new ROLL/RPL, it could be better to target that those would be native BIER apps (app sends bitstring. Maybe not the router-bits, but just the receiver bits. Root adds the router bits).
- Do not understand existing RPL header compression schemes.
 - May want to think about a “pseudo IP multicast” header that is compressed away:
 - If whole network just has 16 applications, each one gets one multicast group, we compress this in the header to 4 bit, but the BIER bitstring actually decides which receivers get the packet. The 4-bit group number is then only in the receiver nodes to decide which IP multicast application gets the packet (translated back to an IP multicast group address).

5. BIER vs. BIER-TE

- Assumption:
 - BIER-TE approach required for non-storing mode ? (true ?)
- Storing mode:
 - Compare benefits of BIER vs. BIER-TE to decide which one a particular deployment should use
 - BIER-TE benefit now is its explicit-hop-by-hop path selection
 - Vs. flexible BIER ECMP options (if those are possible in underlying RPL..) Not sure.. Is underlying RPL routes not only tree, is there ECMP ?
- In any case:
 - A few reference topologies would be great for the discussion
 - Especially when they represent extremes of typical deployments (maximum, average, minimum size of networks, #nodes, how many routers vs. endpoints, etc. pp)

Closing remarks

- Happy to continue to help if there is ongoing interest:
 - No good sales pitch for the solution yet:
 - Can not make judgement calls how useful BIER/BIER-TE will be yet for ROLL/RPL, but would be great if that could be worked out first via the most simple quantitative examples comparing what we could achieve with BIER/BIER-TE vs. without it.
 - This is where reference topology examples may come in.
 - This type of sales pitch example should help WG to decide if work is beneficial enough.
- I would primarily like to see if there is any interest in the native BIER model in the ROLL/RPL space.
 - I think it should be great for wide range of group-control apps (lights on/off)
 - This is a topic BIER WG has not tackled yet, so RPL/ROLL could lead the way
 - Of course: any change/improvement to multicast in apps takes 20 years (SSM: 1999).
 - ROLL should not have to do the Application level BIER side. Should be done in BIER-WG (IMHO)

Thank You!

Efficient route invalidation for RPL

[draft-ietf-roll-efficient-npdao-03](#)

Rahul, Pascal, Rabi, Zhen
IETF102, Montreal

History:

IETF95 - Presented the problem statement

IETF96 - Presented existing solutions based on comments rcvd and why those fall short

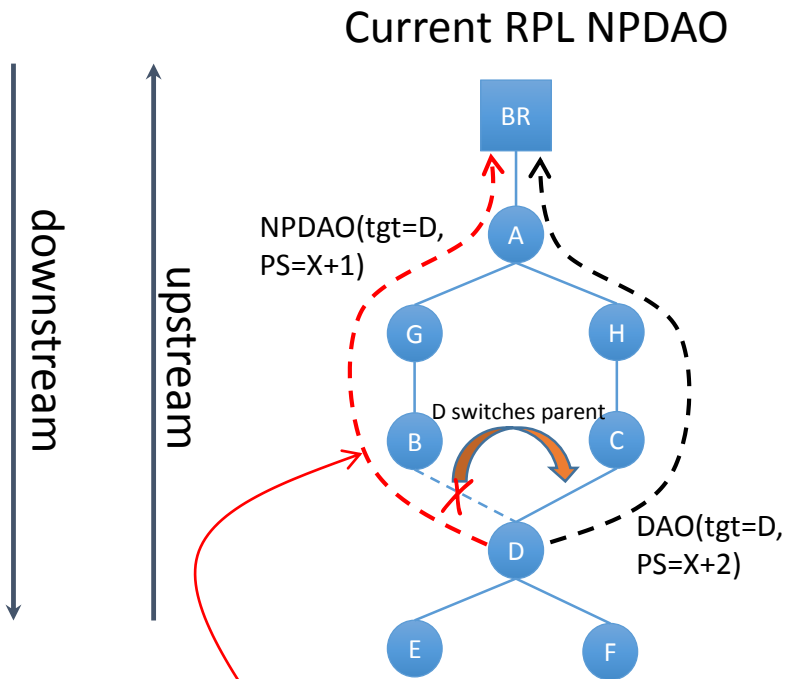
IETF98 – Presented new solution for improving route invalidation

IETF99 – adopted as WG document , thank you for the review

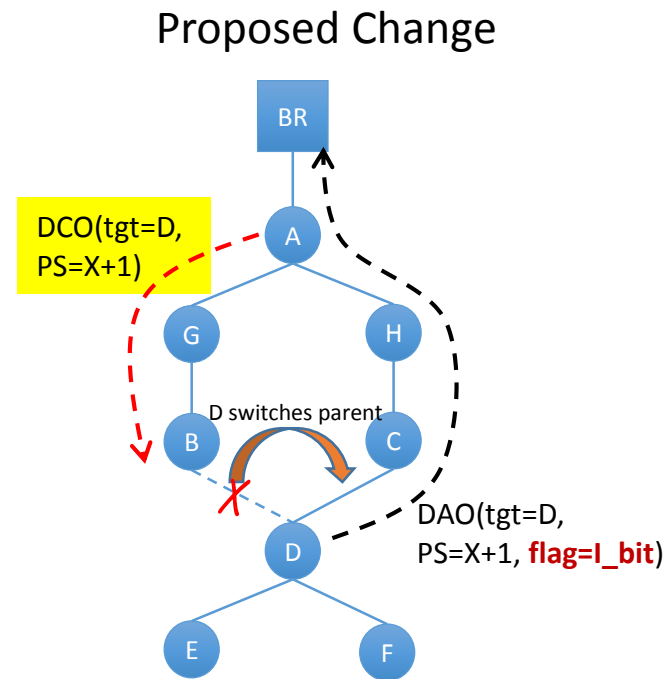
IETF100 – Changes to message codes

IETF101 – Implementation Performance report

Recap: the problem and the solution



NP-DAO via broken links will cause many problems such as reachability and efficiency



- Send the DAO via the new parent;
- Common parent to trigger the DCO to invalidate the previous path

PS = PathSequence
Tgt = Target

Updates

- Secure DCO/DCO-ACK added
- Implementation aspects:
 - Huawei has piloted this draft implementation, 2 years ago
 - Contiki open implementation and performance report was shared before IETF101
- Questions?
- WGLC?

Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)

draft-ietf-roll-aodv-rpl-04

IETF 102, Montreal

Satish Anamalamudi <satishnaidu80@gmail.com>

Mingui Zhang <zhangmingui@huawei.com>

AR. Sangi <sangi_bahrian@yahoo.com>

Charlie Perkins <charles.perkins@earthlink.net>

S.V.R Anand <anand@ece.iisc.ernet.in>

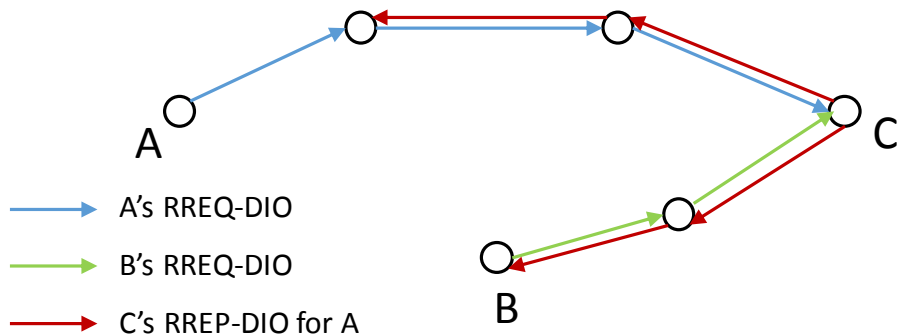
Liu Bing <remy.liubing@huawei.com>

'T' bit removed from the RREP for two reasons

1. In hop-by-hop mode, to build the route entry, the destination address (OrigNode's address) MUST be known by the intermediate nodes

	Source Address	Destination Address	Next hop
Upward route (by RREQ-DIO)	TargNode's address in the AODV-RPL Target Option	OrigNode's address in the DODAGID field of RREQ-DIO	The source address in RREQ-DIO
Downward route (by RREP-DIO)	Indeterminate if the Target Option is elided	TargNode's address in the DODAGID field of RREP-DIO	The source address in RREP-DIO

2. Two OrigNodes, same local RPLInstanceID, same TargNode. The RREP-DIOs for the two OrigNodes can't be distinguished.

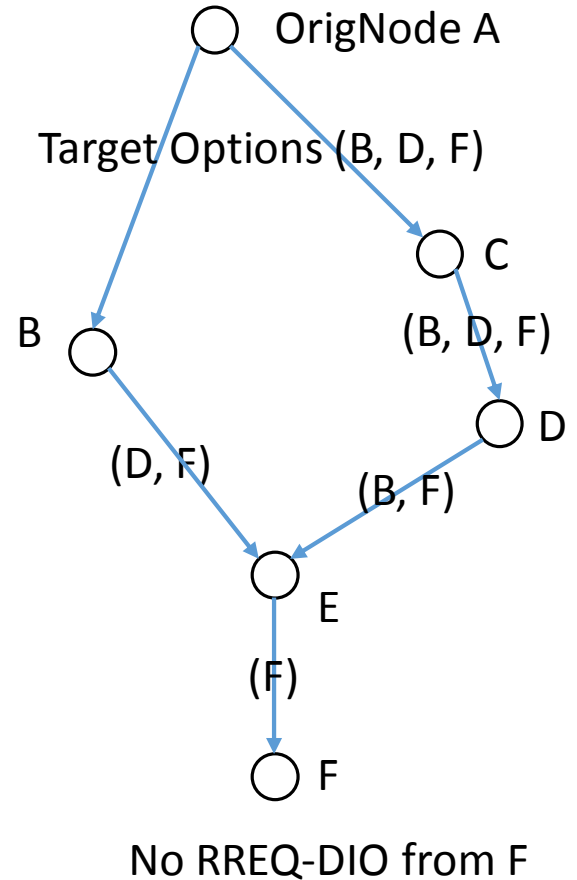


If the OrigNode's address is not included explicitly:

A RREP-DIO from C to A arrives at B, B thinks the downward route is found, but the route follows A's requirements.

Multi-target RREQ-DIO processing

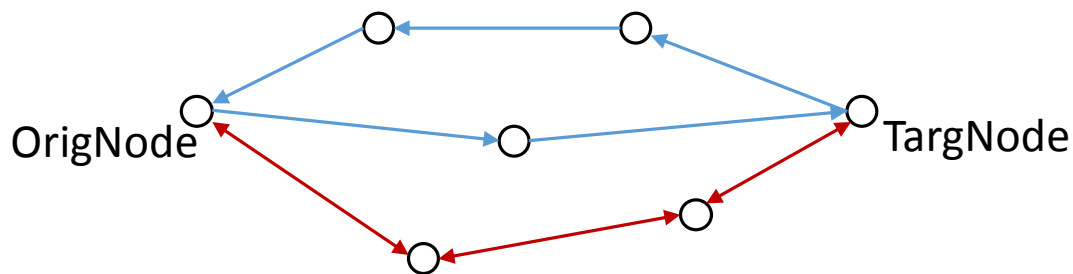
- Multiple Target Options can be included in the OrigNode's RREQ-DIO
- A TargNode can be an intermediate node to other TargNodes
- A TargNode MUST delete its own related Target Option before retransmit the RREQ-DIO.
- When RREQ-DIOs with lists of different Target Options are received, the intersection of the lists MUST be taken.
- When a TargNode receives a RREQ-DIO and it is the only target left, the TargNode MUST NOT retransmit the RREQ-DIO.



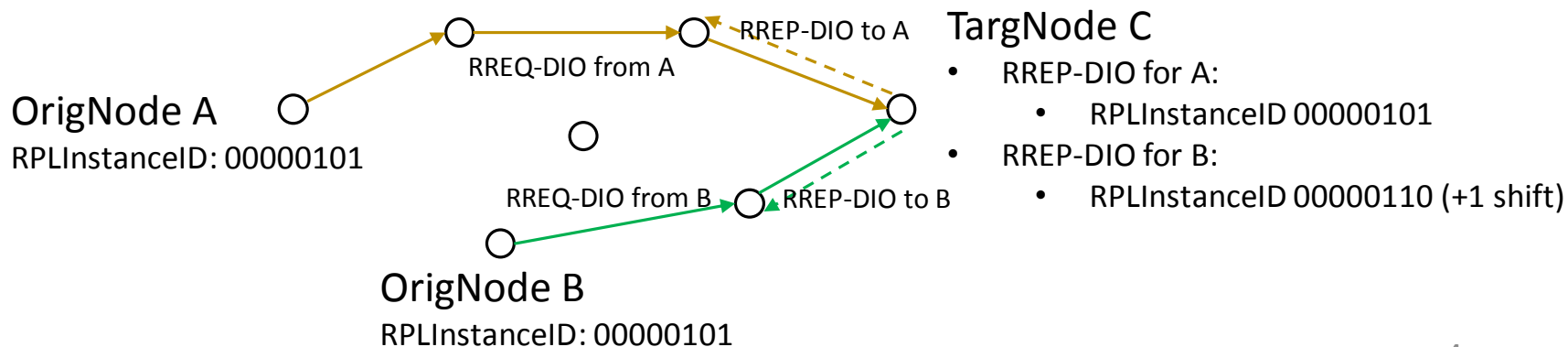
RPLInstanceID pairing in RREQ and RREP

- If multiple RPLInstances exist between the same pair of OrigNode and TargNode, pairing is necessary to match a RREP-instance to the right RREQ-instance. For example, using the same RPLInstanceID.

Different constraints may result in 2 instances



- Shift the RPLInstanceID to another number if the one to be used by the TargNode is already occupied. The original RPLInstance ID can be recovered when necessary.



Disallow multicast address in the AODV-RPL Target Option

- Multicast is complicated compared to multiple Target Options.
- The broadcast of RREQ-DIO would continue even though all the TargNodes in the multicast group have been reached.
- What would the Destination Sequence Number be for a multicast address?

Root initiated routing state in RPL

[draft-ietf-dao-projection](#)

Pascal Thubert
IETF 102

Montreal, July 2018

Root initiated routing state in RPL

P.Thubert

IETF 101

London

Changes Highlights

- Invited Rahul to work on loop avoidance at IETF 101
- New text on use with parcimony to protect devices
 - Getting knowledge of device capability is out of scope. Is that OK?
- Split storing and non storing P-DAO
 - Generic term Route Projection Options (RPO)
 - Via Information Option vs. SR-VIO
- Complex: Rules to concatenate routes and avoid loops
 - SR-VIO can be loose if another route to next hop exists (SR only?)
- Still need to revisit the MOP, 3 bits, gets saturated

Discussions

How is the topology known to the root?

How are the node capabilities known to the root?

Complexity of mixed modes and route concatenation

MOP saturation

Compression of the Via Info option (so far full addresses)

Loop avoidance

- in particular for loose and not end to end route
- Recommend Setting the 'O' bit

<RFC6550>: "Down 'O': 1-bit flag indicating whether the packet is expected to progress Up or Down. A router sets the 'O' flag when the packet is expected to progress Down (using DAO routes), and clears it when forwarding toward the DODAG root to a node with a lower Rank). A host or RPL leaf node MUST set the 'O' flag to 0."

RPL Observations

[draft-rahul-roll-rpl-observations-01](#)

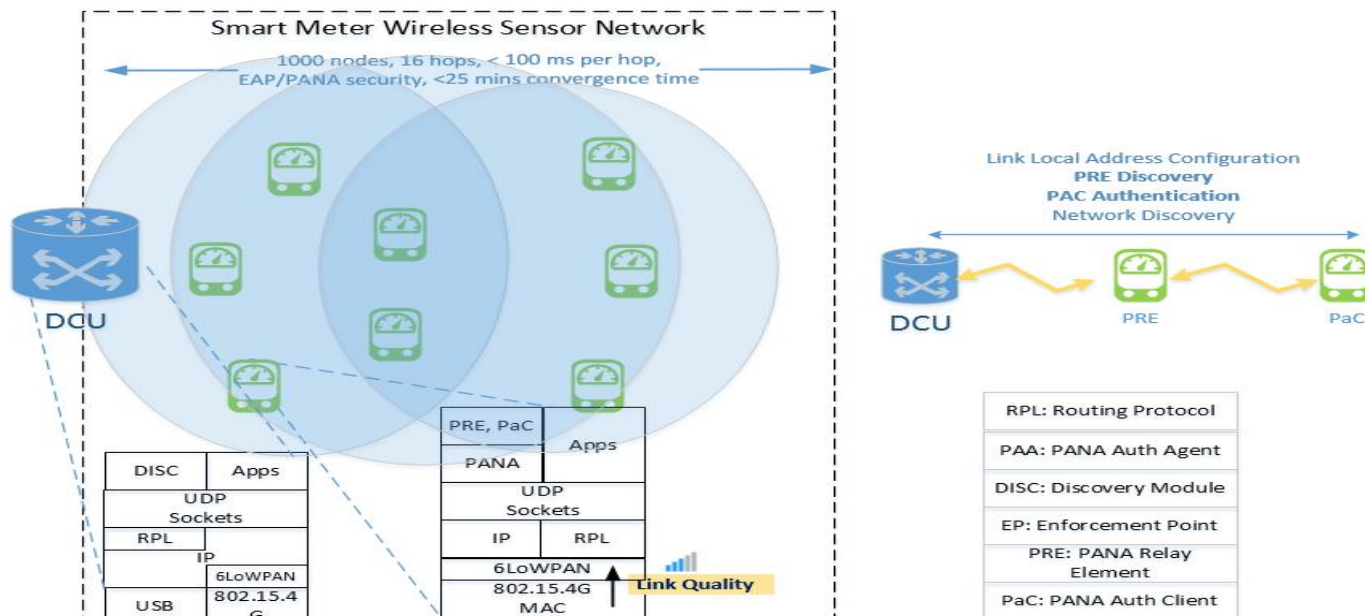
- Rahul, Rabi, YueFeng@ Huawei
IETF102, Montreal

IETF101: Presented first draft

IETF102: Updated based on disc on ML (Thanks to Michael, Pascal)

Background for the observations ...

- Background for the observations
 - Observations were made during solution implementation based on 802.15.4
 - Mostly for storing MOP
- We had some sort of implementation in place for the problems
 - But we don't believe our solutions are optimal/best



DTSN: A lollipop counter or no?

- 6550 says No

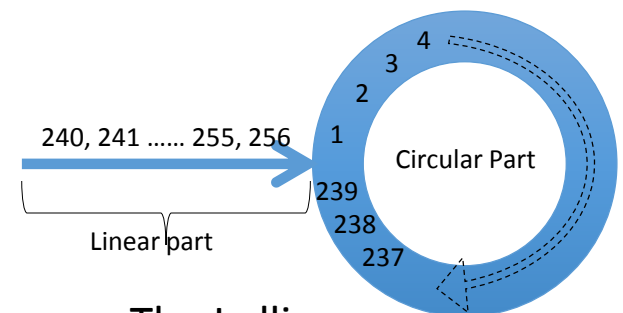
- Section 7 talks about “Sequence Counters” but does not include DTSN as part of it
- Thanks Michael for pointing this out

- Implementations are clearly confused

- Contiki considers DTSN as a lollipop counter
- RIOT does not

- What should it be?

- Consider DTSN not to be a lollipop counter:
 - On reset, start DTSN from a random value
 - Works ok, unless on reboot the node uses the same random value which was its last DTSN prior to reboot
 - Only way to avoid this is to backup in flash on every DTSN update (which is costly)
- Consider DTSN to be a lollipop counter:
 - On reset starts from 240 (based on e.g. in Section 7.2)
 - Need to backup the DTSN in flash for straight part
 - Once it moves into circular region then no need to backup in flash.
 - No failure case involved and flash write is limited to straight part

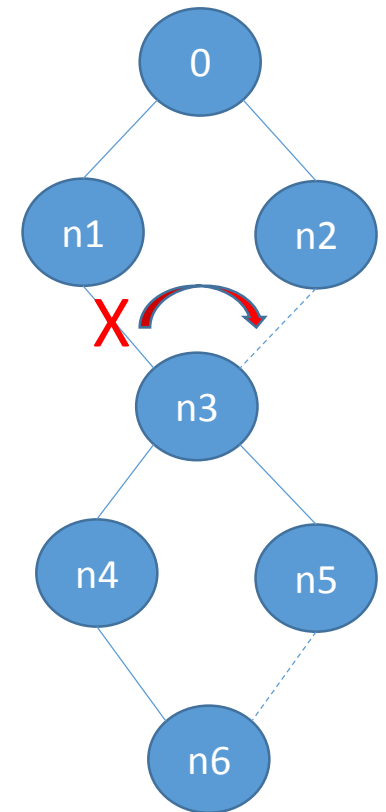


The Lollipop

- Clarifications needed in 6550

The Problem: DTSN in storing mode

- Problems to handle
 - Dependent nodes route update
 - Impacts downstream route availability
- DTSN (DAO Trigger Sequence Number)
 - Decides if DAO should be sent
 - Decider element for RPL Control Traffic
- Problem in storing MOP only
- Tradeoff downstream route-availability vs control overhead



Implementer's Dilemma1

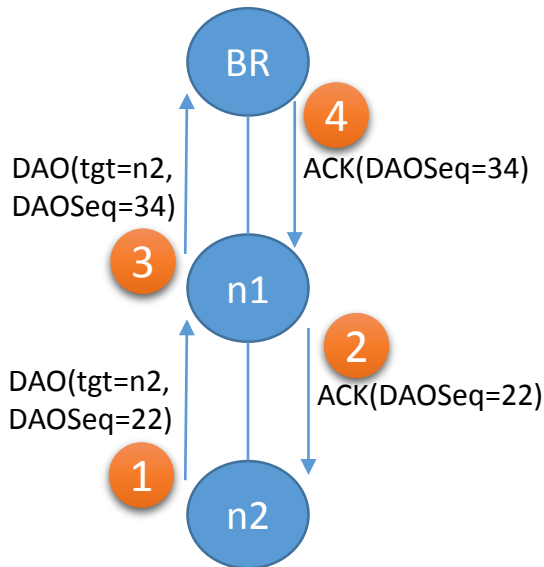
- Should DTSN be incremented with every DIO trickle timer interval?
 - What happens if you do?
 - DAO traffic is too high
 - What happens if you do don't?
 - DAO redundancy is too low. High probability of DAO not reaching BR.
 - With increase in hops, the probability of DAO success drops sharply.

Dilemma2

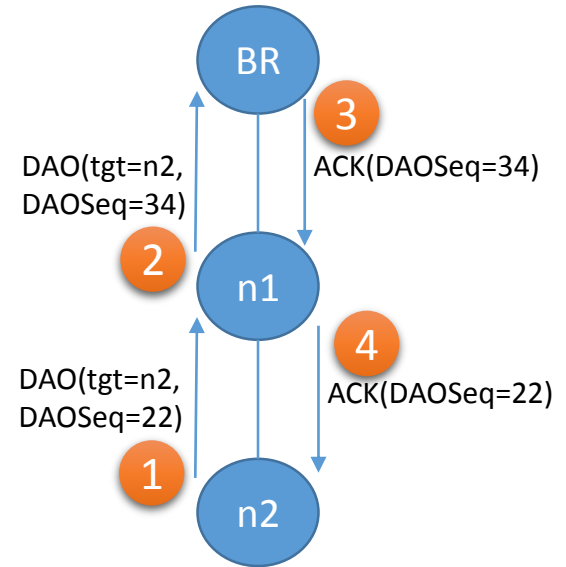
- On parent switch, should node increment its DTSN ?
 - Yes, of-course. Otherwise how would child nodes update their paths.
 - Should child nodes in turn even reset DIO trickle timer and increment DTSN?
 - How would sub-child updates their paths?

DAO-ACK: Multiple interpretations

Hop-by-hop ACK



End-to-End ACK



Pros:

1. No additional RAM
2. Very easy handling. No state.

Cons:

1. Does not help target determine if the DAO has reached BR.
2. Negative ACK from grand-parents cannot be propagated back.

RIOT implements this.

Pros:

1. Helps target determine if the DAO has reached BR.

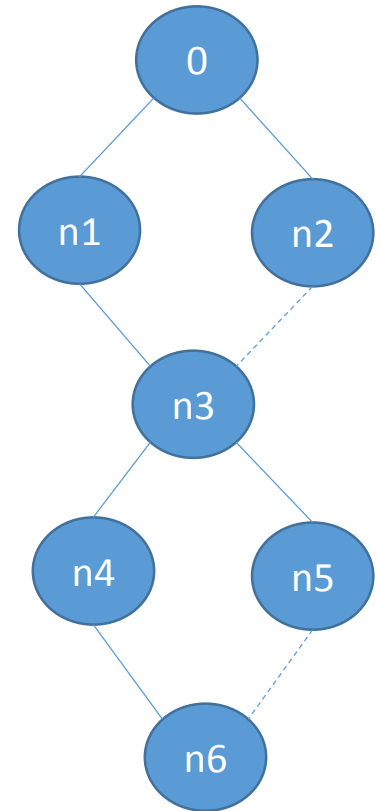
Cons:

1. 6LRs need to maintain DAOSeq in routing entry. Thus 1B per routing entry.
2. Managing DAO-ACK timeout is non-trivial.

Contiki(new version) implements this.

Another Interpretation

- Thanks Pascal for the discussion
- A 6LR when it responds with a DAO-ACK accepts the responsibility to forward the DAO to its upstream parent
 - i.e. it may retry DAO
- This means 6LR needs to maintain state for retransmission in case of no-response
 - Can get easily overwhelmed with if the sub-DODAG rooted at that 6LR is big
- Negative ACK can't be propagated back to the child

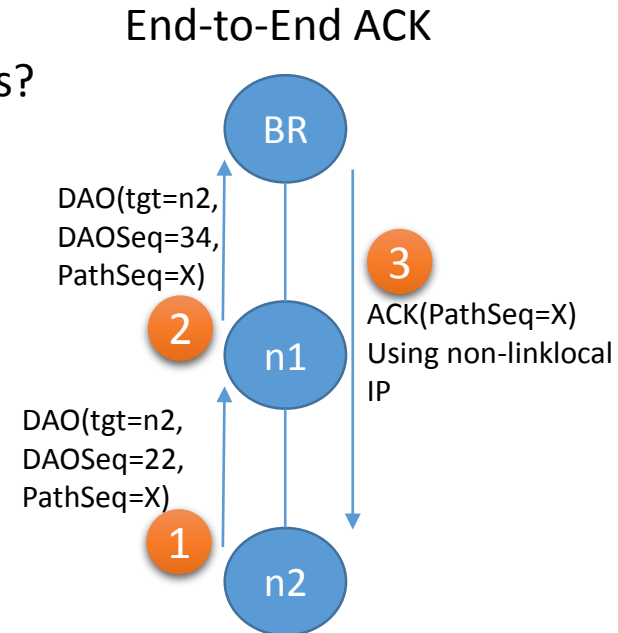


DAO-ACK and aggregated targets: How to ACK?

- DAO-ACK is for DAO message, not individual targets within DAO
 - Also, ACK cannot carry any options as per existing RPL spec
- If multiple targets in a DAO and if subset of targets fail, then how to ACK?
- RPL is not clear on how to handle aggregated targets
 - It certainly allows, but does not do failure handling...
 - RIOT implementation currently sends aggregated targets.
 - Contiki does not work with aggregated targets.
 - Thus interop between them is not possible today (at multi-hops)...

DAO Retransmission and DAO-Ack

- DAO-Ack is important because
 - Only way for node to know that the E2E path is established...
- In hop-by-hop case
 - What happens if DAO/DAO-Ack fails on ancestor links?
- Can we ACK end-to-end using global IP address?
 - No RAM requirement
 - Reduced handling on the intermediate 6LRs
 - But ACKs can't be aggregated in this case



Lot already discussed on ML (Oct 2015):

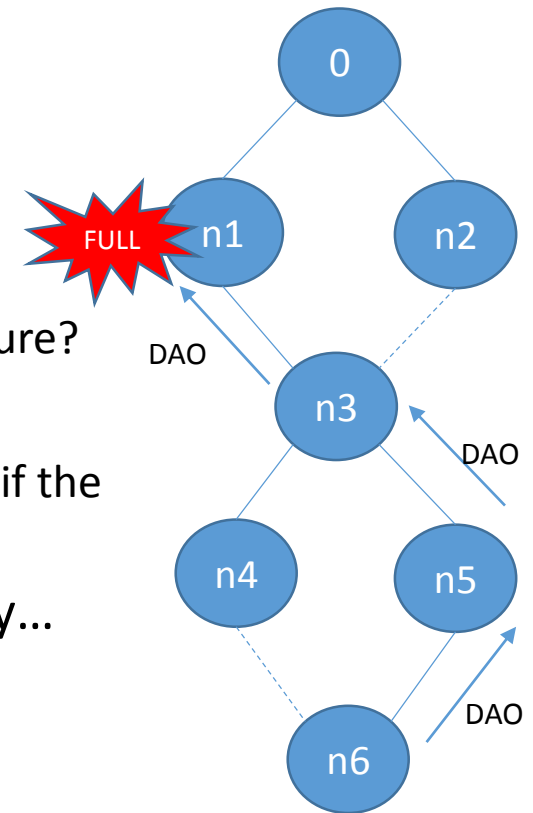
<https://www.ietf.org/mail-archive/web/roll/current/msg09469.html>

Handling node reboots

- RPL State information needs to be maintained across node reboots
 - For e.g. DTSN, PathSequence
- Losing this state across reboot could result in serious loss of connectivity
- Clarification on use of persistent storage

Handling resource unavailability

- Neighbor cache table and Routing Table
- Handling neighbor cache entry full scenario
 - Basic handling there currently
 - DAO-NACK and NA status!=0...
 - It's not enough though...
 - How to avoid connecting to same neighbor in the future?
- Handling routing table full scenario
 - No multi-level proactive feedback, i.e. what happens if the ancestor node does not have space?
- DIO does not signal resource availability currently...



Other Points:

- Should Transit Information be Optional?
 - RPL Transit information carries
 - Path Sequence and Path Lifetime
 - Parent Address (for NS-MOP)
- Aggregated Target Container
 - Aggregation can be optional but should the reception be mandated?

How to do it?

Point	How to handle?	Remarks
DTSN Counter Operation	Errata? (after deciding whether it is lollipop or not)	Can a statement on flash usage be added?
Transit Information been optional	Errata?	
DAO-Ack Semantics	Need solution and doc update or new doc?	Does WG accept that an E2E Ack is needed?
Aggregated DAO target handling.. DAO-Acking for aggregated targets	Could be same as above document	May be an errata/clarification?
DTSN Increment in storing MOP	Need a solution	No clear way of solving this currently.
Handling resource unavailability	Work-in-progress in 6TiSCH	6TiSCH has two documents from Michael which attempts to handle it

RPL DAG Metric Container Node State and Attribute object type extension

draft-koutsiamanis-roll-nsa-extension-02

Remous-Aris Koutsiamanis: aris@ariskou.com

Georgios Z. Papadopoulos

Nicolas Montavont

Pascal Thubert

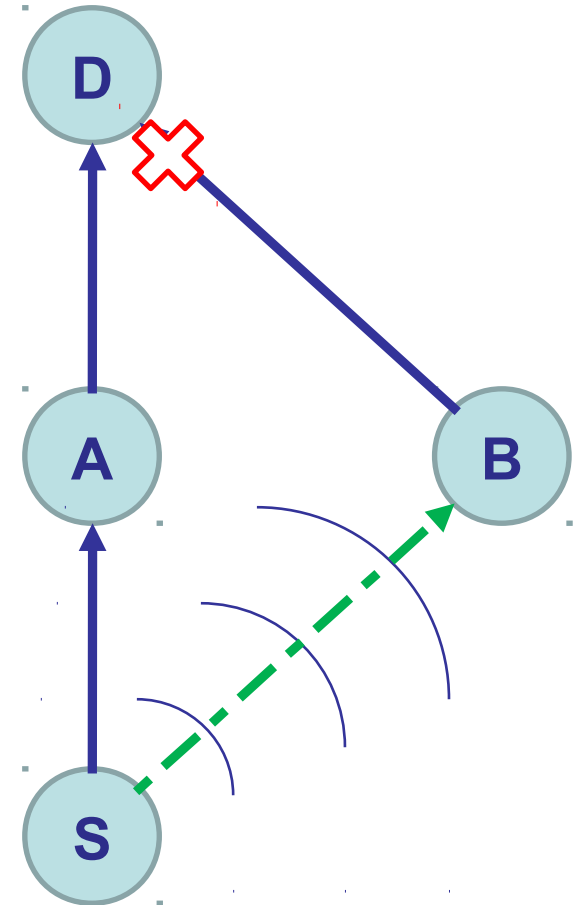
ROLL@IETF102

New since -01

- **Publish code**
 - **Contiki OS implementation**
 - **Wireshark dissectors**
 - **But, without decompression**
- **Renamed poorly chosen acronyms**
- **Issues:**
 1. **IPv6 addresses size**
 2. **“Bad” parents**
 3. **Flooding**

Toward Determinism

- **Reliable communication**
- **Low jitter performance**
- **Packet Replication Elimination**
 - Replication
 - Elimination
 - Promiscuous overhearing



Requirements [1]

- **Alternative Parent Selection**
 - RPL DODAG Information Object (DIO) message format **SHOULD** be extended
 - routing protocol should be extended to allow for 6TiSCH nodes to select AP(s)
- **Promiscuous Overhearing**
 - 6top Protocol should be extended to allow a cell reservation with two receivers
 - 6P ADD Request Format should be transmitted either twice or once in multicast
- **Cells without ACKs**
 - only one parent **MUST** acknowledge the data packet
 - Or an efficient way for double ACKS
- **Packet Elimination**
 - Tagging Packets for Flow Identification

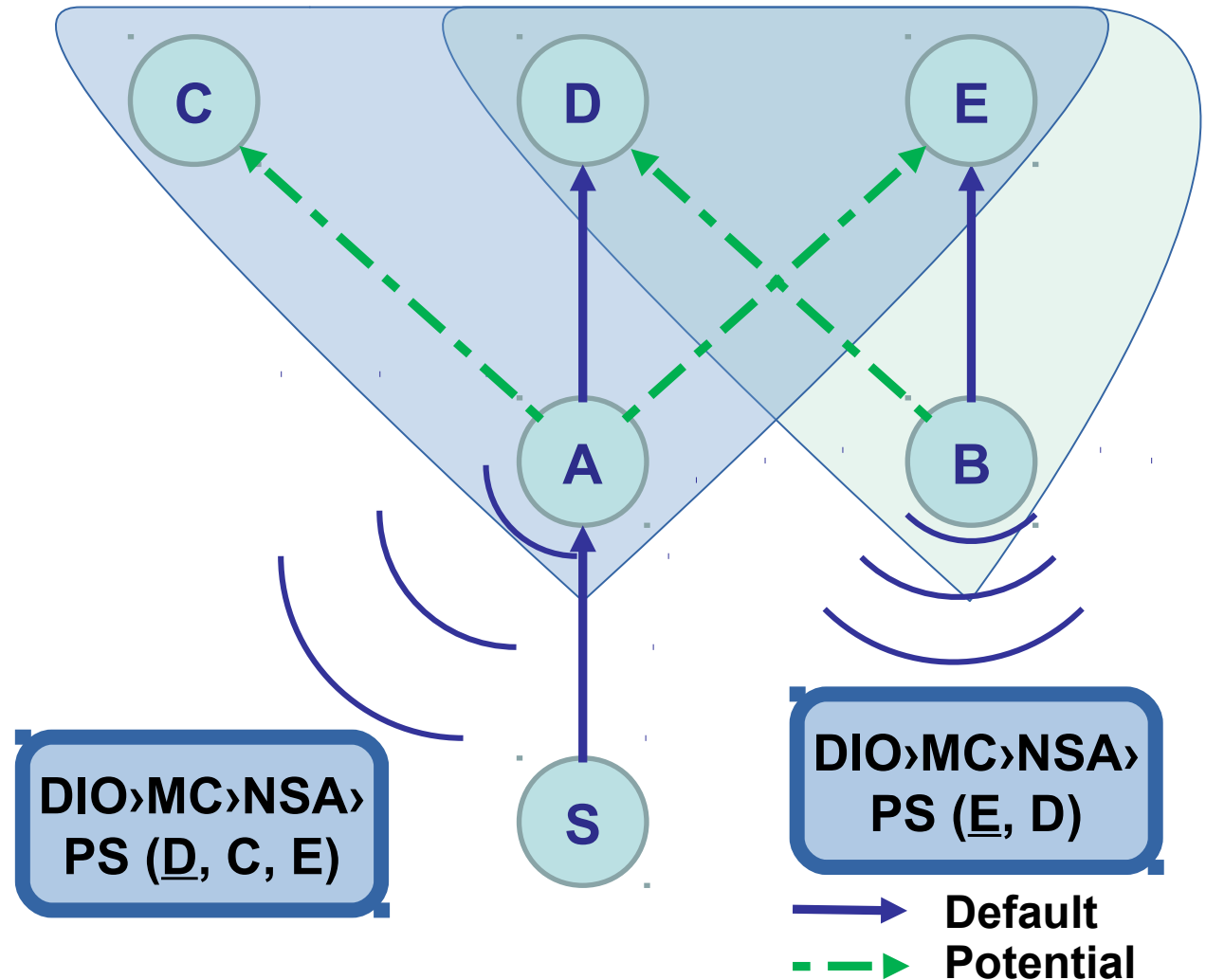
[1] G. Z. Papadopoulos, N. Montavont, and P. Thubert, "Exploiting Packet Replication and Elimination in Complex Tracks in 6tisch LLNs," Working Draft, IETF Secretariat, Internet-Draft draft-papadopoulos-6tisch-pre-reqs-01, 27 December 2017.

Alternative Parent Selection

- **Draft enables Alternative Parent Selection mechanism**
- **Allows selecting alternative parent with common ancestor**

Parent Selection - DIO Messages

- Parent Set A:
 - {D, C, E}
- Parent set B:
 - {E, D}



DIO Format Example

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
RPLInstanceID										Version Number										Rank											
G	o	MOP			Prf			DTSN										Flags					Reserved								
DODAGID																															
DAGMC Type (2)										DAGMC Length																					
DAG Metric Container data																															

MC Format Example (1)

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 (1)										Res Flags					P	C	O	R	A					Prec					Length (bytes)				
Res					Flags					A	O	PS type (TBD1)					PS Length (bytes)																
PS IPv6 address(es) ...																																	

- **Parent Set (PS)**

- Node State and Attributes Option
- PS type = 1 (8 bits)
- PS Length = # of PS addresses x IPv6 address size (8 bits)
- PS IPv6 addresses = 1 or more IPv6 addresses

MC Capture

- Implemented in Contiki OS
- Modified Wireshark

```

  - ICMPv6 RPL Option (DAG Metric container)
    - Type: DAG Metric container (2)
    - Length: 40
    - Routing Metric/Constraint Type: Node State and Attribute (1)
      - Flags: 0x0201, Flag C
        - 0000 0... .. = Reserved Flags: 0x00
        - .... .0.. .. = Flag P: Not set
        - .... ..1. .. = Flag C: Set
        - .... ...0 .. = Flag O: Not set
        - .... .... 0... = Flag R: Not set
        - .... .... .000 .. = A Field: 0x0
        - .... .... .... 0001 = Precedence field: 0x1
      - Metric Length: 36
      - Node State and Attribute Object: 0x0000
        - 0000 0000 .... .. = Reserved field: 0x00
        - .... .... 0000 00.. = Flags: 0x00
        - .... .... .... ..0. = Flag A: Not set
        - .... .... .... ...0 = Flag O: Not set
      - Node State and Attribute Optional TLV: 1
        - Node State and Attribute Optional TLV Type: 1
        - Node State and Attribute Optional TLV Length: 32
        - Raw Data: fe8000000000000000204000400040004fe80000000000000...
```

1. Issue: Extension overhead

- IPv6 addresses

```

- ICMPv6 RPL Option (DAG Metric container)
  - Type: DAG Metric container (2)
  - Length: 40
  - Routing Metric/Constraint Type: Node State and Attribute (1)
    - Flags: 0x0201, Flag C
      - 0000 0... .. = Reserved Flags: 0x00
      - .... .0.. .... = Flag P: Not set
      - .... ..1. .... = Flag C: Set
      - .... ...0 .... = Flag O: Not set
      - .... .... 0... .. = Flag R: Not set
      - .... .... .000 .... = A Field: 0x0
      - .... .... .... 0001 = Precedence field: 0x1
    - Metric Length: 36
    - Node State and Attribute Object: 0x0000
      - 0000 0000 .... .. = Reserved field: 0x00
      - .... .... 0000 00.. = Flags: 0x00
      - .... .... .... ..0. = Flag A: Not set
      - .... .... .... ....0 = Flag O: Not set
    - Node State and Attribute Optional TLV: 1
      - Node State and Attribute Optional TLV Type: 1
      - Node State and Attribute Optional TLV Length: 32
      - Raw Data: fe800000000000000204000400040004fe8000000000000000...
```

32 bytes = 2 IPv6 addresses

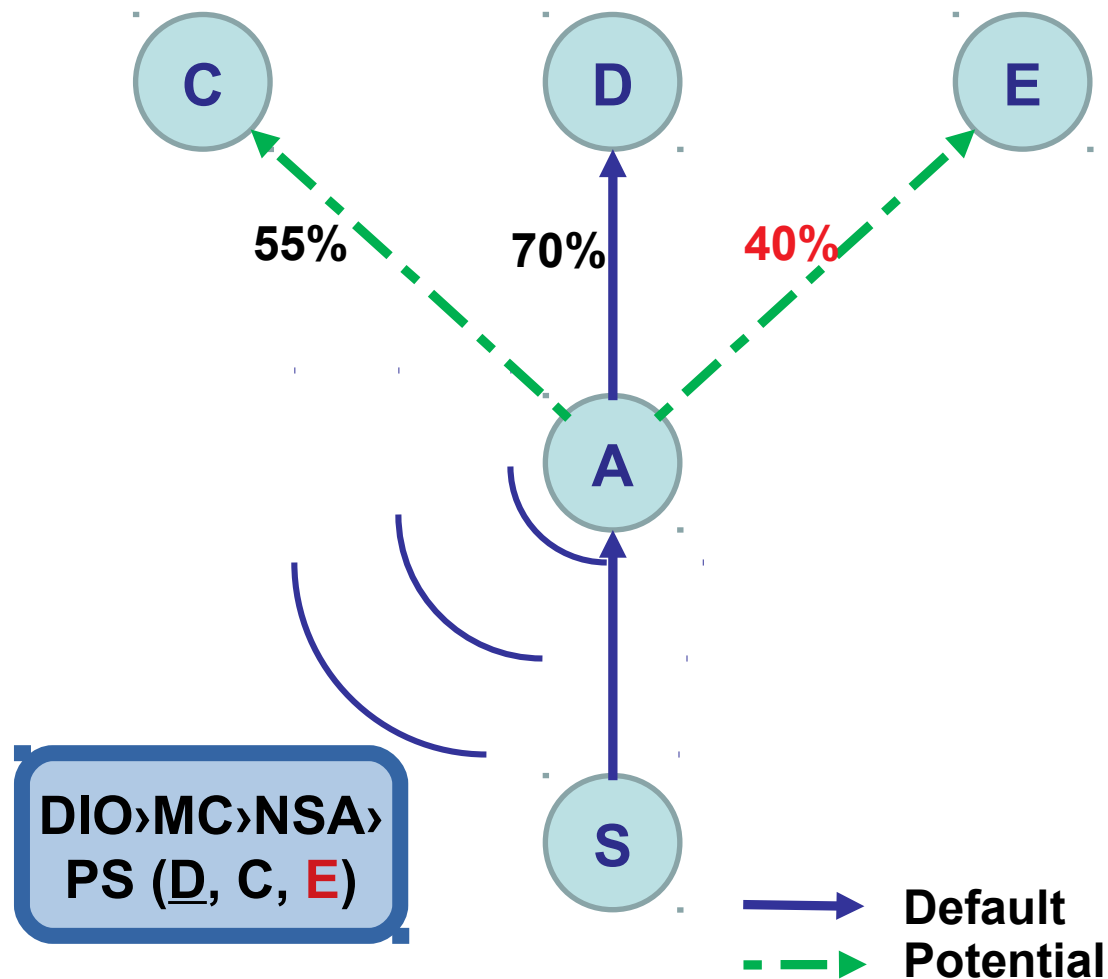
1. Solution: IPv6 Address Compression

- **Use 6LoWPAN Routing Header (6LoRH) (RFC8138)***
 - PS IPv6 addresses
 - By definition within same DODAG
 - **SHOULD Use Source Routing Header 6LoRH method**
 - As little as 2 bytes per address
- **Work in progress: To be implemented in Contiki OS**

* Thanks to Michael Richardson for the pointer

2. Issue: Bad parents

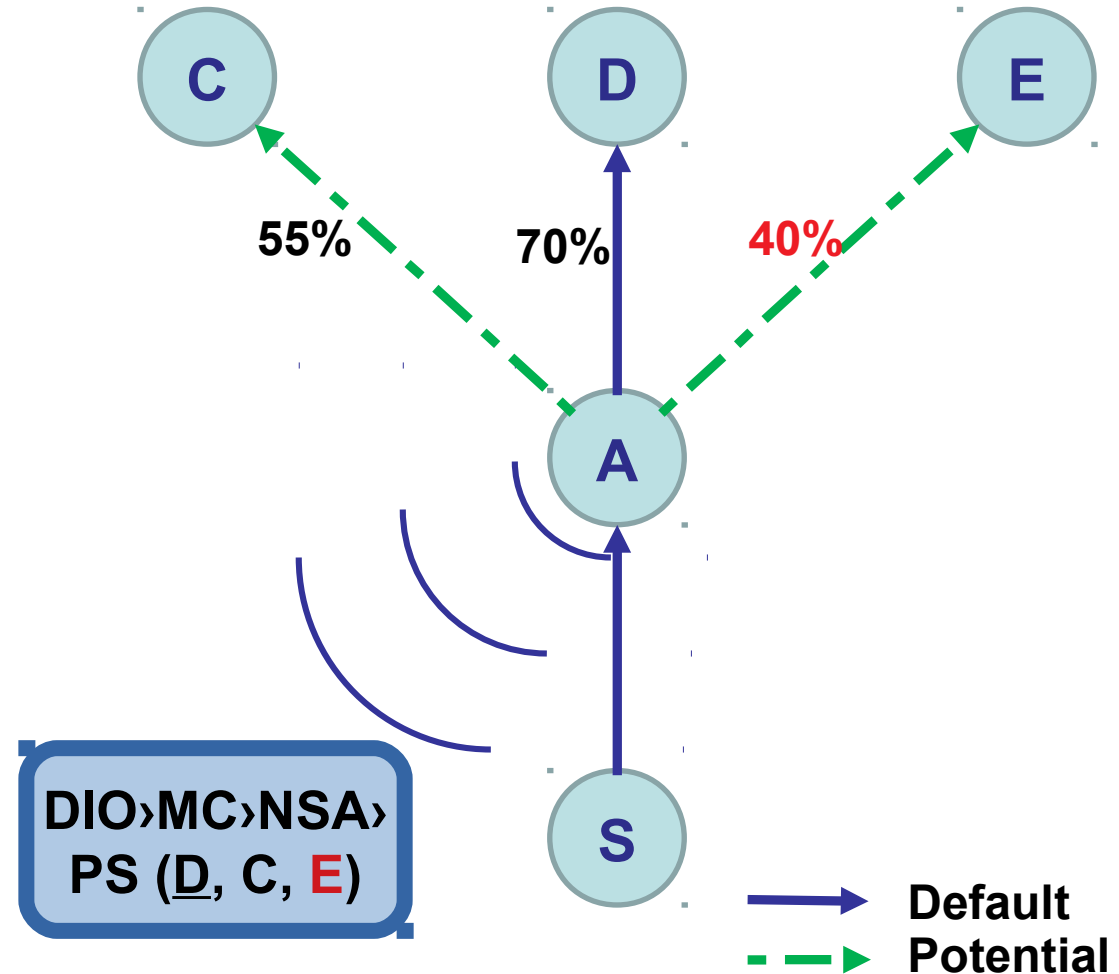
- Use of PS is independent of metric used
 - PS is a constraint
- Possible to announce *bad* parents



2. Solution: Additional Constraint

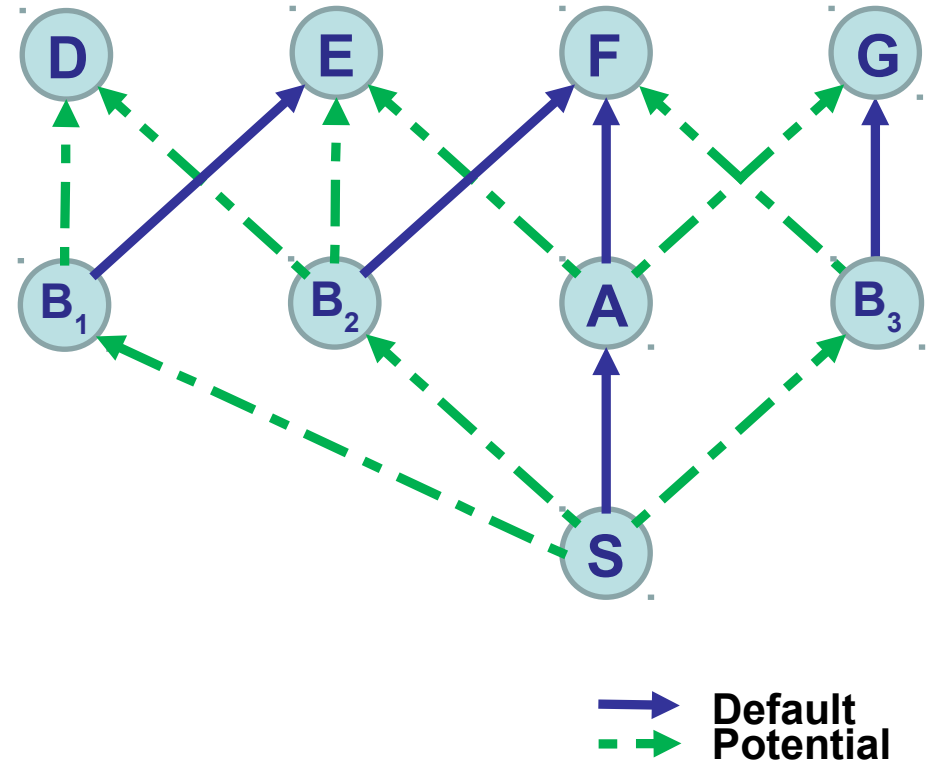
- Avoid selecting *bad* parents just because of common ancestor
 - Add constraint on metric*
 - Filter out, e.g. low ETX parents ($\geq 40\%$)
 - Also, set PS in OF preference order

* Thanks to Oana Iova for the idea



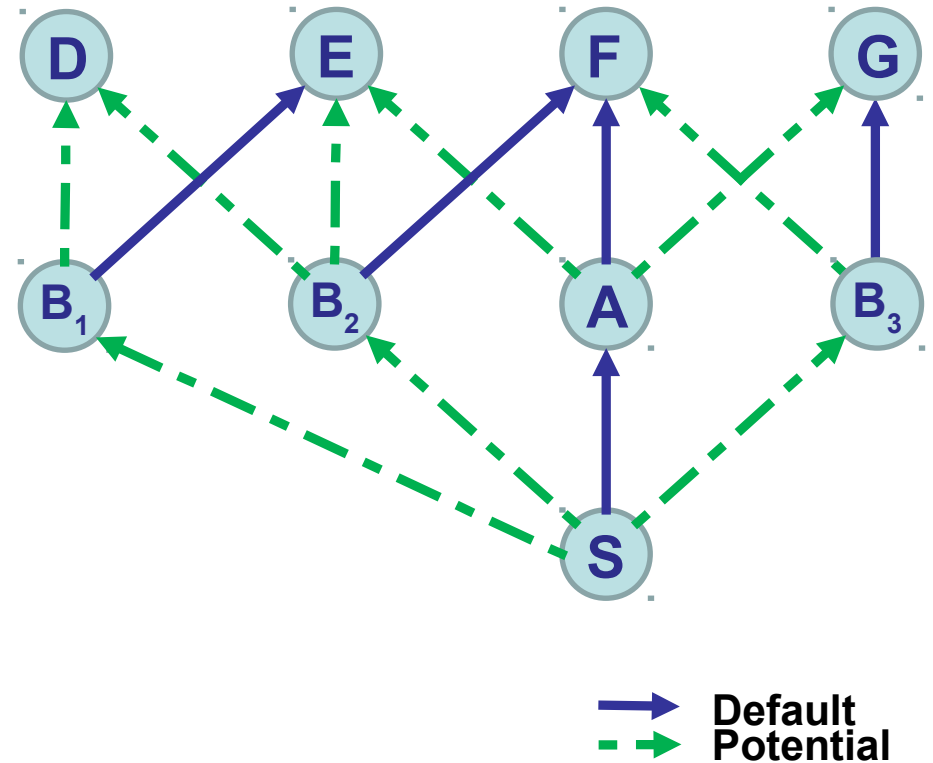
3. Issue: Flooding

- At least 3 ways to implement constraint
 - “Strict”
 - $PP(PP) = PP(AP)$
 - $PP(A) = F$
 - $PP(B_1) = E$ ✘
 - $PP(B_2) = F$ ✔
 - $PP(B_3) = G$ ✘



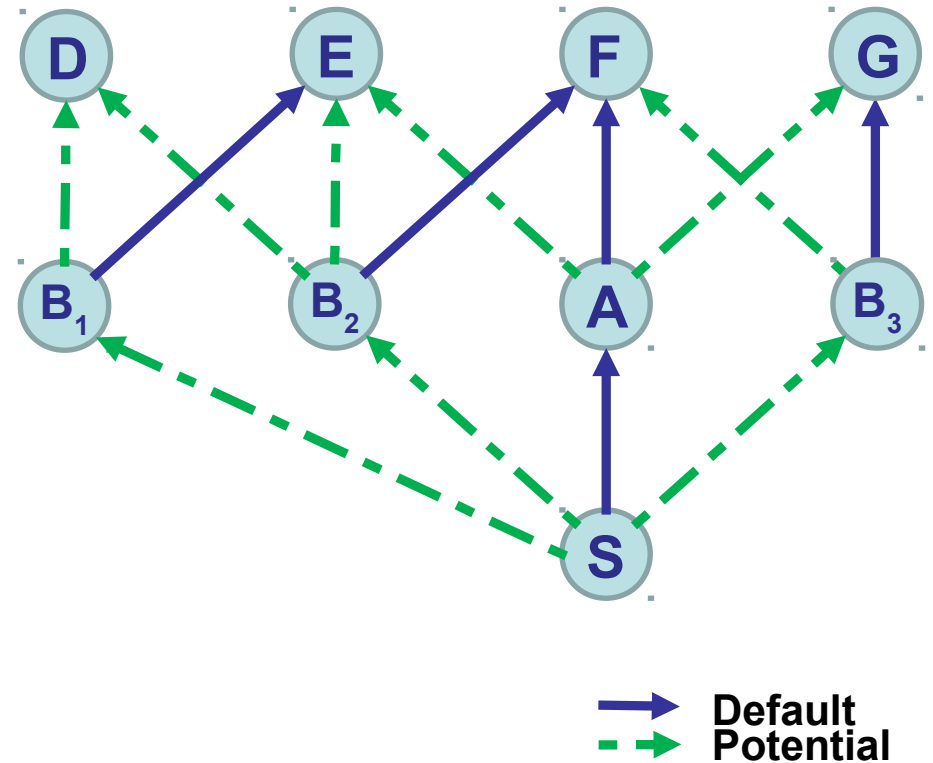
3. Issue: Flooding

- At least 3 ways to implement constraint
 - “Strict”: $PP(PP) = PP(AP)$
 - “Medium”
 - $PP(PP) \in PS(AP)$
 - $PP(A) = F$
 - $PS(B_1) = (\underline{E}, D) \times$
 - $PS(B_2) = (\underline{E}, D, E) \checkmark$
 - $PS(B_3) = (\underline{G}, F) \checkmark$



3. Issue: Flooding

- At least 3 ways to implement constraint
 - “Strict”: $PP(PP) = PP(AP)$
 - “Medium”: $PP(PP) \in PS(AP)$
 - “Relaxed”
 - $PS(PP) \cap PS(AP) \neq \emptyset$
 - $PS(A) = (\underline{E}, E, G)$
 - $PS(B_1) = (\underline{E}, D) \checkmark$
 - $PS(B_2) = (\underline{E}, D, E) \checkmark$
 - $PS(B_3) = (\underline{G}, F) \checkmark$



3. Solution: Control Divergence

- **“Relaxed” leads to flooding**
 - **Experiments published in AdHoc-Now 2018***
 - **Use “Medium” or “Strict” versions**
 - **Control “divergence” caused by overhearing**
 - **Use PS of children in parents**
 - **Add parameter:**
 - **nth first parents in PS: Forward when overhearing**
 - **Rest: Drop!**

* [Thanks to Tomás Lagos Jenschke](#)

Feedback

- **Road forward**
 - Presented during the last 3 IETF meetings
 - **draft-koutsiamanis-roll-nsa-extension-03** is on the way
 - We received and addressed comments
 - **Thanks to Derek, Michael, Oana, Tomas**
 - **1 Journal: IEEE Transactions on Industrial Informatics**
 - **2 Conferences: IEEE ICC 2017, AdHoc-Now 2018**
 - **1 Poster: ACM EWSN 2017**
 - **1 Dependability Competition: ACM EWSN 2017**
- **Path to adoption by WG?**

Thanks!

Questions?

ROLL@IETF102

Traffic-Aware Objective Function

draft-ji-roll-traffic-aware-objective-function-01

Chenyang Ji

Remous-Aris Koutsiamanis aris@ariskou.com

Georgios Z. Papadopoulos

Diego Dujovne

Nicolas Montavont

ROLL@IETF102

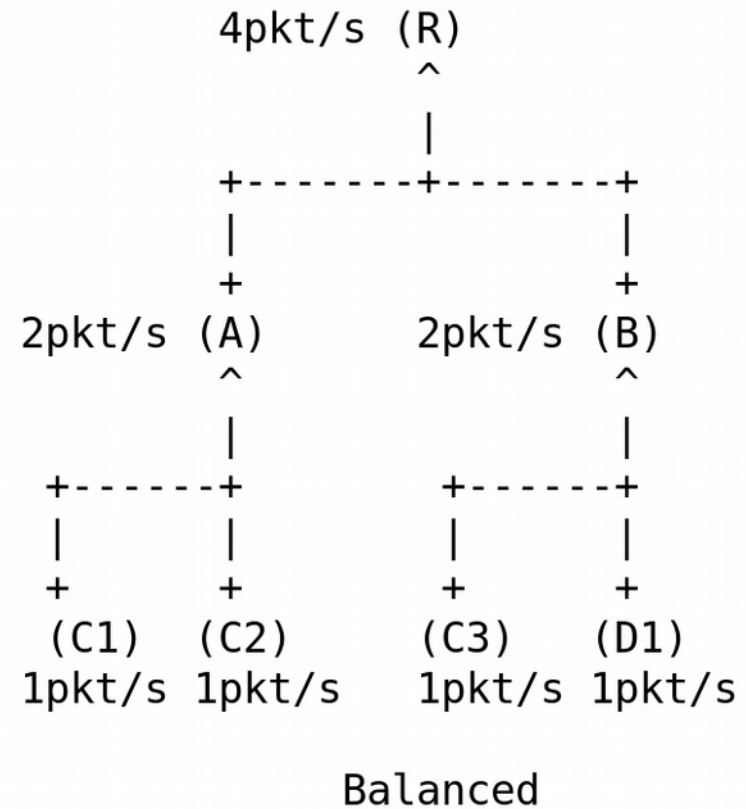
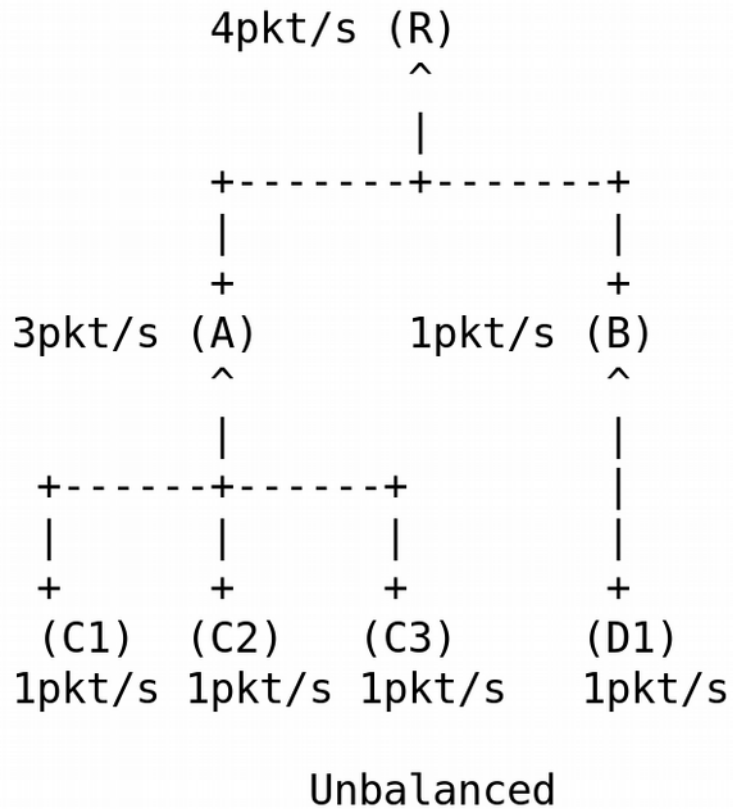
Standardisation Efforts

- **Objective Function → Preferred Parent**
 - OF0
 - MRHOF
 - *Load balanced OF (LB-OF)*

Problem statement

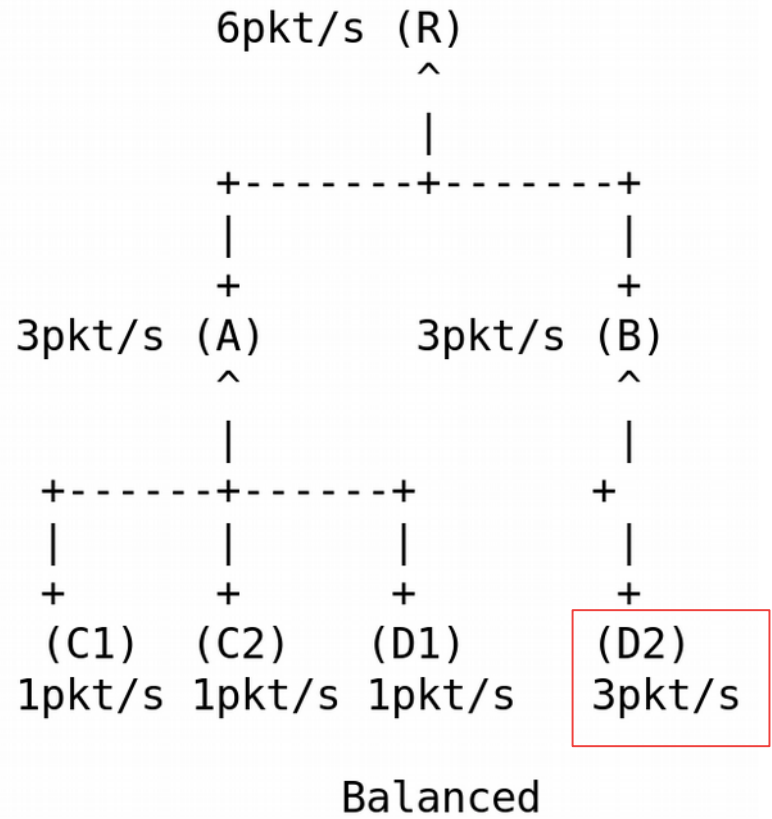
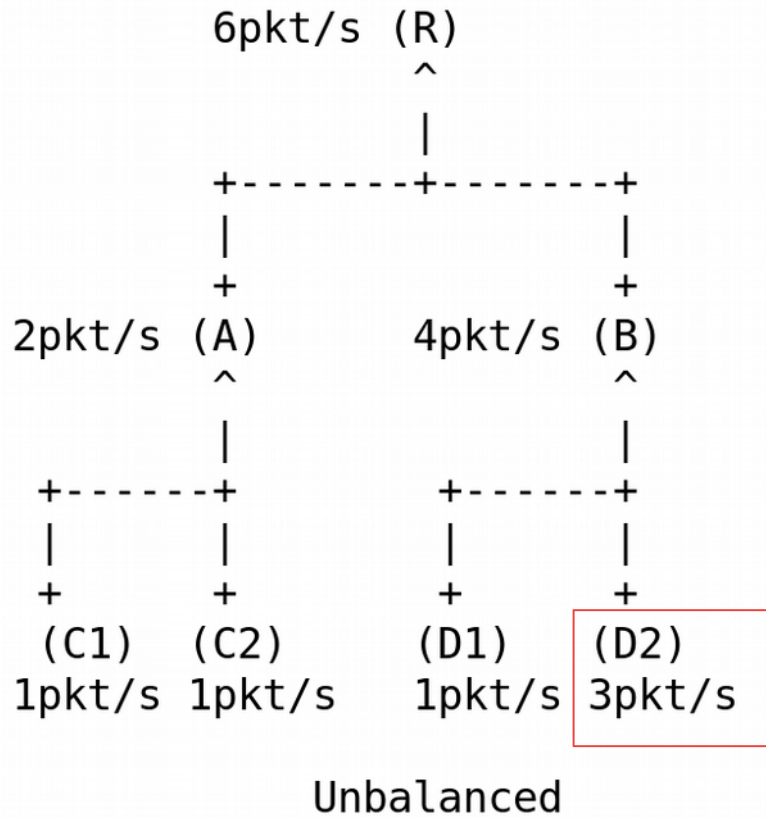
- **Using standard OFs (OF0, MRHOF) leads to unbalanced network:**
 - **Some nodes overloaded (forwarding)**
 - **Lower network and node lifetime**
 - **Higher packet losses (queueing)**
 - **Higher packet delay (queueing)**

Examples (1)



Nodes with same TX requirements

Examples (2)



Nodes with different TX requirements

Traffic-Aware OF

- **New metric**
 - **Packet Transmission Rate (PTR) per node**
 - **Alternatively, cumulative**
 - **Data packets sent per time unit**
 - **Alternatively, octets per time unit?**
- **Traffic-Aware OF**
 - **Least PTR → Preferred parent**
 - **Recommended use as with MRHOF**
- **Preliminary results presented at IETF 101**

DIO Format Example

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
RPLInstanceID										Version Number										Rank											
G	o	MOP				Prf				DTSN										Flags					Reserved						
DODAGID																															
DAGMC Type (2)										DAGMC Length																					
DAG Metric Container data																															

Packet Transmission Rate

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 (TBD1)										Res Flags					P	C	O	R	A	Prec					Length (bytes)						
Packet Transmission Rate (PTR)																															

- **Node Metric Object (PTR)**
 - 2 octets – unsigned integer

Issues (1)

- **Neighbour Table size limitations**
 - **Contiki uses 8 by default**
 - **Options if parent has full neighbour table**
 - **Parent: DAO-ACK with rejection Status**
 - **Use with/integrate Child Count Metric***

***draft-hou-roll-rpl-parent-selection-00**

Issues (2)

- **Relation with wider issues**
 - **Parent Selection**
 - **Load Balancing**
- **Related upcoming work:**
 - **DODAG selection & balancing**
 - **Reuse of functionality / Modularity**

Issues (3)

- **Aspects of network lifecycle**
 - **Join to network (EB)**
 - *draft-richardson-6tisch-enrollment-enhanced-beacon-01*
 - **Select RPL instance (DIO)**
 - **Select DODAG (DIO)**
 - *draft in progress*
 - **Select Preferred Parent (DIO)**
 - *draft-ji-roll-traffic-aware-objective-function-01*
- **Position in lifecycle**
 - **Reuse of functionality / Modularity**

Thanks!
Questions?

ROLL@IETF102

ROLL working group

YANG MPL model
draft-ietf-roll-mpl-yang-02

Peter van der Stok

Recent changes

YANG doctor review; thanks to Radi Krejci

- Added a section on Network Management Data Architecture (NMDA)
- Updated the Description texts
- Added units where appropriate
- Three modules (ops, seeds, statistics) “augment” domain module
- Assigned SIDs to YANG identifiers to reduce payload

SID: YANG Schema Item iDentifier

TODO

More reviews

Augmented modules should relate to domain instances of domain module

WGLC

REMINDER MODULE contents

Domain spans MC addresses
MC addresses are assigned to interfaces

```
+--rw mpl-domain
  +--rw domains* [domainID]
    +--rw domainID          uint16
    +--rw MClist*          yang:ipv6-address
  +--rw addresses* [MCaddress]
    +--rw MCaddress        yang:ipv6-address
    +--rw interfaces*     string
```

* : list

[key] : key attribute(s) of list

Operational parameters

Per domain assign MPL parameters

```
+--rw mpl-op
  +--rw SE_LIFETIME                               uint16
  +--rw PROACTIVE_FORWARDING                     boolean
  +--rw SEED_SET_ENTRY_LIFETIME                 uint64
  +--rw mpl-parameter* [domainID]
    +--rw domainID                               uint16
    +--rw DATA_MESSAGE_IMIN                    uint16
    +--rw DATA_MESSAGE_IMAX                    uint16
    +--rw DATA_MESSAGE_K                       uint16
    +--rw DATA_MESSAGE_TIMER_EXPIRATIONS       uint16
    +--rw CONTROL_MESSAGE_IMIN                  uint16
    +--rw CONTROL_MESSAGE_IMAX                  uint16
    +--rw CONTROL_MESSAGE_K                     uint16
    +--rw CONTROL_MESSAGE_TIMER_EXPIRATIONS     uint16
    +--rw MC_address*                            yang:ipv6-address
```

Operational statistics

Per domain and seed progress in packets

```
+--ro mpl-seeds* [seedID, domainID]
  +--ro seedID          uint64
  +--ro domainID        uint16
  +--ro local           boolean
  +--ro life-time       uint64
  +--ro min-seqno       uint8
  +--ro data-number     uint8
  +--ro control-number  uint8
  +--ro nr-of-timers    uint8
  +--ro seed_timers* [seqno]
    +--ro seqno         uint8
    +--ro I             uint8
    +--ro c             uint8
    +--ro e             uint8
    +--ro t             uint8
```

Operational statistics

Per domain and seed, MPL counters

```
+--ro mpl-statistics* [seedID, domainID]
  +--ro seedID                               uint64
  +--ro domainID                             uint16
  +--ro c-too-high                           uint64
  +--ro nr-forwarded                         uint64
  +--ro nr-not-forwarded                     uint64
  +--ro nr-of-messages-received              uint64
  +--ro nr-of-copies-received                uint64
  +--ro nr-of-messages-forwarded            uint64
  +--ro nr-of-copies-forwarded              uint64
  +--ro nr-of-refused                        uint64
  +--ro nr-of-notreceived                    uint64
  +--ro nr-of-missing                         uint64
  +--ro nr-of-inconsistent-data              uint64
  +--ro nr-of-consistent-data                uint64
  +--ro nr-of-inconsistent-control          uint64
  +--ro nr-of-consistent-control            uint64
```

Routing for RPL Leaves

draft-thubert-roll-unaware-leaves-05

Pascal Thubert

IETF 102

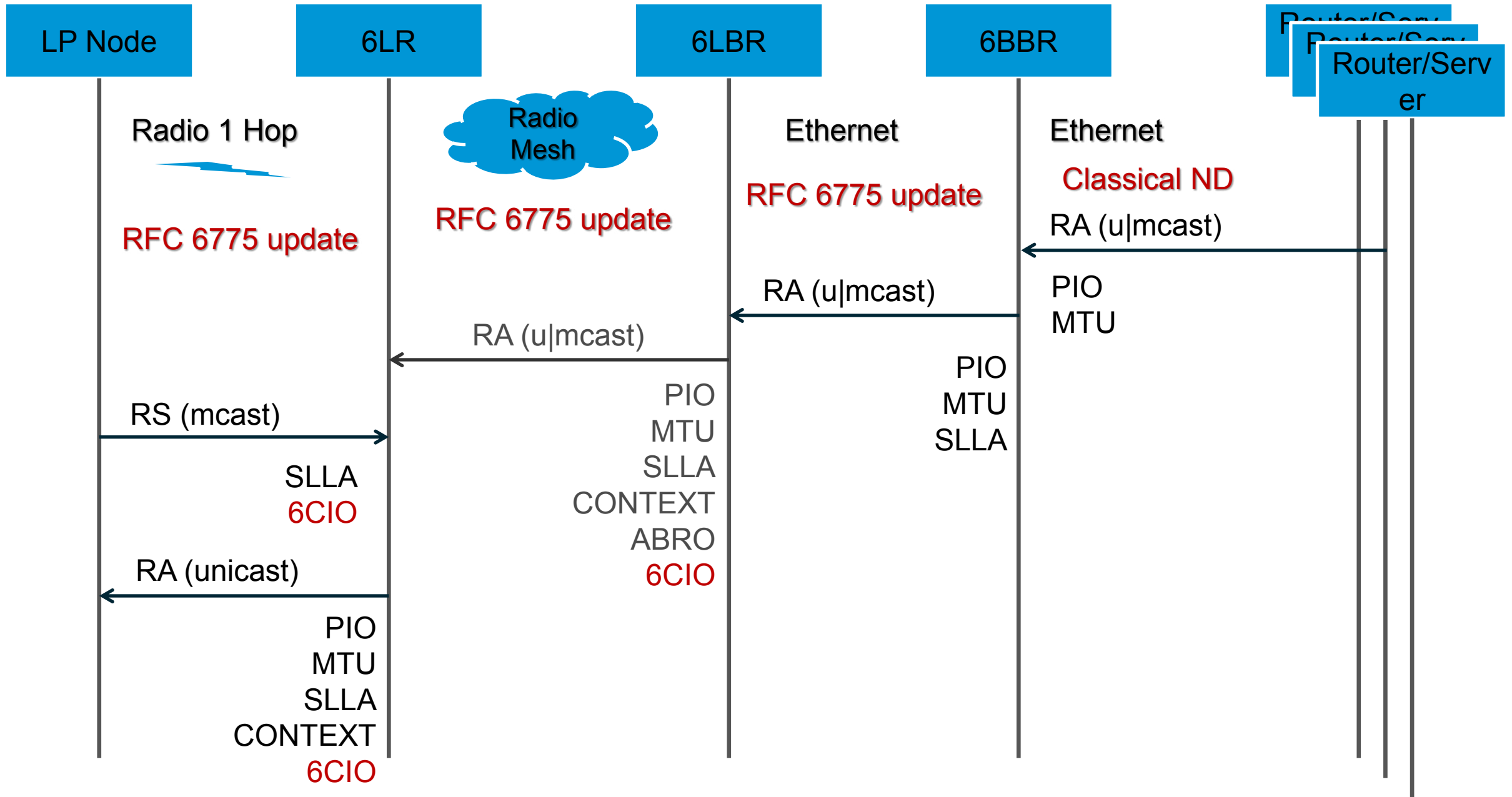
Montreal

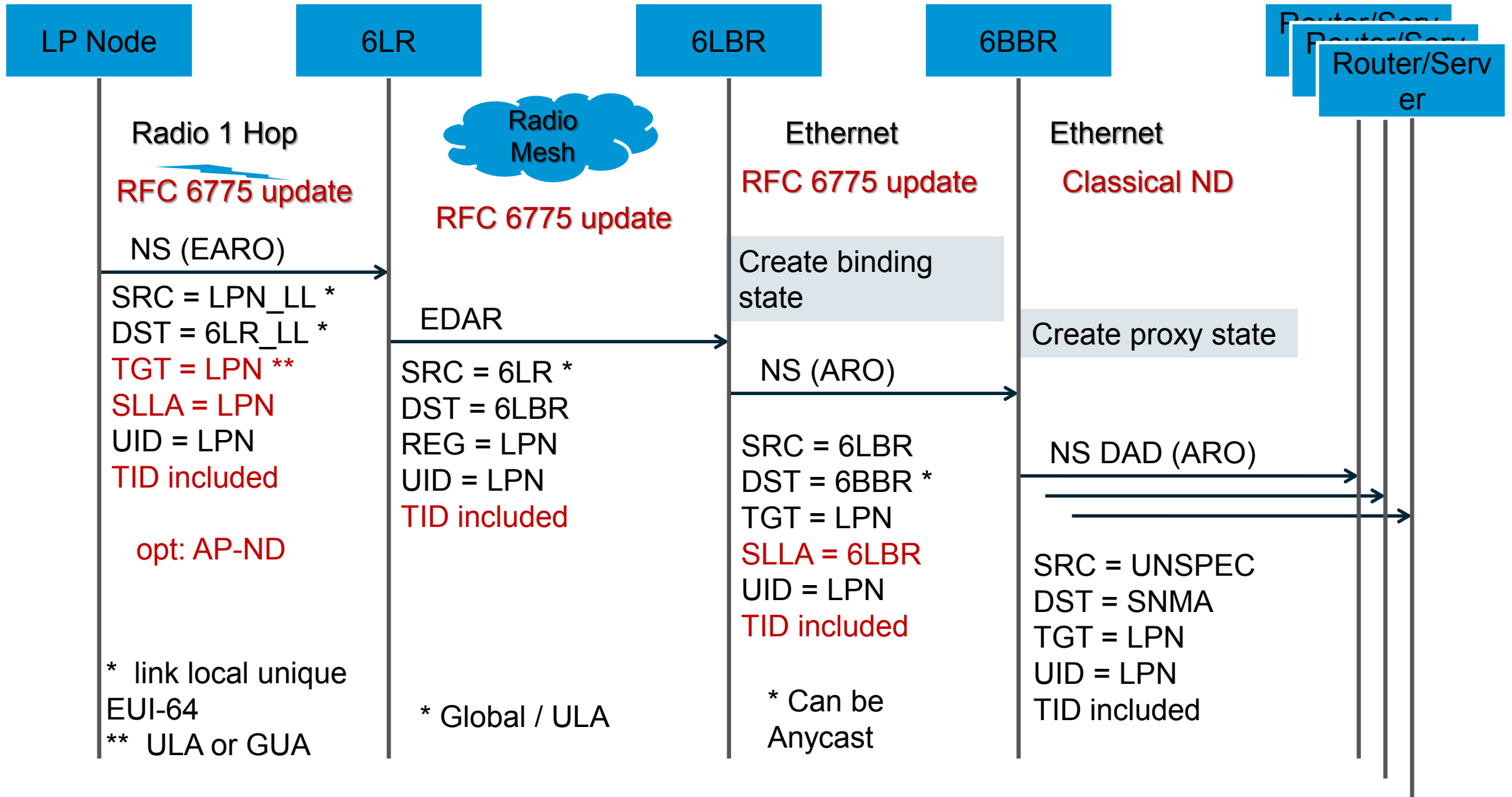
Unmet expectations

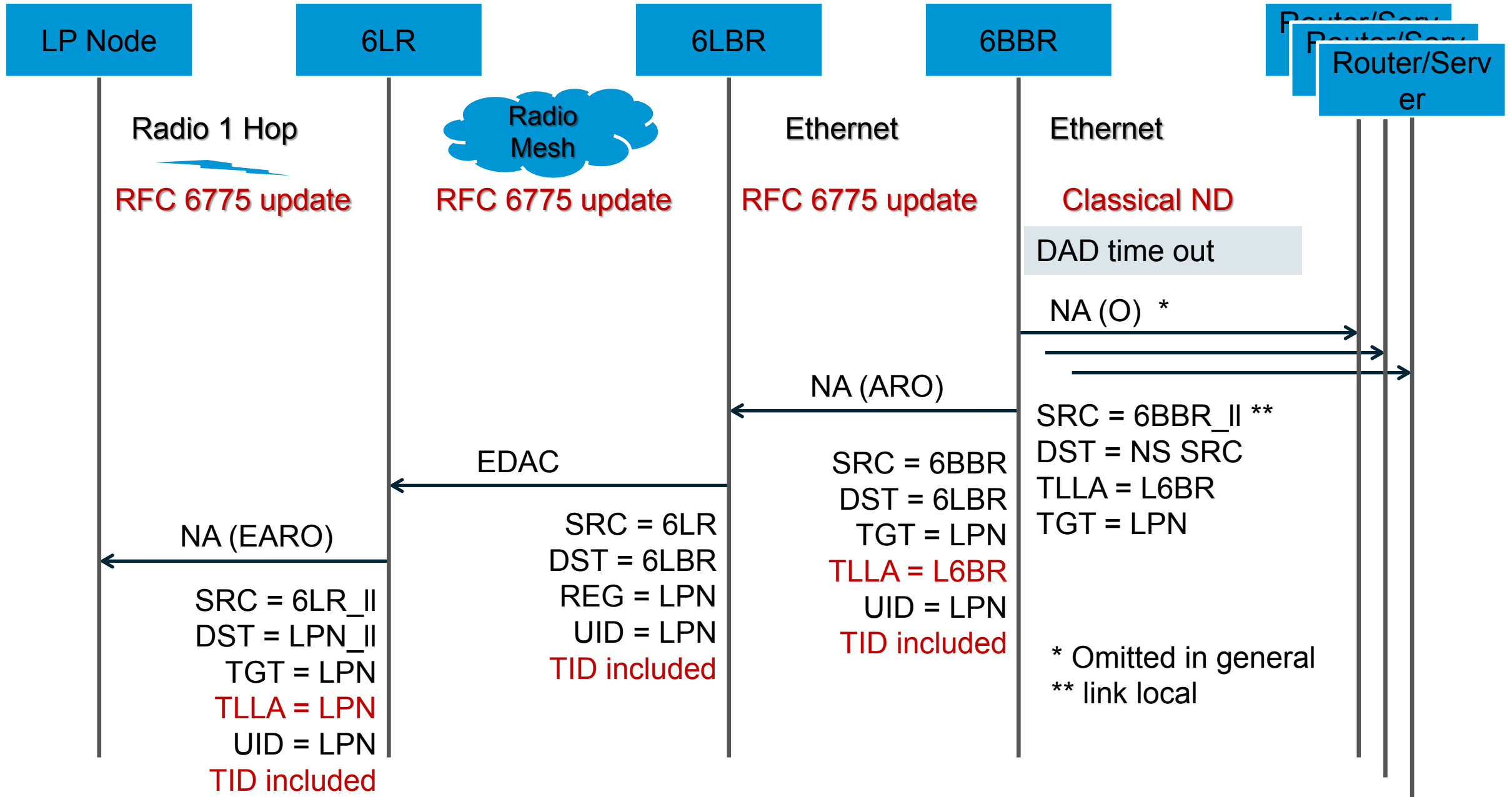
- Connectivity for a Non-RPL aware node in a RPL domain
Forwarding is described but not the control plane
- Integration of the EDA Exchange (EDAR/EDAC) used as keep-alive with the RPL signaling to avoid duplication
At the moment both are needed periodically This spec uses a common lifetime and the EDA exchange is proxied
- Separation of the RPL Root and the 6LBR and proxy registration to the 6BBR
The RPL root proxies the EDA with the 6LBR and the NS(EARO) with the 6BBR

RFC 6775 Update

P.Thubert, E. Nordmark, S. Chakrabarti, C. Perkins







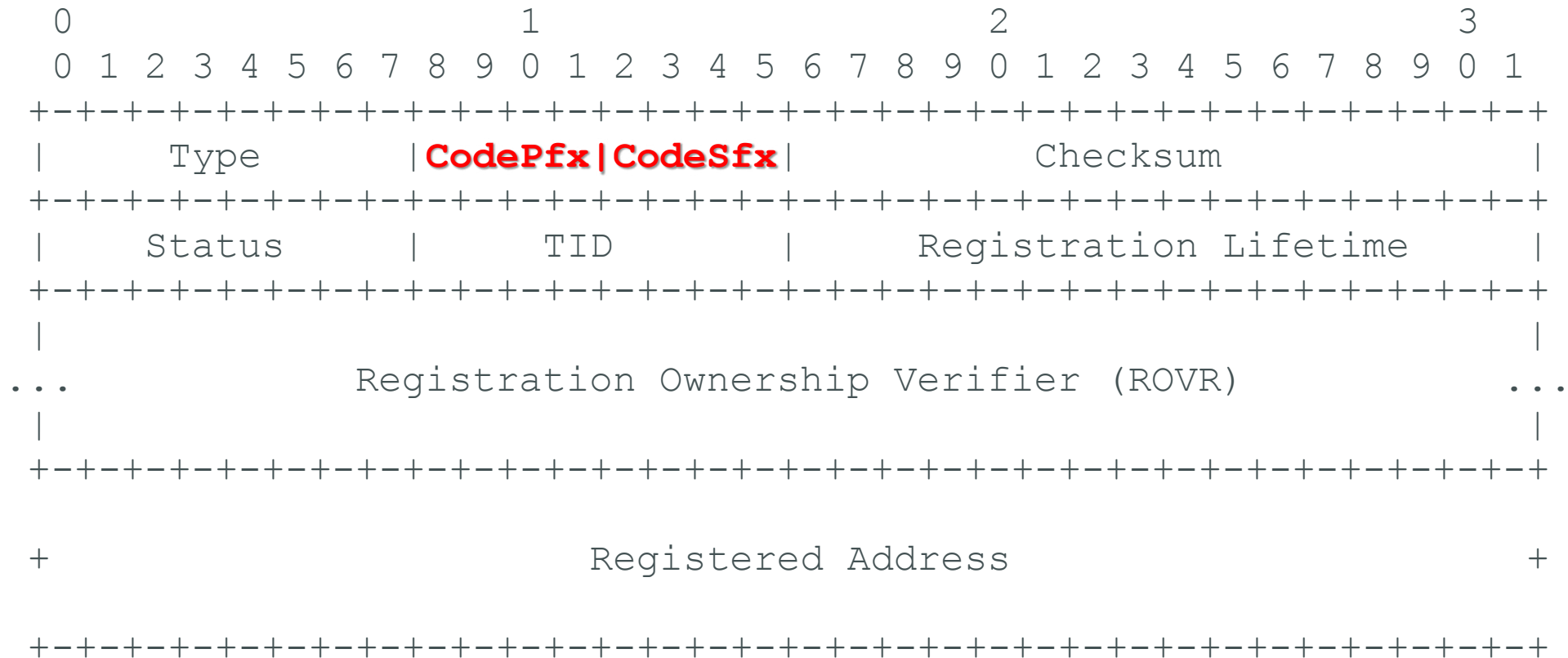
Charlie Perkins

- As an author and native speaker, Charlie made a final pass on the language and the organization
- Found that text was repeated, other was scattered
- Fixed the language, regrouped items
- E.g., took functional text out of the definition, to appropriate section
- Also removed extraneous references
- Work happened over draft 19-21

Issue 1: EDAR / EDAC extensibility

- The size of the ROVR was inferred from the size of the message
- Did not leave a possibility to insert options
- This might be desirable in the future, e.g., MAC Address option for a MAP server
- Long discussion, tried multiple possibilities
- Ended up with split ICMP Code, similar to what we discussed with Adrian Farrell
- Added in draft -20

RFC 6775 update new features: ICMP code split



Code: The ICMP Code [RFC4443] for Duplicate Address Messages is split in two 4-bit fields, the Code Prefix and the Code Suffix.

RFC 6775 update new features: ICMP code split

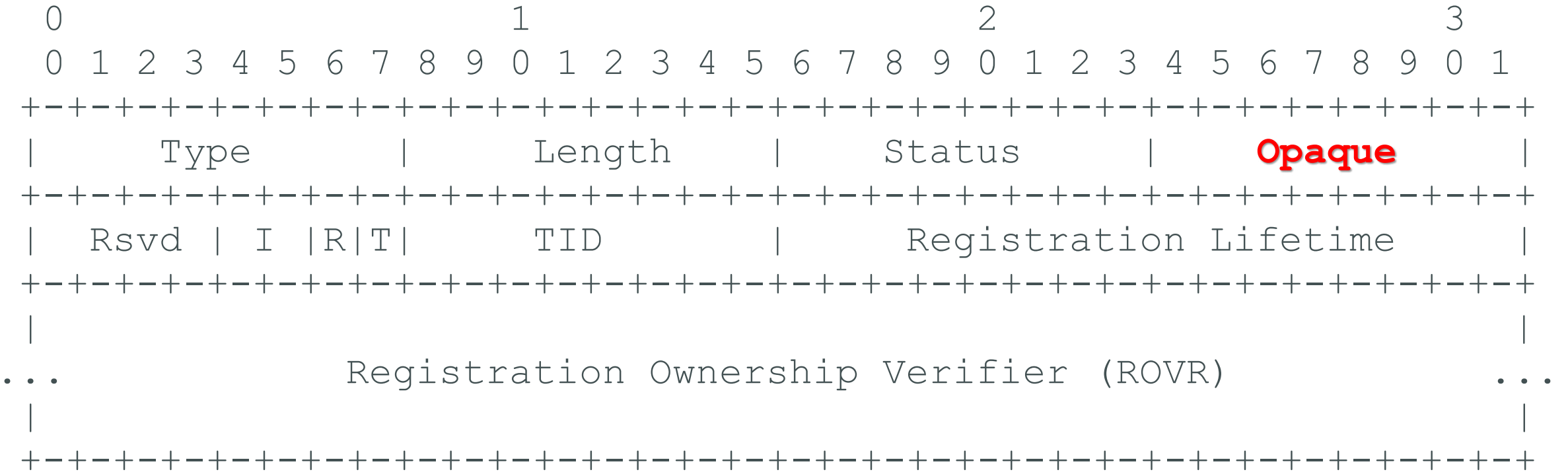
Code:

The ICMP Code [RFC4443] for Duplicate Address Messages is split in two 4-bit fields, the Code Prefix and the Code Suffix. The Code Prefix MUST be set to zero by the sender and MUST be ignored by the receiver. A non-null value of the Code Suffix indicates support for this specification. It MUST be set to 1 when operating in a backward-compatible mode, indicating a ROVR size of 64 bits. It MAY be 2, 3 or 4, denoting a ROVR size of 128, 192, and 256 bits, respectively.

Issue 2: Enabling Other Routing Registrars

- 6BBR is only one possible routing registrar. Others include RPL [I-D.thubert-roll-unaware-leaves] and RIFT [I-D.ietf-rift-rift]
- Resolution to use a generic term as opposed to mention 6BBR specifically
- Also allow an opaque field. RPL uses it for instance ID.
- Added in draft -19
- Generalization to the term « routing registrars » in -21

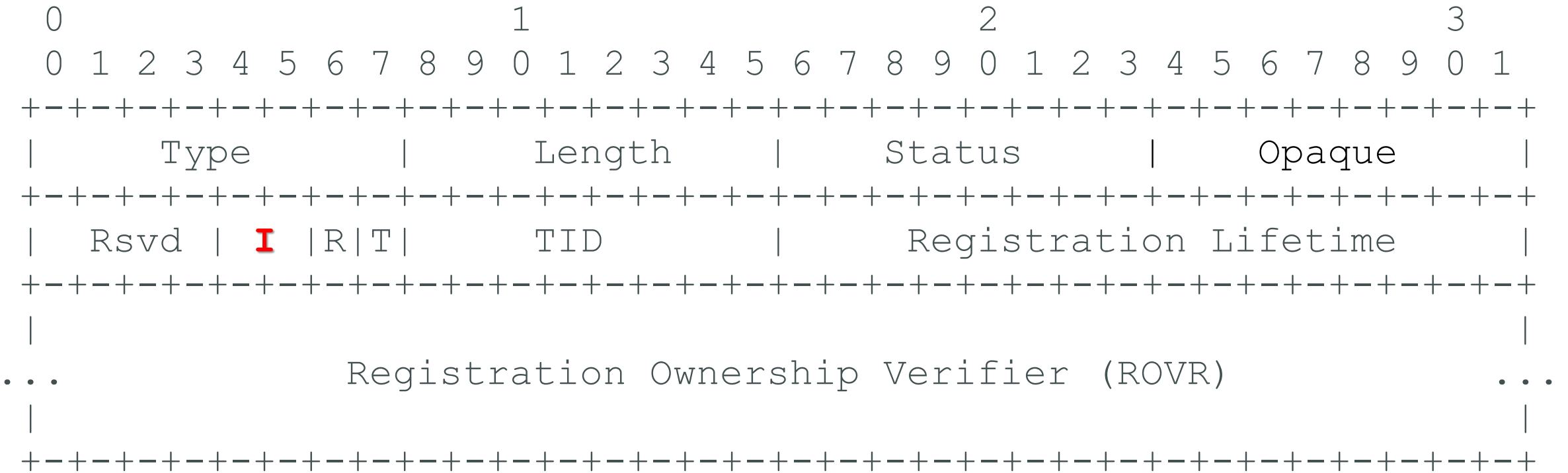
RFC 6775 update new features: the Opaque field



Opaque:

An octet opaque to ND; the 6LN MAY pass it transparently to another process. It MUST be set to zero when not used.

RFC 6775 update new features: the I field



I: Two-bit Integer: A value of zero indicates that the Opaque field carries an abstract index that is used to decide in which routing topology the address is expected to be injected.

draft-thubert-roll-unaware- leaves

P.Thubert

IETF 102

Montreal

Terminology

- RFC 6550:
 - A RPL leaf may understand RPL
 - But does not Act as a router
- This draft: A RPL-unaware leaf does not implement anything specific to RPL, but it **MUST** support draft-rfc6775-update

Notes on the 'R' flag (defined in draft-rfc-6775-update)

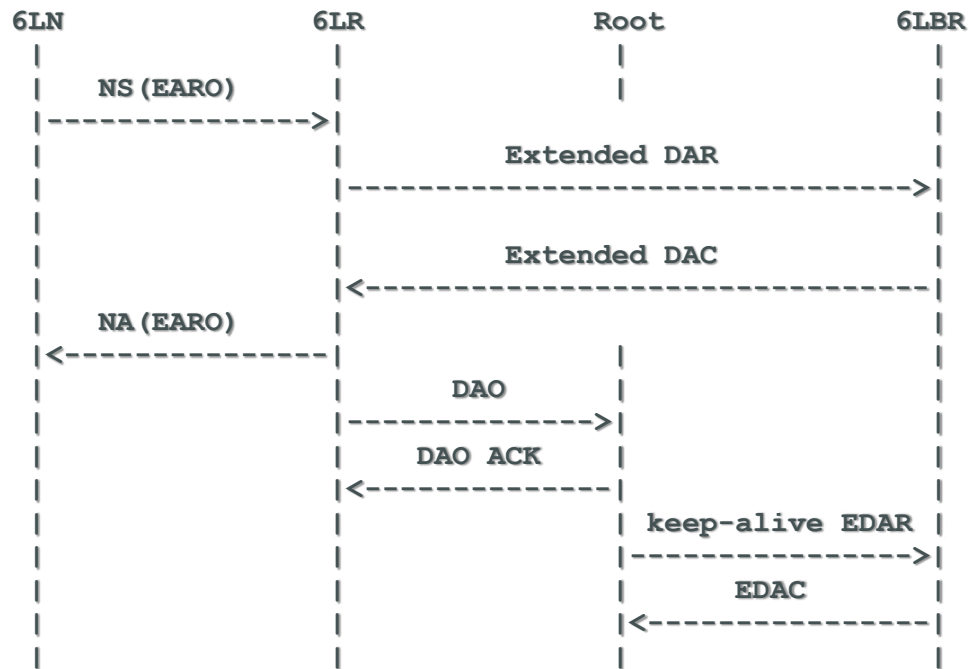
- A RPL Unaware Leaf does not know that there is routing in place and that the routing is RPL; draft-thubert-roll-unaware-leaves does not require anything from the Leaf.
- draft-rfc-6775-update specifies a new flag in the EARO, the 'R' flag.
- If the 'R' flag is set, the Registering Node expects that the 6LR ensures reachability for the Registered Address, e.g., by means of routing or proxying ND.
- Conversely, when it is not set, the 'R' flag indicates that the Registering Node is a router, which for instance participates to RPL and that it will take care of injecting its Address over the routing protocol by itself.
- A 6LN that acts only as a host, when registering, **MUST** set the 'R' to indicate that it is not a router and that it will not handle its own reachability.
- A 6LR that manages its reachability **SHOULD NOT** set the 'R' flag; if it does, routes towards this router may be installed on its behalf and may interfere with those it injects.

Mapping Fields from RPL DAO to NS(EARO) and EDA

- The Registered Address in a RPL Target Option is a direct match to the Registered Address field of the EDAR message and in the Target field of the NS, respectively
- EARO's TID is a direct match to Path Sequence in Transit Information option (TIO)
- **NEW: EARO's opaque field carries the RPLInstanceID, 0 means 6LR's default**
- EARO's Lifetime unit is 60s. RPL uses Lifetime Units that is passed in the DODAG Configuration Option. Converting EARO to DAO and back requires mapping of units.
- The Registration Ownership Verifier (ROVR) field in keep-alive EDAR messages by the Root is set to 64-bits of all ones to indicate that it is not provided. It is obtained in the EDAC from the 6LBR and used in proxy registration.
 - Q: Should we carry it in a RPL option in DAO messages?

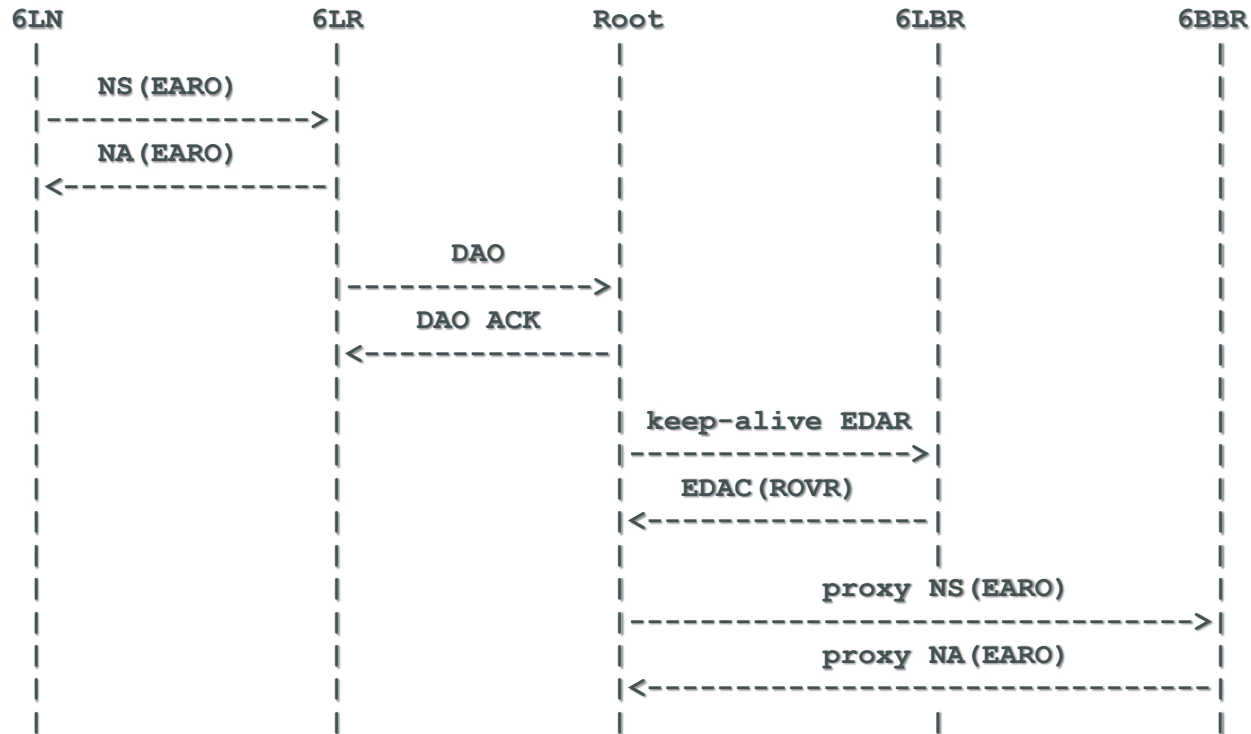
First registration

- Upon the first registration, the EDAR / EDAC populates a state in the 6LBR including the ROVR field and the 6LR sends a first DAO message.
- The RPL Root acts as a proxy on behalf of the 6LR upon the reception of the DAO propagation initiated at the 6LR. **Should we allow splitting from the 6LBR, e.g.:**



EDA (DAR, DAC) message Proxying

- Upon the renewal of a 6lowPAN ND registration: if the 'R' flag is set, the 6LR injects a DAO targeting the Registered Address, and refrains from sending a DAR message.
- With a Root/6LBR split that could give:



Discussion

- Should we force that the RPL root is 6LBR?

Open Mic