Wednesday 07/24 - 10:00-12:00

IETF 105 ROLL

Routing over Low-Power And Lossy Networks

Chairs:
Peter van der Stok
Ines Robles

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

- 10:00 -12:00 Wednesday Morning session I
- Remote Participation
  - Jabber Room: xmpp:roll@jabber.ietf.org?join
  - Meetecho: https://www.meetecho.com/ietf105/roll
  - Etherpad: https://etherpad.tools.ietf.org/p/notes-ietf-105-roll
  - Slides: http://datatracker.ietf.org/meeting/105/materials
- Jabber Scribe: **Please volunteer**
- Minutes taker: **MILLION THANKS** to Dominique :-)
- **Please sign blue sheets :-)**

Agenda

Wednesday, July 24, 2019 (EDT) Morning session I

10:00 - 10:05	Introduction	Ines/Peter
10:05 - 10:15	draft-ietf-roll-dao-projection	Pascal
10:15 - 10:25	draft-ietf-roll-unaware-leaves	Pascal
10:25 - 10:35	draft-ietf-roll-nsa-extension-04	Georgios
10:35 - 10:50	draft-rahul-roll-mop-ext	Rahul (remotely)
10:50 - 11: 00	draft-thubert-roll-turnon-rfc8138	Pascal
11:00 - 11: 55	Updates in ROLL BIER	Pascal - Carsten
11:55 - 12:00	OPEN FLOOR	All

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

DELAYED

Submitted

Progressing

State of Active Internet-Drafts

Draft	Status
draft-ietf-roll-aodv-rpl-07	Submitted to IESG for Publication
draft-ietf-roll-dao-projection-06	Discussion today
Draft-ietf-roll-forw-select-00 (Expired)	On hold
draft-ietf-roll-useofrplinfo-31	RFC Queue
Draft-ietf-roll-dis-modifications-00 (Expired)	To be continued
Draft-ietf-roll-mpl-yang-02 (Expired)	On hold
Draft-ietf-roll-bier-ccast-01 (Expired)	Bier-roll design team takes over
draft-ietf-roll-efficient-npdao-15	RFC Queue
draft-ietf-roll-rpl-observations-01	Used as model to develop further drafts
draft-ietf-roll-nsa-extension-04	Discussion today
draft-ietf-roll-unaware-leaves-02	Discussion today

Related Internet-Drafts

Draft	Status
draft-rahul-roll-mop-ext-01	Discussion today :-)
draft-thubert-roll-turnon-rfc8138-03	
draft-koutsiamanis-roll-traffic-aware-of-00	In Progress ?
draft-audeoudh-rpl-asymmetric-links-00	

Open tickets

Ticket	Summary	Component
#179	Security considerations for dao projection	dao-projection
#180	13 issues to address in dao projection draft (lifetime, MOP, retransmissions, route cleanup)	dao-projection
#187	New version of RFC6550 - Topics to include	rpl
#188	Should 6LBR be included into the DODAG root?	rpl
#189	RPL and new MOP?	dao-projection

https://trac.ietf.org/trac/roll/report/2

Root initiated routing state in RPL

draft-ietf-roll-dao-projection

Pascal Thubert

IETF 105

Montreal

Changes Highlights

- New flag in the RPI to indicate projected route
  - Needed for error processing
- Text on RFC 8138, Need for compression & RPI update

Updating the RPI (RFC 6550 & RFC 6553)

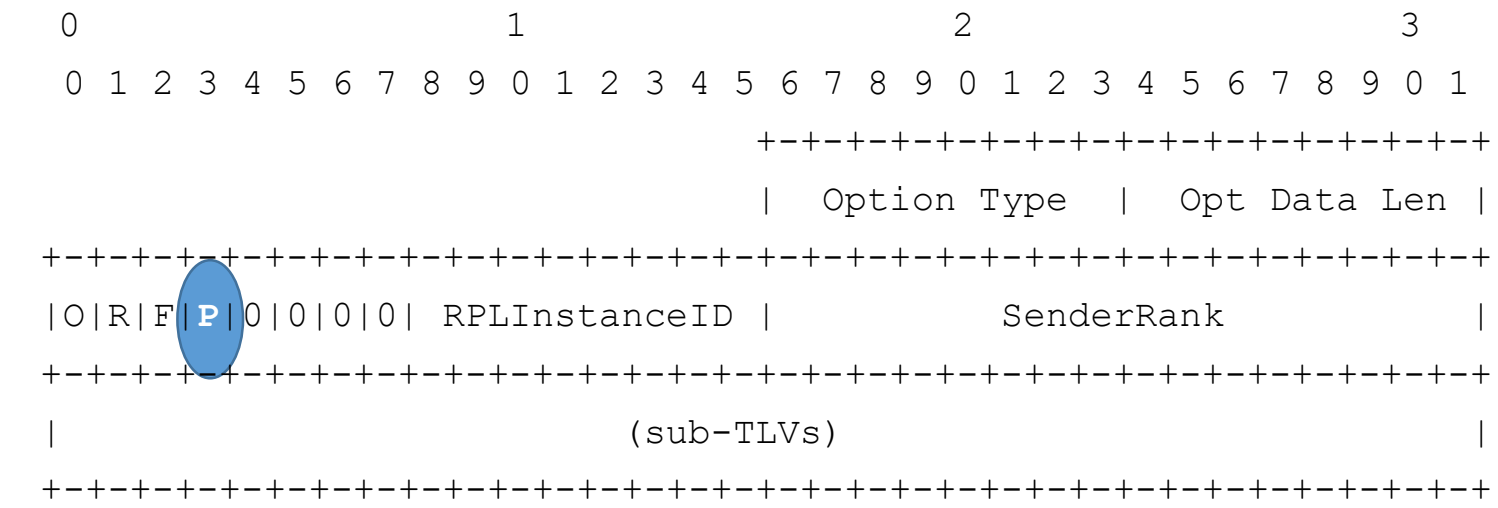


Figure 6: RPL Option

New fields:

P: 1-bit flag; indicates that the packet is routed along a projected route.

On RFC 8138 usage: ERPI

- RPI flags and Rank are useless along the projected route.
- On the other hand, we could use a new flag in the RPI to indicated that the route is a projected route in the non-storing case.
- Introduced an additional ERPI encoding that :
  - Is inspired from the existing RPI compression but is Elective
  - Does not contain a Rank information nor the flags
  - May Indicate projected route, another thread on that

On RFC 8138 usage: RPI (cont.)

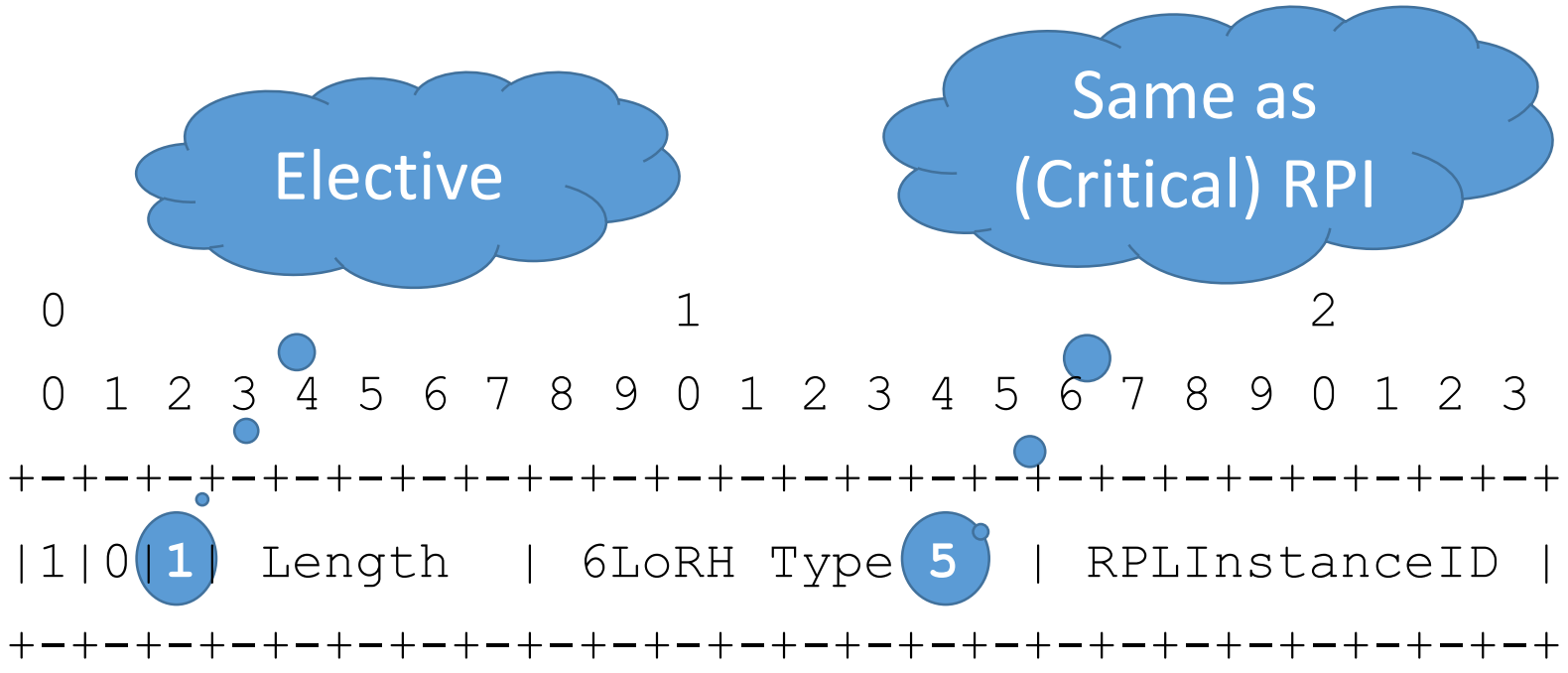


Figure 5: A ERPI-6LoRH carrying a RPLInstanceID

On RFC 8138 usage: IP in IP

- An implementation is used to
  - IP in IP with SRH from the root to the last node indicated in the SRH or
  - IP in IP with no SRH from the node indicated as encapsulator to (implicitely) the root.
- For DAO projection we will use an SRH from the node indicated as encapsulator to the last node indicated in the SRH, which is a mix of the above. It is compatible with RFC8138 but possibly untried with existing code.
- Provided an example packet that has IP in IP and SRH and none of the outer IPs is the root

On RFC 8138 usage: IP in IP (cont.)

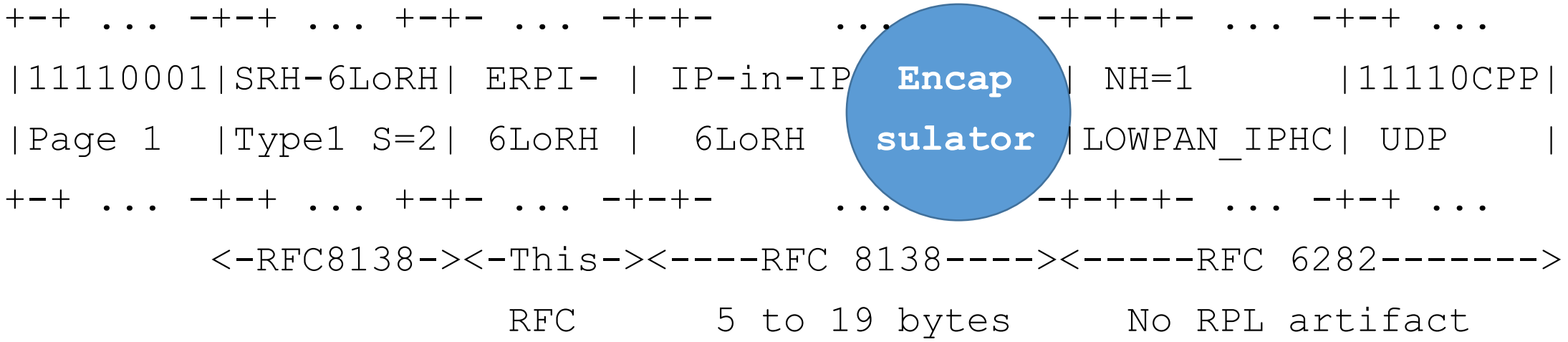


Figure 3: Example Compressed Packet with SRH.

Discussions needed to progress

- How is the topology known to the root?
  - Could use external management or Non-Storing DAO
  - Time to do the RPL control plane compression
  - Suggestion to add a peer info option similar to transit info option
  - Suggestion to use connected dominating set of RPL parent (Cisco IPR)
- How are the node capabilities known to the root?
  - Suggestion to add a node capability option (in mop ext? ref mop ext?)
- Complexity of mixed modes and route concatenation
- MOP saturation: addressed by draft-rahul-roll-mop-ext
- Compression of the Via Info option (so far full addresses)
  - Time to do the RPL control plane compression

Discussions needed to progress

- How is the topology known to the root?
  - Could use external management or Non-Storing DAO
  - Suggestion to add a peer info option similar to transit info option
  - Suggestion to use connected dominating set of RPL parent (Cisco IPR)
  - Time to do the RPL topology description draft
    (in ROLL, in RAW?)

Discussions needed to progress (cont.)

- How are the node capabilities known to the root?
  - Suggestion to add a node capability option
  - (in mop ext? or in this draft with a ref/dependency on mop ext?)
- Complexity of mixed modes and route concatenation
- MOP saturation: addressed by draft-rahul-roll-mop-ext
- Compression of the Via Info option (so far full addresses)
  - Time to do the RPL control plane compression

Routing for RPL Leaves

draft-ietf-roll-unaware-leaves

Pascal Thubert

IETF 105

Montreal

6lo standard work

A proactive setting of proxy/routing state to avoid multicast due to reactive Duplicate address detection and lookup in IPv6 ND

- RFC 8505 (Issued 11/2018)
  - The registration mechanism for proxy and routing services
  - Analogous to a Wi-Fi association but at Layer 3
- draft-ietf-6lo-backbone-router (WGLC complete 1/25)
  - Federates 6lo meshes over a high-speed backbone
  - ND proxy analogous to Wi-Fi bridging but at Layer 3
- draft-ietf-6lo-ap-nd (WGLC complete 3/26)
  - Protects addresses against theft (Crypto ID in registration)
- draft-thubert-6lo-unicast-lookup
  - Provides a 6LBR on the backbone to speed up DAD and lookup
- draft-thubert-6man-ipv6-over-wireless (new draft)
  - IPv6 ND vs. WiND applicability to wireless networks

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
  - Could be used to replace 6LBR with DHCPv6 server
  - Note: Missing TID in DAO to do a full proxy operation

Terminology

- RFC 6550:
  - A RPL leaf may understands 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 RFC 8505,
  - and it MUST ask the 6LR for abstract reachability services

Status

- Now WG document
- Forces IP in IP to the parent 6LR
  - If leaf does not support RPL artifacts or RFC 8138
  - It is preferred that leaf supports artifacts to save IP in IP in storing mode
  - In Non-Storing Mode, encapsulation to parent is ‘E’ flag.
- In Non-Storing Mode, Keep-alive EDAR nested in DAO/DAO-ACK
  - But not in Storing Mode since parent acknowledges before prop. DAO
- Work mostly complete. Anything missing?

ML questions: Should we discuss MPL in the draft?

1. The HbH option cannot be ignored because of the first bits
2. A MPL unaware leaf will drop the packet due to the HbH

If we want to serve MULs we need to either

- use a new value for the option (e.g., x6D => x2D)
- encapsulate IP in IP and each router that has unaware leaves broadcasts a decapsulated packet to its leaves
- add a capability bit with the registration

backup

Notes on the ‘R’ flag (defined in RFC 8505)

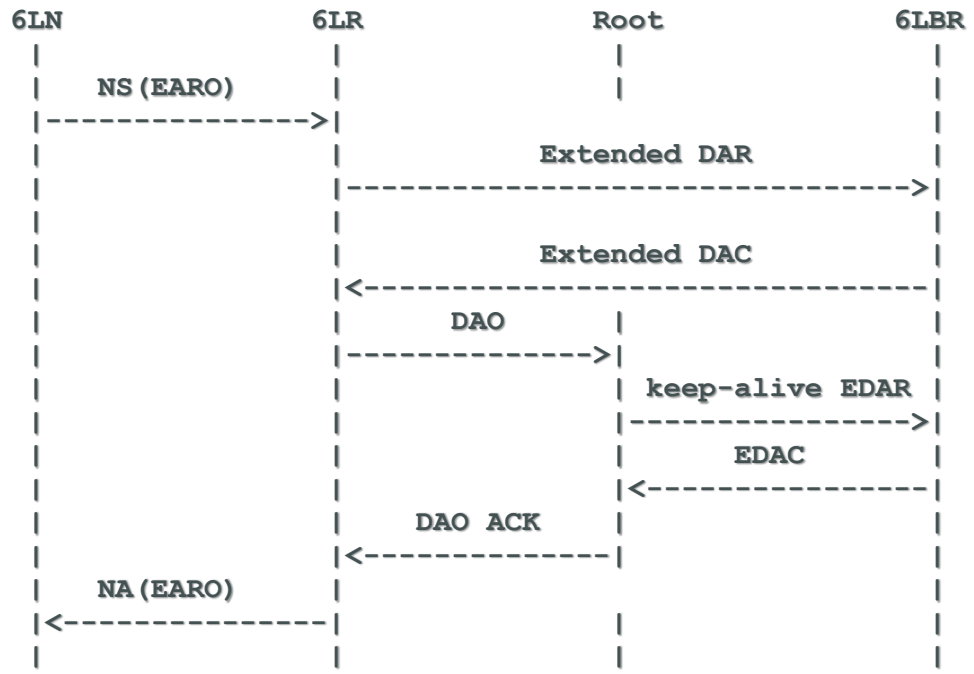
- A RPL Unaware Leaf does not know that there is routing in place and that the routing is RPL; draft-thubert-roll-unaware-leaves does not require anything from the Leaf.
- RFC 8505 specifies a new flag in the EARO, the 'R' flag.
- If the 'R' flag is set, the Registering Node expects that the 6LR ensures 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)
- 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?

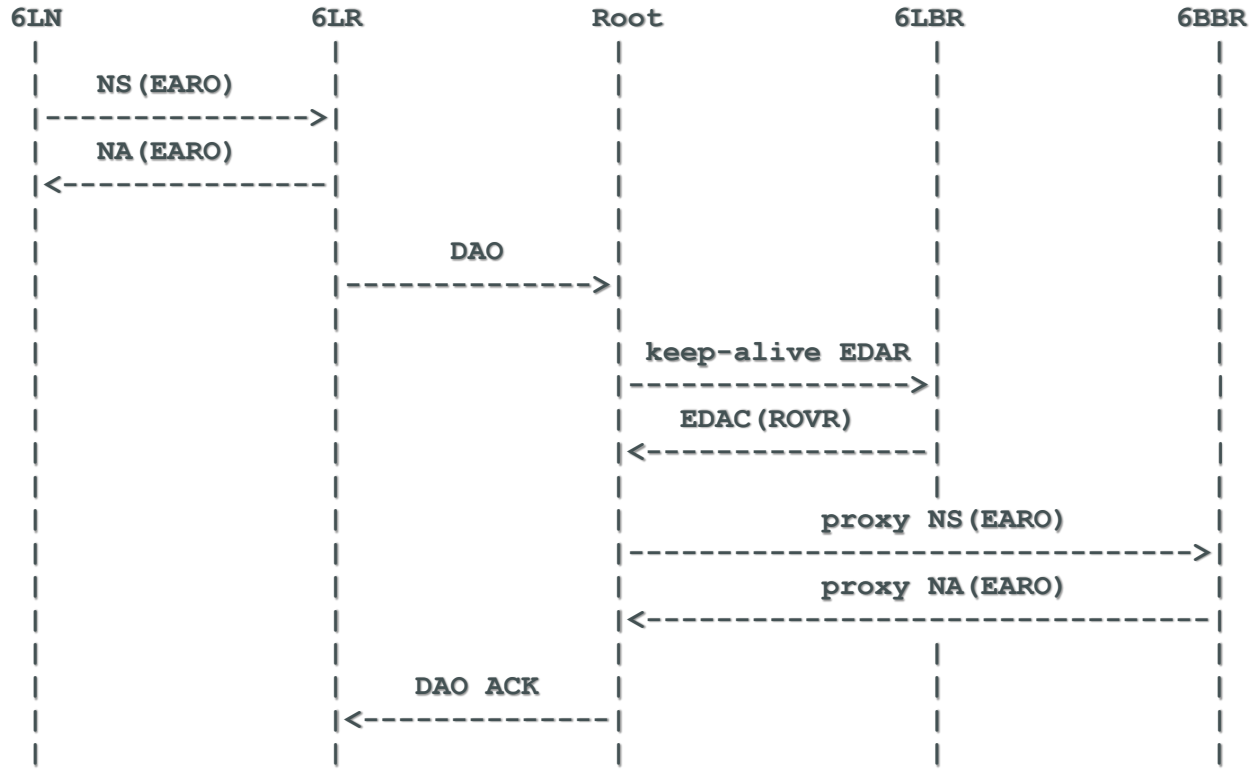
First registration

- 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.



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, the proxy keep alive flow is like:



Common Ancestor Objective Functions and Parent Set DAG Metric Container Extension

draft-ietf-roll-nsa-extension-04

Remous-Aris Koutsiamanis
Georgios Z. Papadopoulos
Nicolas Montavont
Pascal Thubert

ROLL@IETF105

Version -04 : updates since -01

- Changes addressing feedback from Dominique, Diego, Rahul, and Pascal (thank you so much!)
  - Added description of 3 CA OFs in draft (changed title to reflect that)
    - Common Ancestor Objective Functions and Parent Set DAG Metric Container Extension
    - The CA pattern, now is prescriptive, not just an example
    - The OFs extend MRHOF
    - Described as “diff” from the behaviour of MRHOF
  - Removed all references to 6LoRH-like compression of Parent Set
  - Lots of readability improvements
    - special thanks to Dominique and Diego

Road Forward

- Any extra reviews (especially of the definition of the CA OFs)
- WGLC?
- Code is available here:
  - Contiki NSA extension https://github.com/ariskou/contiki/tree/draft-ietf-roll-nsa-extension
  - Wireshark dissectors (for the optional TLV, i.e., PS): https://code.wireshark.org/review/gitweb?p=wireshark.git;a=commit;h=e2f6ba229f45d8ccae2a6405e0ef41f1e61da138

Thanks!

Questions?

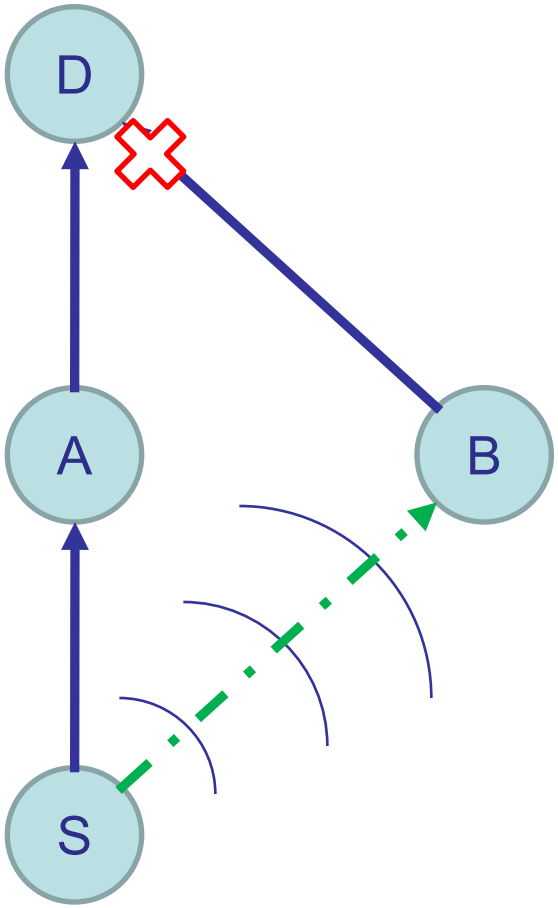
Backup

Brief Overview

- Goal: “Determinism”
  - High reliability
  - Low jitter
  - Bounded delay
- How:
  - Send multiple copies of packets (PRE) over different paths
- Challenges:
  - Avoid flooding with copies
  - Keep jitter low
- This draft:
  - Path information in DIO to help with challenges
  - OFs which use the path information to control PRE

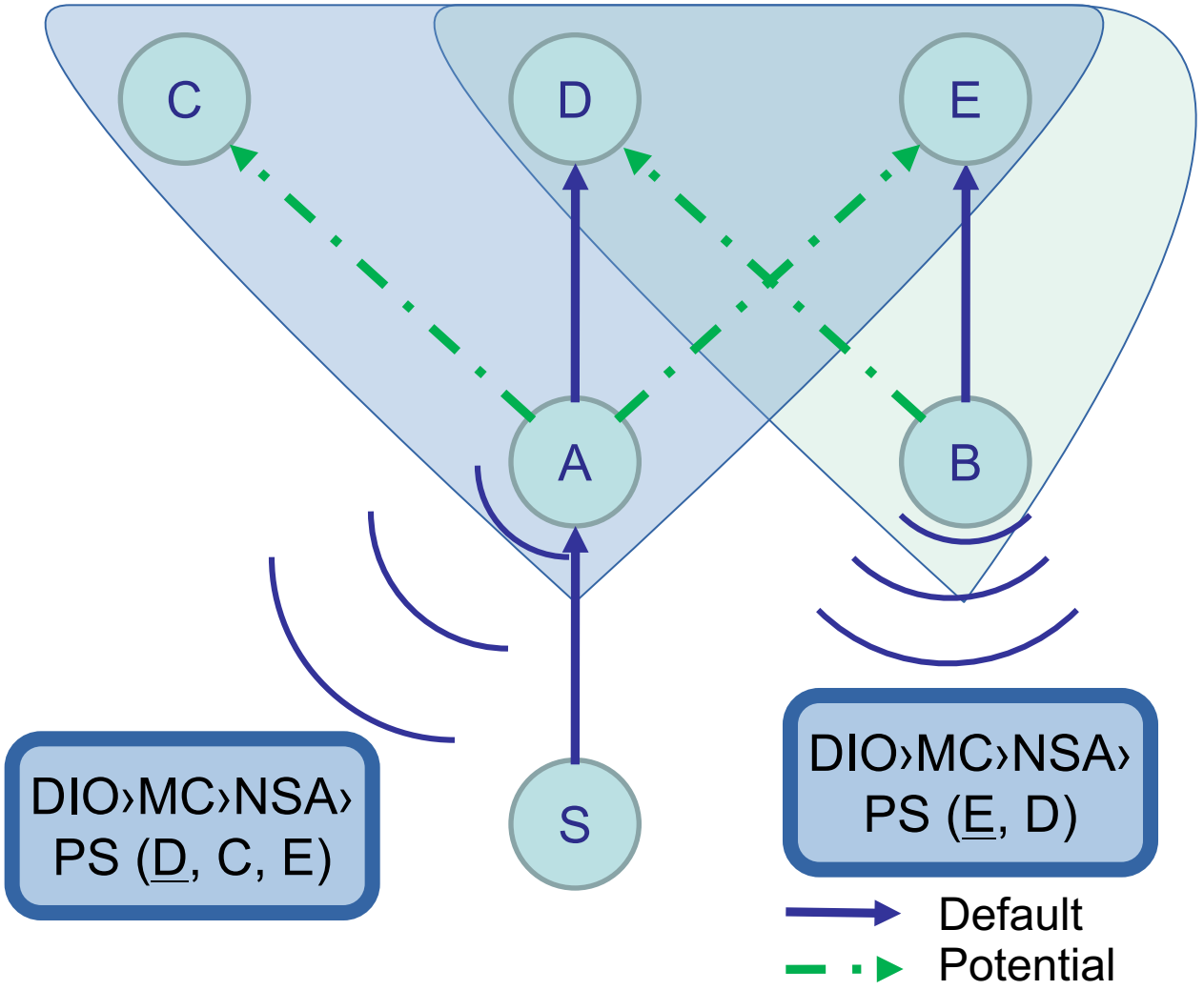
PRE over 6TiSCH

- Low jitter → bounded delay
- Reliable communication
- Packet Replication Elimination
  - Replication
  - Elimination
  - Promiscuous Overhearing (optionally)

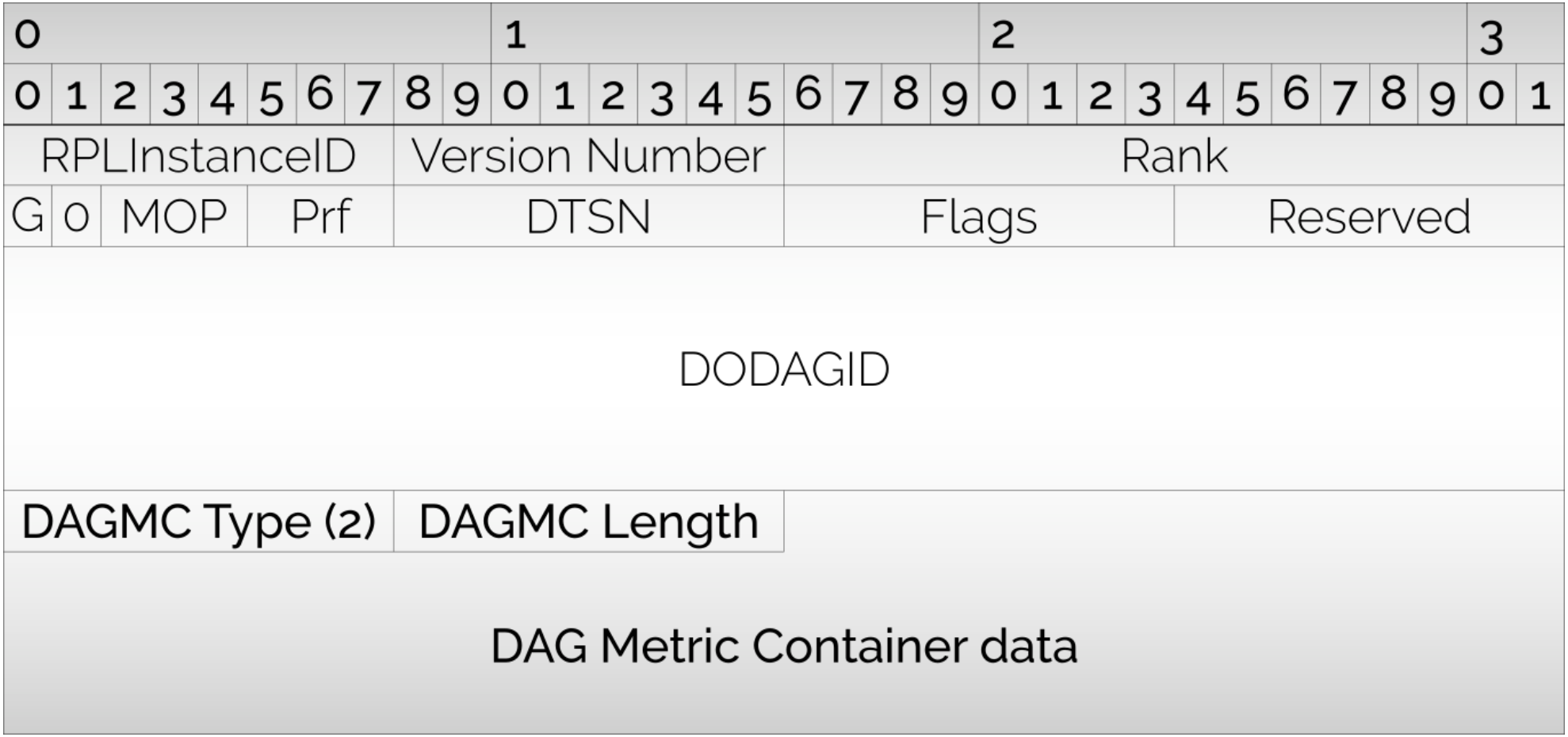


Parent Selection - DIO Messages

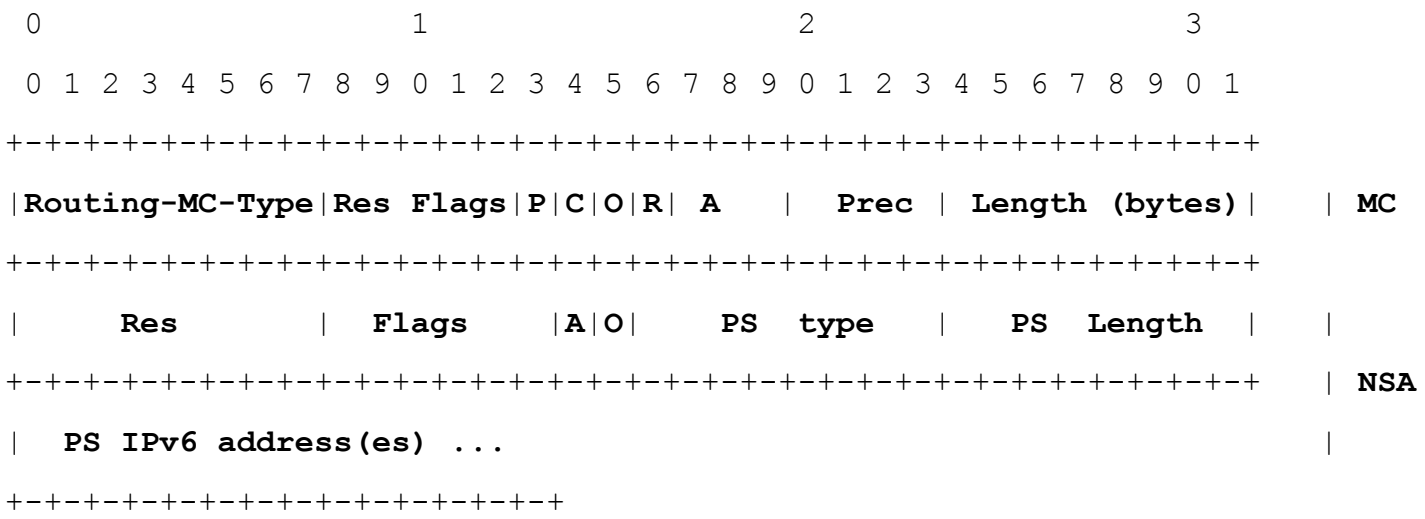
- Parent Set A:
  - {D, C, E}
- Parent set B:
  - {E, D}



DIO Format Example



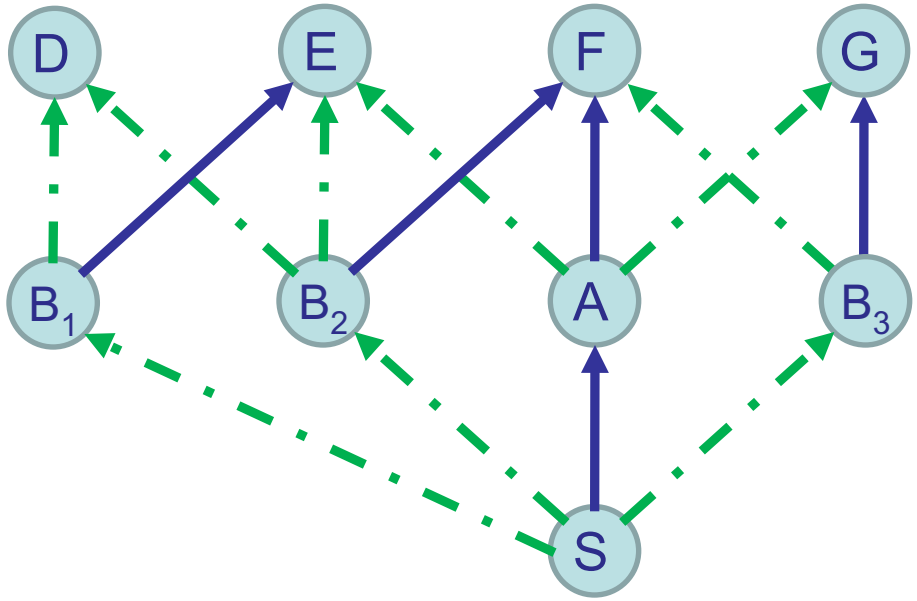
MC Format Example (1)



- 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

Implementation of CA: Medium

- PP(PP) ∈ PS(AP)
  - PP(A) = F
  - PS(B_1) = (E, D) ✖
  - PS(B_2) = (F, D, E) ✔
  - PS(B_3) = (G, F) ✔



Preferred Parent (PP)

Alternative Parent (AP)

MOP Extension & Capabilities

draft-rahul-roll-mop-ext-01

- Rahul, Pascal @ IETF105, Montreal

Need of MOP-extension?

- Mode of Operation (MOP)
  - Mandates primitives to be supported by the 6LRs
  - 3-bits in size
  - Already exhausted

MOP	Used for
0	No downward routes
1	Non-storing
2	Storing with no mcast
3	Storing with mcast
4	P2P-RPL
5,6,7 (Unused)	(AODV-RPL, P-DAO-NS, P-DAO-Storing)

Already in Contention

MOP Extension

- MOPex Option
  - New RPL Control message option
  - Applicable only if base DIO-MOP = 0x7
  - Final MOP = base MOP + MOPex

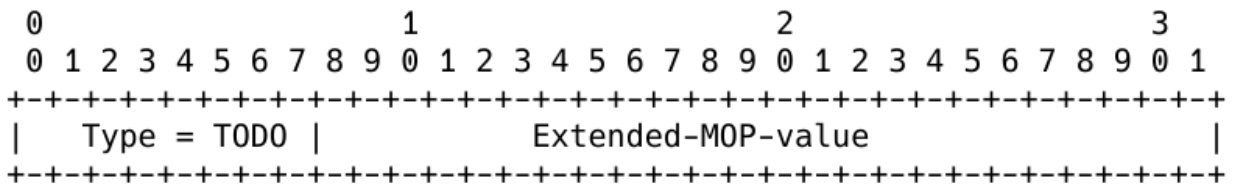


Figure 1: Extended MOP Option

Base MOP	MOPex	Final MOP
0	NA	0
1	NA	1
:	:	:
6	NA	6
7	0	7
7	1	8
7	2	9
:	:	:

Table 1: Final MOP calculation

Introducing Capabilities

- Capabilities indicate the set of features supported
  - Could be mandatory or ***optional***
    - Specs defining new capability indicate whether it is mandatory/optional
- Why MOP is not sufficient?
  - MOP ***mandates*** primitives needed by the routers
  - Unlike MOP, Capabilities can be ***negotiated***,
    - using DIO/DAO/DAO-ACK

Capabilities (Caps) Option

- Defined as new RPL Control message option
  - Can be part of DIO/DAO/ACK

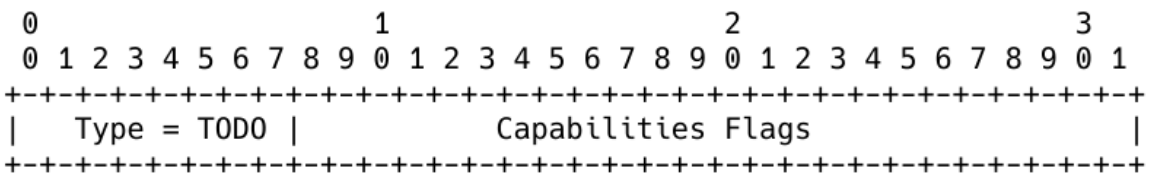
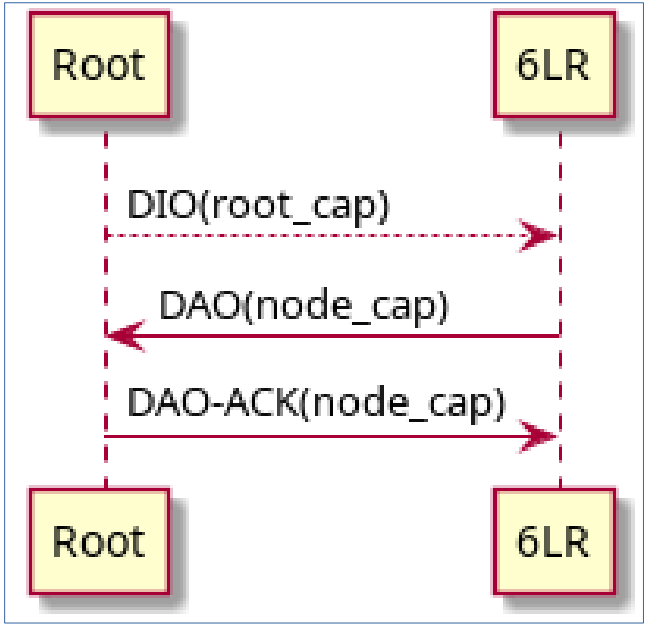


Figure 2: Capabilities Option

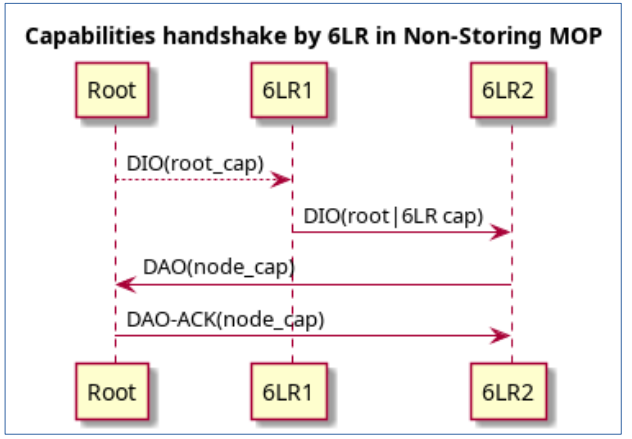
Use-case for Root

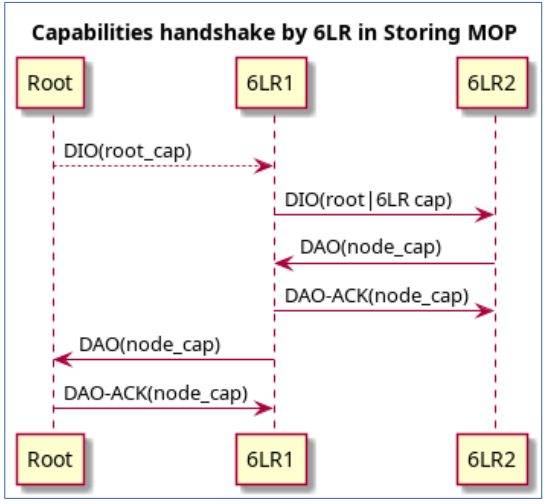
- Used by root
  - In DIO: Inform all the 6LR/6LN of root's capabilities
  - In DAO-ACK: Inform the 6LR/6LN about accepted capabilities from Root.



Use-case for 6LR/6LN

- Used by 6LR/6LN
  - In DIO: Inform its child nodes about its capabilities (for 6LRs)
  - In DAO: Inform parent/ancestors/ root of this 6LR's capabilities



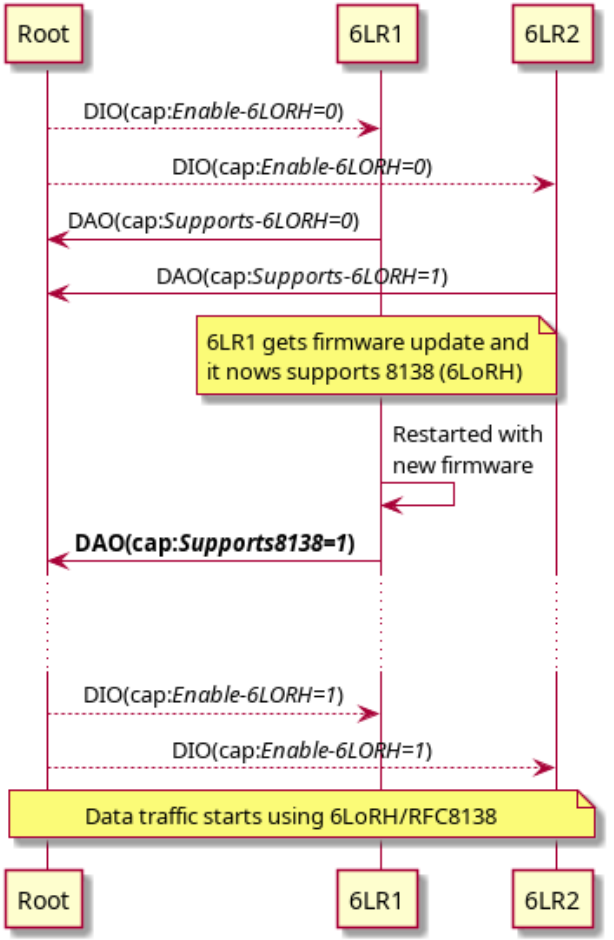


Points to ponder

- CAPs can work with existing MOPs
  - CAPs and MOPs are not dependent on each other
- Reducing CAPs control overhead
  - Eliding mechanism similar to DIO Configuration Option?
- How to reduce control overhead of MOPex?

Turnon-6LoRH, a use-case

- Can be handled without changing RFC 6550?
  - Note that this is not what is suggested in the draft, currently.



ACK

- Thanks to Georgious for the review
- Updates
  - Clarification: what if MOPex option is absent but the base MOP is 7.
  - Made explicit: CAP and MOPex are mutually exclusive
  - Added detailed IANA considerations

Enabling RFC8138 in brown field

draft-thubert-roll-turnon-rfc8138

Pascal Thubert

IETF 105

Montreal

RFC 8138

- RFC 8138
  - Compresses RPL artifacts
  - Fine in a new deployment
  - But no possible coexistence and no migration scenario

- vs. draft-ietf-roll-useofrplinfo
  - In contrast, UseOfRPLInfo sets bit 3 in DODAG Configuration option
  - Indicates switch to RPI 0x23
  - Possible coexistence

draft-thubert-roll-turnon-rfc8138

- Adds another flag (T) in the DODAG Configuration Option
  - Avoids a flag day
  - Nodes are migrated to new code with flag off
  - When all nodes support RFC 8138, T flag can be turned on
  - At turn on nodes that do not support RFC 8138 can only be leaves

Operation

A node that supports this specification SHOULD source packets in the compressed form using [RFC8138] if the new T flag is set in the RPL configuration option from its parents.

Failure to do so will result in larger packets, yields higher risks of loss and may cause a fragmentation.

A node that supports this specification SHOULD refrain from sourcing packets in the compressed form using [RFC8138] if the T flag is reset.

This behavior can be overridden by a configuration of the node in order to cope with intermediate implementations of the root that support [RFC8138] but not this specification and cannot set the T flag.

Regardless of the setting of the bit, the node MUST forward a packet in the form it was received, compressed or uncompressed.

Transition

The T flag in the DODAG Configuration option prevents Flag Days that might otherwise include truck rolls.

Operators can reflash over the air and then restart the devices asynchronously, keeping the network globally alive.

A network can only be migrated to this specification if all nodes support [RFC8138] or the remaining nodes that do not support [RFC8138] operate as only as leaves.

Failure to observe that rule may cause the remaining node to receives compressed packets that they can neither un-compress nor forward.

Transition

A new MOP or a new OF would satisfy the same need since it forces the legacy nodes to be leaves only

Supposedly capabilities will also enable that but needs new work

If all possible routers are not capable of RFC 8138, zones may be isolated.
Draft proposes single and double instance scenarios that enforce leaf behavior

Also need to encapsulate to the parent, same as for an unaware leaf. Needs work.

IETF 104 Prague
ROLL-BIER Design Team

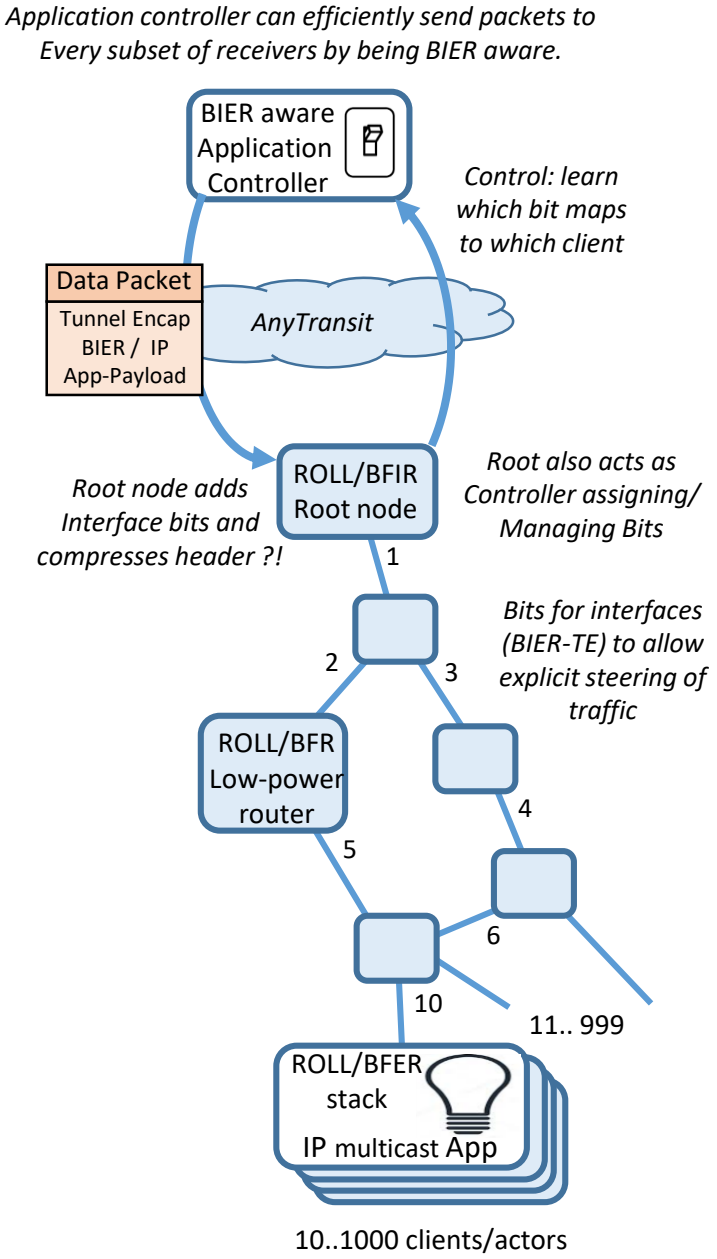
For he DT: toerless eckert (Huawei), <tte@cs.fau.de>

BIER in ROLL Vision

Design Team

- Consider joining/collaborating in BIER design team in ROLL-WG:
- Email: roll-bier-dt@ietf.org (normal subscribe)
- https://trac.ietf.org/trac/roll/wiki/roll-bier-dt (to be filled)
- Issues: tte@cs.fau.de

- What could be cool about this (if design team decides to do it) ?
- End-to-end BIER (with TE) in low-power networks (e.g.: building control)
  - Example: Application Controller sends BIER packet to subset of clients (lightbulbs)
  - Each client is BFER (has a bit)
  - Every packet can address a separate subset of actors through bitstring
  - Only controller app needs to be BIER aware. Receivers can think its just IP multicast.
- BIER TE bits to save power/memory
  - Routers are low-power (memory/CPU). Do not want to keep large routing table (1000 lightbulbs). Links are low power too.
  - Every interface has a bit. Routers only need to route on bits to directly connected downstream neighbors.
- No ASIC constraints. Everything is software
  - Headers/Bitstring can be compressed (loss free, lossy (bloom) to support long bitstrings.
  - Should result in header more compact than existing ROLL/RPL headers even for unicast: Only hop-by-hop bits sets to one receiver: Would also be used for unicast forwarding.



Since IETF103

- Just few meetings of design team towards end of 2018, one? In january 2019
  - Did not find cycles to agree on an offsite meeting to accelerate progress
  - But useful time spent to bring each othre up to speed on details.
  - See latest notes from:
    - https://etherpad.tools.ietf.org/p/roll-bier

Relevant docs

- draft-thubert-roll-bier
  - Proposed arch of BIER via ROLL
    - Core – not IP Multicast overlay, not L2.5 encap
    - Intends to support BIER/BIER-TE x explicit/bloom-filter mode of operations
- draft-ietf-roll-ccast
  - BIER for ROLL with bloom filters. Expect separate (IP multicast) overlay to handle false positives
- draft-thubert-6lo-bier-dispatch
  - L2.5 proposed encap for BIER packets
  - Comparable layer/function to MPLS/Ether BIER encap (RFC8296) ?!
    - Aka: would not use RFC8296 because in 6LO networks it all about compression
- draft-thubert-roll-unaware-leaves
  - Not covering BIER, but introduces type of nodes we want to support via roll-bier too.

- draft-other-apologies-forgetting-you
  - Please help complete this list

Since IETF103

- Proposal
  - Try to capture what we worked out into existing draft(s) if necessary new ones
  - Bottom up option:
    - Draft-thubert-roll-bier
      - does not include IP multicast layer – fine
      - How can we fix this up so we all like it ? (clearer division normative, vs informative)
      - Carsten: merge draft-ietf-roll-ccast – does it make sense?
      - Just define that false positives is part of the service provided with bloom filters
    - Which doc do we write the IP multicast relevant pieces into ?

The end

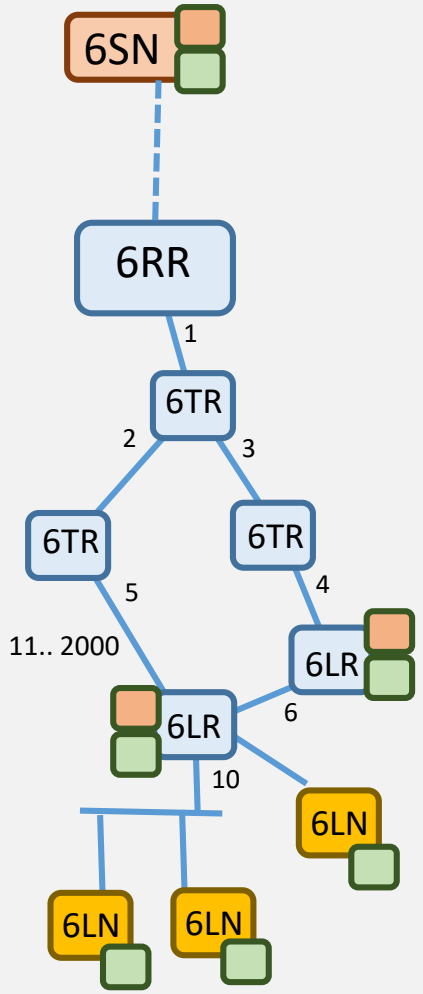
Remaining slides from IETF103 in case they are needed

Framework/ Terminology

ROLL+BIER+6LoRH

- Tunnel Mode
  - BIER, BIER-TE: Bits assigned to 6LR
  - BIER-TE: Bits also assigned to adjacencies/6TR
  - No bits assigned to 6LN
- Services
  - BIER 6SN-> 6LR + BA
  - IP unicast 6SN -> 6LR/6LN + IA
- 6LoRH + BIER
  - Compress/uncompress unicast/multicast packets
  - 6RR … 6LR
  - 6LR ... 6LN ???

Framework



Terminology

6SN = Server Node / Application
Not running ROLL/BIER
able to send Multicst
packets with
BIER bitstring to target
set of destinations explicit

RPL + BIER routers
Roles:
6RR = root.
needs to support 6LoRH for BIER
6TR = transit – no bit in BIER mode
6LR = leaf (LoWPAN) Router
bit in BIER mode,
connects to 6LN
needs to support 6LoRH BIER

6LN = Lightweight Node
no knowledge of ROLL/BIER
MAY want to support
6LoRH with BIER
extensions for compression

BA = BIER aware app able to send/rcv packets with bitstring, no IP multicast needed

IA = IP unicast/multicast app. No BIER awareness

Continuing to limit work scope by eliminating “below the line / do not work” options

- Only consider IP multicast payload for ROLL-BIER multicast now
  - Required for 6LN (non ROLL-BIER capable) nodes
  - For 6LR receivers we do primarily care that we can address them directly from the sender via bitstring, but not so important to get rid of potentially unnecessary IP Multicast header
    - IP Multicast header should be compressed also by 6LoRH ?!
- No new “faked” IP packets (see later slide)
- Only “transport mode” to 6LR, only “tunnel mode” to 6LN
  - See explanations later
- Only consider BIER-TE semantic, not BIER ?
  - TBD. Maybe we can start defining common forwarding and add BIER control plane later (when seen necessary)

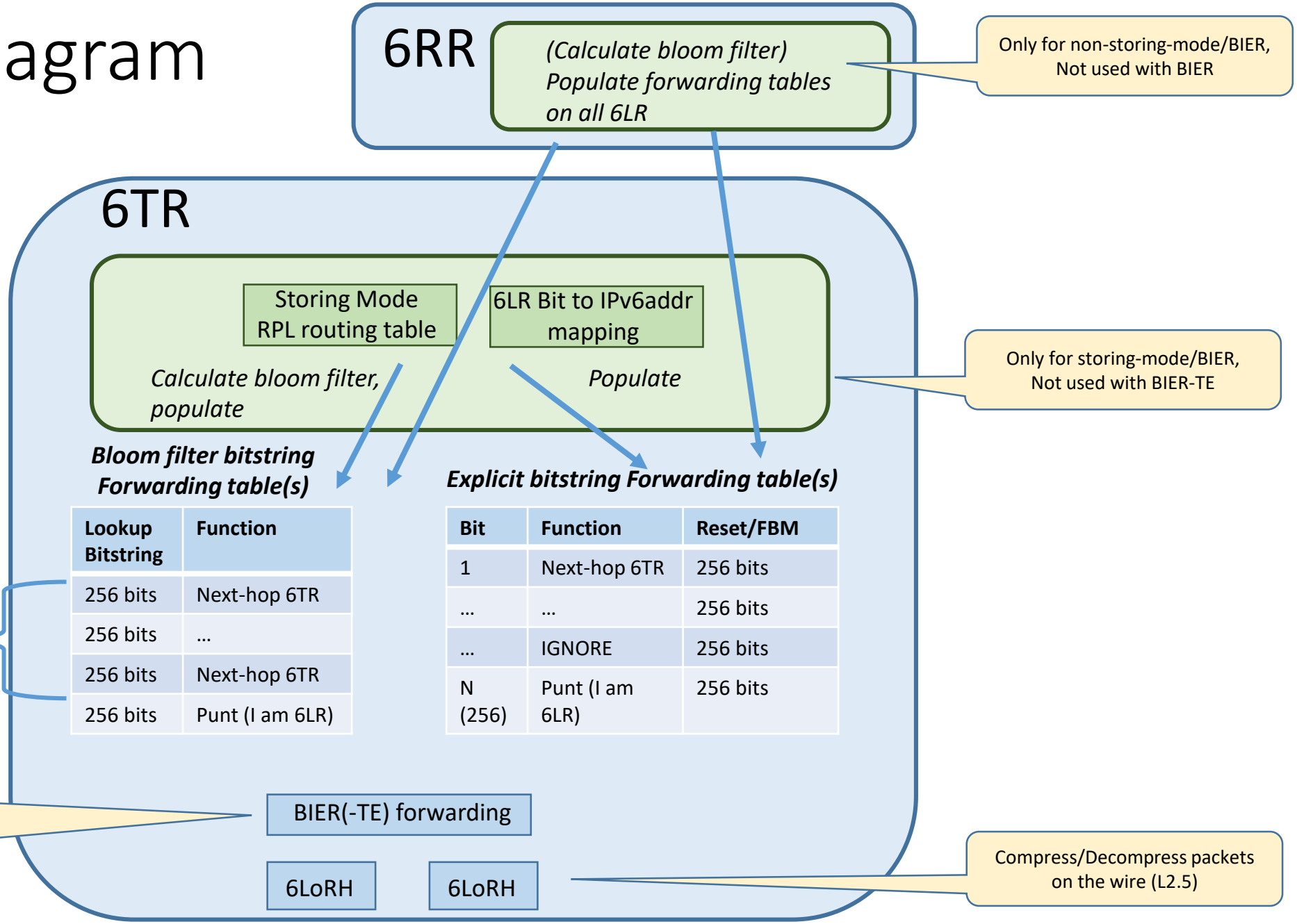
Details, Stack options

- First option: “Tunnel Mode” to 6LN, “Transport mode to 6LR”. Common header
  - Lowest layer is 6LoRH with BIER bitstring. Also all RPL artefact.
  - Next: 6LoPAN compressed IP packet (RFC6282)
- Transport Mode to 6LR:
  - 6LoRH indicates this mode, compres/uncompress destination address
  - Save 6LoPAN state for Dst address and compression field in 6LoPAN header (compressed state)
- Tunnel Mode to 6LN:
    - *Simple IP packet without any RPL/BIER artefacts.*
      - *High compression of IP address through state on every RPL hop for src/dst addresses (stateful)*
      - *Stateless – assumes MAC address derived IPv6 address.*
        - *But only works single L2 hop because it relies on the L2 destination “MAC”.*
      - *Aka: Stateful compression + BIER is big benefit because with BIER only the 6RR and 6LR would need to have state for the addresses because 6TR would just forward based on bits.*
      - *Need to show in slides how BIER benefit applies here:*
          - *No changes required for 6LoPAN to get desired benefit*
      - *FUTURE: define details of “transport mode” options.*

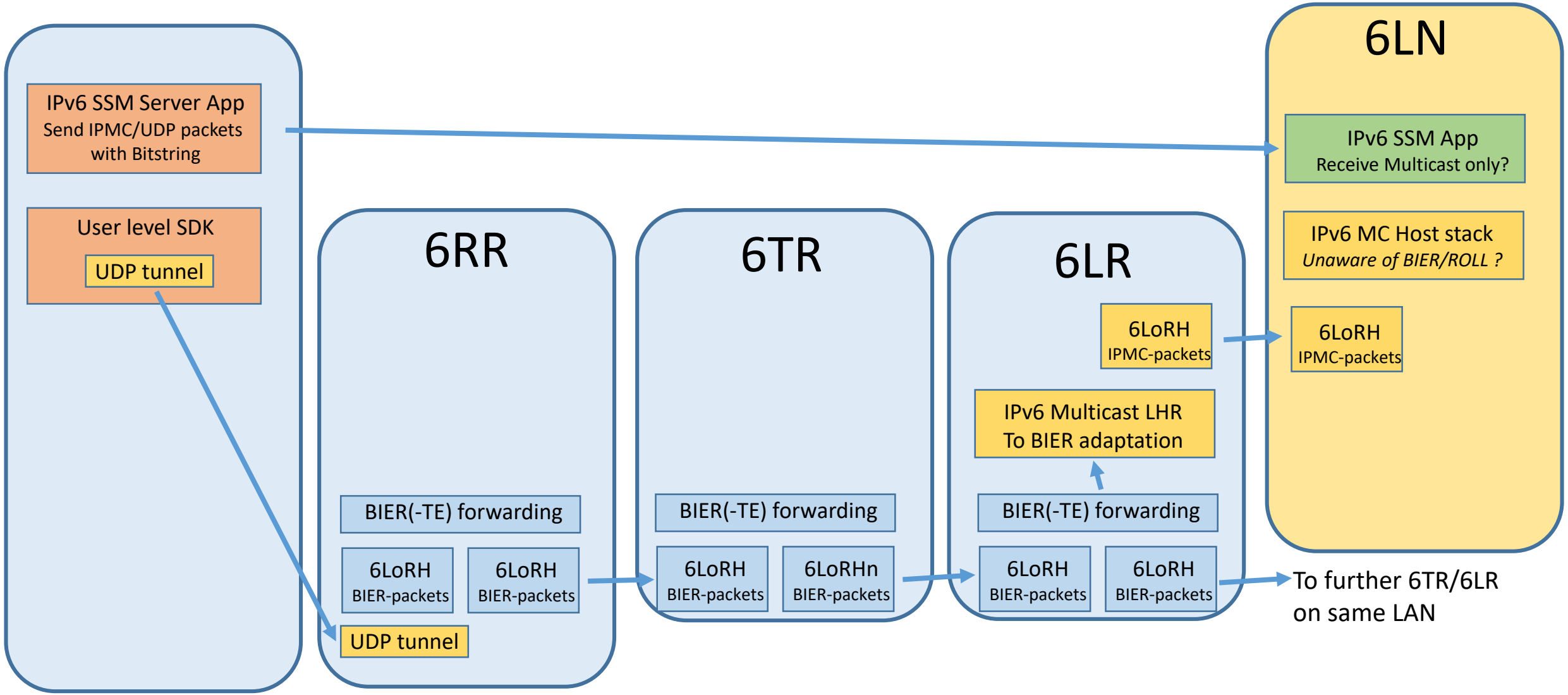
6RR/6LR diagram

Forwarding table does Not need to distinguish Between BIER/BIER-TE

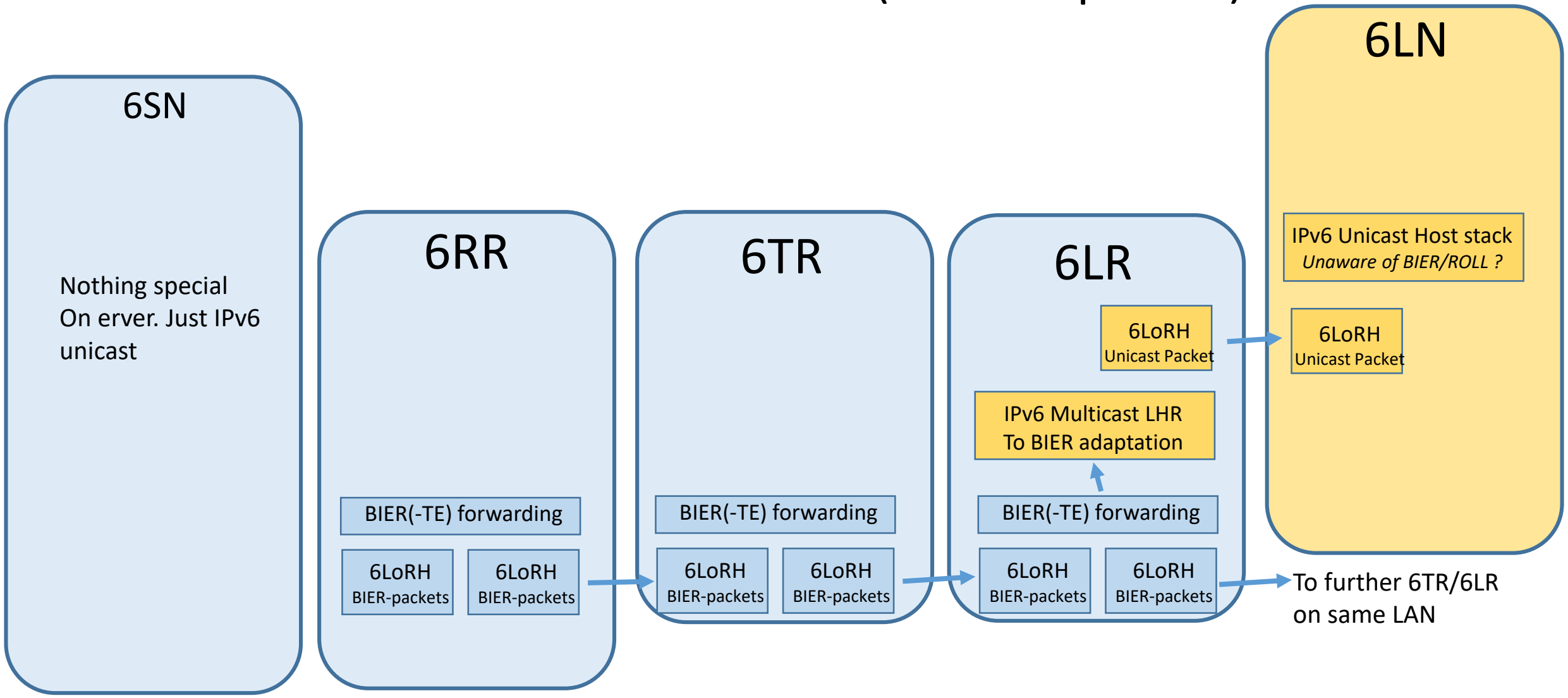
Difference just in how Next-hop entries/FBM for Bits are created



IP multicast over ROLL-BIER



IP unicast over ROLL-BIER (incomplete)



Bit-Reset/ Bloom Filter considerations (not reviewed)

- With bloom filter bitstring, bits can not be reset
- BIER logic
  - Leads to duplicates. Other radio-network issues can also lead to duplicates, so maybe not a big issue – radio networks MUST be prepared to deal with duplicates
  - Routing protocol microloops are avoided by reset. If RPL can have microloops they would only be protected against by TTL expiry (eg.: > 100 packet replications possible).
  - False positives in bloom filter can create more duplicates and more replicated packet when there is a routing microloop
- With BIER-TE
  - No routing protocol involved = no IGP microloops.
  - False positives can create duplicates AND looping (like badly set bits by BIER-TE controller can create loops too)
- Solutions ?
  - Have ROLL-BIER domain TTL (6LoRH TTL ?) that is set to be as small as possible: longest path to any receiver. Calculated by root node ?!
    - With common network diameter < 8 ?! This should be good enough ?! (4 bits enough to encode ?)
  - Duplicate elimination by packet-ID ? Probably takes too much memory…
- What to do ?
  - Reset bits when using explicit bitstring, not-reset only when using bloom-filter bitstring ?!
  - Figure out we can live without resets for all modes ?

BIER consideration (not reviewed)

- With bitstring length N, we need N * N bits for FBM – reset bits
  - With bloom filters, we can not have FBM (reset bits). But we try to resolve that issue
  - Do we want to forego FBM for non-bloom filter ?
  - Less problems to solve than with bloom filter: No false positives! Just more duplicates, routing microloop duplicates.
- Simple option ?
  - Make FBM a “SHOULD” – low-end devices could optimiz them away.
- Quantify benefit of explicit BIER bitstring (not BIER-TE)
  - Unicast: Do we save anything over existing storing mode with 6LoRH ? (header already well compressed)
    - Maybe there is NO reason to use BIER bitstring for unicast (only BIER-TE)
  - Multicast: assume existing storing mode network (aka: overhead of unicast routes acceptable)
    - BIER avoids to introduce another multicast routing protocol (PIM, RPL-multicast state,…)
    - More efficient use of bits (no bits used for adjacencies, just 6LR

BIER consideration (2, not reviewed)

- Option position summary ? All need to be validated/quantified.
- Non-storing mode networks
  - Use BIER-TE == non-storing. Unicast+multicast.
  - Can further minimize state avoiding FBM memory
  - Smaller networks use explicit bitstring. Larger ones use bloom filter.
- Networks with storing mode routers
  - Use BIER, maybe just multicast. No benefit? to use for unicast.
  - Smaller networks use explicit bitstrings, larger ones use bloom filter.

Bitstring / BIER considerations

- Explicit BIER-TE bitstring allows to save memory on 6TR/6LR:
  - Do not need RPL routing entries for all 6TR/6LR.
- With BIER, we do need RPL routing table for all 6TR/6LR
  - What do we save (why BIER) ? Unclear / need to better caracterize.
  - Comparison complex ? Because we also have to take savings from 6LoRH into account.
- Aka: Do we need to consider Explicit Bitstring + BIER ?
  - Forwarding plane can be the same BIER/BIER-TE. Eliminating BIER just reduces control plane work to consider (RPL -> create BIER forwarding table entries).
  - BIER/BIER-TE forwarding more similar than outside of ROLL:

Open

- With BA (bier aware application) we could send non-IPv6 payload. Any benefit in that ?
  - No ? 6LoRH would compress IP header away so there is no benefit eliminating IP ??
- Unicast between 6LR/6LN ? Unicast 6LR/6LN towards 6RR ?
  - How do we deal with that ?

- IEF Multicast likes SSM.
  - SSM could help to address 6SN (which server needs to know about IGMP memberships). Part of

Refuse

Refuse - Does not work

- Does not work: Multicasting unicast packets
  - Aka: unicast packet replicated via bitstring to more than one destination
  - Could recreate packet with each destinations unicast dest-addr, but:
    - No architecturally clean solution to do so – e.g.: UDP checksum calculation
  - Make destination address a “unicast-group-address” – IPv6 unicast address same on all nodes
    - Avoid problems like UDP checksum
    - But would just reinvent IP multicast service with IP unicast addresses (which we will have).
    - No added value.

Refuse - Below the line (now)

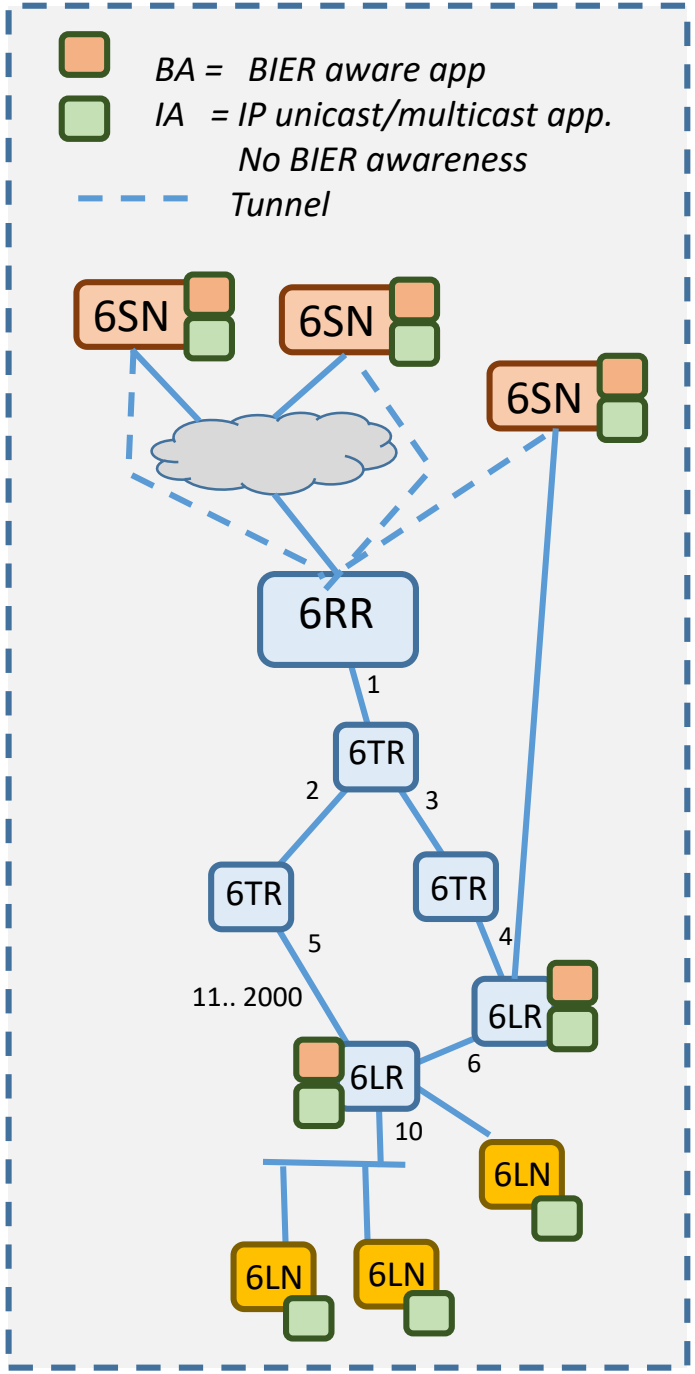
- Using BIER to carry non-IPv6 packets
  - Idea was: BIER packets can directly carry any payload, no IPv6-multicast heade required, eliminate overhead/complexity of IPv6
  - Claim: 6LoRH can compress the IPv6 headers so much that there is not enough initial value in designing header stacks to eliminate IPv6.
  - But: We need to make sure 6LoRH will also equally well compresses the IPv6 multicast header
  - Also eliminates need to figure out how to deal with false-positives for non-IPv6—multicast BIER packets
- Below the line: Transport mode
  - In transport mode, 6LN nodes will have bits assigned to them in the bitstring. This introduces the need for them to be more aware of BIER, e.g.: introduce more BIER awareness into 6LoRH all the way into 6LN and receiver applications.
  - For unicast
    - Will work fine but below line until we have evidence that it could provide better compression than tunnel mode with 6LoRH+BIER compression.
  - Transport mode for multicast
    - Just too much work trying to figure out in first round how to build a lightweight node that ALSO has to be aware of BIER bitstrings.

TO BE DISCUSSED

IP Multicast layer

IP Multicast layer overview

- Primary target for IP Multicast in ROLL-BIER
  - Efficient sending/replication of multicast packets from IP multicast sender (app) on 6SN to IP multicast receiver (app) on 6LN/6LR
  - App on 6SN should be able to explicitly indicate set of receivers. Key benefit not possible with BIER not possible with standard IP multicast.
    - Example from design-team slide: send “ON”/“OFF” message to subset ot light-bulbs that should be switched “ON”/”OFF”
- Not needed: Send IP multicast packets from 6LR/6LN
  - Application client/server mode:
  - Application has one or more server (6SN) sending IP multicast through tunnel into 6RR. Clients (6LR/6LN) can just receive IP multicast.
  - Client->client multicast “emulated” by client sending unicast to Server and then server sends IP multicast
    - Client may receive its own IP multicast packet back, so application need to be able to recognize/filter packets from “itself”
    - Client on 6LR can be excluded from bitstring by 6SN, so this additional filtering only required for 6LN



IP Multicast layer: SSM only

- Majority of IETF Multicast experts prefer SSM IP Multicast over classical ASM IP Multicast
  - Proposal follows that direction
  - SSM: receivers send (S,G) membership instead of (G) membership (MLDv2).
  - S = sender (Server) IP unicast address.
- Benefits
  - If we want to avoid implement directly client->client IP multicast in ROLL-BIER, we need Client/Server model, only servers allowed to send IP multicast, client unicasting to server if they need to be able to send IP multicast (previousslide). This is exactly the model how IP multicast would be done with SSM.
  - SSM avoid need to coordinate IP multicast group addresses across applications (potentially big operational issue).
  - When SSM IP multicast application on a sender wants to create a new SSM stream, it simply needs to allocate an SSM IP multicast group that is unused on this sender (like allocating an TCP port unused for a new unicast service).
  - Client discovery of SSM (S,G) ? Use same scheme a for Unicast server discovery in the absence of IP multicast (e.g.: DNS). As necessary, extend discovery to also discover IP Multicast group (e.g.: Put Server IP unicast address and group-address into DNS. E.g.: group-address in TXT record).
  - Ideal: New IP multicast application on 6SN can allocate new local IPv6 address (independent from IPv6 address used by other IP multicast app running on same 6SN)
    - Benefit: Applications that can have their own SSM IP Multicast sender IPv6 addresses can use static/well-known IPv6 Multicast group application – no need for group discovery by receivers.
    - Best method for local address allocation ? IPv6 privacy addresses ?... ?

IP Multicast layer: MLDv2

- Why do we even need MLDv2 ?
- Not for 6LR: Receiver application on 6LR will receive multicast packet because of bit in BIER bitstring. No IGMP/IP Multicast needed.
- Receiver on 6LN will receive link-layer multicast packet across access-subnet with potentially many 6LN.
  - (S,G) IP multicast header indicates whether packet is of interest to receiver
  - 6LR needs to know whether to send packet to a particular subnet. Some signaling needed to learn this from receivers. No need to reinvent wheel: MLDv2 does this.
  - Receivers need to open link-layer filter (eg: destination) to receive multicast and send packets up to right application. OS-level MLDv2/multicast-socket-API does this. No need to reinvent the wheel.
- BUT!! Above reasoning to reuse/not-modify 6LR->6LN IP multicast (SSM) is ONLY true if IP Multicast packet actually contains IPv6 Multicast header.
  - If we want compressed packet on 6LR->6LN link, then the above rules may not be true: If new OS-level code is needed on 6LR/6LN to support 6LoRH compressed IP multicast packets, then we should re-evaluate what the most pragmatic way is to get a working solution.
  - Unclear what exists today wrt. 6LoRH/IP-multicast/compressed-packets.

IP Multicast layer: proposal (1) - INCOMPLETE

- 6LN:
  - Signaling: SSM subset of RFC3810 (MLDv2), host side
    - Note: This should also be subset of “lightweight IGMPv3/MLDv2” (RFC5790)
  - Only need to be able to receive, not send IP multicast
  - 6LoRH (extension?) to receive compressed IP multicast packets
- 6LR:
  - Signaling: SSM subset of RFC3810 (MLDv2), router side
  - (Modified/Extended) SSM subset of RFC4605 (IGMP/MLD proxy routing)
    - 6LR aggregates SSM receiver state from all subnets: 6LN + internal virtual subnet for IP Multicast receiver application on 6LR itself.
    - 6R sends aggregated membership state to 6SN (join/leave): HOW? (later slide)
    - Any BIER packet with 6LR “bit” set or 6LR comb filter match will be passed to MLD proxy routing forwarding as coming ”from upstream”
    - NO receiving of IP multicast packets from downstream nodes (6LR, 6LN) – just discard
- 6TR:
  - Do not participate in P multicast layer. Just forward BIER. Become 6LR when BIER bitmask/comb-filter entry exists for them. Every 6LR is also always 6TR (forwards BIER independent of processing IP Multicast)
- 6RR:
  - Just like 6TR except that it also must be able to receive tunneled IP Multicast + (pseudo) BIER header from 6SN.

IP Multicast layer: issue

- Simple: let 6SN “just” send SSM IP multicast. No BIER improvements
  - Receipt of IP multicast ONLY determined by MLDv2 membership from receivers (6LN / 6LR).
  - Foregoes application benefits of using BIER (sender determines receivers)
- Problem: Can not get BIER benefit for 6LN without non-heuristic bitstring transport mode”
  - Give BIER bits to 6LN, carry bitstring all the way to 6LN
  - Can not use heuristic likely not useful here: need to be able to eliminate false positives at application level.
- Design for ONLY SSM IP multicast different / simpler as if we want to introduce application layer BIER benefits to application

IP Multicast ONLY model

- 6LR send aggregate MLD membership (join/leave) to RR.
- 6LR keeps (S,G) -> { 6LRi } state.
- Keep table of Bits(6LRi) -> bitstring:
  - bitstring with one bit (non-bloom, BIER)
  - bitstring with > 1 bits (non-bloom, BIER-TE):
    - BIER bit for 6LR plus bits for each hop towards it.
  - bitstring with > 1 bits (bloom BIER/BIER-TE). Bloom compression bitstring result for 6LR (BIER) plus Bloom compression bitstring result for each hop towards it (BIER-TE).
- 6SN just send IP multicast to RR via some tunnel
  - RR may perform RPF check (source address must be 6SN unicast address).
  - Then encaps/forwards BIER/BIER-TE:
    - Loop up (S,G) of packet, Ors Bits(6LRi) for all { 6LRi }. Sends packets
- At minimum same type of hop-bits determination for BIER-TE / non-storing mode as used for unicast non-storing mode today. Just encoded as direct/bloom-filter bits.

IP Multicast + 6SN BIER model

- 6SN should indicate set of destinations instead of “just” MLDv2 join state
- As long as we assume we always use IPv6 multicast packet / MLD, the set of destinations indicated by 6SN must be subset of the “joiners”
  - Simple app example: Each application just has ONE IPv6 multicast group, all application members on 6LR / 6LN join to that group. But 6SN only indicates subset of those application clients for each packet.
  - Does not work for 6LN without transport mode (prior slide).
- Most simple? Hybrid solution (introduce BIER benefit to 6SN with minimum additional work ?)
  - RR forwards (S,G) -> { 6LR } state changes to 6SN. Use of SSN makes this easy: Each 6SN only gets updates for those (Si,G) with its own Si.
  - 6SN send IP Multicast packet with additional BIER pseudo-header
    - Indicating subset of 6LR to which packet should go
  - 6RR then only OR’s bitstrings for this subset as opposed to full { 6LR } it maintains
- INCOMPLETE

Next steps ?!

- Push discussion about BIER interface for 6SN to BIER-WG ?!
  - ROLL should not come up with somethin ROLL specific that does not have to be ROLL specific – better try to find general purpose solution in BIER (ROLL defining requirements)
- SSM IP multicast (ONLY) solution
  - ROLL: (optional) ? compression for 6LoRH extension on access LAN
  - “Tunnel” from 6SN node to 6RR - to send IP Multicast packets from 6SN to 6RR
    - Two cases ?
    - A) 6SN is inside 6LoRH network: simple 6LoRH header extension: one bit to indicate “packet goes to root”, but also encode IP Multicast Group address.
    - B) 6SN is “behind backbone”: Need an actual IP tunnel (path between 6SN/6RR not natively supporting 6LoRH). Just encapsulate same 6LoRH packet inside ?
  - MLDv2 proxy operations on 6LR
    - 6LR -> 6RR covered by ‘draft-ietf-bier-mld-01.txt’, but discovery of 6RR and constraints (only receive, not sending) and 6LR->6LN not covered.

OPEN MIC

THANK YOU!