Monday 5/11 - 9h00-11h00

IETF 103 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

- 9:00-11:00 Monday Morning session I
- Remote Participation
  - Jabber Room: xmpp:roll@jabber.ietf.org?join
  - Meetecho: https://www.meetecho.com/ietf103/roll
  - Etherpad: https://etherpad.tools.ietf.org/p/notes-ietf-103-roll
  - Minutes taker: **MILLION THANKS** to Dominique B.
- Jabber Scribe: **MILLION THANKS** to Rahul J. and Michael R.
- **Please sign blue sheets :-)**

Agenda

09:00 - 09:05	WG Introduction	(Ines/Peter)
09:05 - 09:35	ROLL-BIER Design Team Status	(Toerless)
09:35 - 09:45	draft-ietf-roll-efficient-npdao-09	(Rahul)
09:45 - 10:00	draft-ietf-roll-rpl-observations-00	(Rahul)
10:00 - 10:10	draft-ietf-roll-aodv-rpl-05	(Charlie)
10:10 - 10:25	draft-ji-roll-traffic-aware-objective-function-02	(Georgios)
10:25 - 10:40	draft-koutsiamanis-roll-nsa-extension-02	(Georgios)
10:40 - 10:55	draft-ietf-roll-dao-projection-04	(Pascal)
	Draft-thubert-roll-unaware-leaves-05	(Pascal)
10:55 - 11:00	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	Authors correcting AD Evaluation
draft-ietf-roll-dis-modifications-00	To be continued
draft-ietf-roll-mpl-yang-02	No comments, stagnated
draft-ietf-roll-bier-ccast-01	Bier-roll design team takes over
draft-ietf-roll-efficient-npdao-03	Wait for IPR declarations
draft-ietf-roll-rpl-observations-00	Discussion today

Related Internet-Drafts

Draft	Status
draft-thubert-roll-unaware-leaves-05	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

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
#186	Useofrplinfo - AD review	useofrplinfo
#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

BIER - ROLL Design Team

IETF 103 Bangkok
ROLL-BIER Design Team

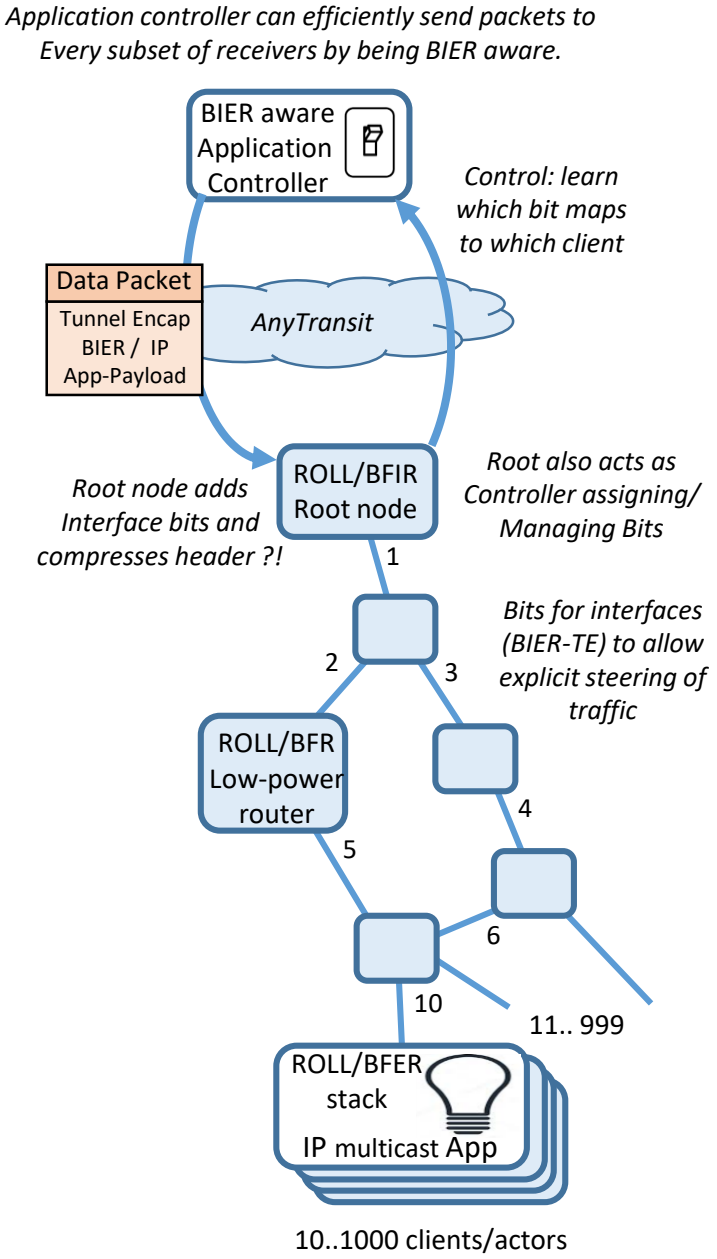
Toerless Eckert (Huawei), <tte@cs.fau.de>

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.



Status of dt

- Slow start but getting more organized
  - Try weekly meeting via IETF webex, Wed. 7 PM GMT
  - until we can get work items assigned to individual stakeholders
    - Getting participants up to speed
  - Will redo role call for meeting time/cycle after IETF103
- Started to put slide deck (this) and notes etc. into github
  - www.guthub.com/toerless/roll-bier (no ROLL WG github repo group ?)
- Worked through bunch of arch level details
  - But in middle of writing them down

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.

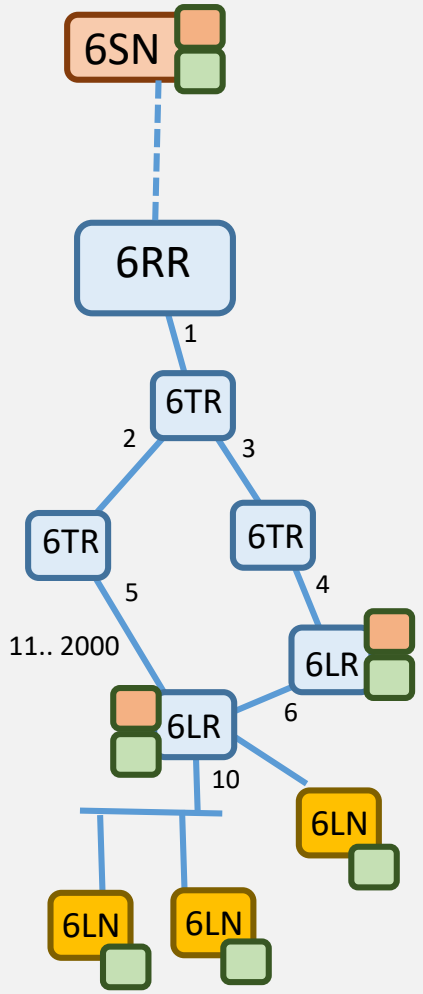
- draft-other-apologies-forgetting-you
  - Please help complete this list

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)

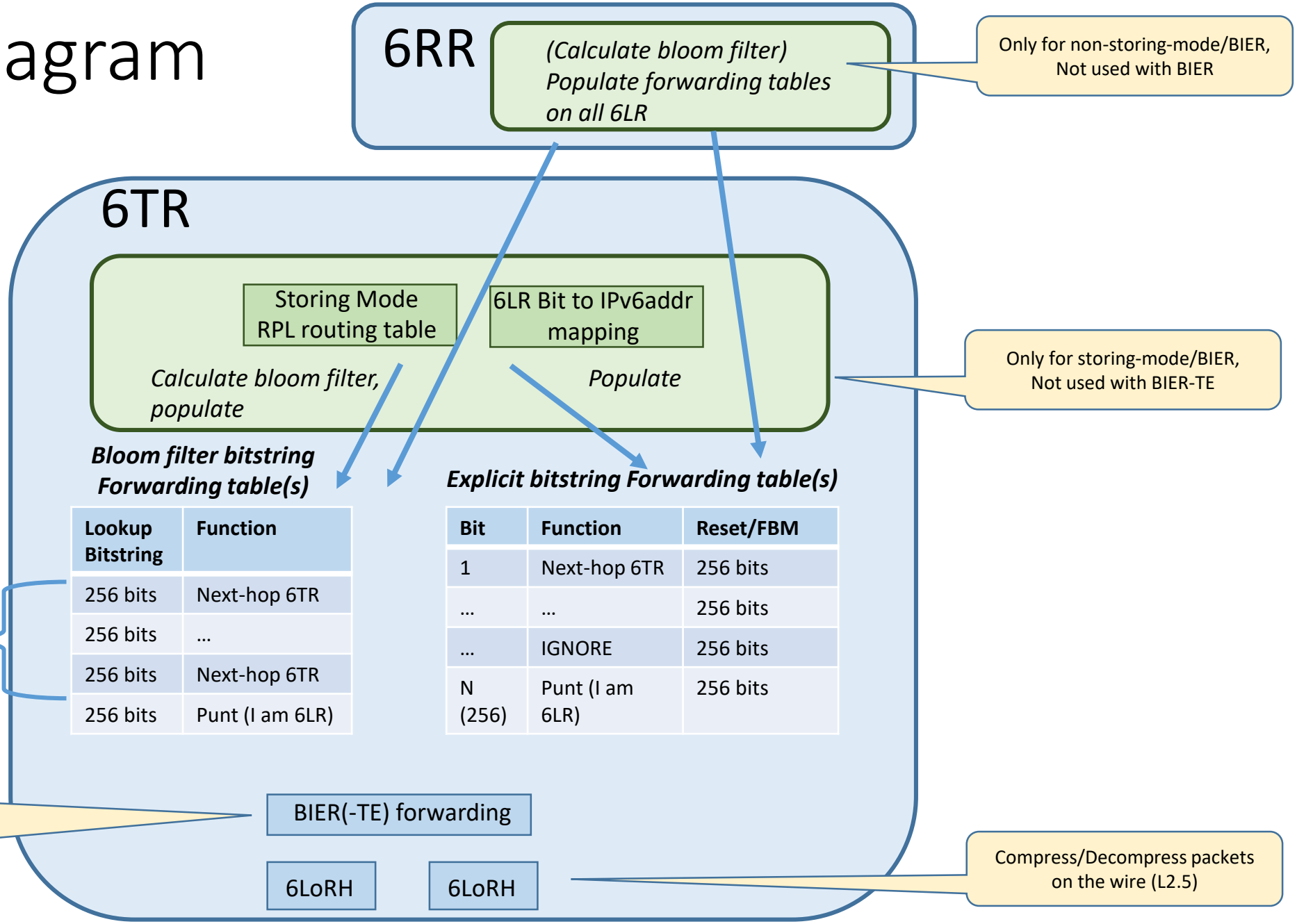
Details, Stack options

- 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.*

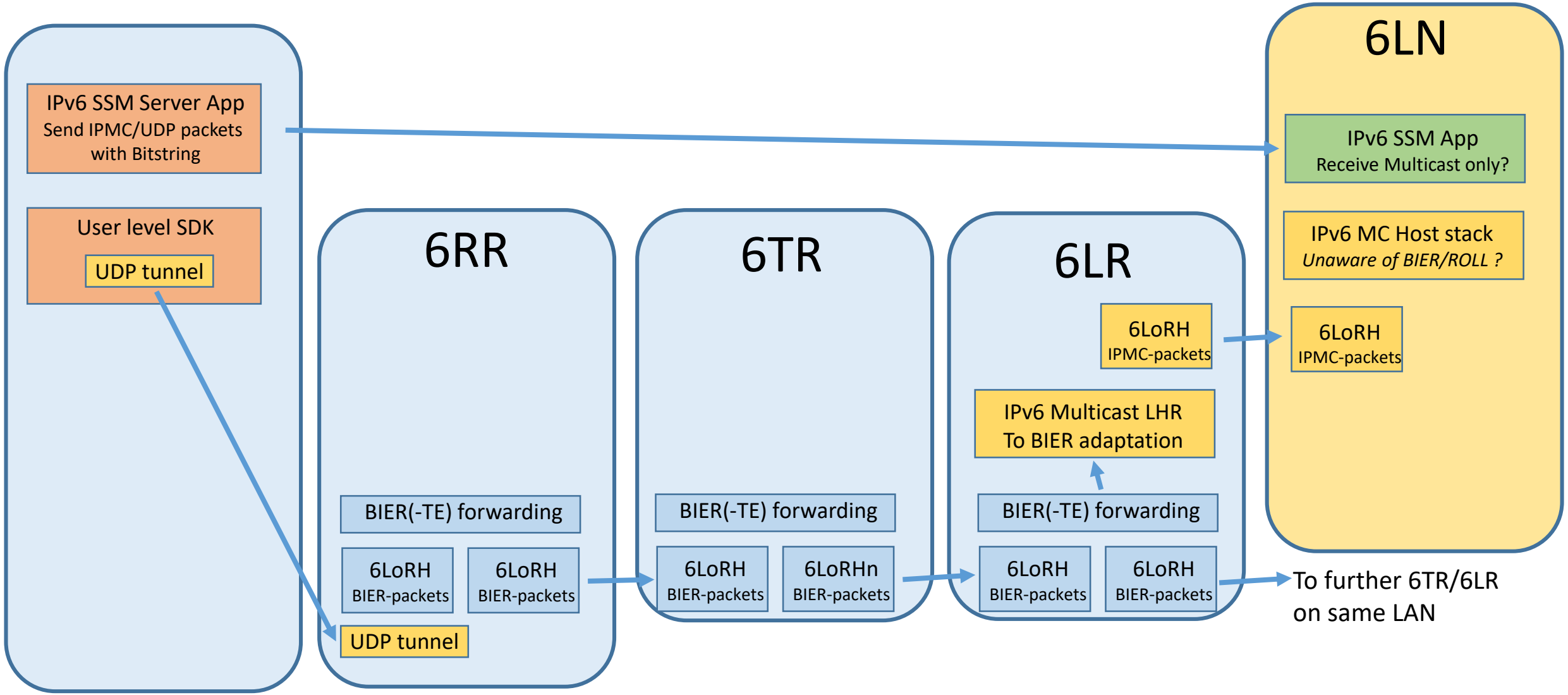
6RR/6LR diagram

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

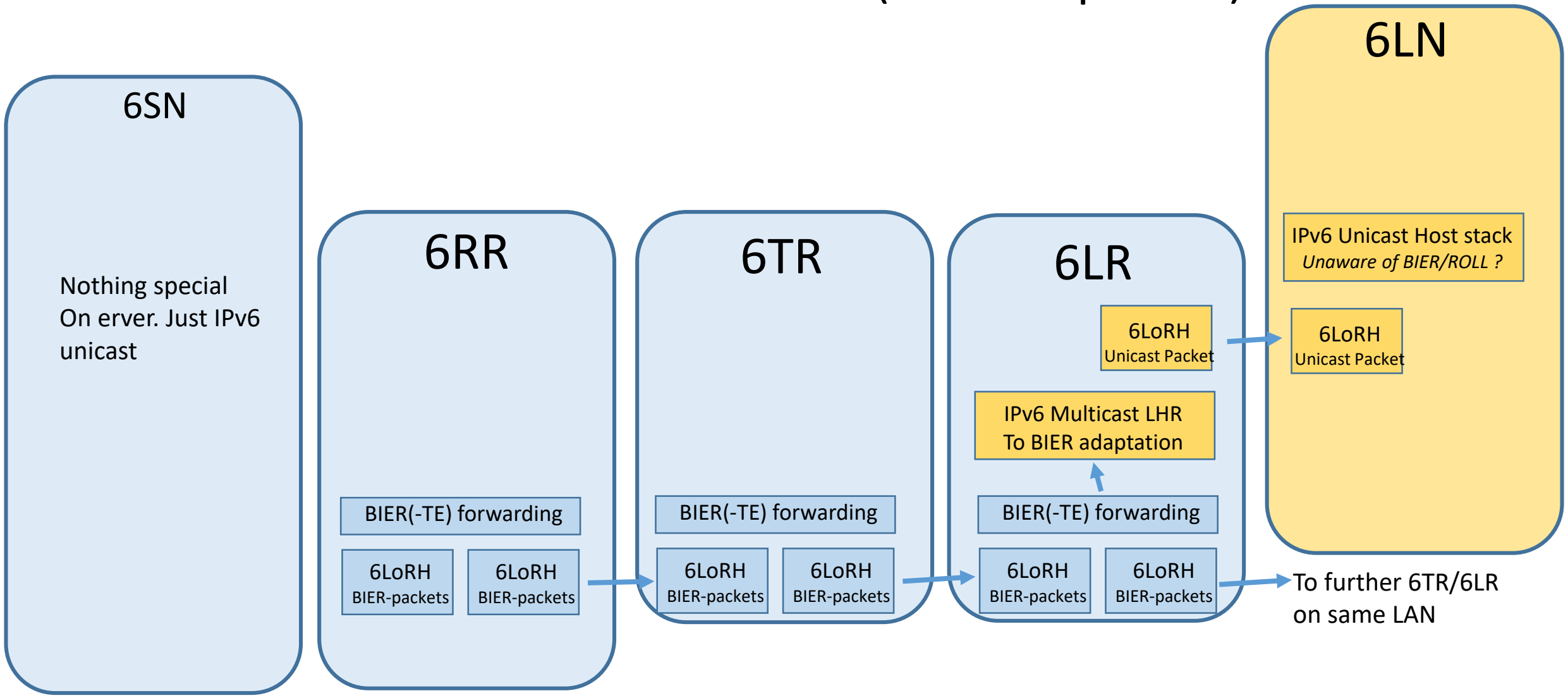
Difference just in how Next-hop entries/FBM for Bits are created



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)

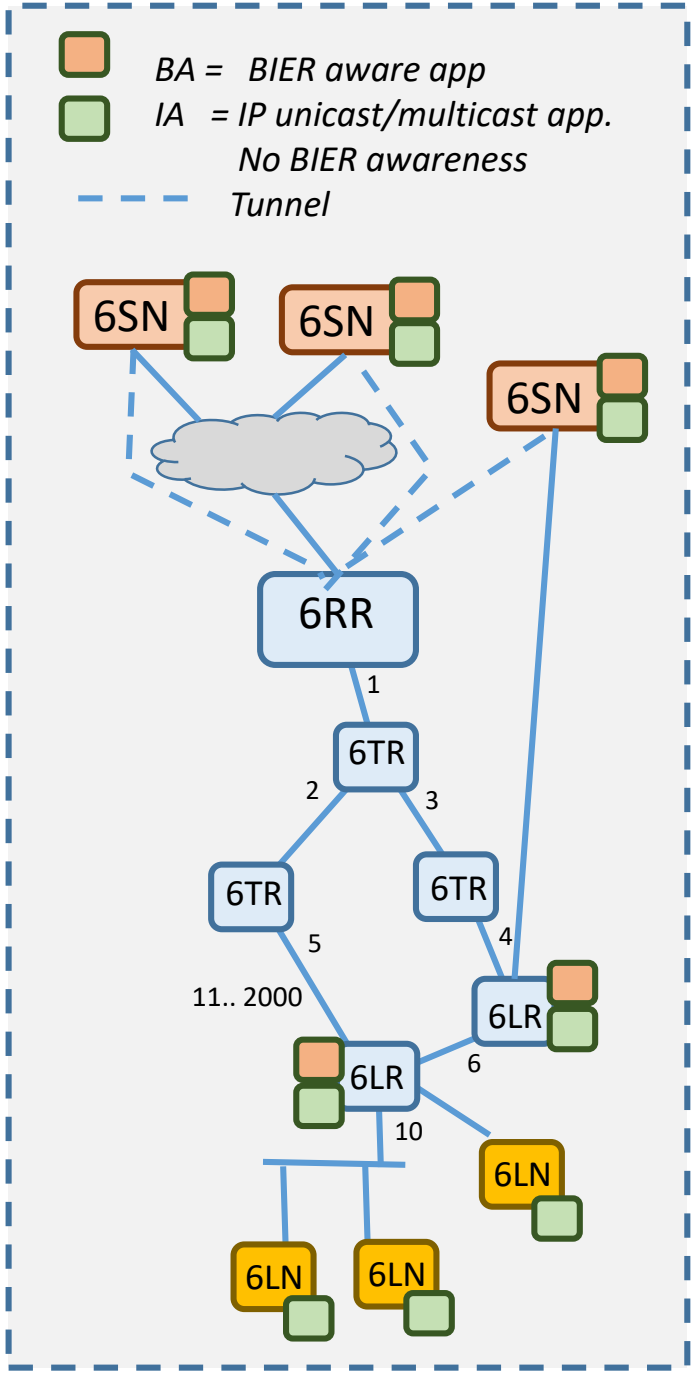
- Using BIER to carry non-IPv6 packets
  - Idea was: BIER packets can directly carry any payload, no IPv6-multicast heade required, eliminate overhead/complexity of IPv6
  - Claim: 6LoRH can compress the IPv6 headers so much that there is not enough initial value in designing header stacks to eliminate 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

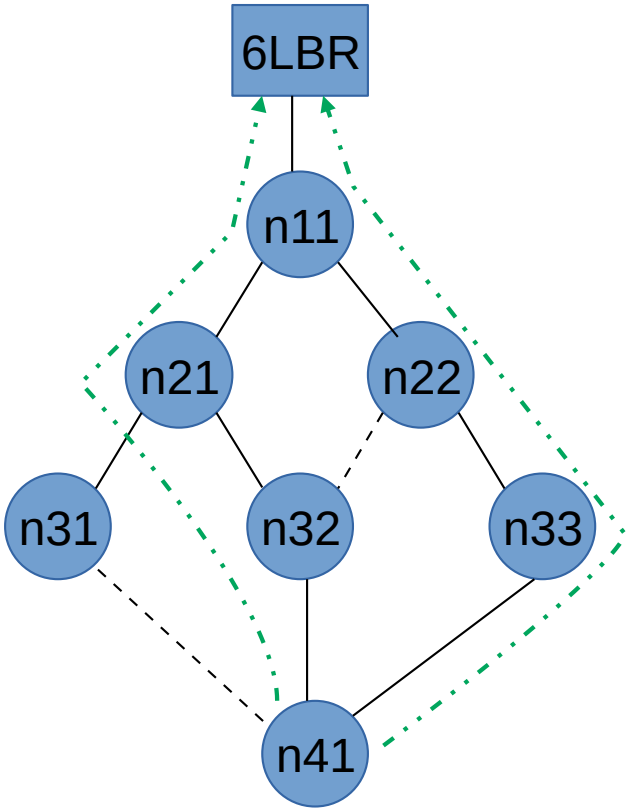
Next steps ?!

- 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.

draft-ietf-roll-efficient-npdao
Updates
IETF103, Bangkok

Updates

- Multiple preferred parent nodes
  - DAOs to multiple preferred parents (used multipath routing for e.g.)
  - DCO handling clarified in this context
  - Example messaging in appendix



Updates

- Security Considerations inline to 6550
- Clarified text for handling of
  - DCOSequence (choosing initial value)
  - “Req#1: Tolerant to link failures...”
    - Georgios and Peter reported ambiguity/confusion in the text. The section was reworded.
- Terminology
  - Consistent usage
  - Duplicated some terms from 6550 for better readability

Acknowledgements

- Major update in -06/07, thanks to Peters review
- Update in -05, Thanks to reviews from Pascal, Georgios and Liubing

RPL Observations

draft-ietf-roll-rpl-observations-00

IETF103, Bangkok

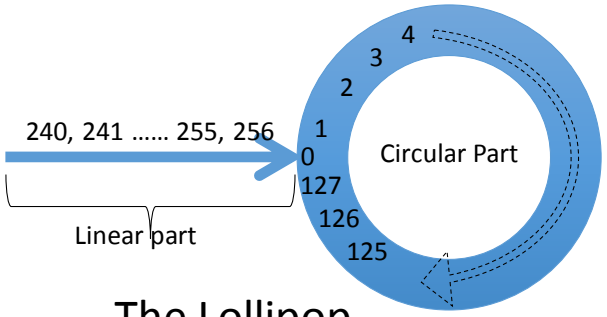
IETF101: Presented first draft

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

IETF103: WG adopted

DTSN: A lollipop counter or no?

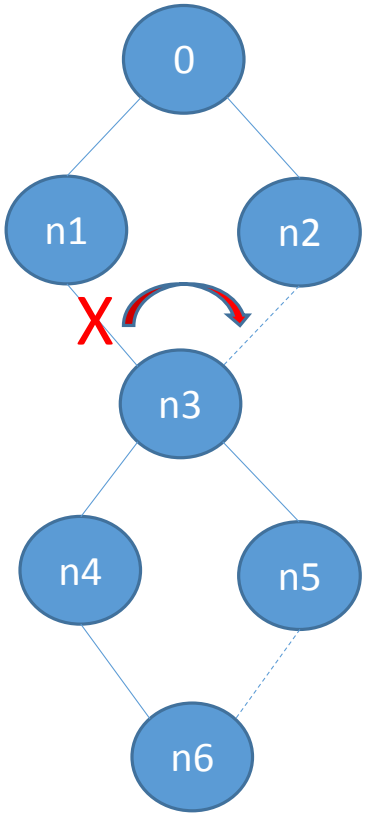
- 6550 says No
  - Section 7 talks about “Sequence Counters” but does not include DTSN as part of it
  - Thanks Michael for pointing this out
- Implementations are clearly confused
  - Contiki considers DTSN as a lollipop counter while RIOT does not
- What should it be?
  - Consider DTSN not to be a 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
- Clarifications needed in 6550



The Lollipop

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

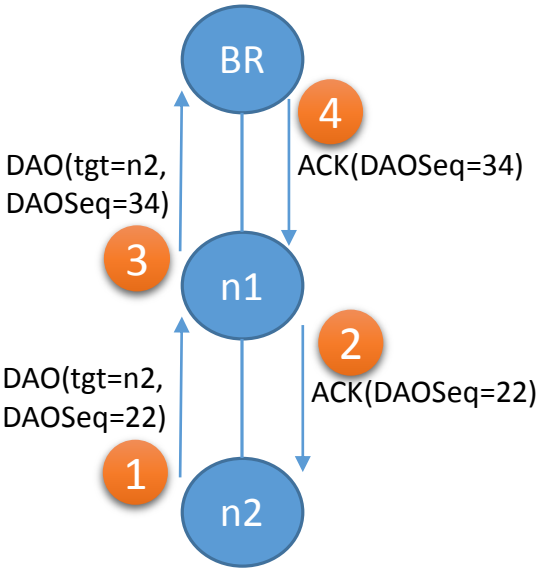
- Should DTSN be incremented with every DIO trickle timer interval?
  - What happens if you do?
    - DAO traffic is too high
  - What happens if you do don’t?
    - DAO redundancy is too low. High probability of DAO not reaching BR.
    - With increase in hops, the probability of DAO success drops sharply.

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



Pros:

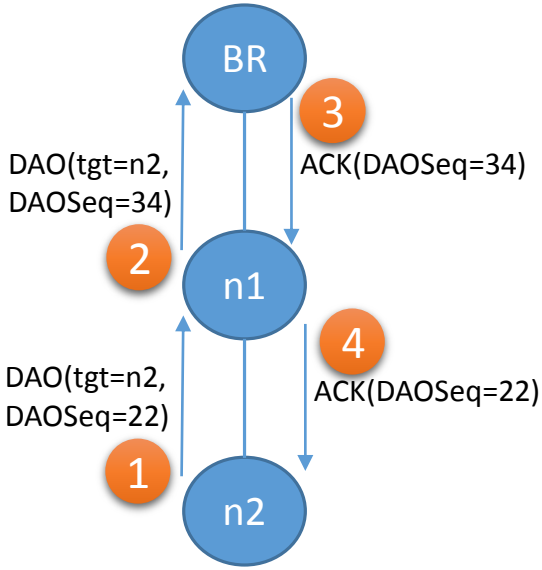
1. No additional RAM
2. Very easy handling. No state.

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.

End-to-End ACK



Pros:

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

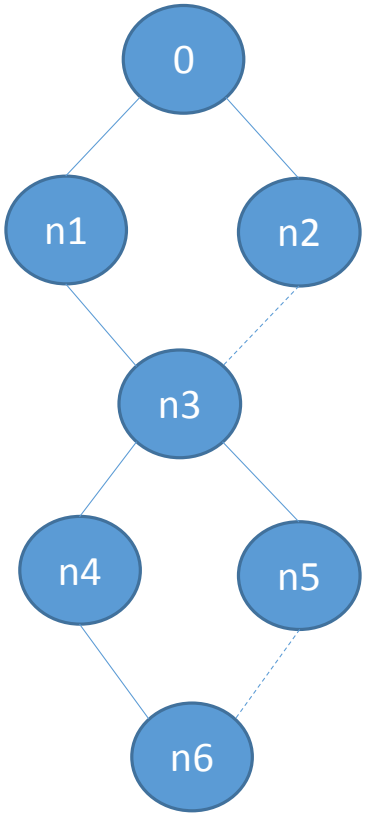
Cons:

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



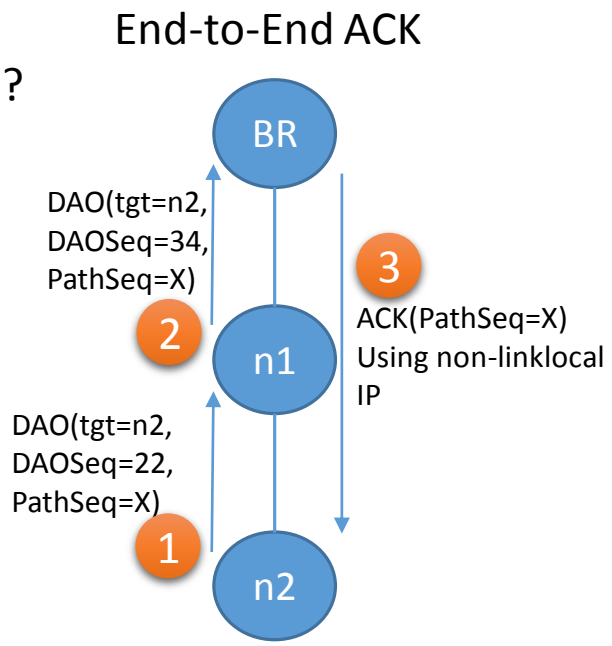
DAO-ACK and aggregated targets: How to ACK?

- 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



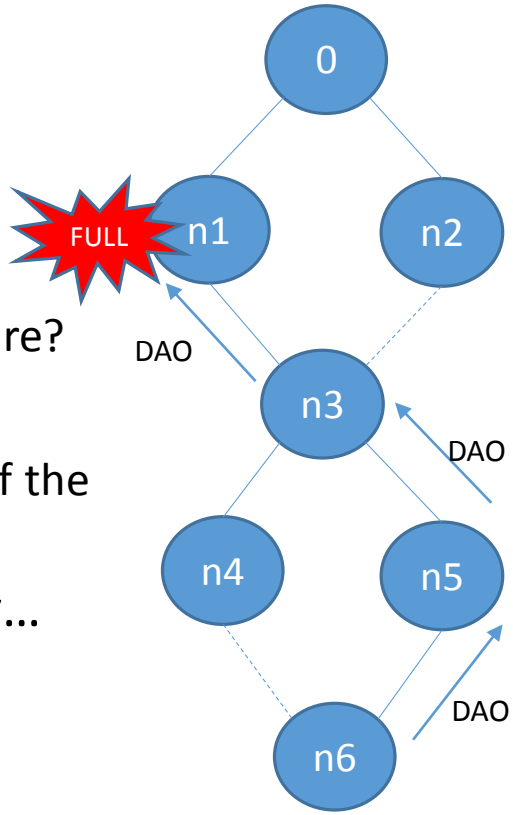
Lot already discussed on ML (Oct 2015):
https://www.ietf.org/mail-archive/web/roll/current/msg09469.html

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
  - Should we inform implementers that not retaining sequence number value in linear part of lollipop counter can cause issues in case of reboots?
- Should deployment provision persistent storage for network stack
  - Even though app does not require any persistent store?

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

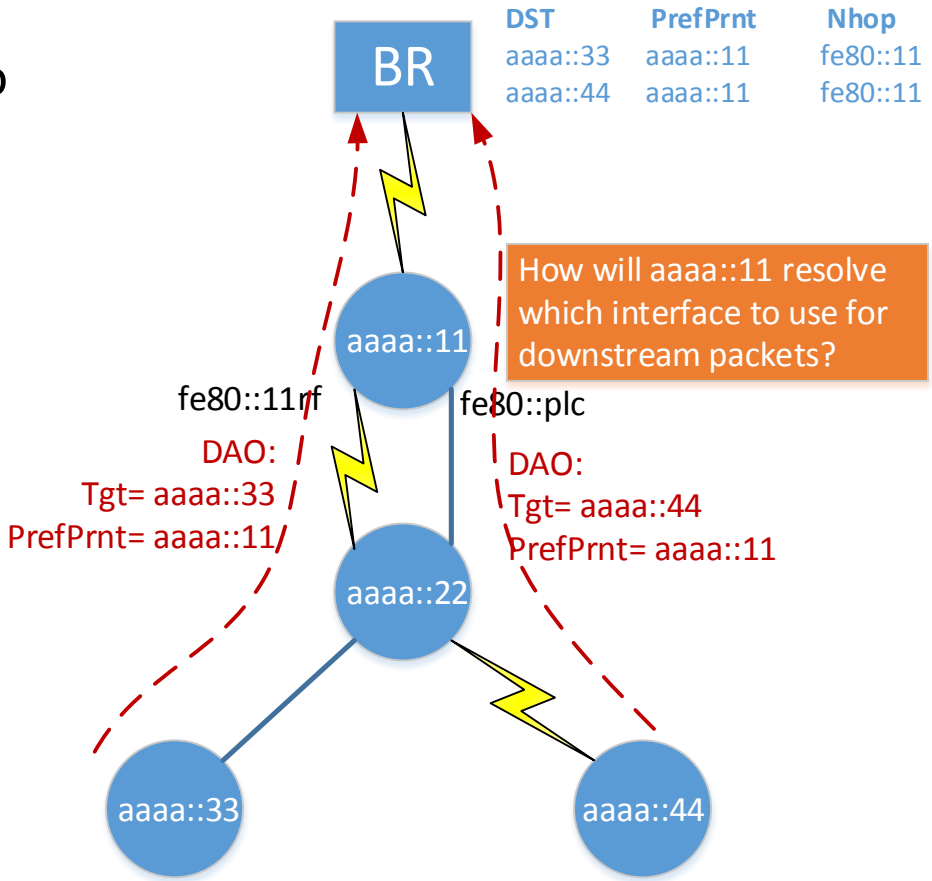
- Should Transit Information be Optional?
  - RPL Transit information carries
    - Path Sequence and Path Lifetime
    - Parent Address (for NS-MOP)
- Aggregated Target Container
  - Aggregation can be optional but should the reception be mandated?

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	6TiSCH has docs which attempts to handle it (?)

Plan

- Next level observations related to
  - For e.g. use of multiple link layer
- The plan is to provide data/stats with the scenarios



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

draft-ietf-roll-aodv-rpl-05

IETF 103, Bangkok

Satish Anamalamudi <satishnaidu80@gmail.com>

Mingui Zhang <zhangmingui@huawei.com>

Charlie Perkins <charles.perkins@earthlink.net>

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

Liu Bing <remy.liubing@huawei.com>

Sequence Numbers

- Why not use DODAGVersionNumber and DTSN as the SeqNo?
  - The VersionNumber is always set to 0 (as in RFC 6997). Once a route is discovered, there is no need to keep the DODAG.
  - The DAO message is not used in AODV-RPL
- Each router maintains its own SeqNo (8 bits lollipop counter)
- OrigNode
  - Increase the SeqNo when it needs to find a new route
  - RREQ-DIO
    - OrigSeq: the incremented SeqNo of OrigNode
    - TargSeq: the SeqNo of TargNode, if known
- Intermediate router (hop-by-hop mode)
  - Update the related route entry when SeqNo is larger
- TargNode
  - The TargNode owns TargSeq, does not need to update it
  - Include its SeqNo in RREP-DIO

Route Lifetime / Invalidation

- The Lifetime values (‘L’ field) have been increased
- Source routing mode
  - Address vectors only exist at the OrigNode and TargNode, and can be refreshed by the Sequence Numbers
- Hop-by-hop mode
  - Case 1: for common routers on the old and the new paths, route entries are updated when receiving RREP/RREQ-DIOs with larger SeqNo.
  - Case 2: for routers only on the old path, route entries wait for expiration.
  - Active invalidation: not necessary
    - Redundant in case 1
    - In case 2, due to unreachability, invalidation message can’t clear all the stale route entries. Thus a lifetime is still needed.

Traffic-Aware Objective Function

draft-ji-roll-traffic-aware-objective-function-03

Chenyang Ji
Remous-Aris Koutsiamanis aris@ariskou.com
Georgios Z. Papadopoulos
Diego Dujovne
Nicolas Montavont

ROLL@IETF103

New since -01

- Changed value used in MC
  - Previously: Packet Transmission Rate (PTR)
    Now: Remaining Throughput (RT)
- Addressed parent and DODAG selection
- Defined configuration of time window for RT calculation
- Described use within enrollment
- Minor:
  - IANA registration requests

Standardisation Efforts

- Objective Function → Preferred Parent
  - OF0
    - ETX
    - No hysteresis
  - MRHOF
    - ETX, Energy, other additive metrics
    - Hysteresis
  - *Load balanced OF (LB-OF)*
    - Uses parent's child-count
    - Hysteresis

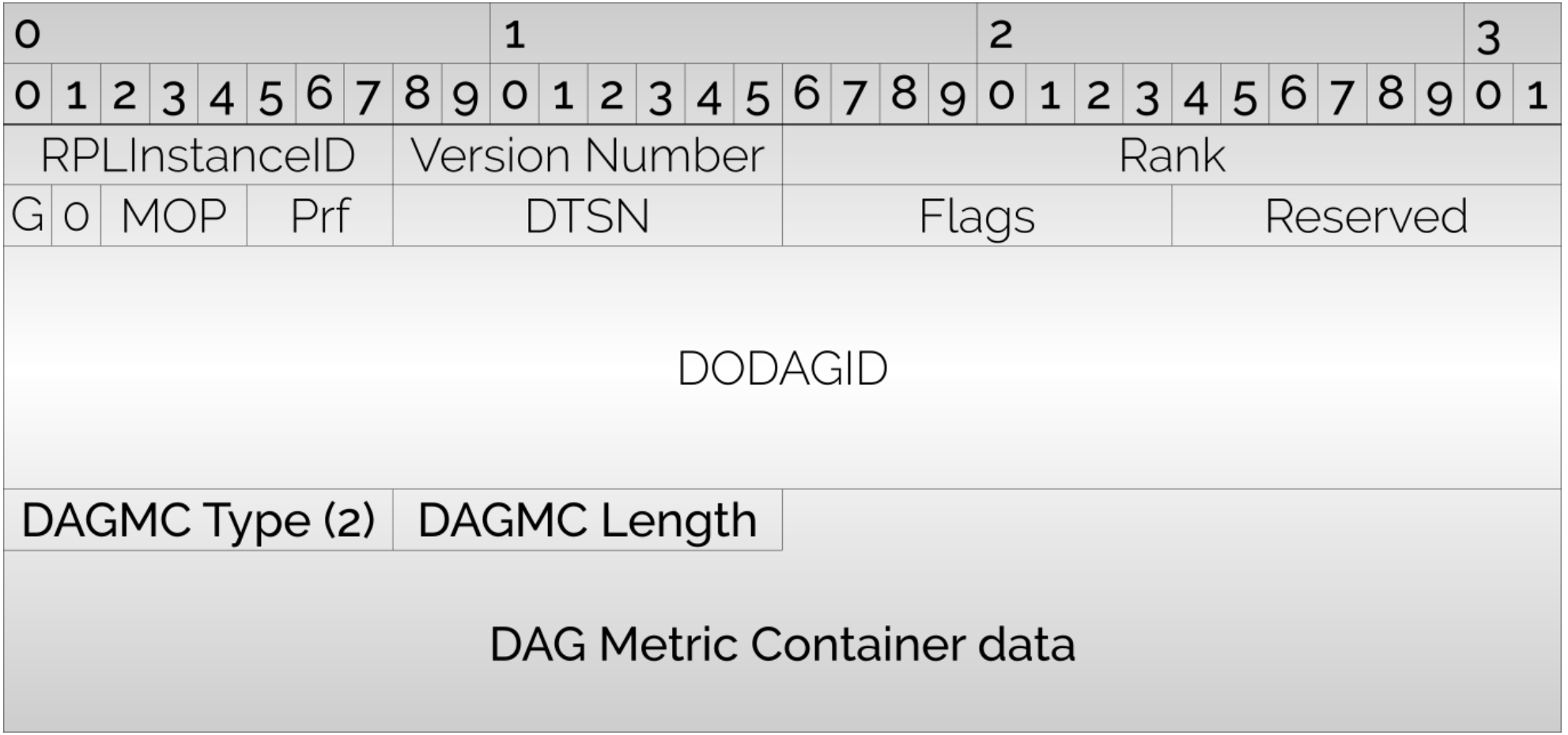
Problem statement

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

Traffic-Aware OF

- New Remaining Throughput (RT) metric
  - Aggregated → Use DAGMC.A=MINIMUM
  - Sliding window tracking of used throughput capacity
    - Window size: THROUGHPUT_PERIOD (time)
  - Calculate RT (in THROUGHPUT_PERIOD)
    - Total Throughput Capacity – Used Throughput
- Traffic-Aware OF
  - Highest RT → Preferred parent
  - Use with MRHOF-style hysteresis

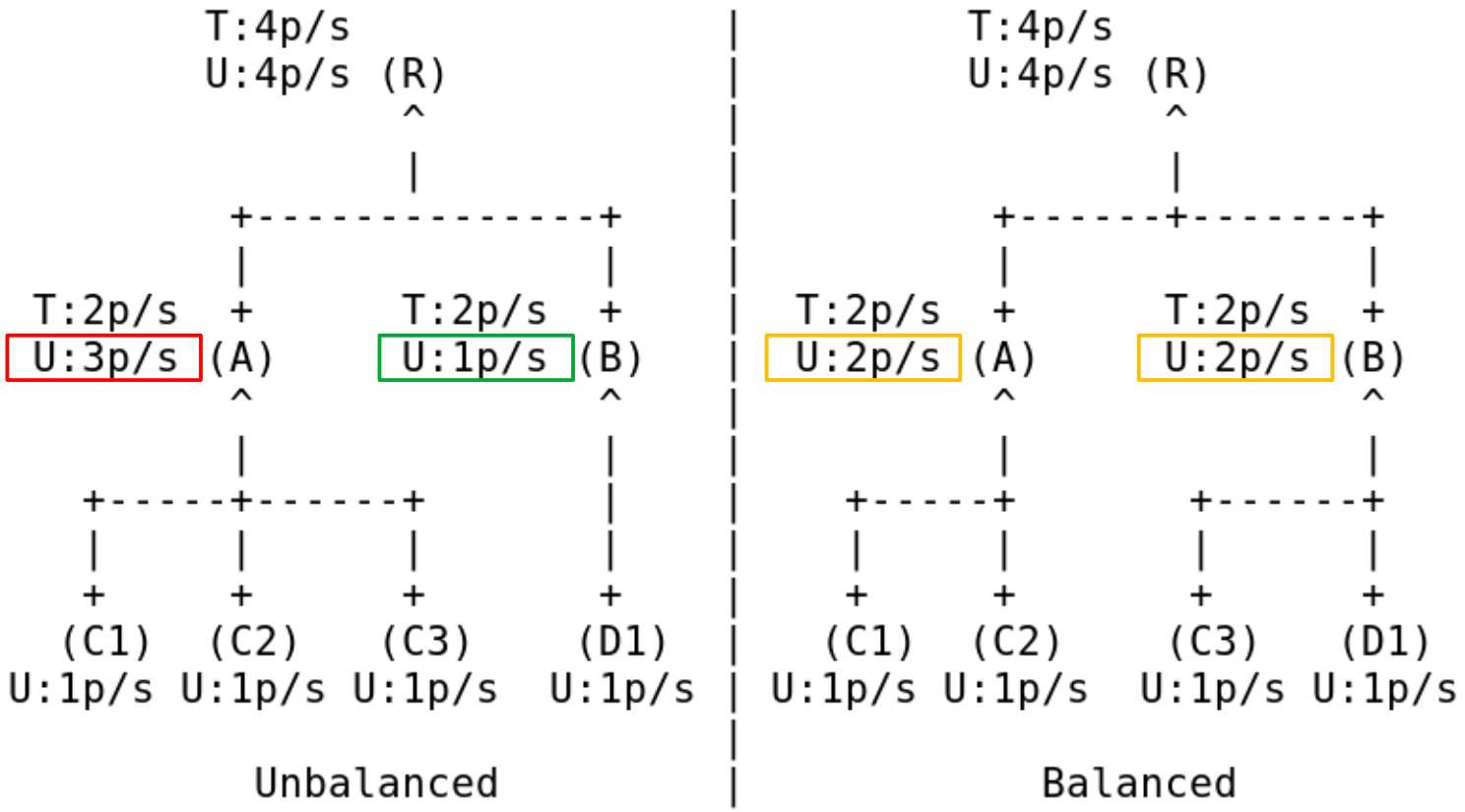
DIO Format Example



Remaining Throughput

- **Node Metric Object (RT)**
  - **2 octets – unsigned integer**
  - **A = 2 = Minimum**

Parent Selection Example (1)

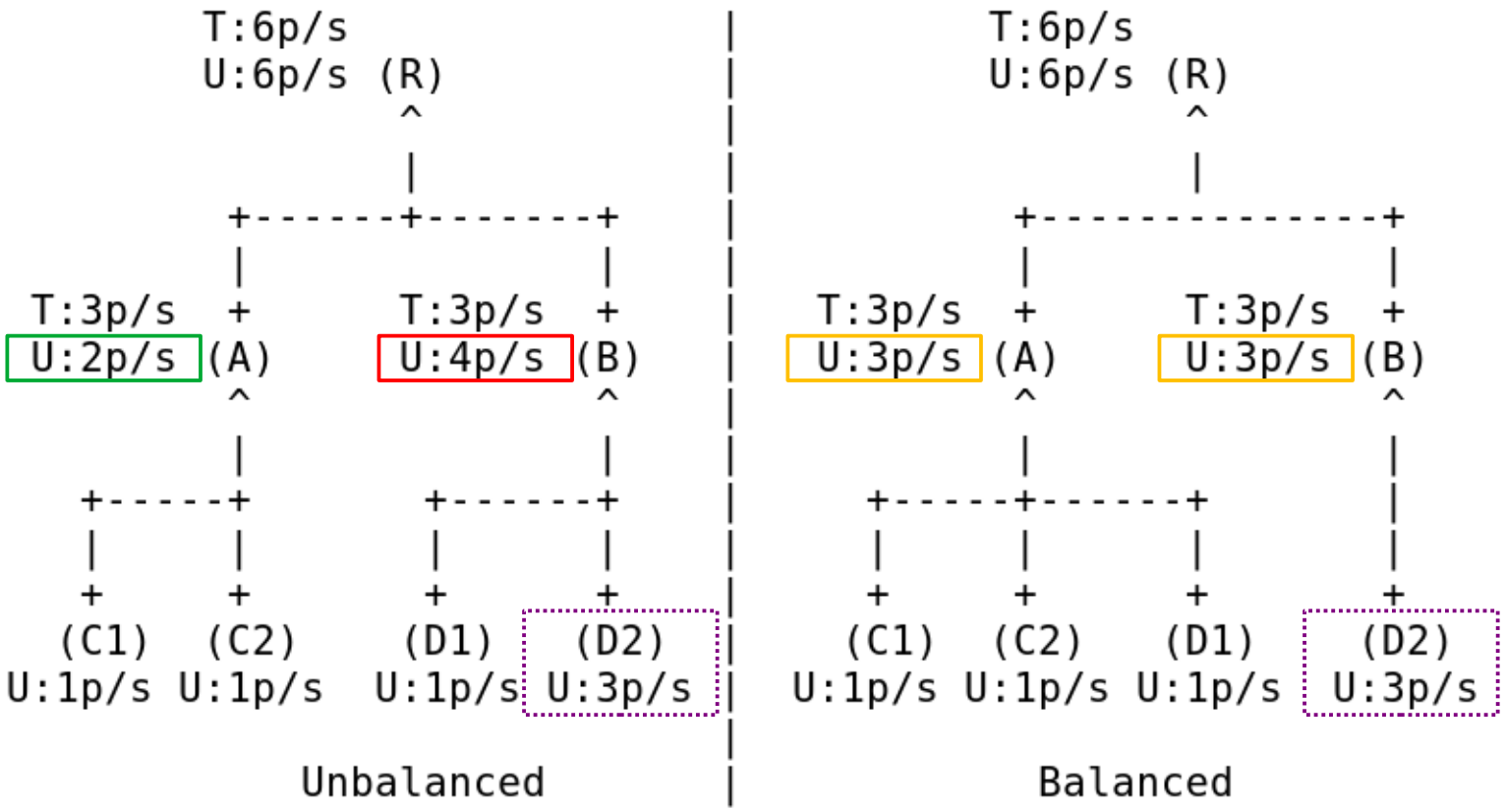


T → Total Throughput Capacity

U → Used Throughput Capacity

- **Nodes with the same TX requirements**

Parent Selection Example (2)

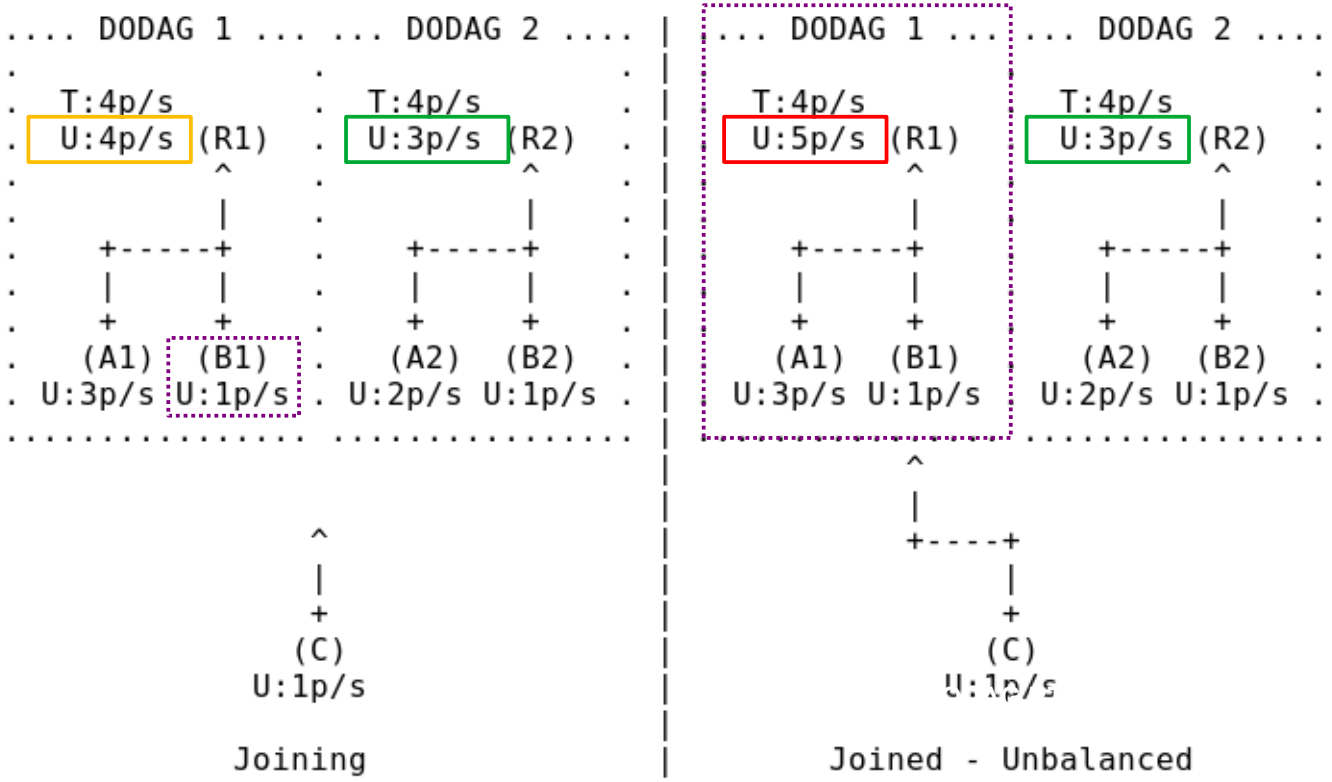


T → Total Throughput Capacity

U → Used Throughput Capacity

- Nodes with **different** TX requirements but **same child count**
- **LB-OF** draft **will not balance** this example

DODAG Selection Example (1)

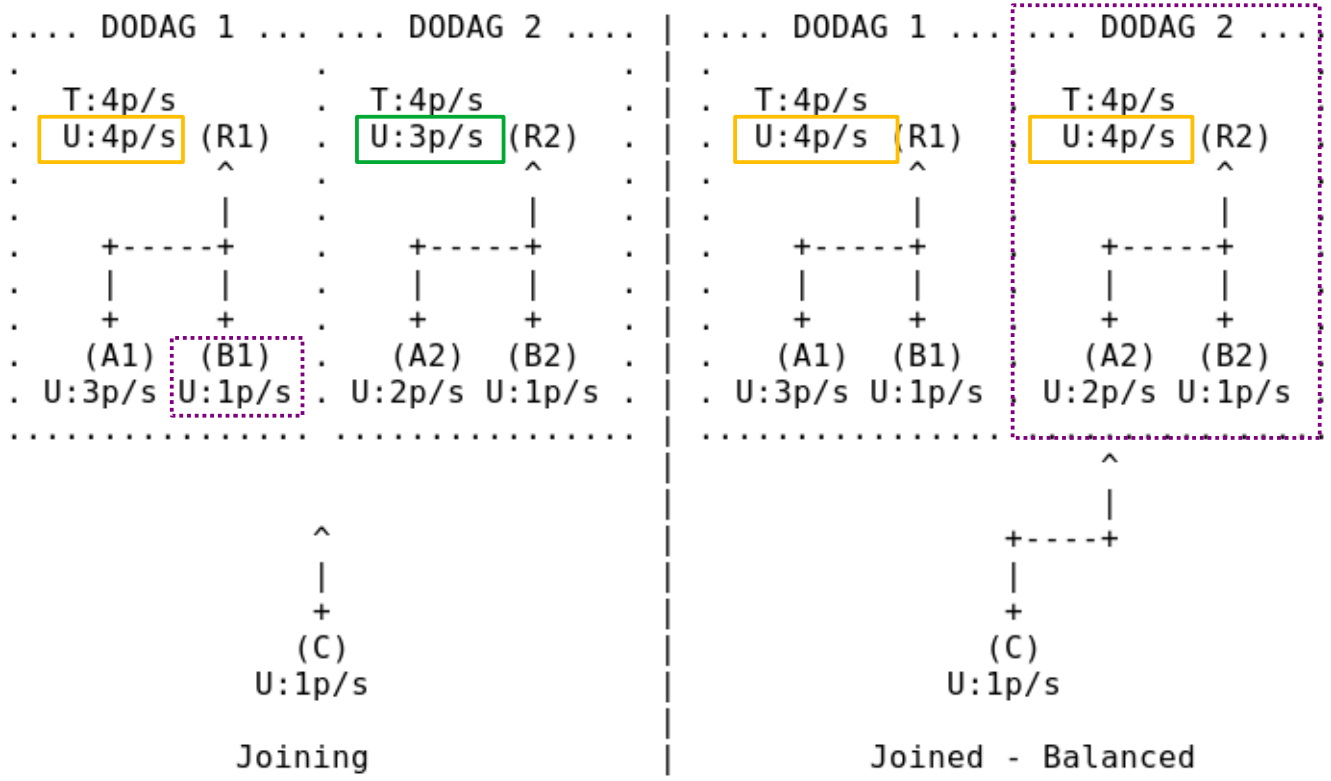


T → Total Throughput Capacity

U → Used Throughput Capacity

- DODAG selection **without RT** leading to **unbalanced traffic**

DODAG Selection Example (2)

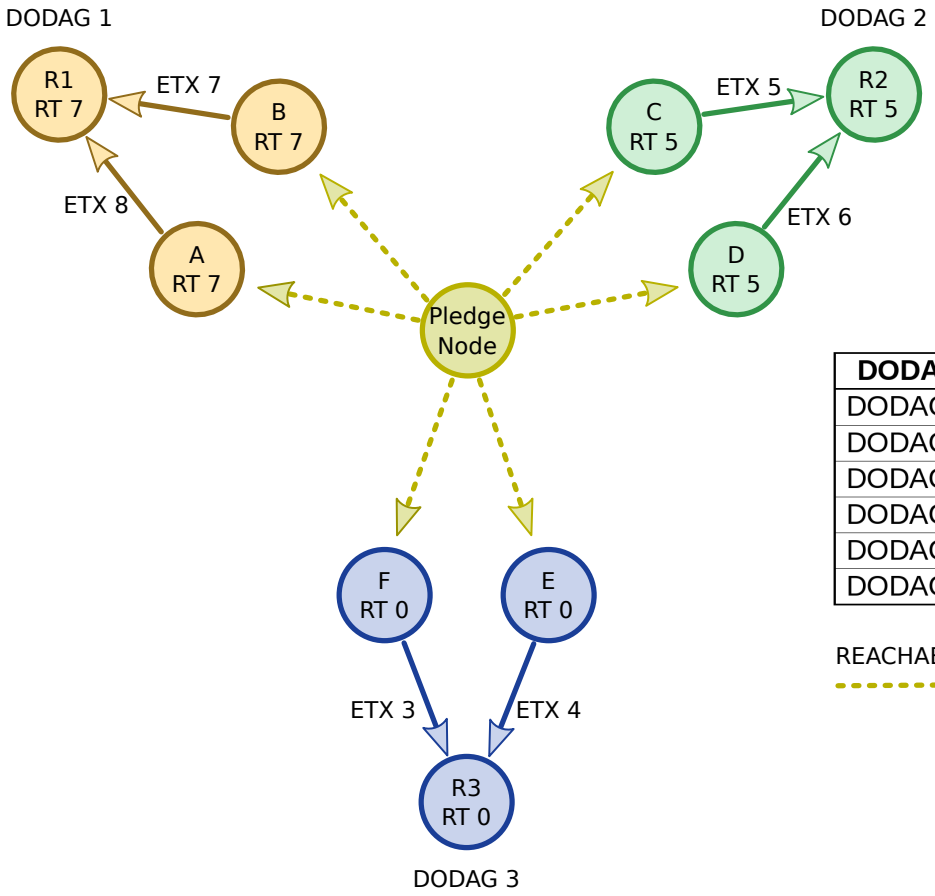


T → Total Throughput Capacity

U → Used Throughput Capacity

- DODAG selection with **RT** leading to **balanced traffic**

Multi-Metric DODAG Selection Example (1)

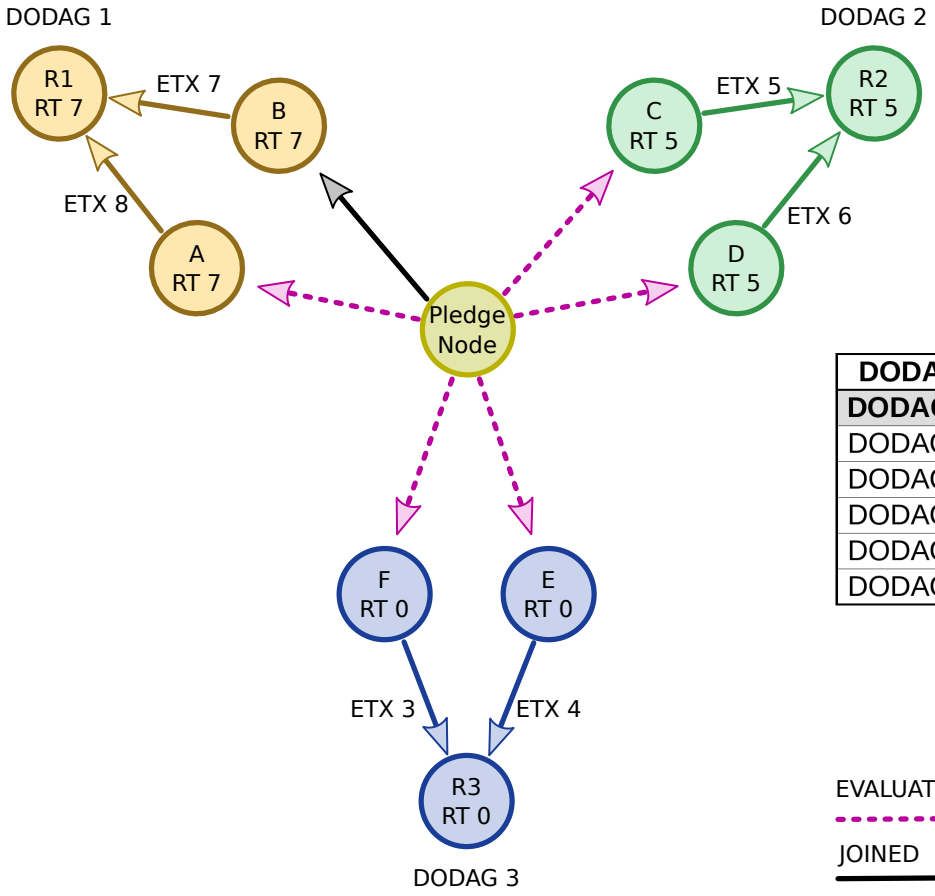


DODAG	Node	RT	ETX
DODAG 1	A	7	8
DODAG 1	B	7	7
DODAG 2	C	5	5
DODAG 2	D	5	6
DODAG 3	E	0	3
DODAG 3	F	0	4

REACHABLE

- Pledge node needs to pick a DODAG

Multi-Metric DODAG Selection Example (2)

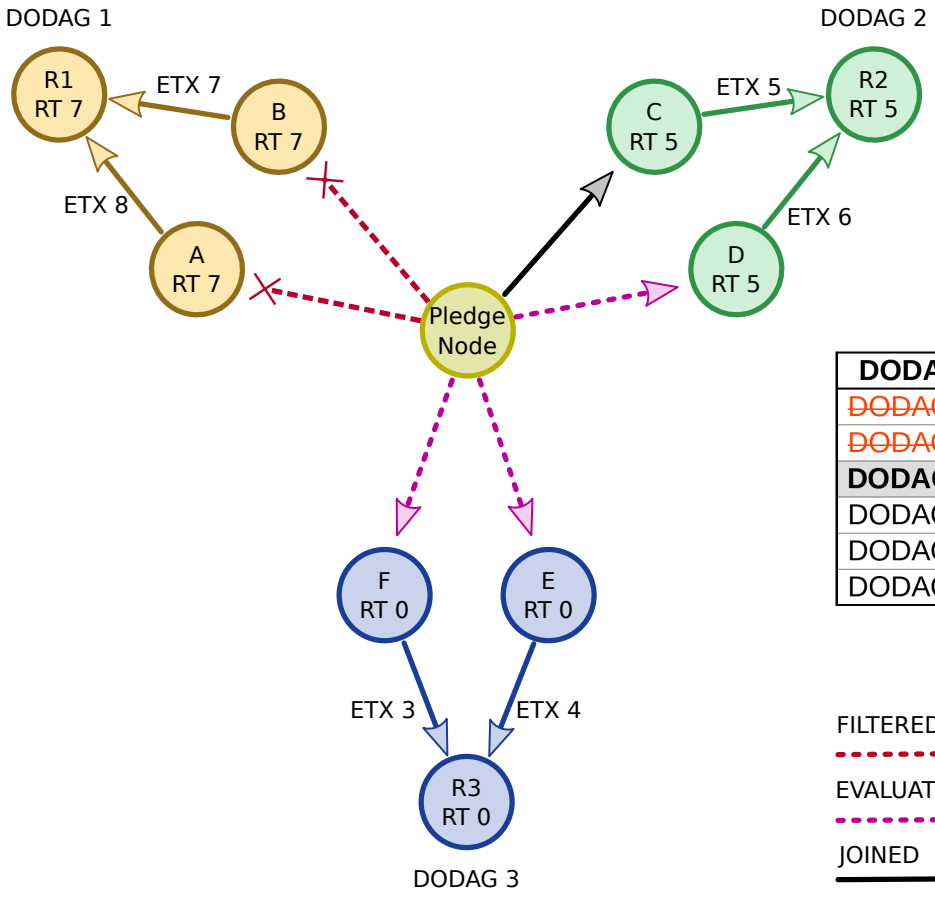


DODAG	Node	RT	ETX
DODAG 1	**B**	**7**	**7**
DODAG 1	A	7	8
DODAG 2	C	5	5
DODAG 2	D	5	6
DODAG 3	E	0	3
DODAG 3	F	0	4

Precedence	
0	1
RT	ETX

- Use **Prec** in MC to **sort** potential parent by **RT** and then by **ETX**

Multi-Metric DODAG Selection Example (3)

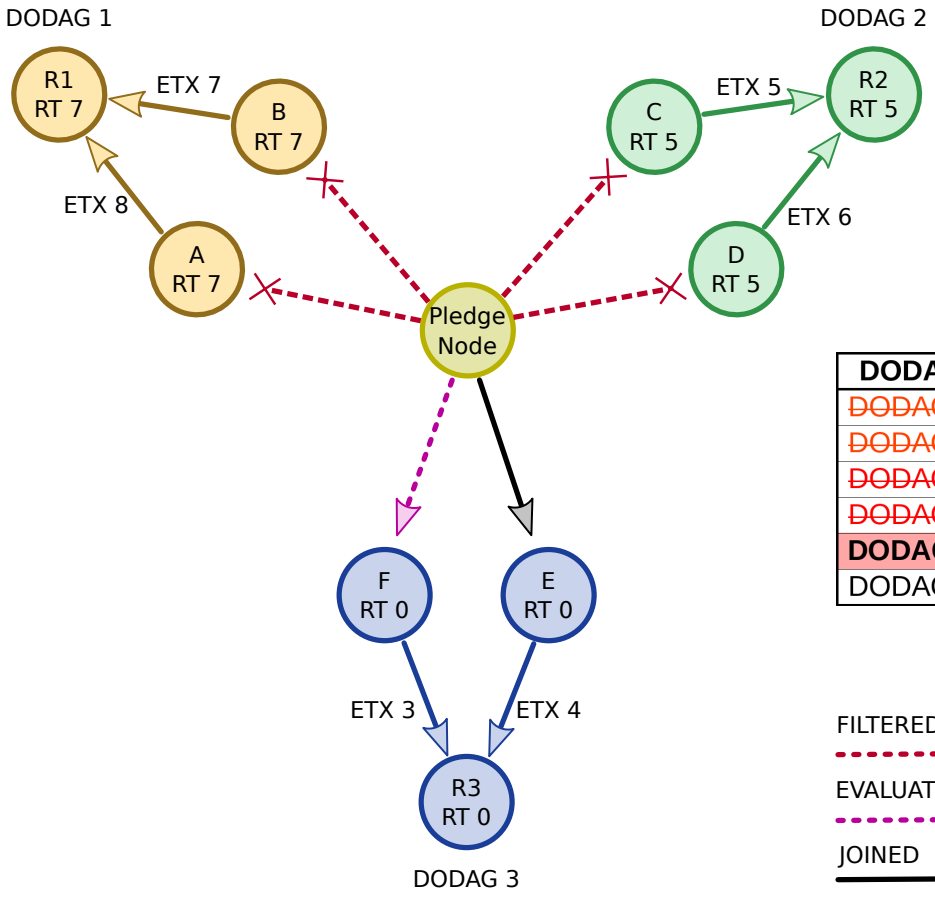


DODAG	Node	RT	ETX
~~DODAG 1~~	~~B~~	~~7~~	~~7~~
~~DODAG 1~~	~~A~~	~~7~~	~~8~~
DODAG 2	**C**	**5**	**5**
DODAG 2	D	5	6
DODAG 3	E	0	3
DODAG 3	F	0	4

Precedence		Constraint
0	1	0
RT	ETX	ETX <7

- Filter out **bad ETX** values with constraint on ETX

Multi-Metric DODAG Selection Example (4)

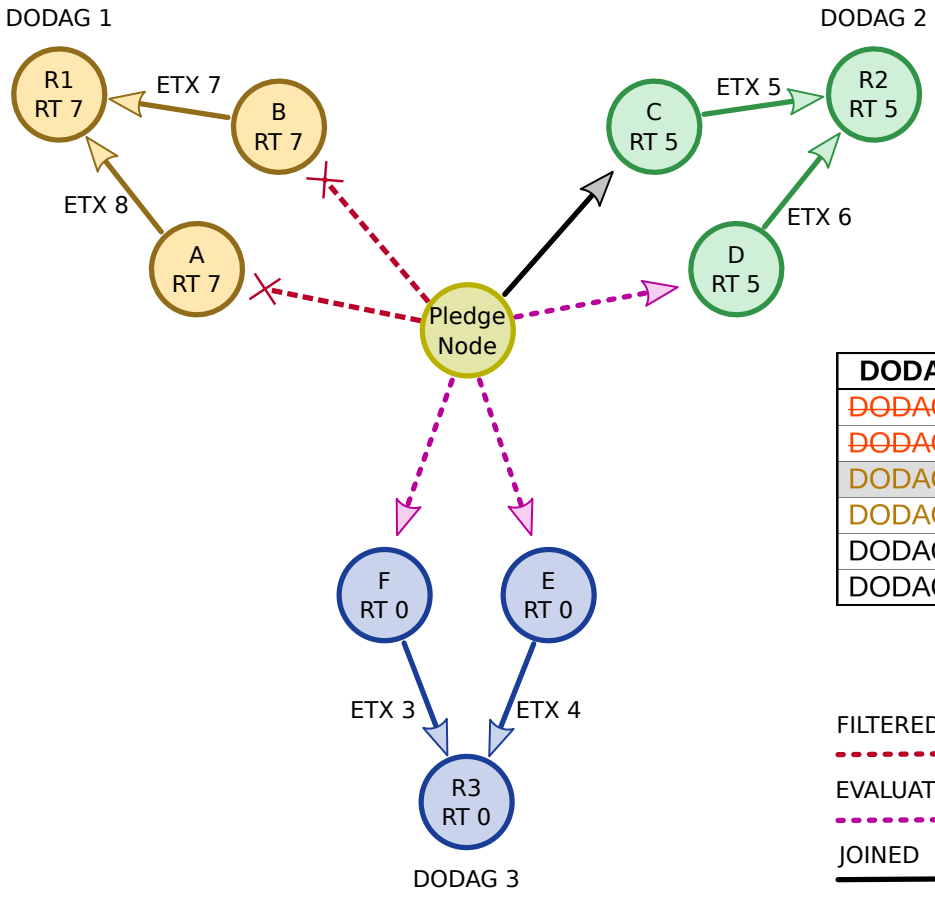


DODAG	Node	RT	ETX
~~DODAG 1~~	~~B~~	~~7~~	~~7~~
~~DODAG 1~~	~~A~~	~~7~~	~~8~~
~~DODAG 2~~	~~C~~	~~5~~	~~5~~
~~DODAG 2~~	~~D~~	~~5~~	~~6~~
DODAG 3	**E**	**0**	**3**
DODAG 3	F	0	4

Precedence		Constraint
0	1	0
RT	ETX	**ETX <5**

- **Too restrictive** constraint leads to **no potential** DODAGs available

Multi-Metric DODAG Selection Example (5)



DODAG	Node	RT	ETX
~~DODAG 1~~	~~B~~	~~7~~	~~7~~
~~DODAG 1~~	~~A~~	~~7~~	~~8~~
DODAG 2	C	5	5
DODAG 2	D	5	6
DODAG 3	E	0	3
DODAG 3	F	0	4

Precedence		*Optional* Constraint
0	1	0
RT	ETX	***ETX <5***

FILTERED

EVALUATED

JOINED

- Use **optional flag** in constraint to have **fallback** in case of no DODAGs available with constraint

Enrollment – Layer 2

- Join network (EB)
  - *draft-richardson-6tisch-enrollment-enhanced-beacon-01*
- Report PAN priority in EB
  - Should it be reported in rank priority?
  - PAN priority = 16 - FLOOR(LOG2(RT + 1))
  - PAN priority is opaque → exact reporting of RT not required
  - LOG2: “compress” the **16-bit RT** into **8-bit PAN priority**
    - Higher accuracy in low RT values, lower accuracy in high RT values
    - RT=1 vs RT=3 → Big difference
    - RT=51 vs RT=53 → Small difference
    - Implementation: 5 shifts, 4 ORs, 1 lookup from 16-byte table

Enrollment – RPL

- Initial DODAG selection
  - Directly pick DODAG with highest remaining throughput capacity
- Use in conjunction with other metrics
  - e.g. ETX

Road forward

- Reviewed by **Derek** and addressed **comments**
- **Presented paper** with results in the GIIS 2018 conference
- **Contiki implementation** close to release

- Next steps
  - Other reviewers?
  - Other changes?
  - Adoption?

Thanks!
Questions?

RPL DAG Metric Container Node State and Attribute object type extension

draft-koutsiamanis-roll-nsa-extension-03

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

ROLL@IETF102

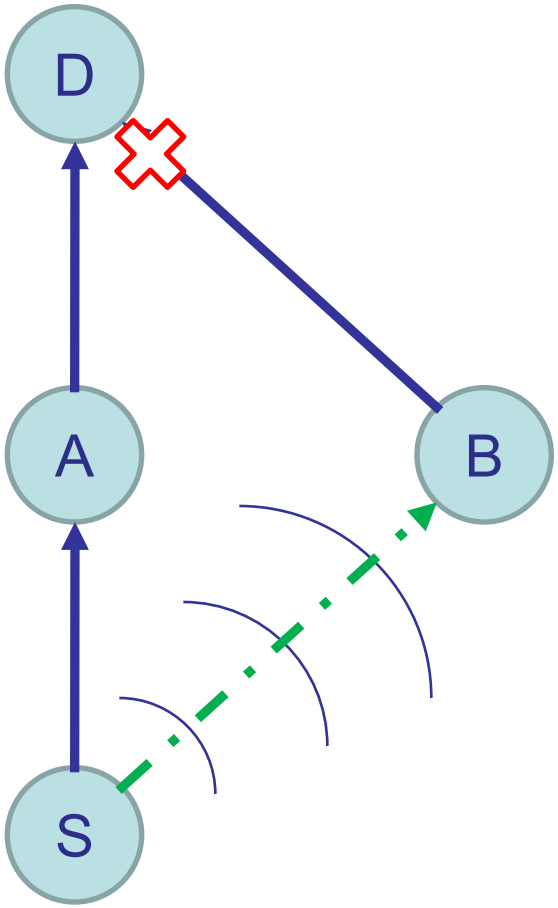
Updates on -03 version

- Reviews from Diego are applied
  - Thank you Diego

- More results

PRE over 6TiSCH

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

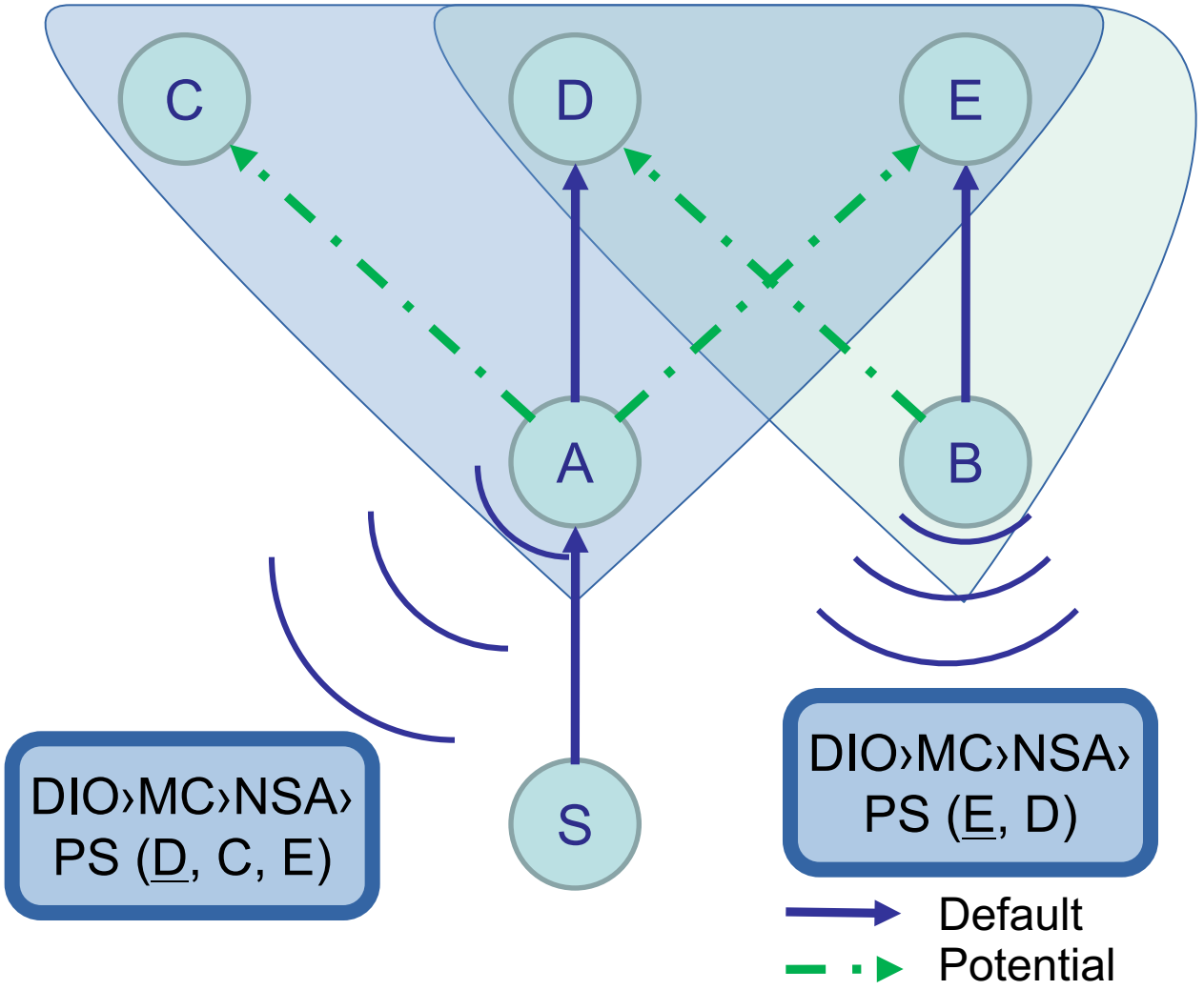


Alternative Parent Selection

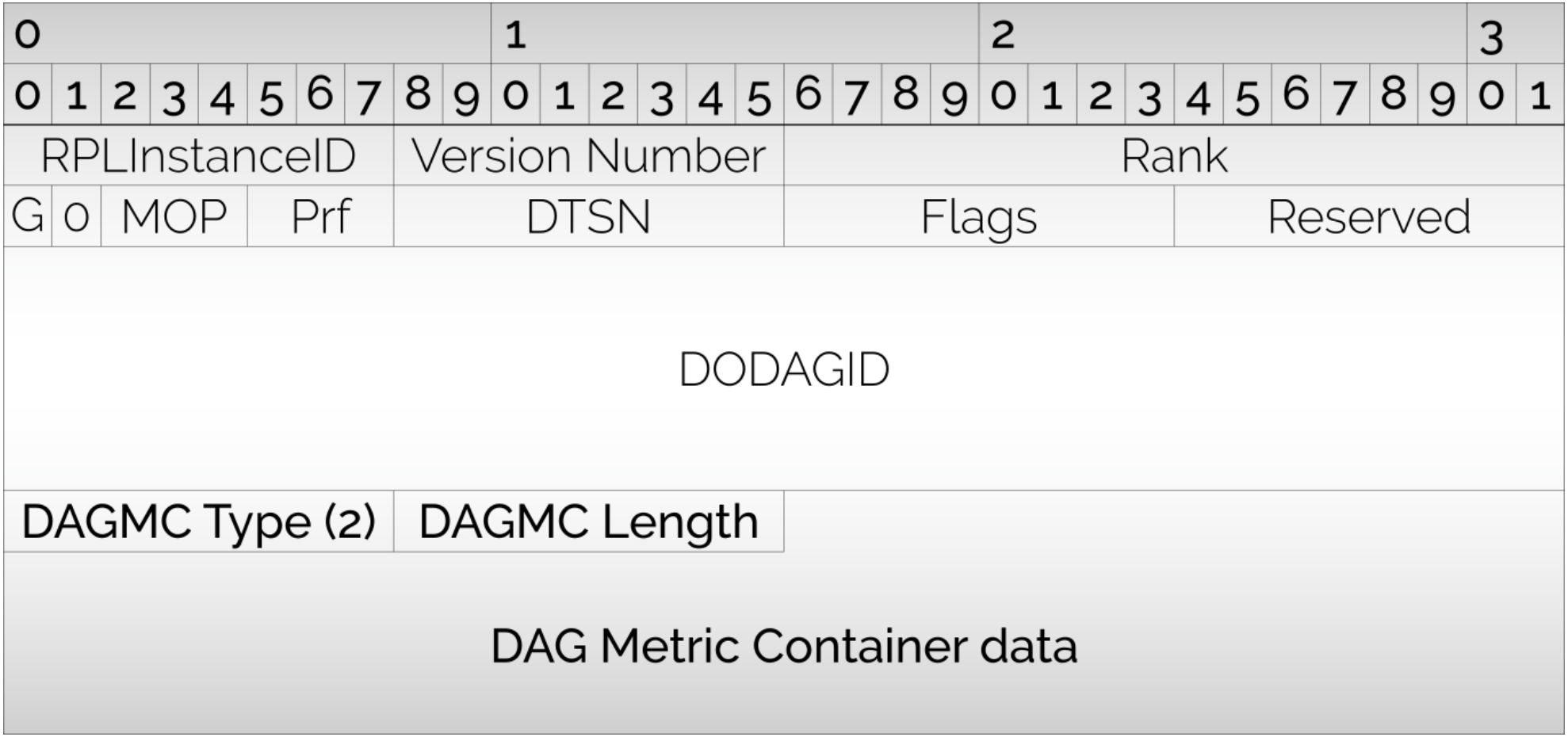
Allows selecting alternative parent with Common Ancestor

Parent Selection - DIO Messages

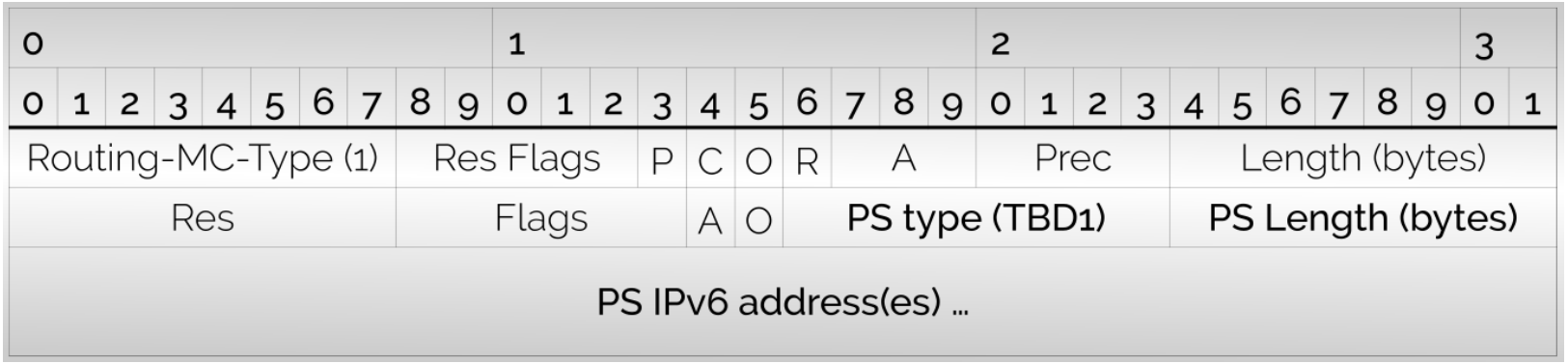
- Parent Set A:
  - {D, C, E}
- Parent set B:
  - {E, D}



DIO Format Example



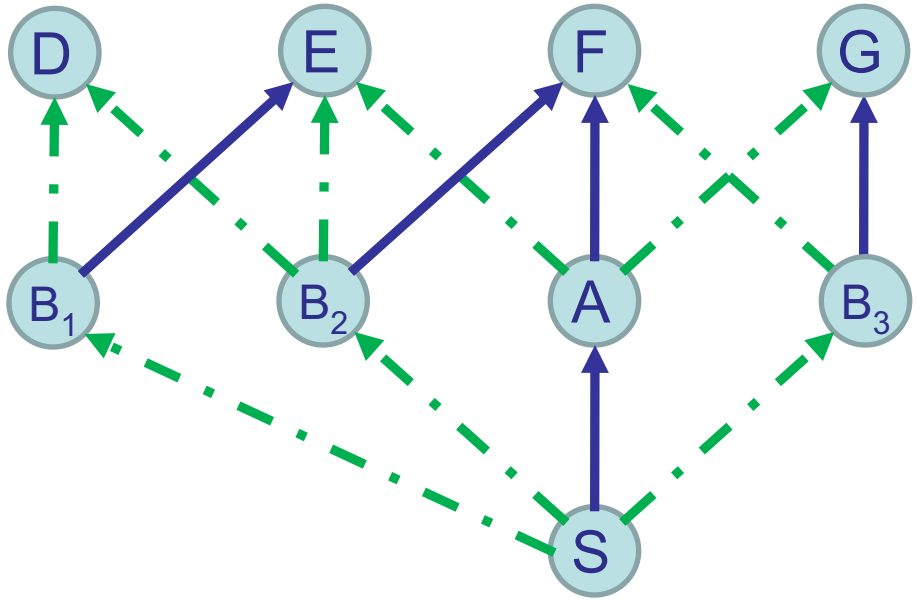
MC Format Example (1)



- 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 : Strict

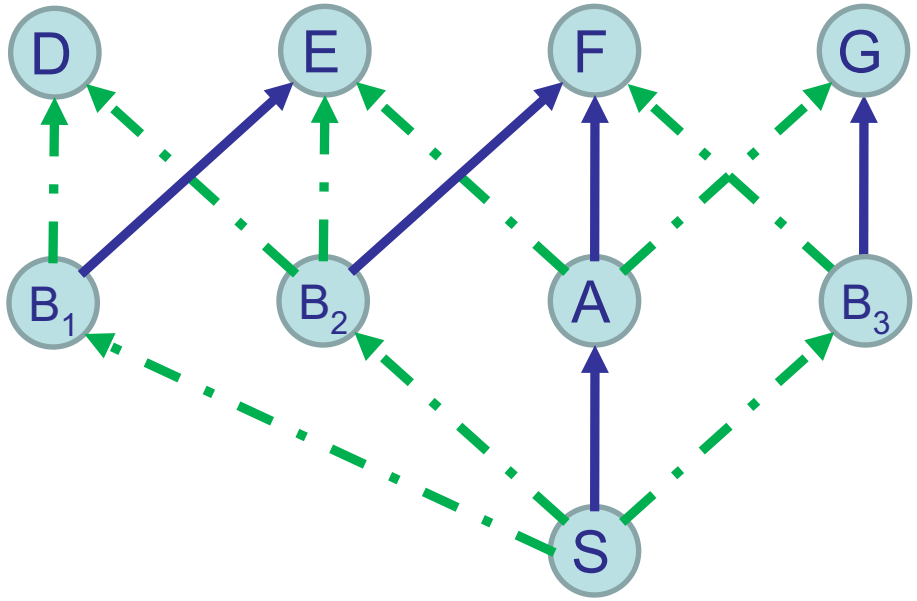
- PP(PP) = PP(AP)
  - PP(A) = F
  - PP(B_1) = E ✖
  - PP(B_2) = F ✔
  - PP(B_3) = G ✖



Preferred Parent (PP)
Alternative Parent (AP)

Implementation of CA : Medium

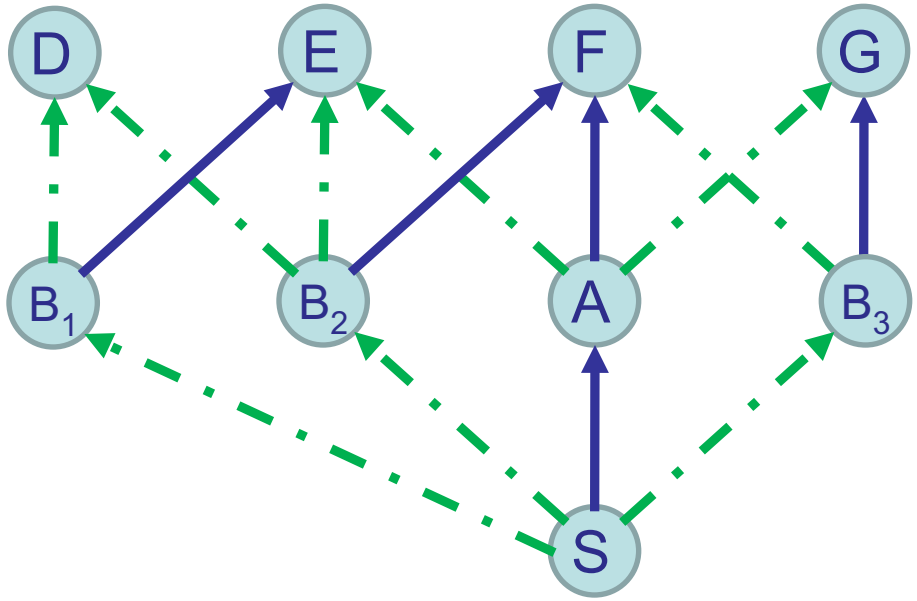
- PP(PP) ∈ PS(AP)
  - PP(A) = F
  - PS(B_1) = (E, D) ✖
  - PS(B_2) = (F, D, E) ✔
  - PS(B_3) = (G, F) ✔



Preferred Parent (PP)
Alternative Parent (AP)

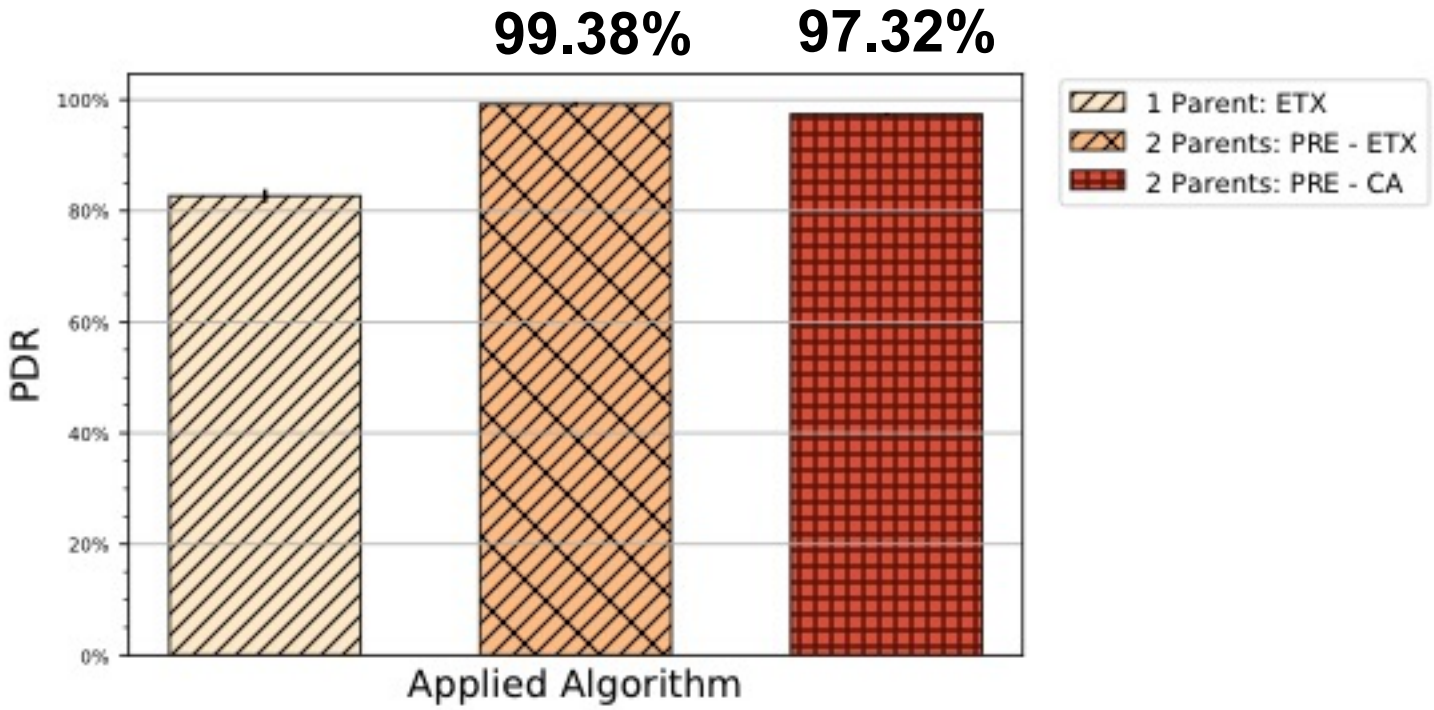
Implementation of CA : Relaxed

- PS(PP) ∩ PS(AP) ≠ ∅
  - PS(A) = (F, E, G)
  - PS(B_1) = (E, D) ✔
  - PS(B_2) = (F, D, E) ✔
  - PS(B_3) = (G, F) ✔



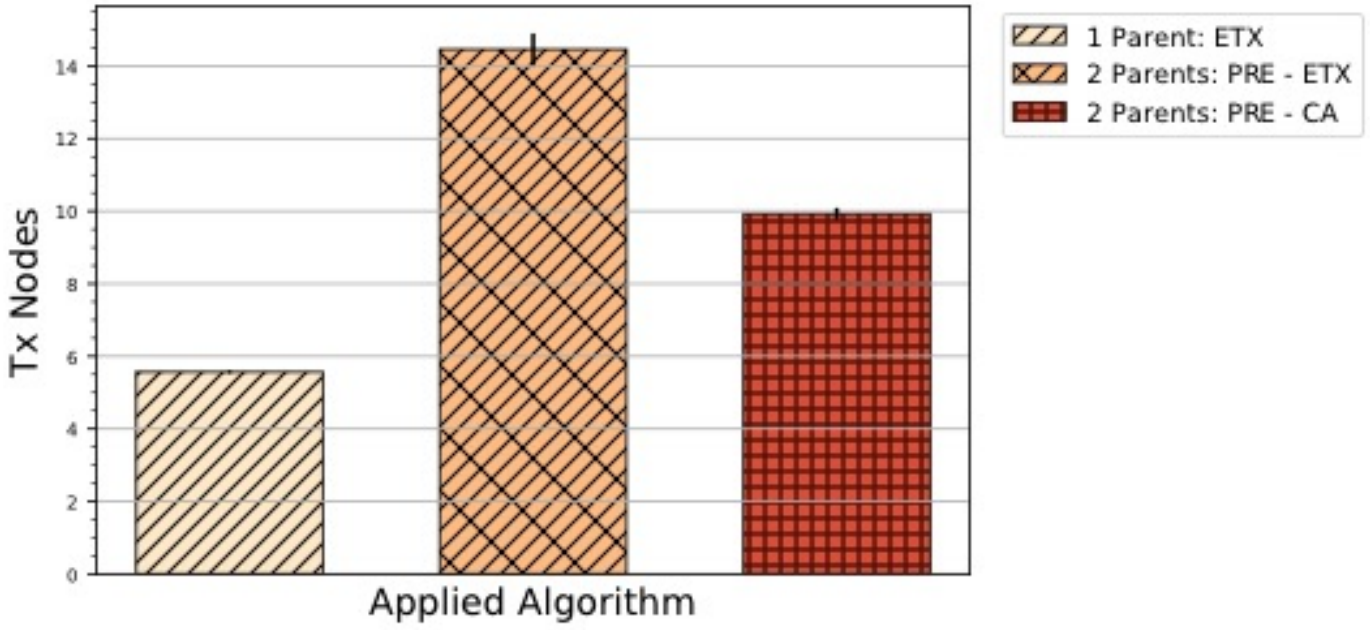
Preferred Parent (PP)
Alternative Parent (AP)

Strict CA: PDR



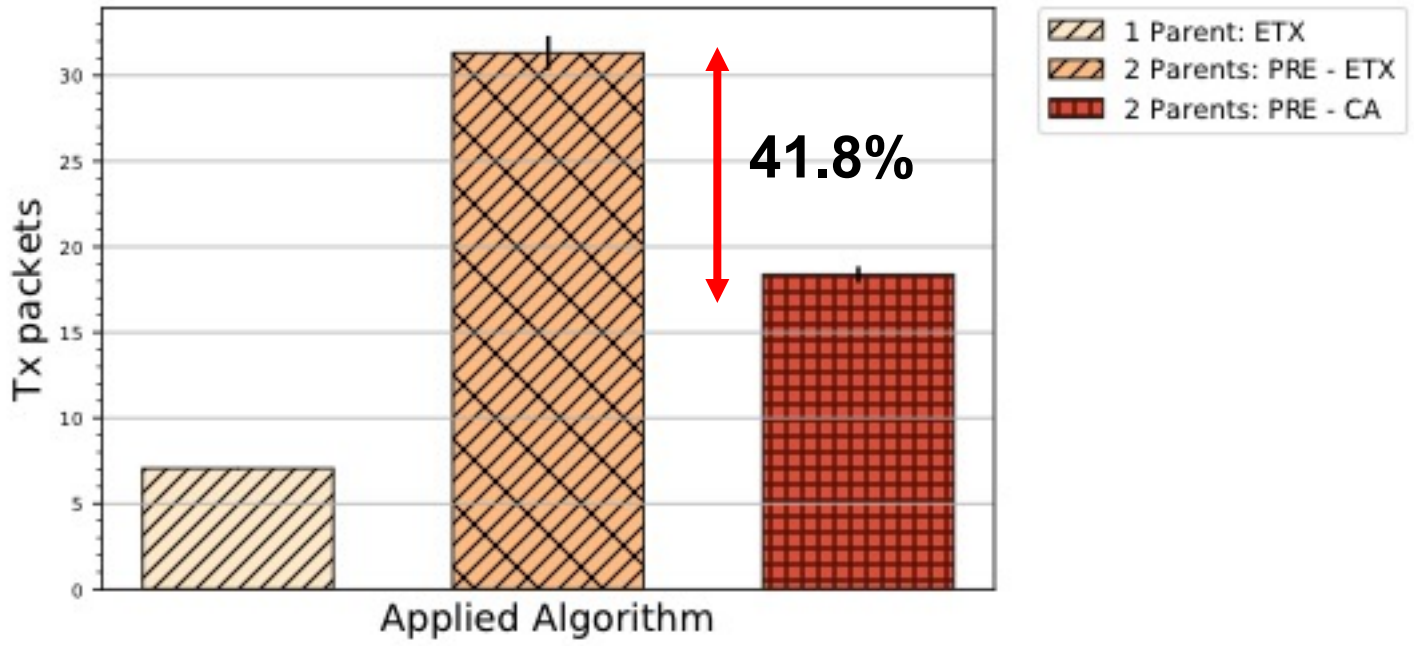
Similar but slightly lower PDR performance for PRE-CA and PRE-ETX

Strict CA: Traversed nodes



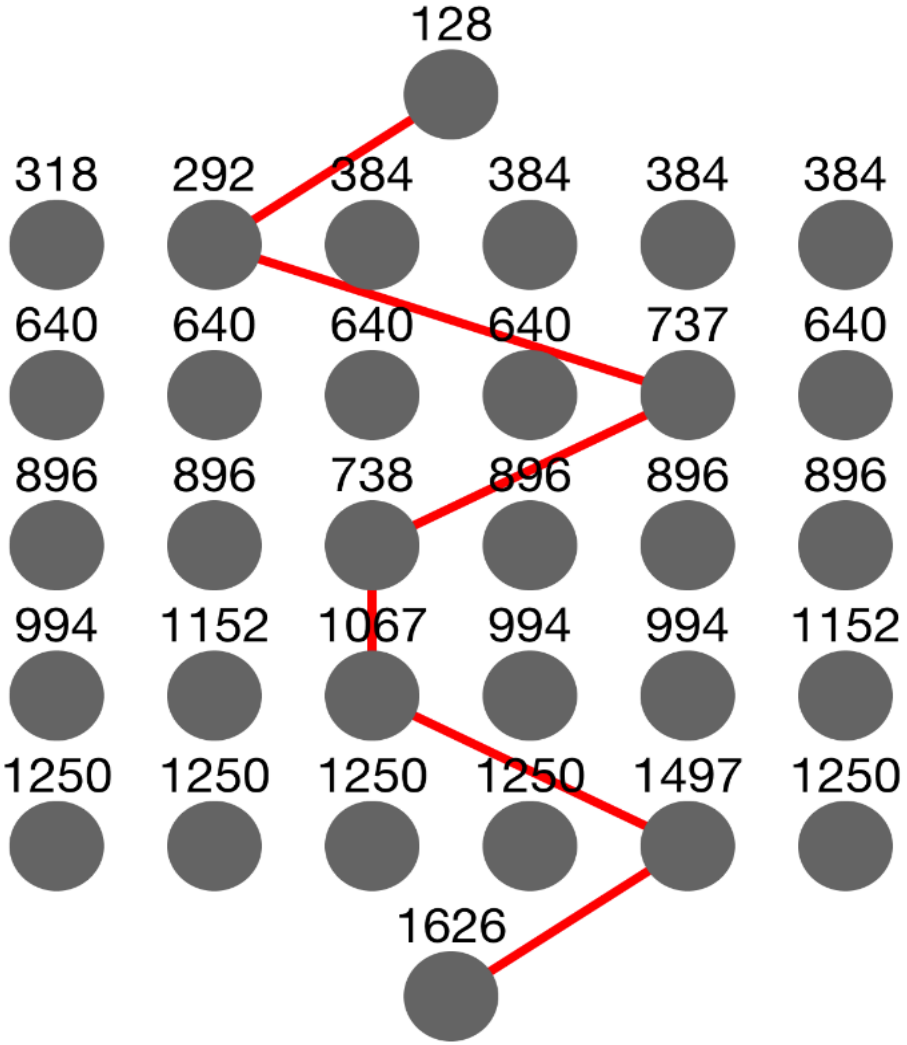
Fewer nodes traversed → energy consumption

Strict CA: Duplications/packet

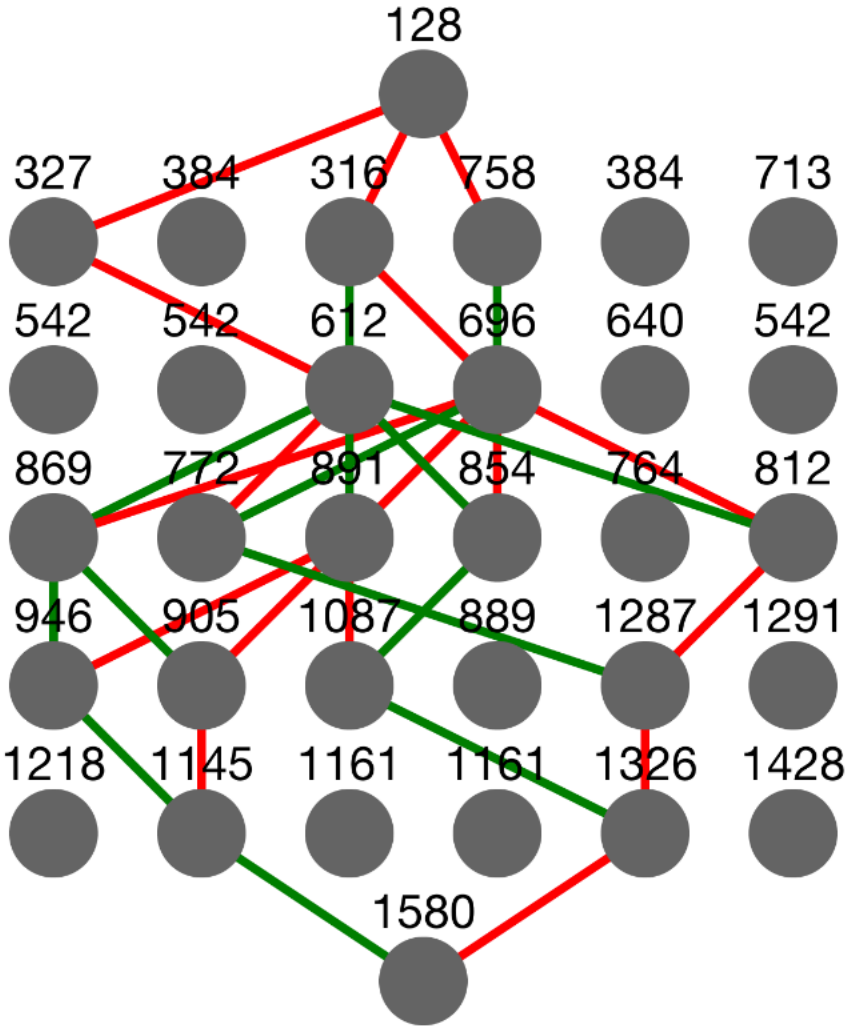


Significantly lower traffic → energy consumption

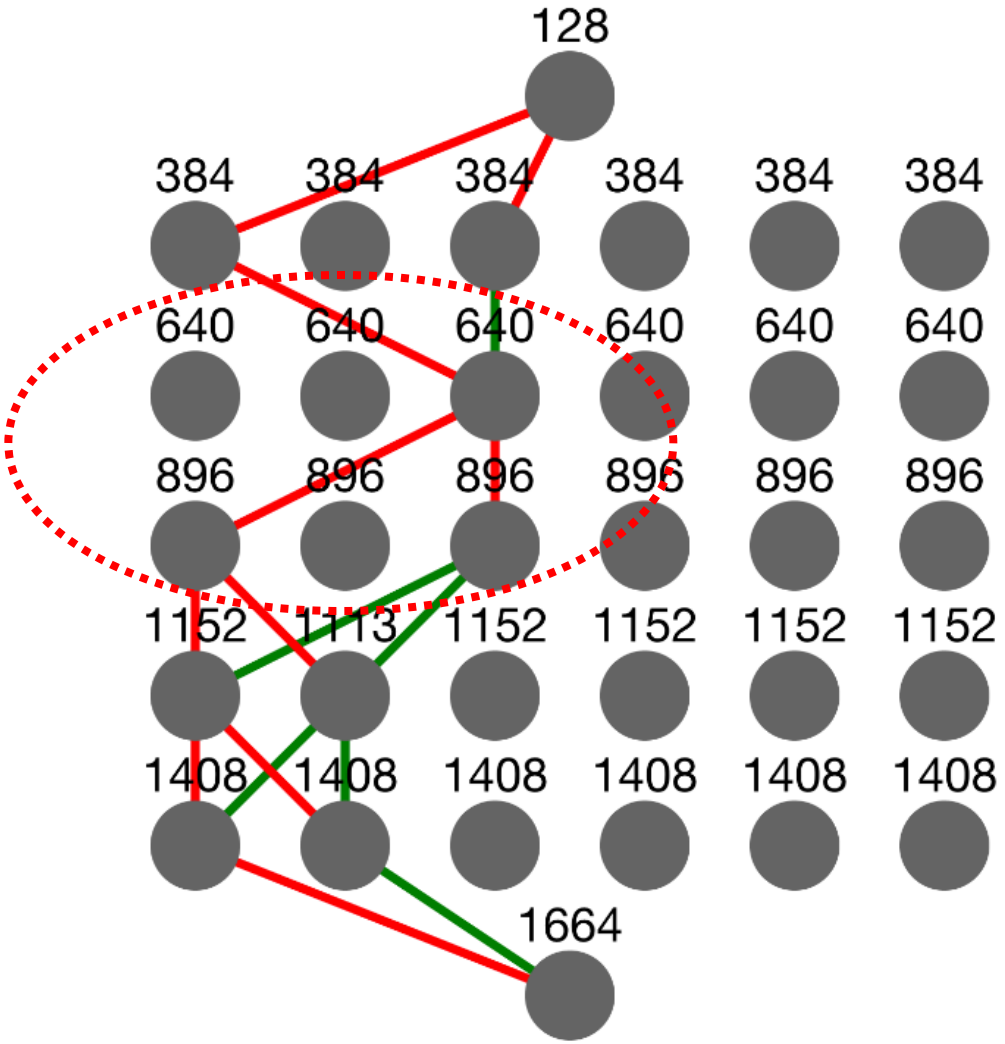
Snapshot: 1 Parent ETX



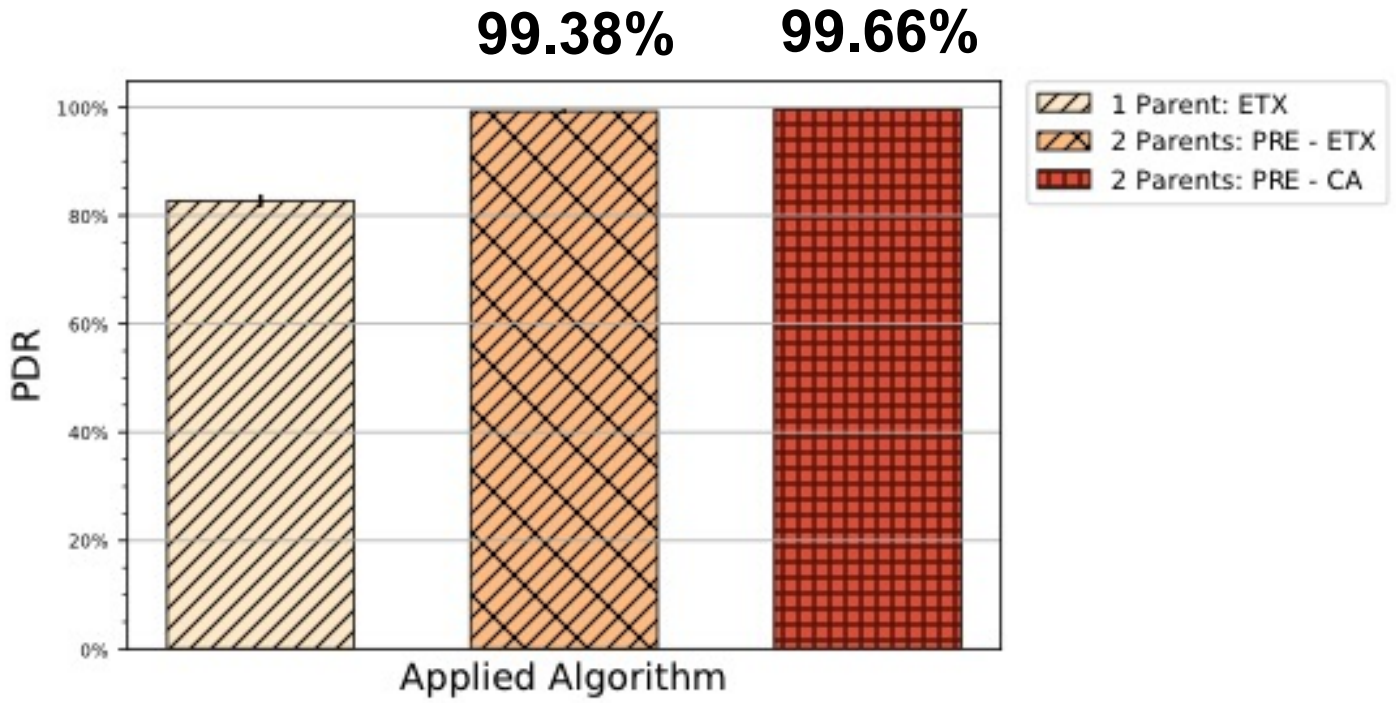
Snapshot: 2 Parents PRE-ETX



Snapshot: 2 Parents PRE-CA Strict

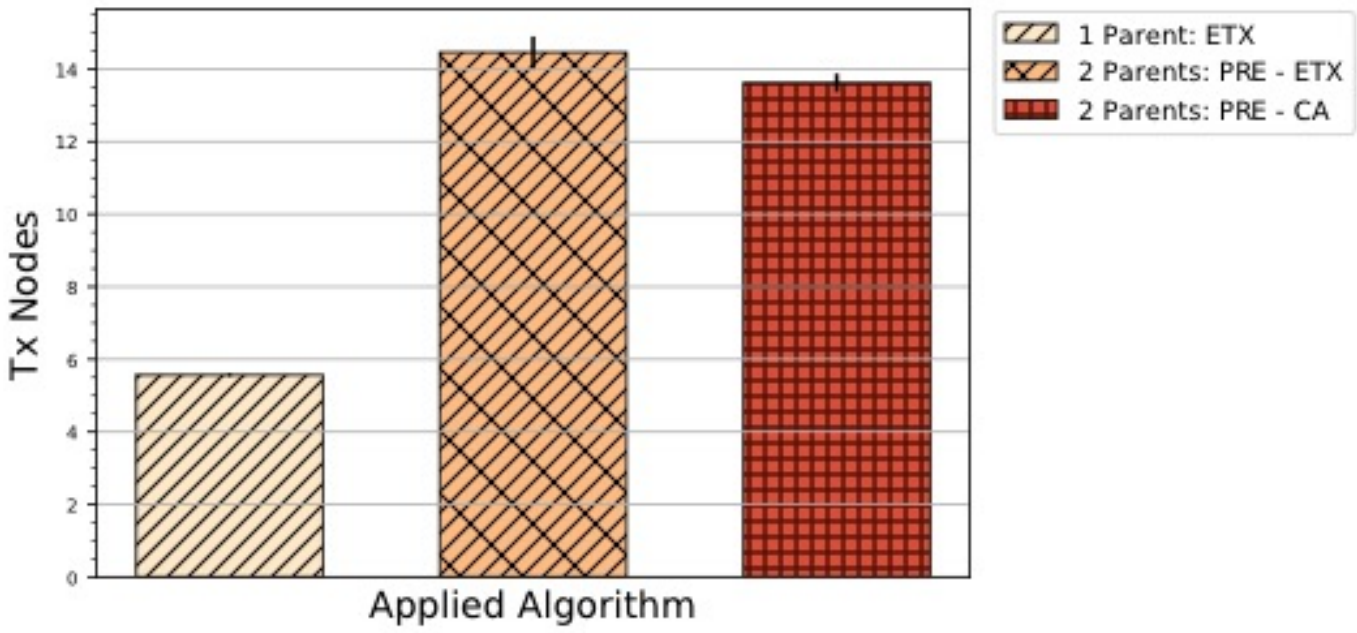


Medium CA: PDR



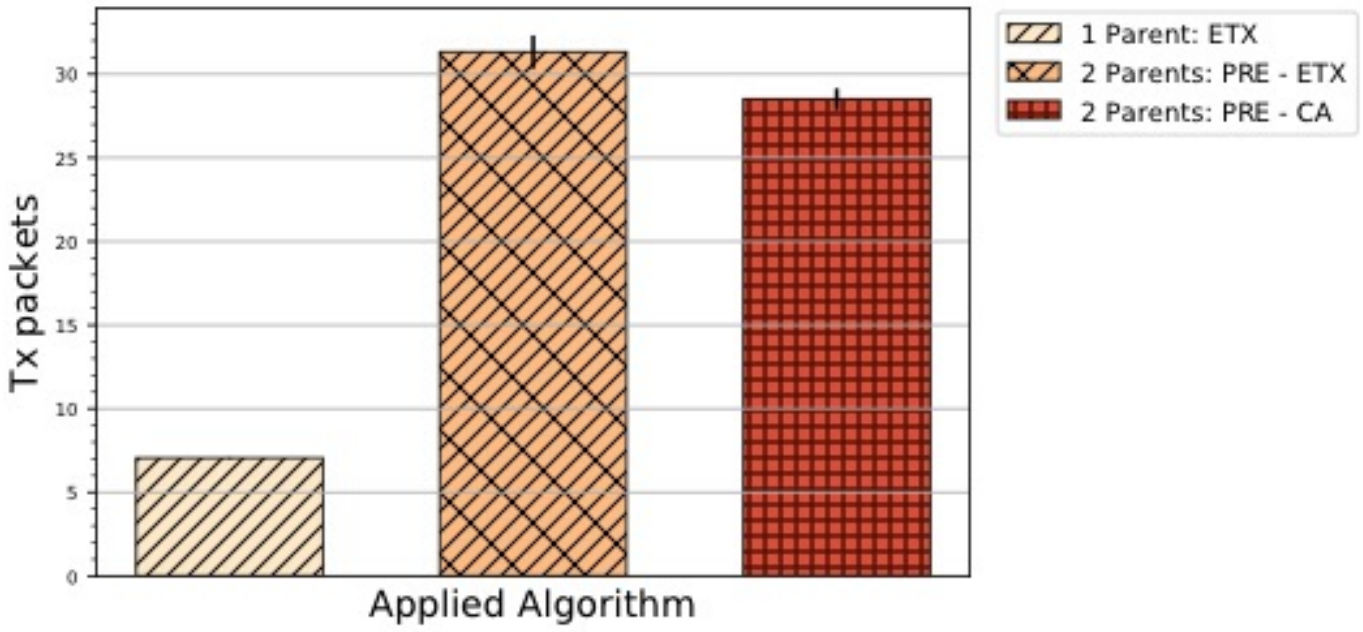
CA comes with equivalent PDR

Medium CA: Traversed nodes



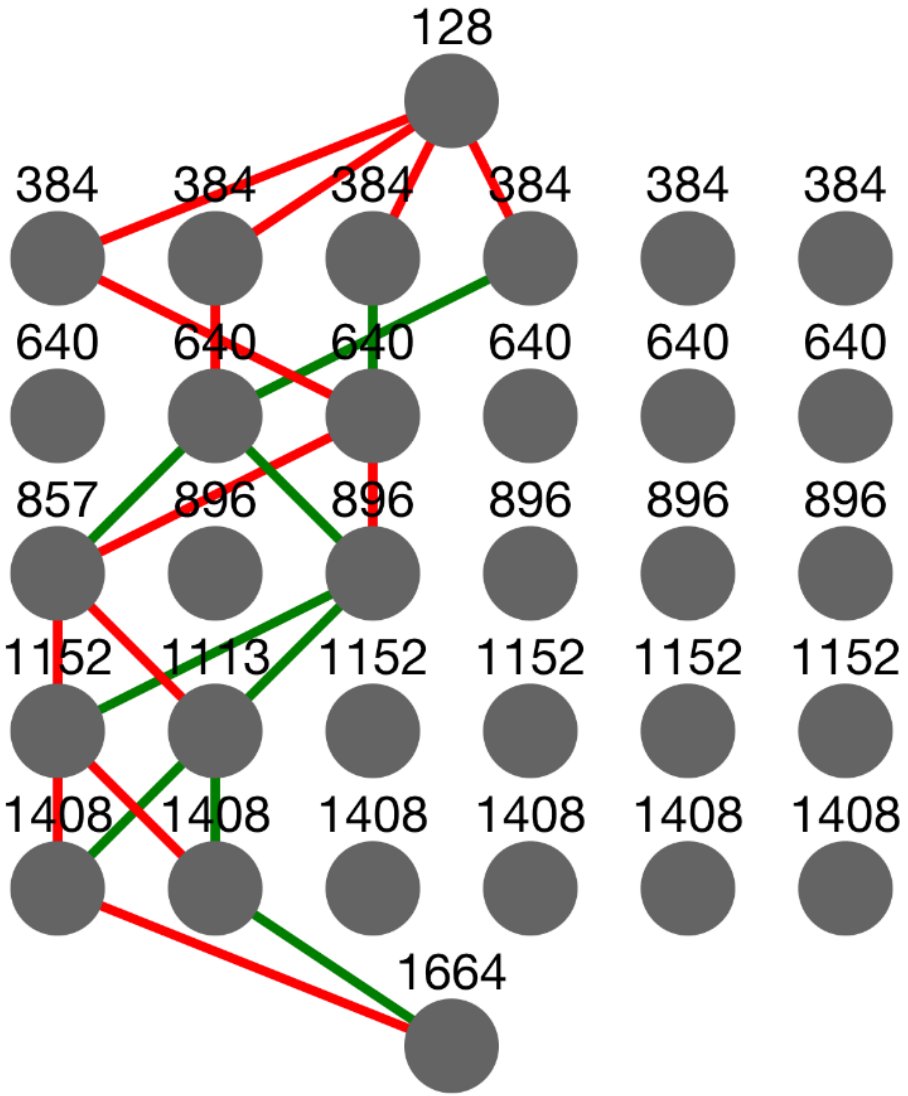
At the cost of partial flooding

Medium CA: Duplications/packet

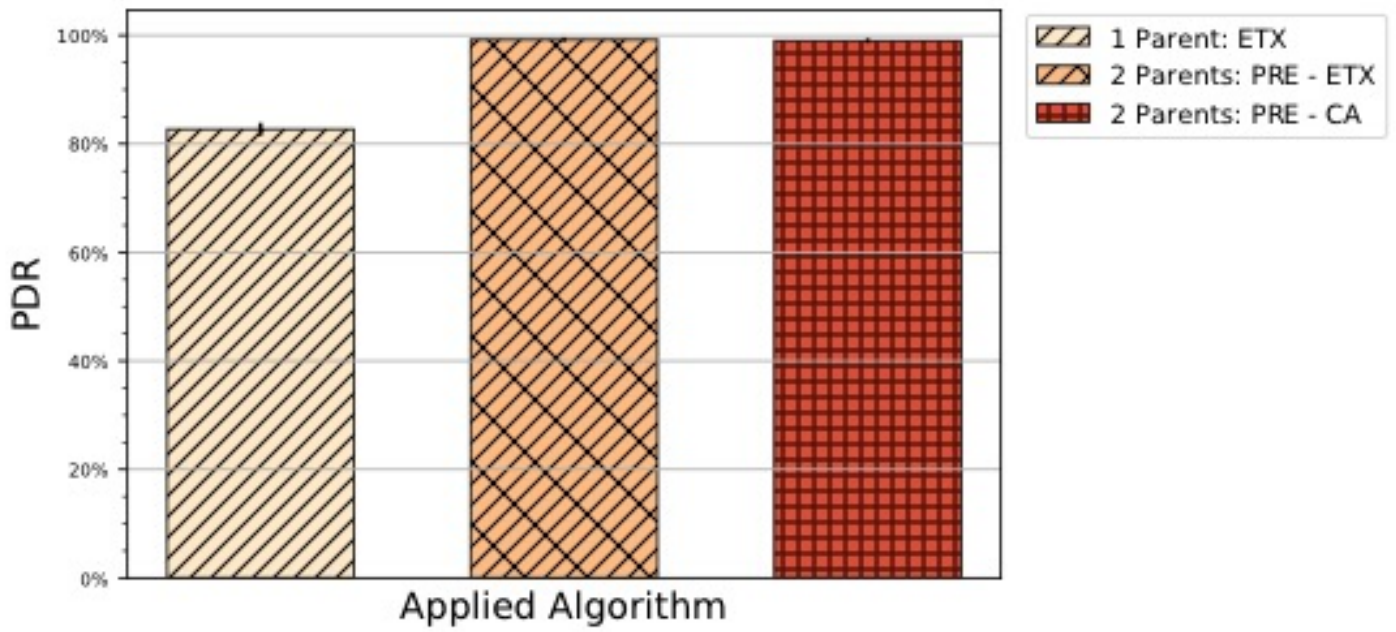


At the cost of energy consumption

Snapshot: 2 Parents PRE-CA Medium

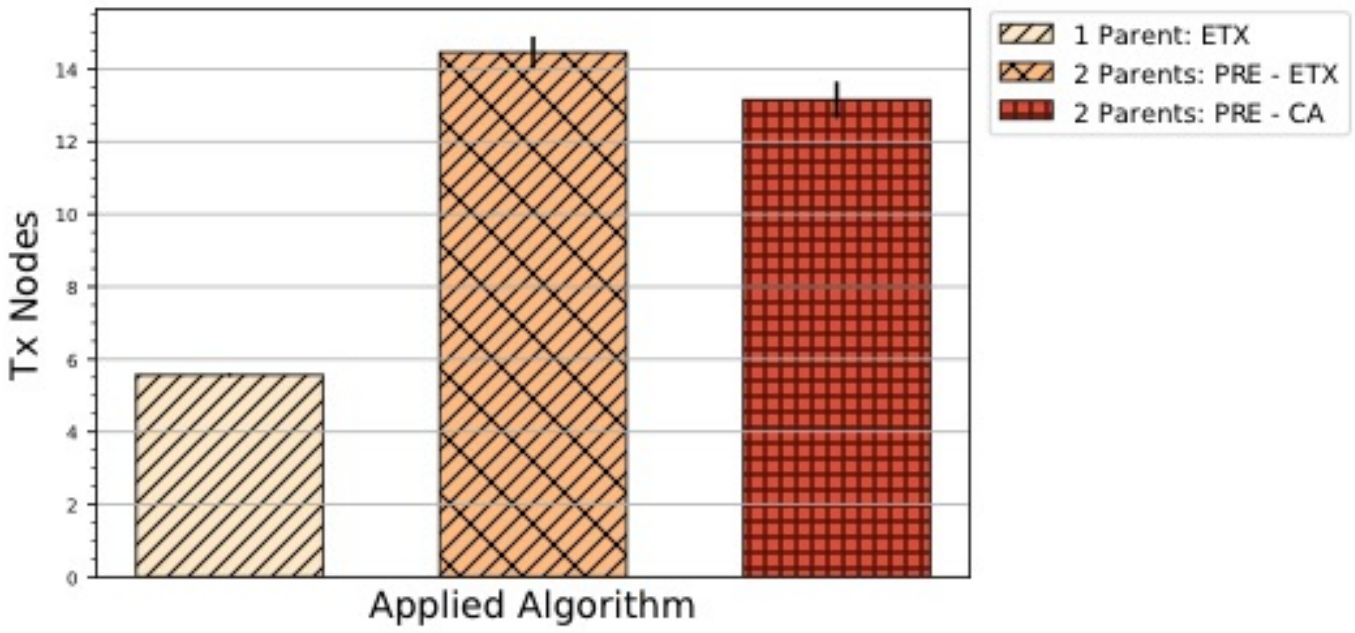


Strict to Medium CA: PDR



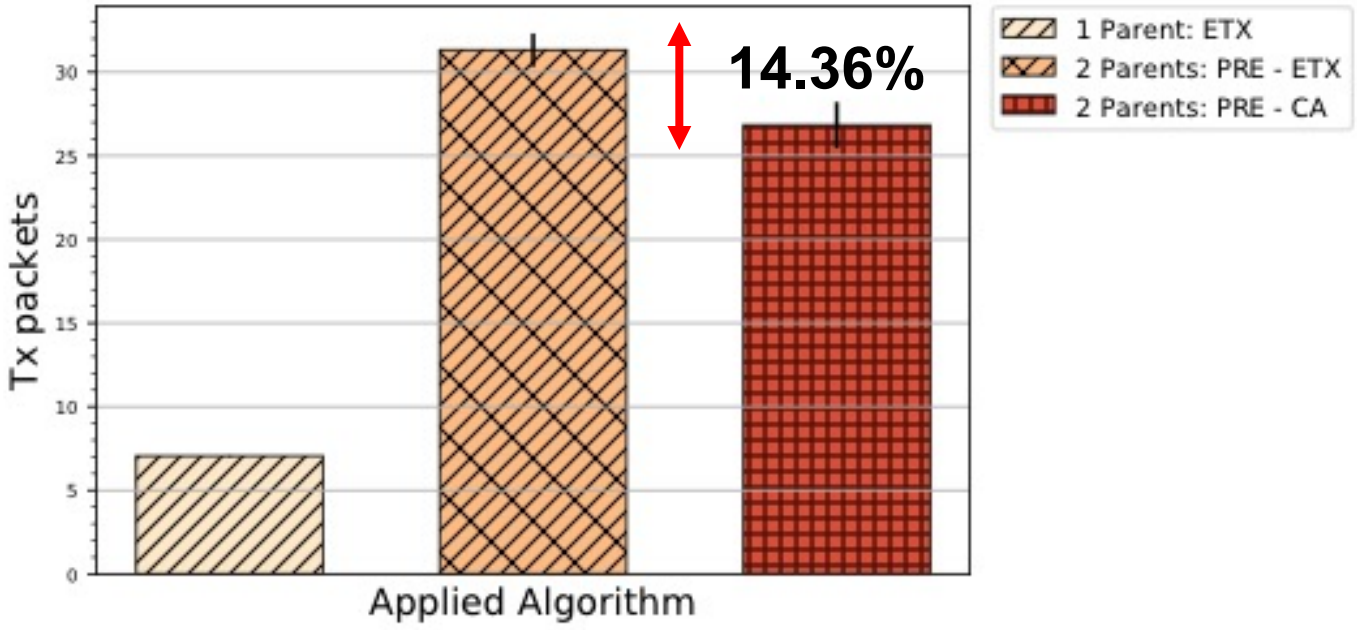
Same PDR performance for PRE-CA and PRE-ETX

Strict to Medium CA: Traversed nodes



Small improvement on # of Traversed nodes

Strict to Medium CA: Duplications/packet



Which leads to 14% improvement on Duplications → reduced energy consumption

Road Forward

- Presented during the last 4 IETF meetings
- Code (or partial code) is available here:
  - Contiki NSA extension https://github.com/ariskou/contiki/tree/draft-koutsiamanis-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
- We received and addressed **reviews** from Diego, Derek (Thank you!)
  - We are looking for more reviews

Road Forward

- Education:
  - 1 Tutorial (2 hours): GIIS 2018 International Conference
  - PRE is used in Teaching (Lab Sessions with Contiki OS) @IMT Atlantique
- Research:
  - 1 Journal: IEEE Transactions on Industrial Informatics, 2018
  - 2 Conferences: AdHoc-Now 2018, IEEE ICC 2017
  - 1 Poster & 1 Dependability Competition: ACM EWSN 2017

Credits

- Thanks to **Tomas** for running (*hundreds of*) **Simulations**
- Thanks to **Diego** and **Derek** for the **Reviews**
- Thanks to **Michael**: input on the **IPv6 compression** (RFC
- Use 6LoWPAN Routing Header (6LoRH) (RFC8138)
- Thanks to **Oana** : adding constraint on metric to not selecting “bad” Alternative Parents → Filter out parents with low ETX

Thanks!

Questions?

Root initiated routing state in RPL

draft-ietf-dao-projection

Pascal Thubert

IETF 103

Bangkok

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)

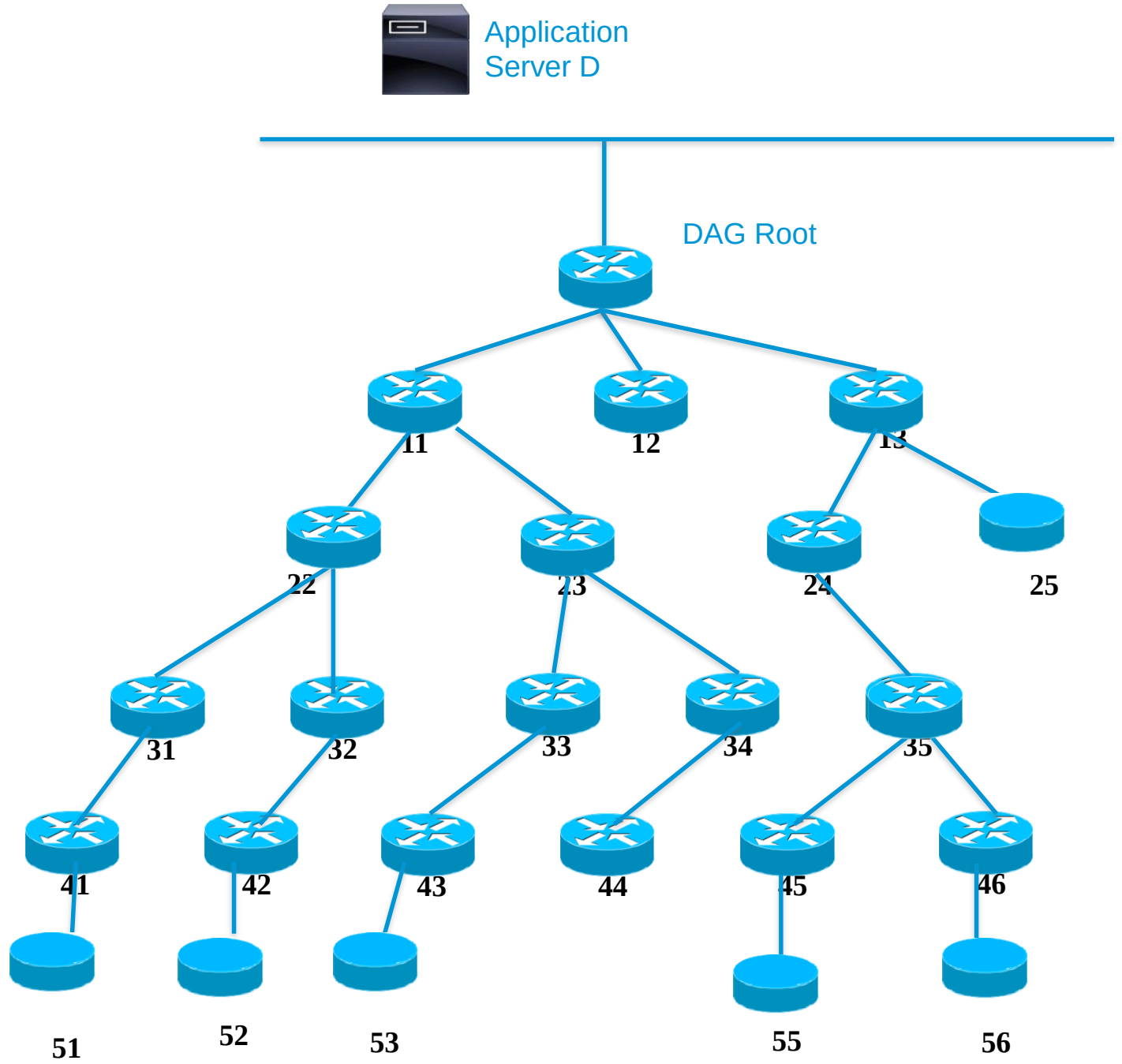
Loop avoidance

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

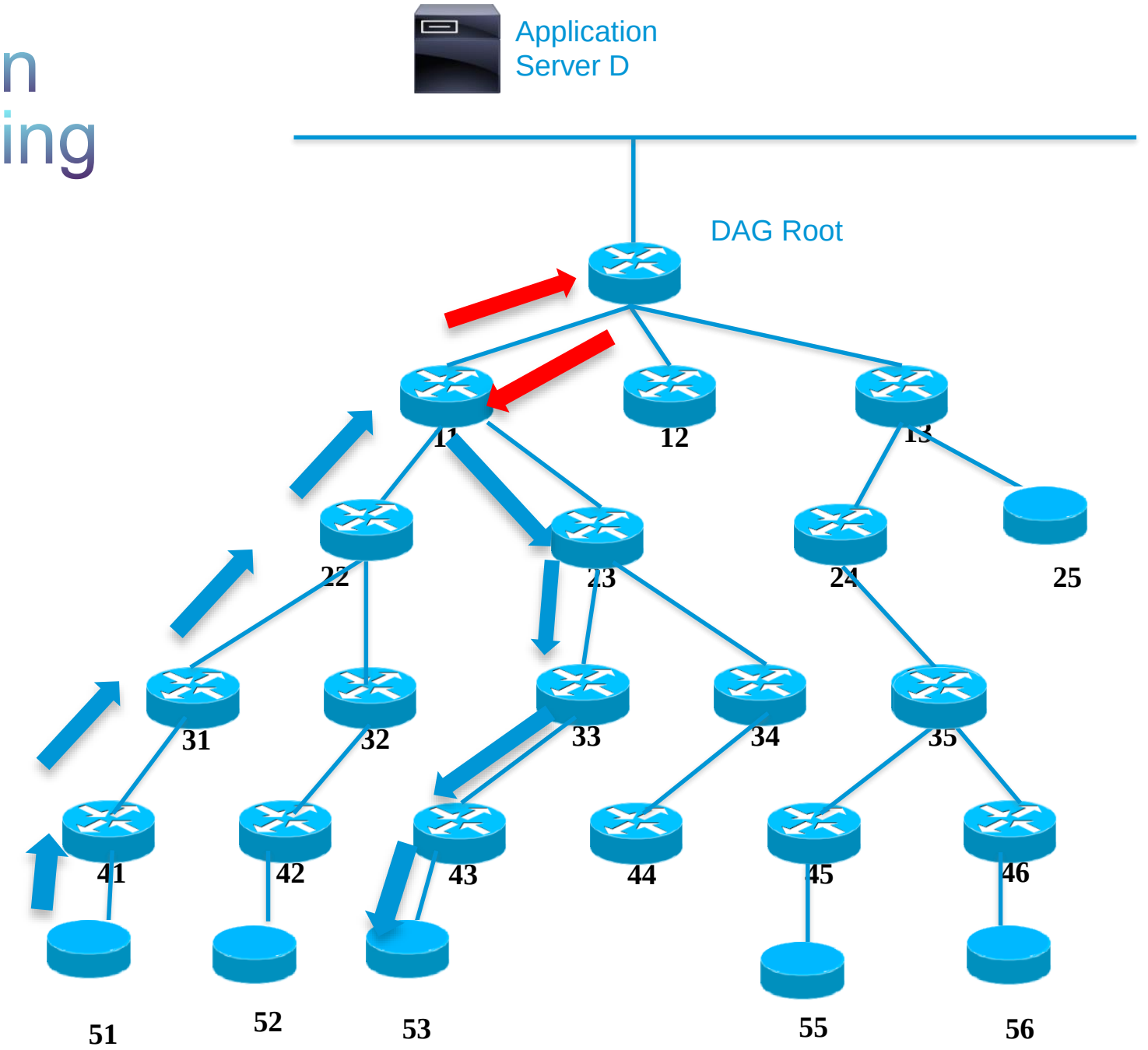
<RFC6550>: “Down 'O': 1-bit flag indicating whether the packet is expected to progress Up or Down. A router sets the 'O' flag when the packet is expected to progress Down (using DAO routes), and clears it when forwarding toward the DODAG root to a node with a lower Rank). A host or RPL leaf node MUST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

Backup

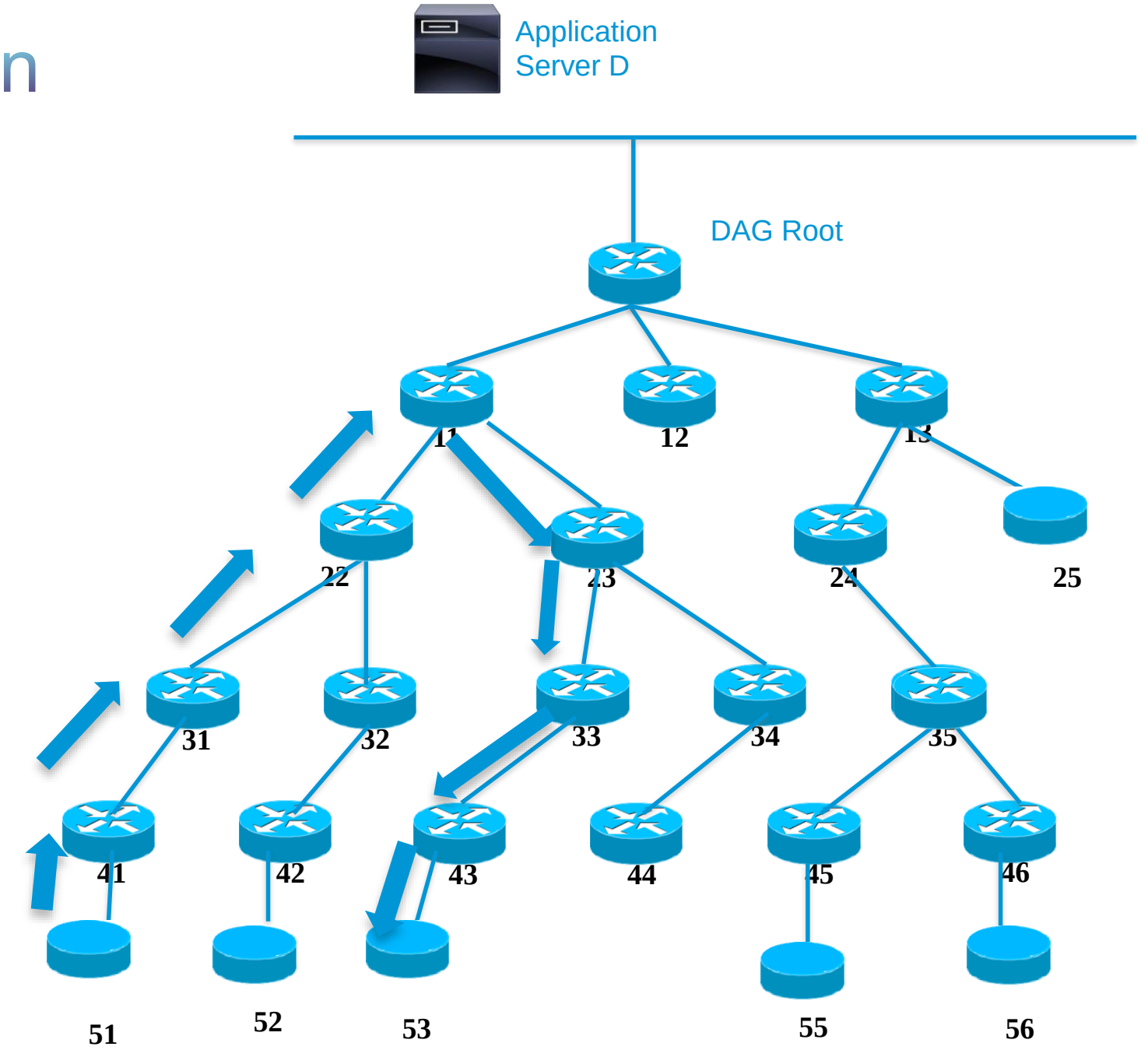
New: non-storing mode transversal route


Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

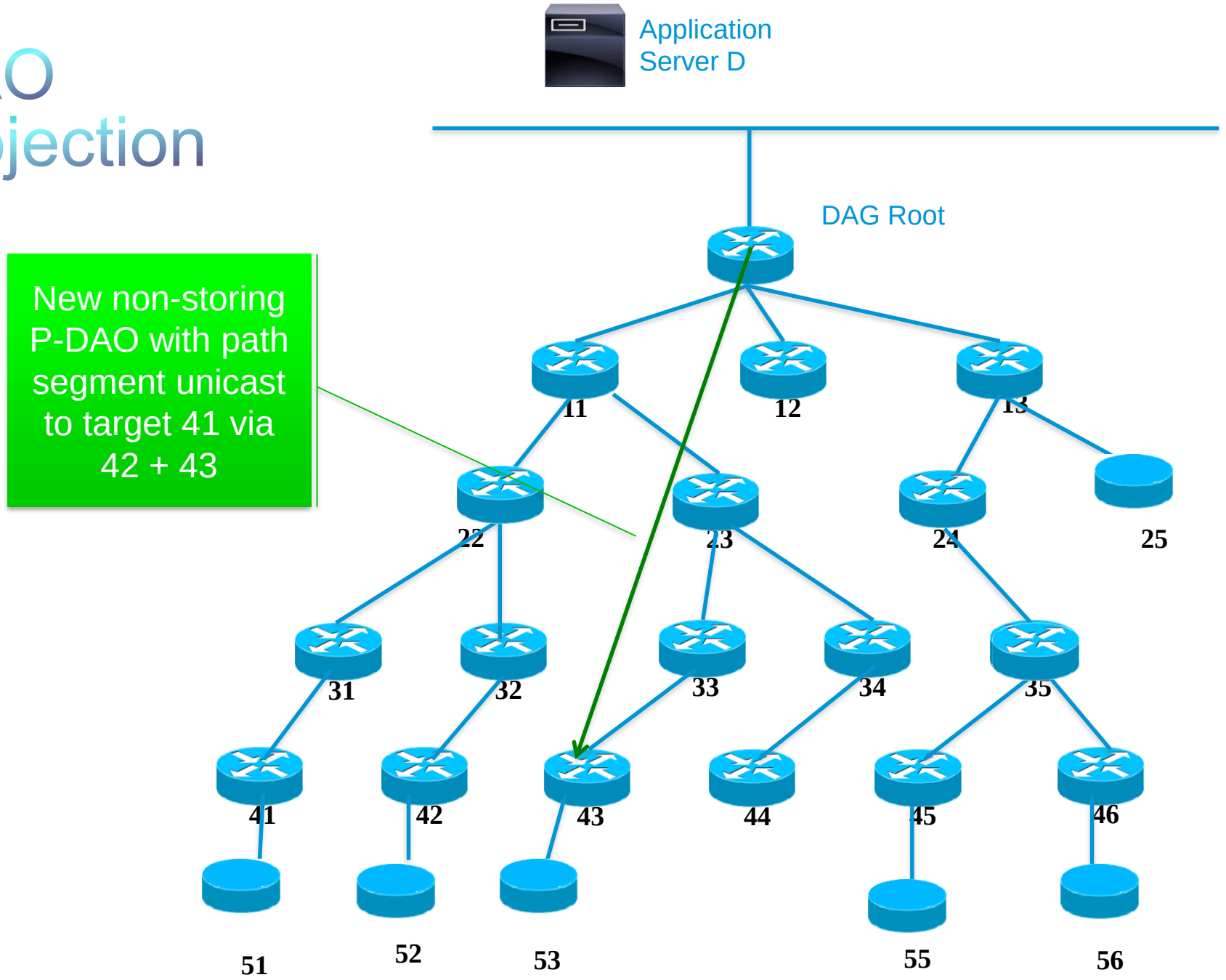
Stretch in non-storing mode



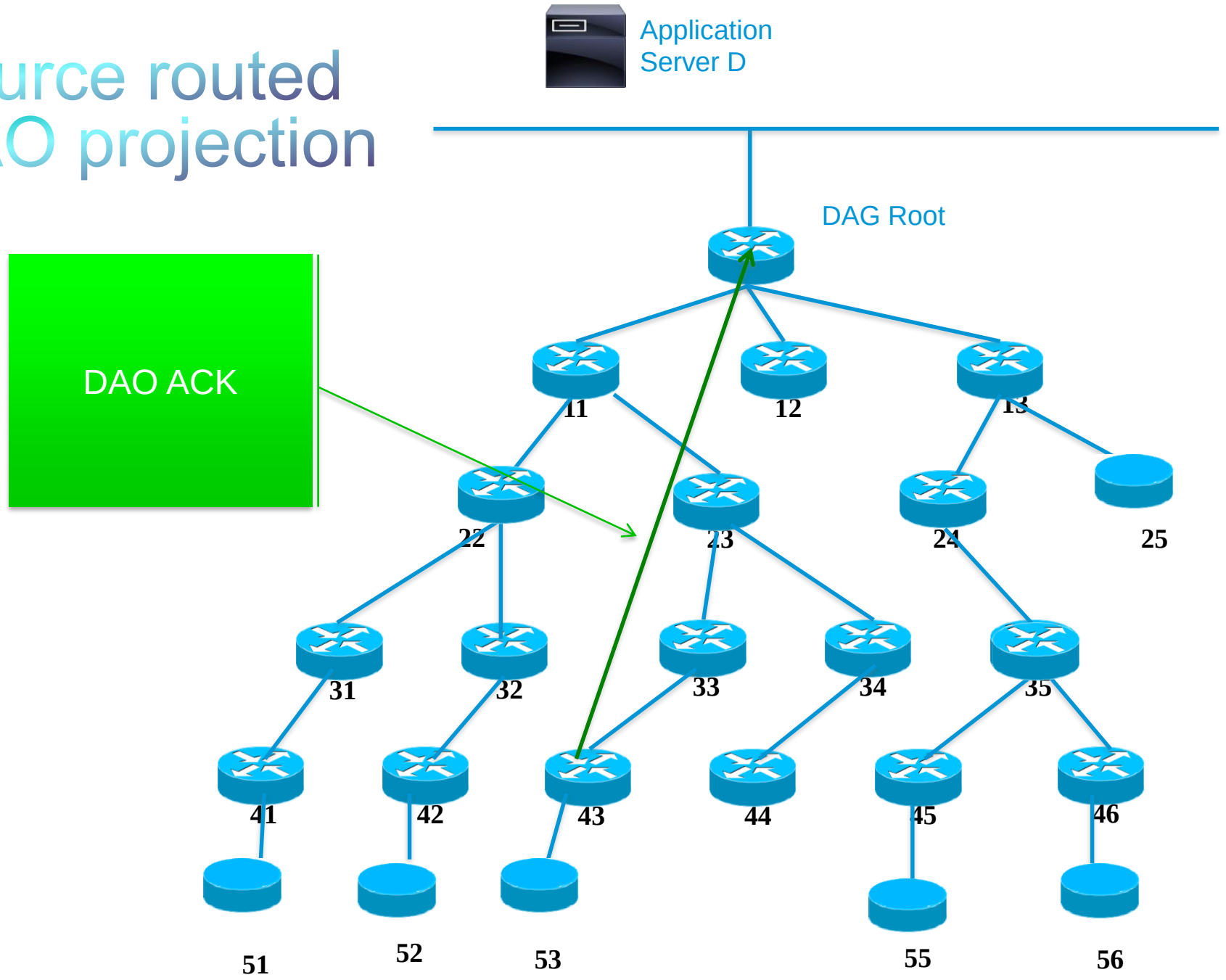
Stretch in storing mode



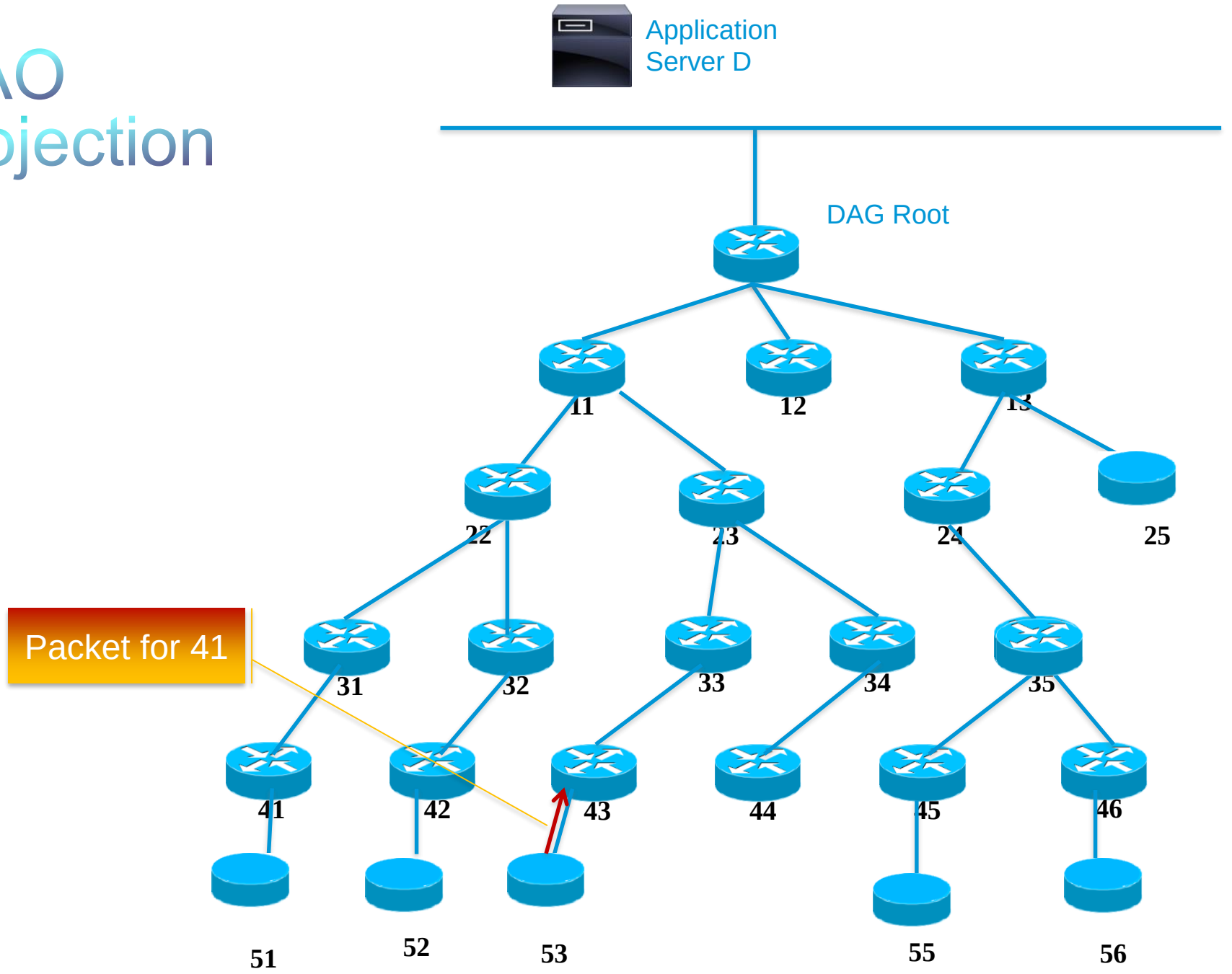
DAO projection



Source routed DAO projection



DAO projection

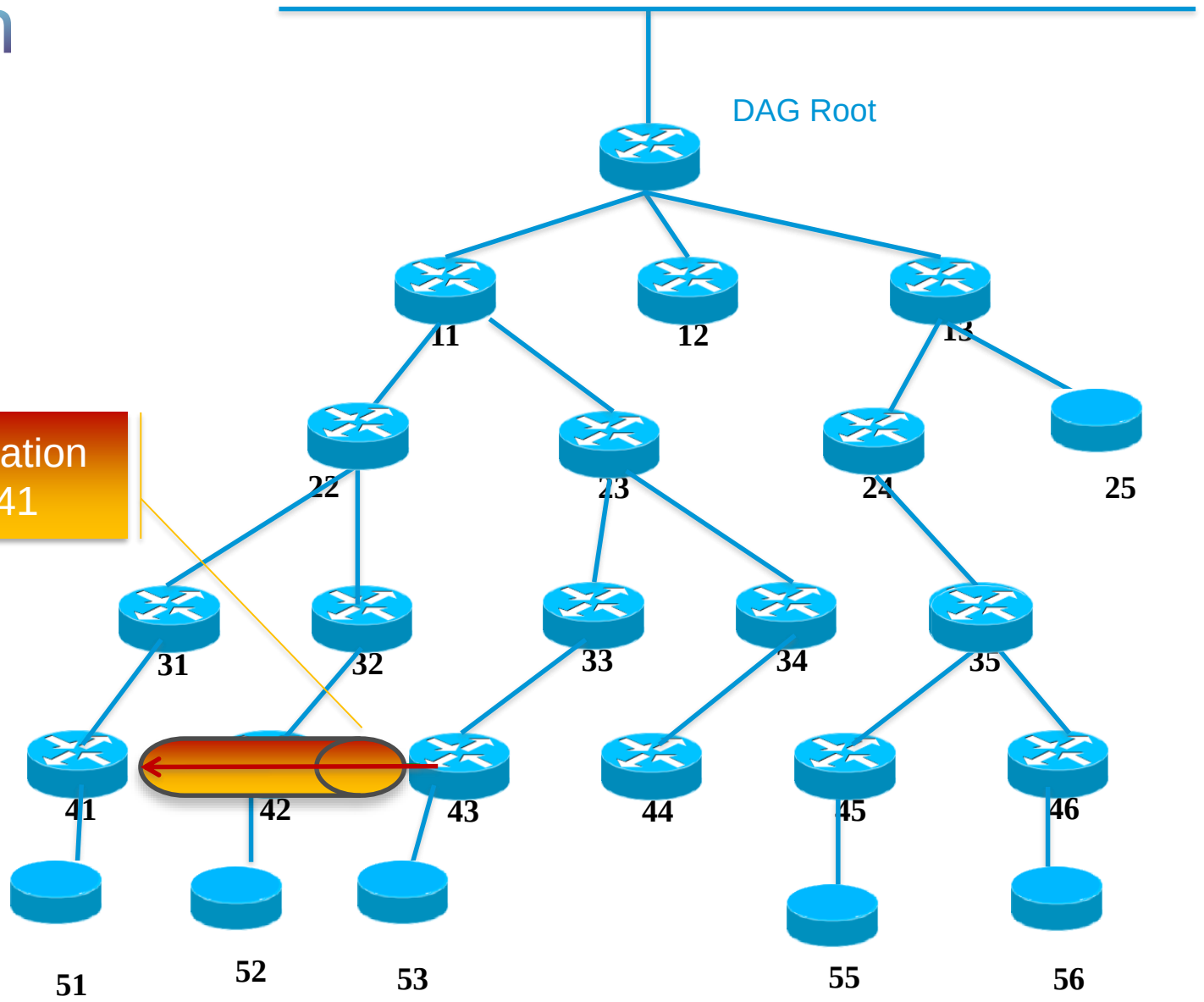


DAO projection

Application Server D

DAG Root

IP in IP encapsulation with SRH 42, 41

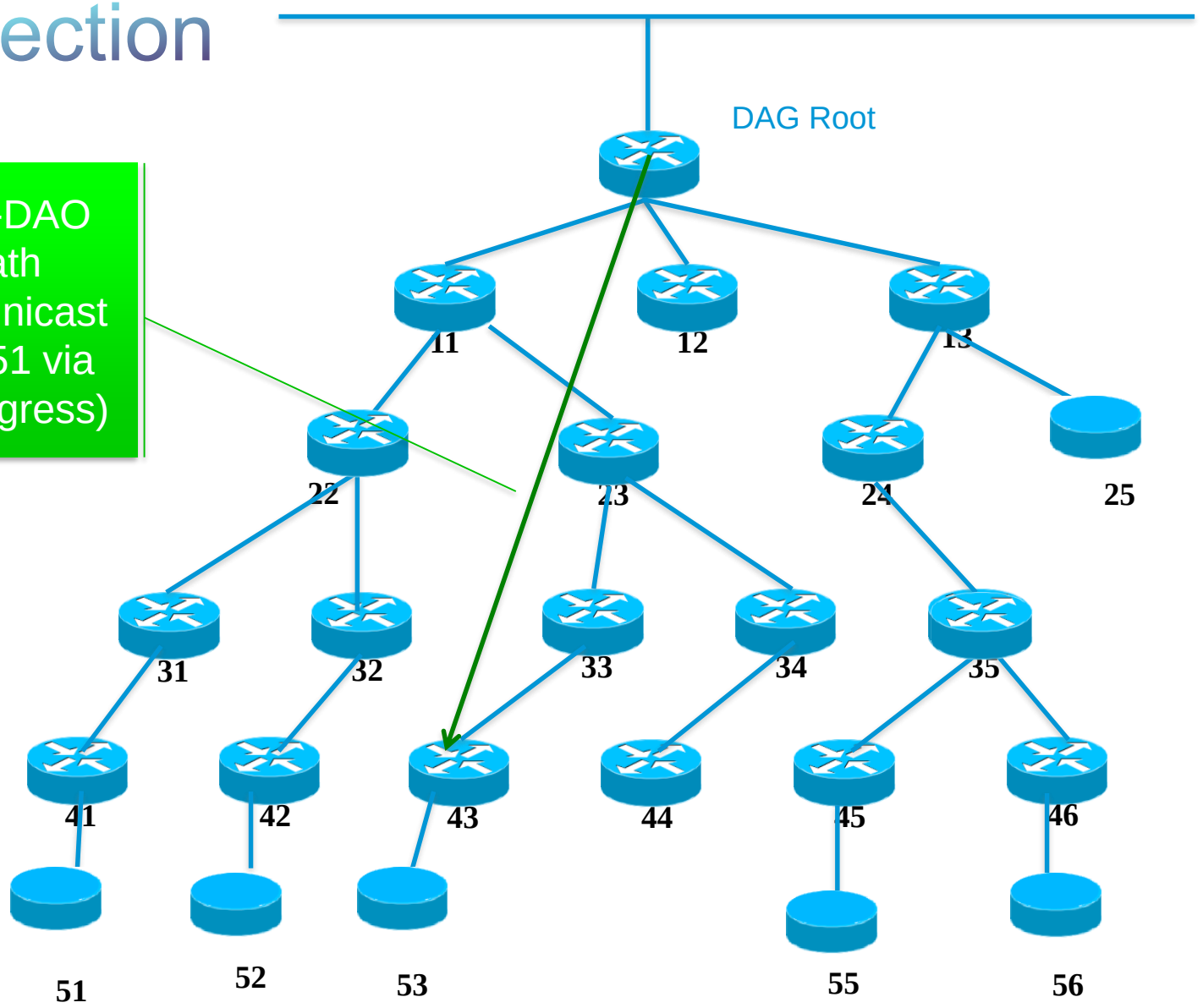


Not done yet
DAO projection

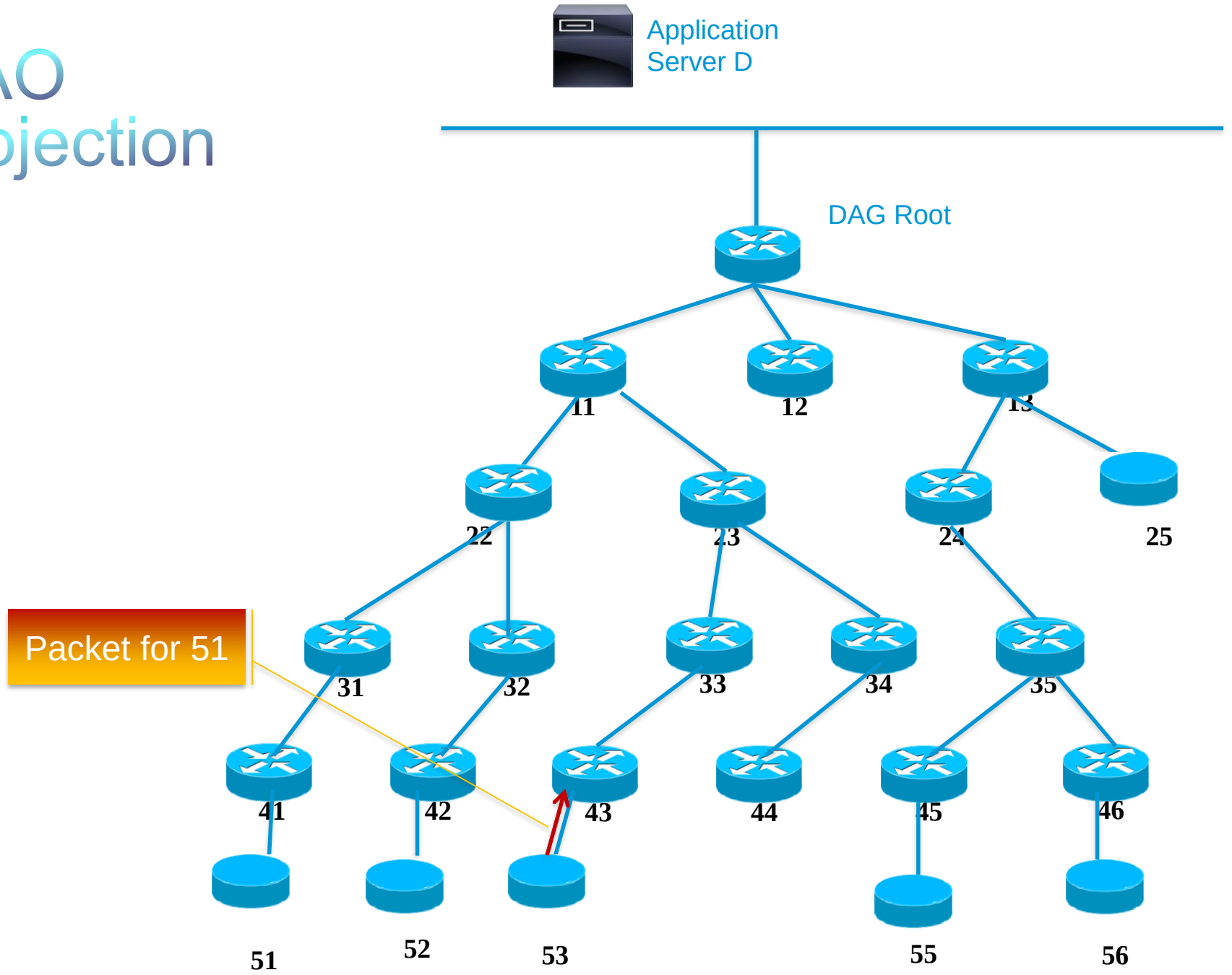
Application
Server D

DAG Root

storing P-DAO with path segment unicast to target 51 via 43==41 (egress)



DAO projection

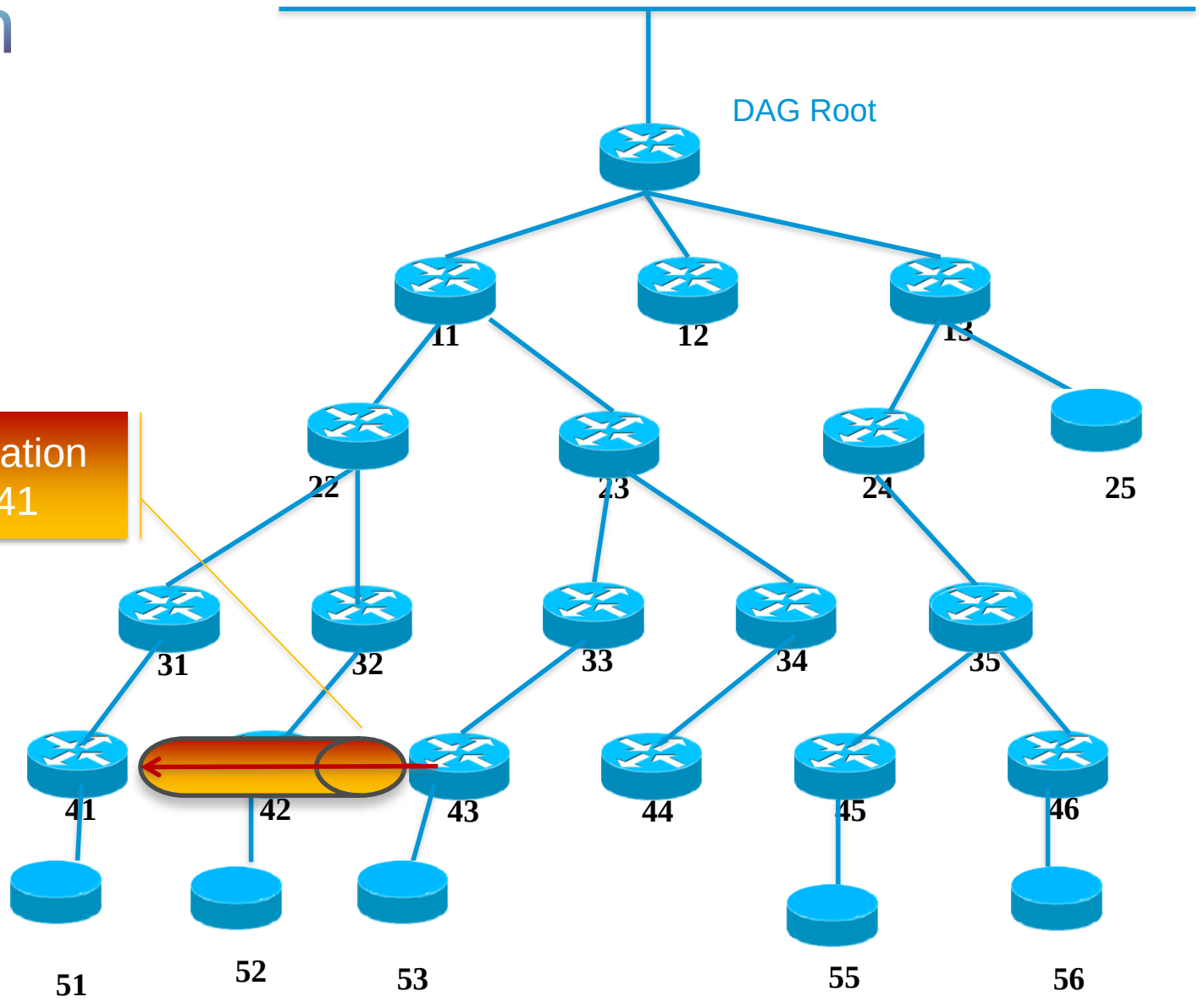


DAO projection

Application Server D

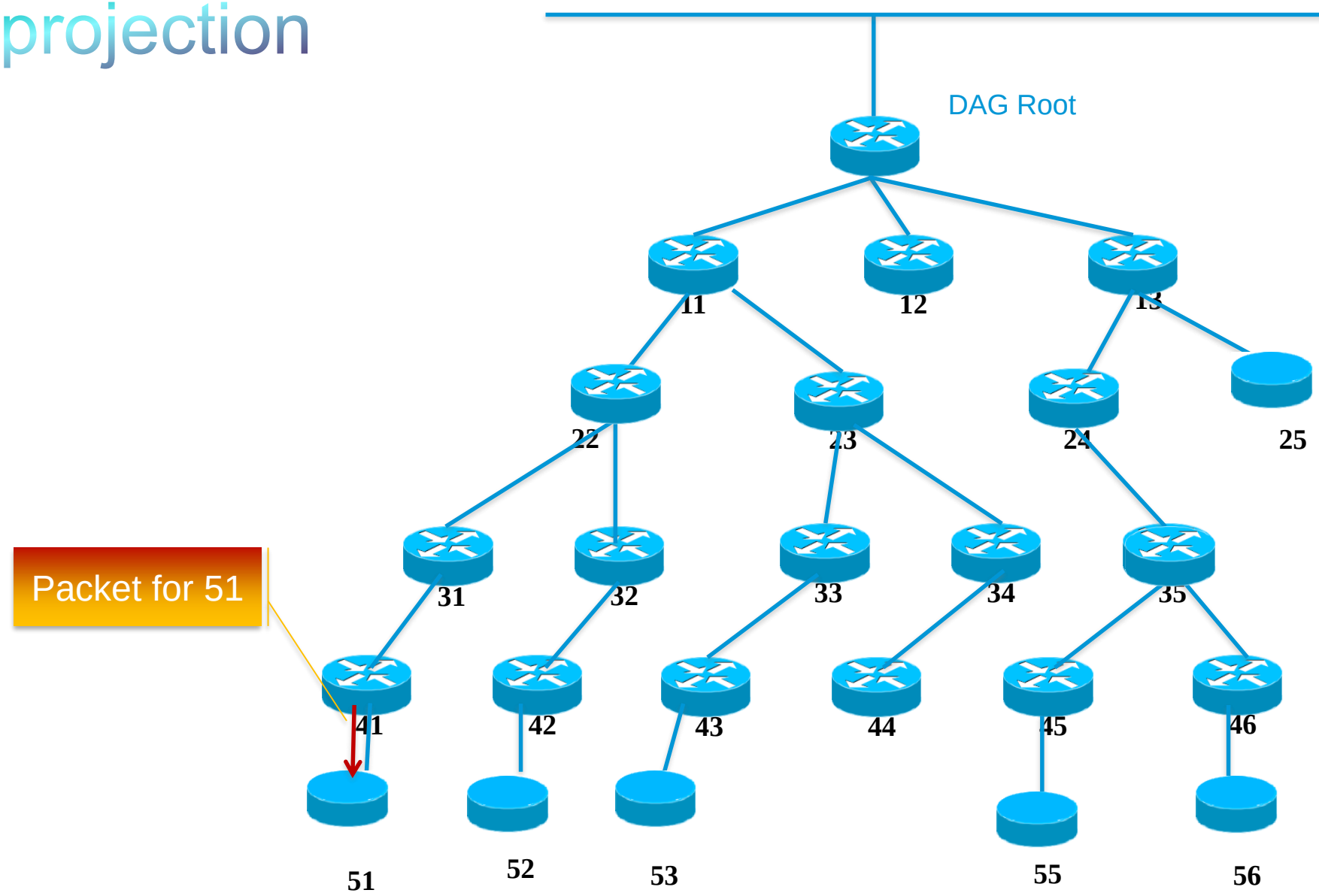
DAG Root

IP in IP encapsulation with SRH 42, 41

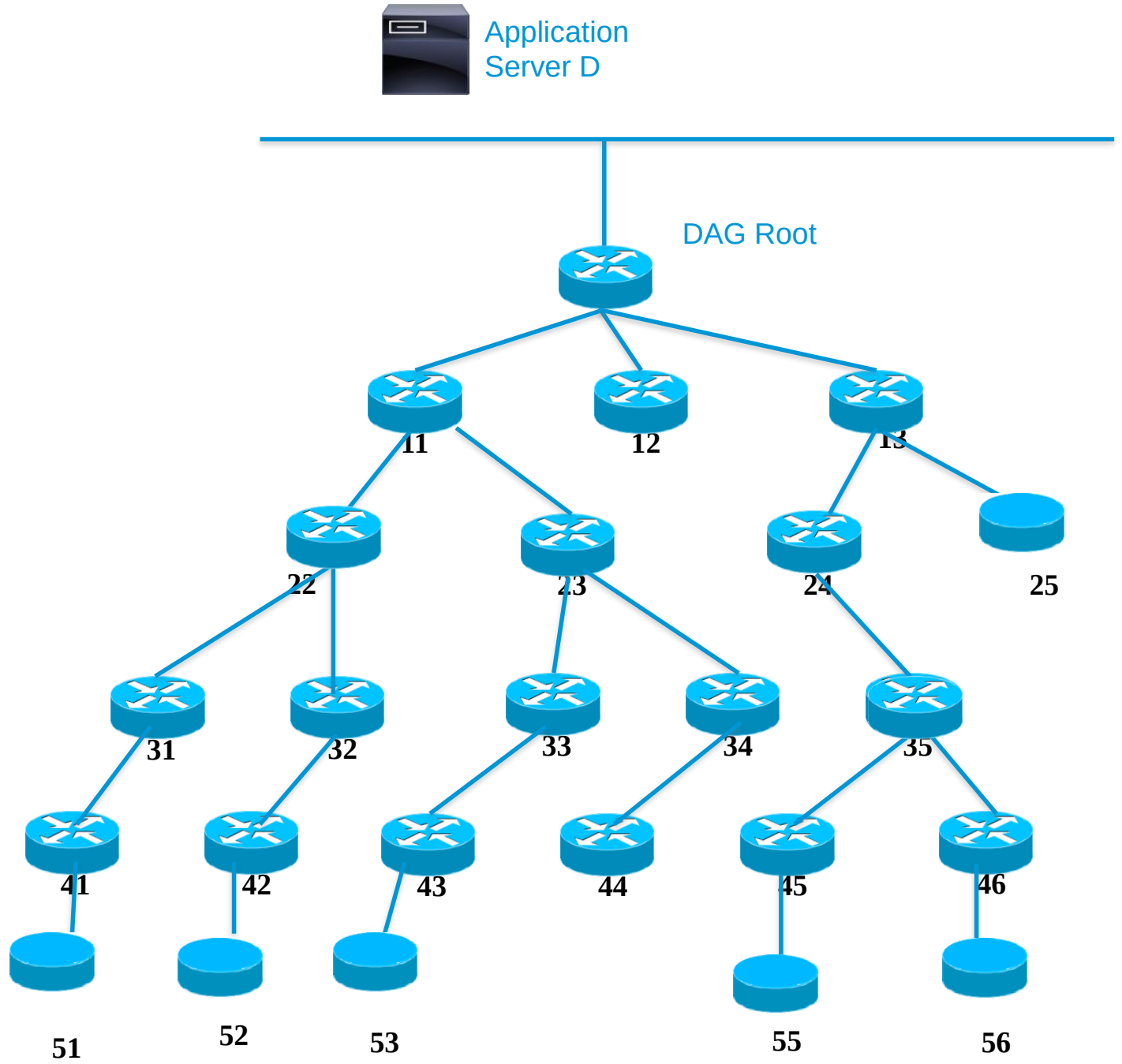


DAO projection

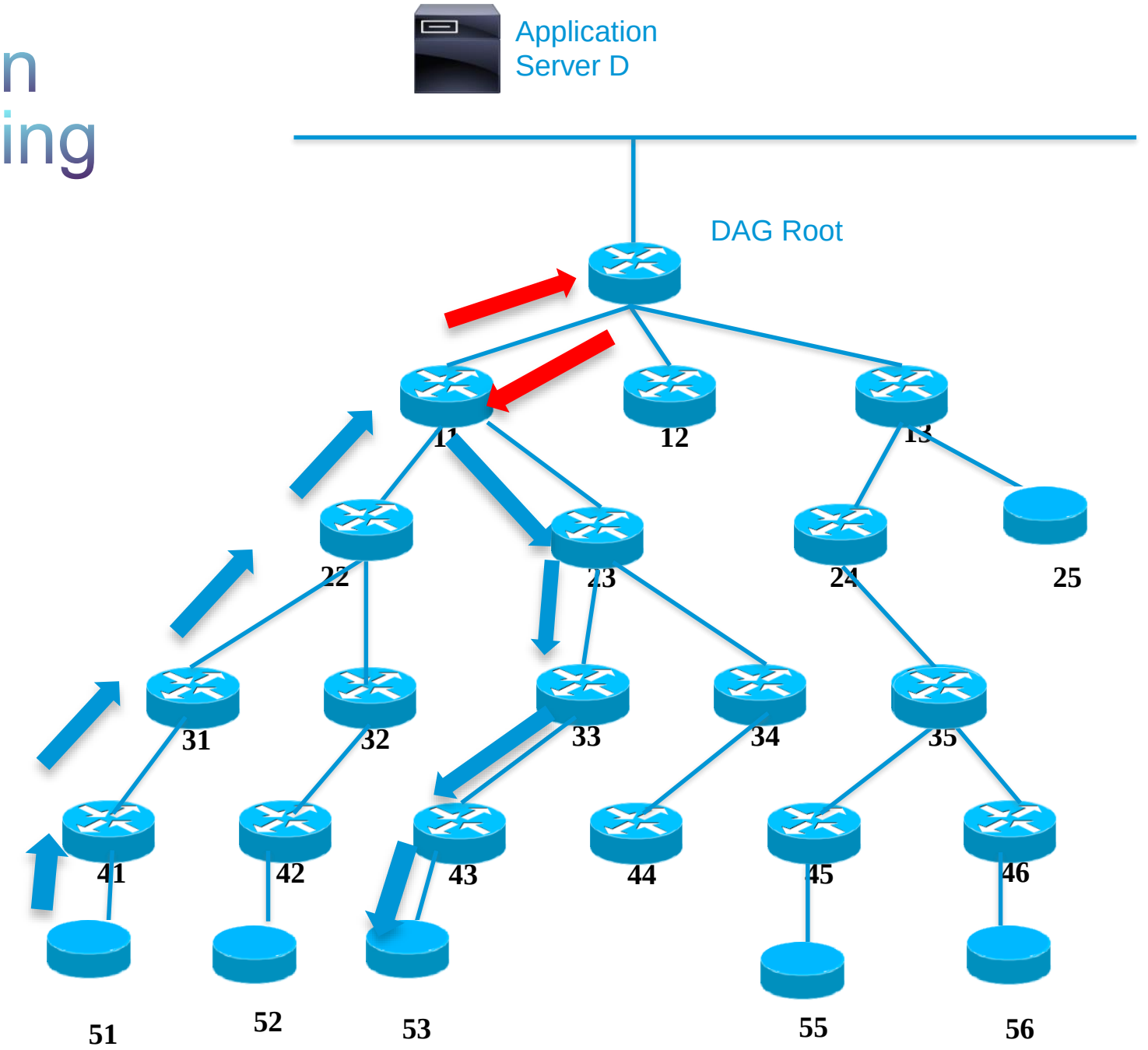
Application Server D



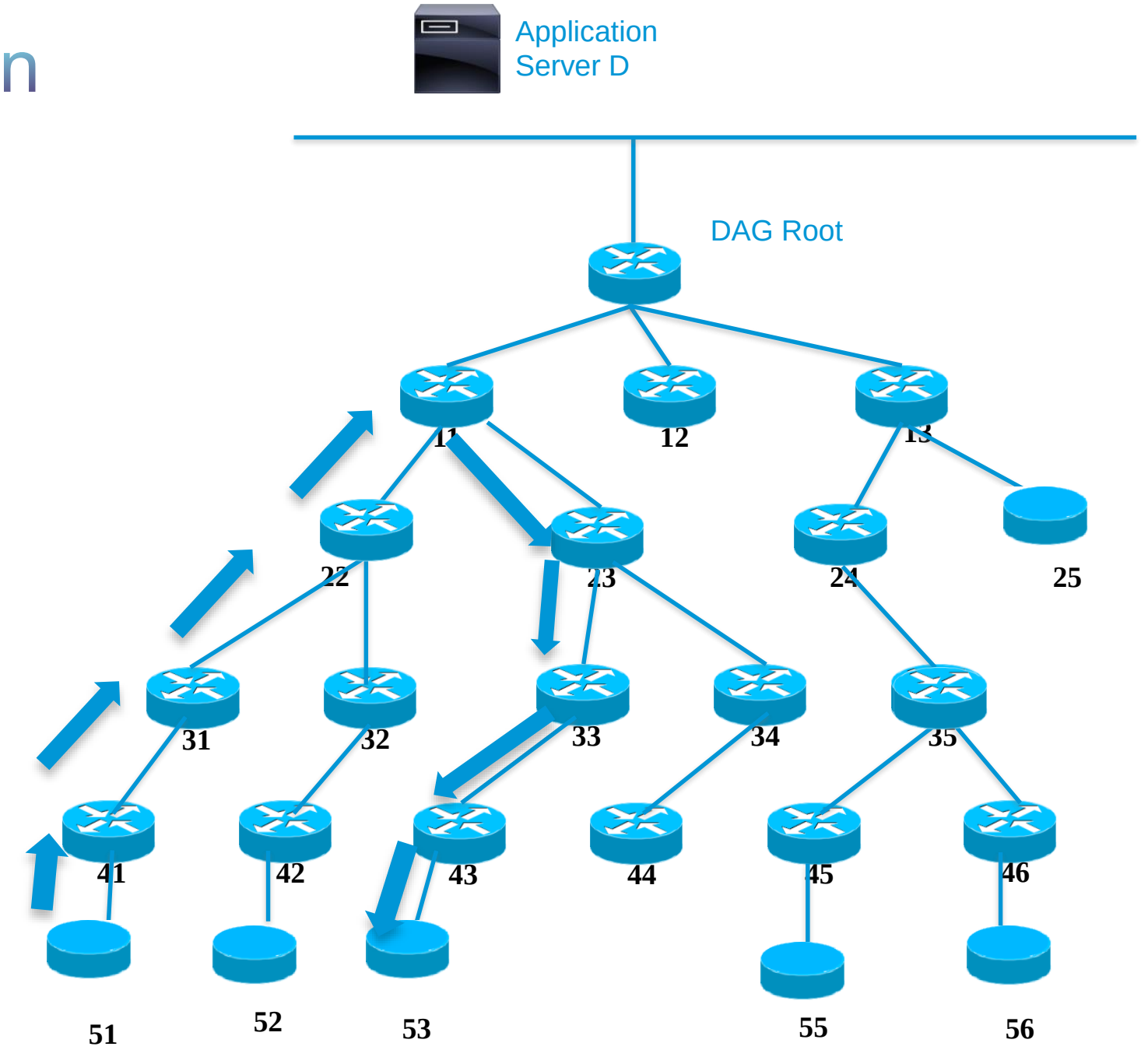
Storing mode transversal route


Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

Stretch in non-storing mode



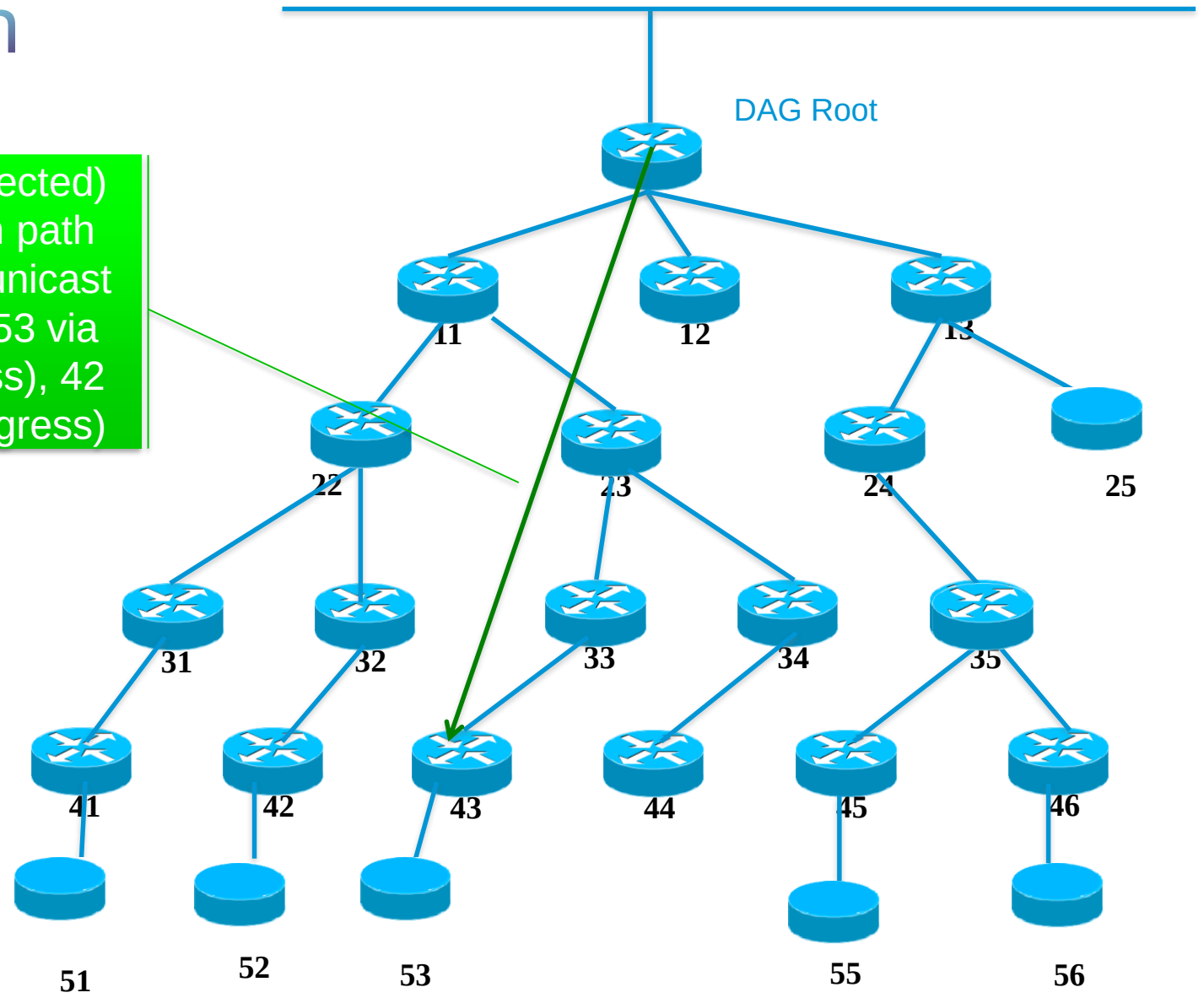
Stretch in storing mode



DAO projection

Application Server D

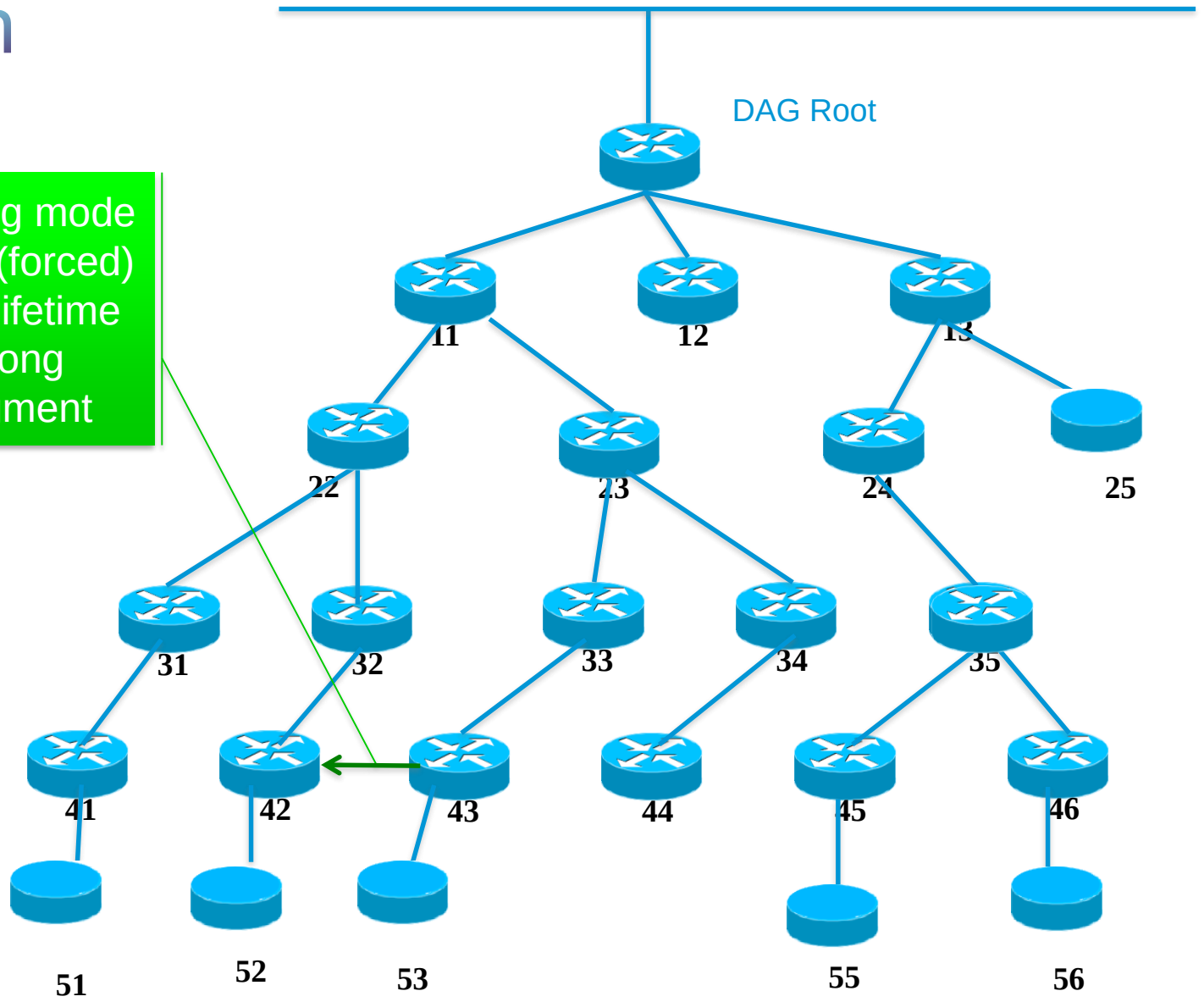
New (projected) DAO with path segment unicast to target 53 via 41 (ingress), 42 and 43 (egress)



DAO projection

Application Server D

Storing mode DAO (forced) with lifetime along segment

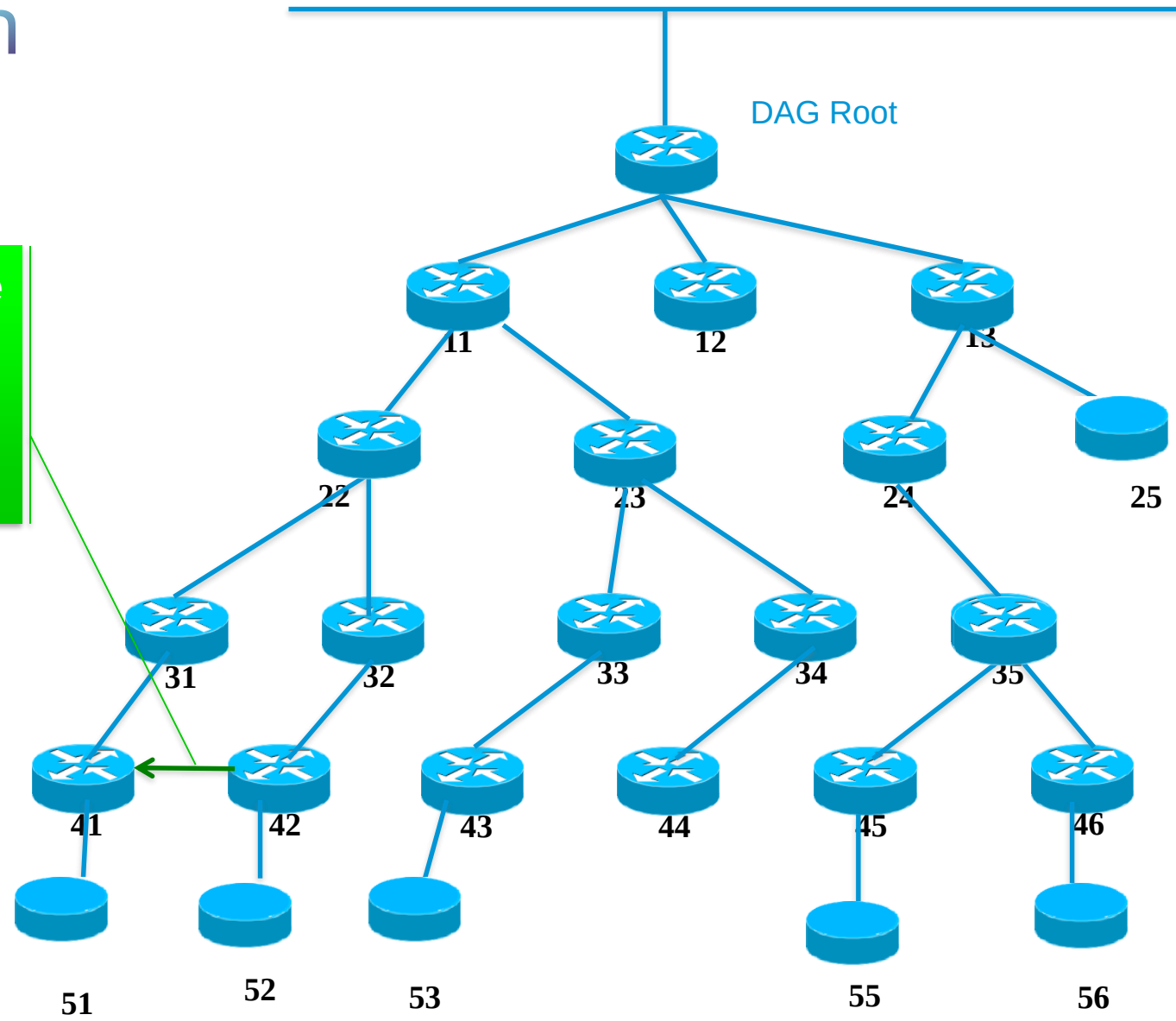


DAO projection

Application Server D

DAG Root

Storing mode DAO (forced) with lifetime along segment

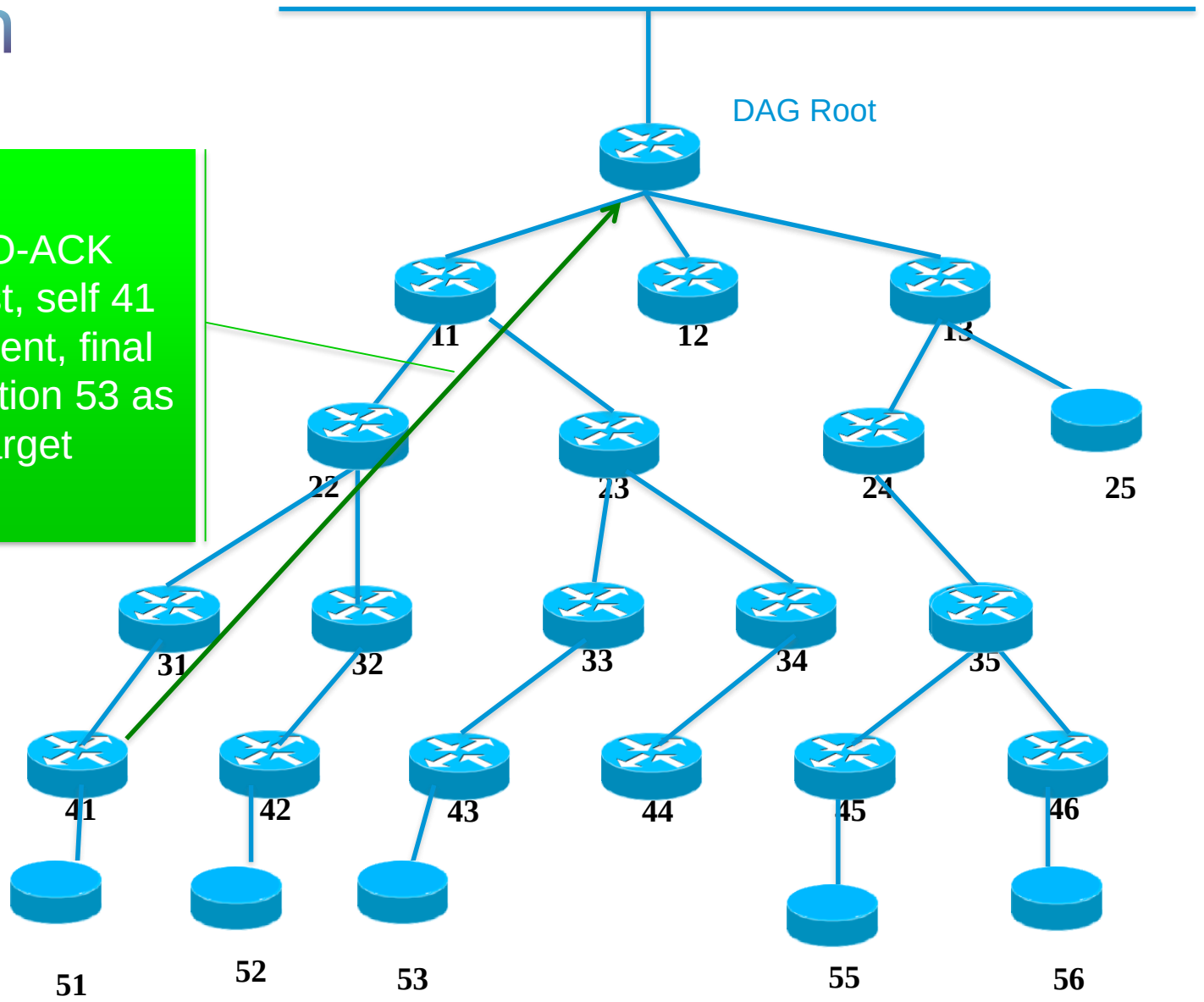


DAO projection

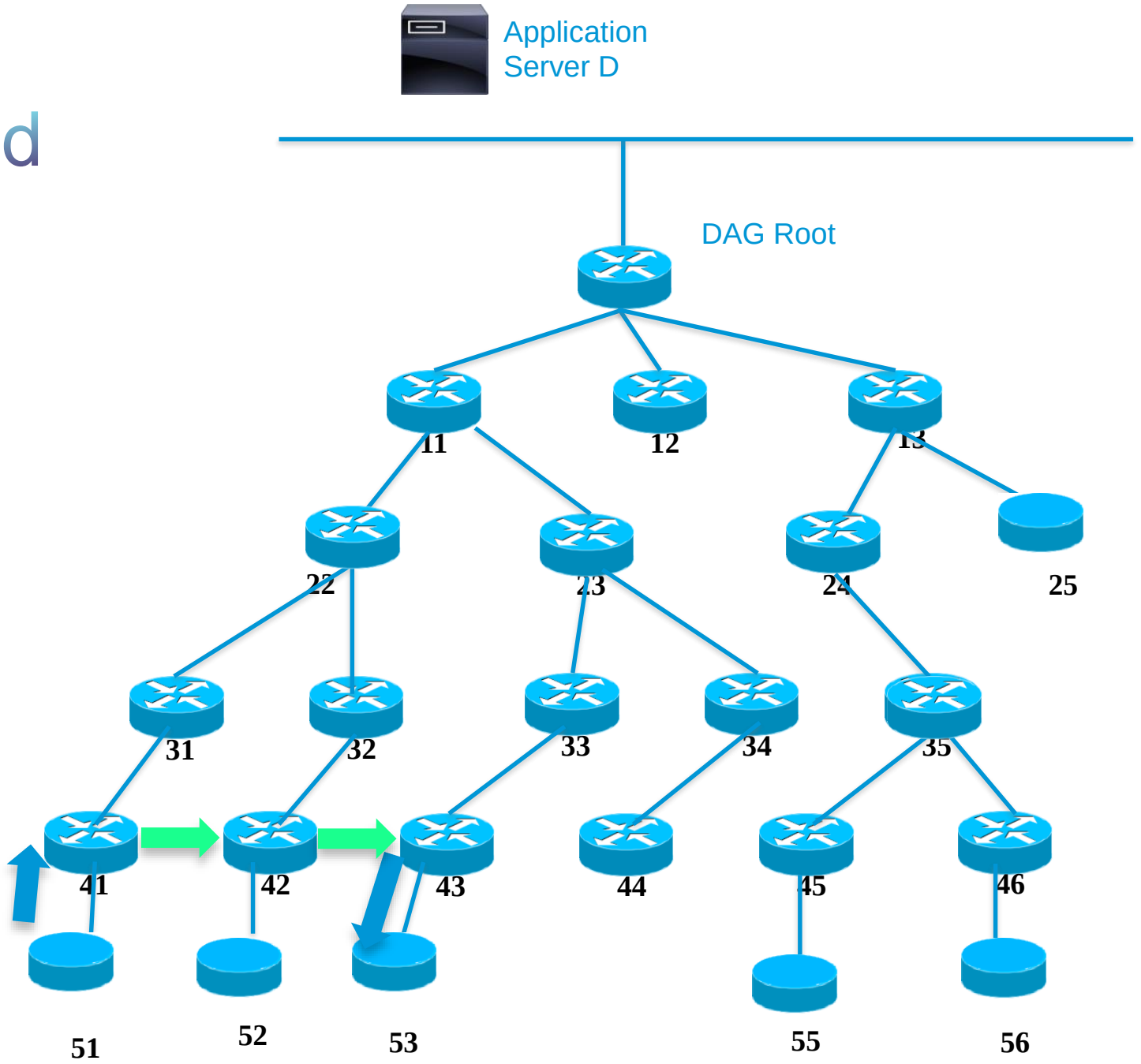
Application Server D

DAG Root

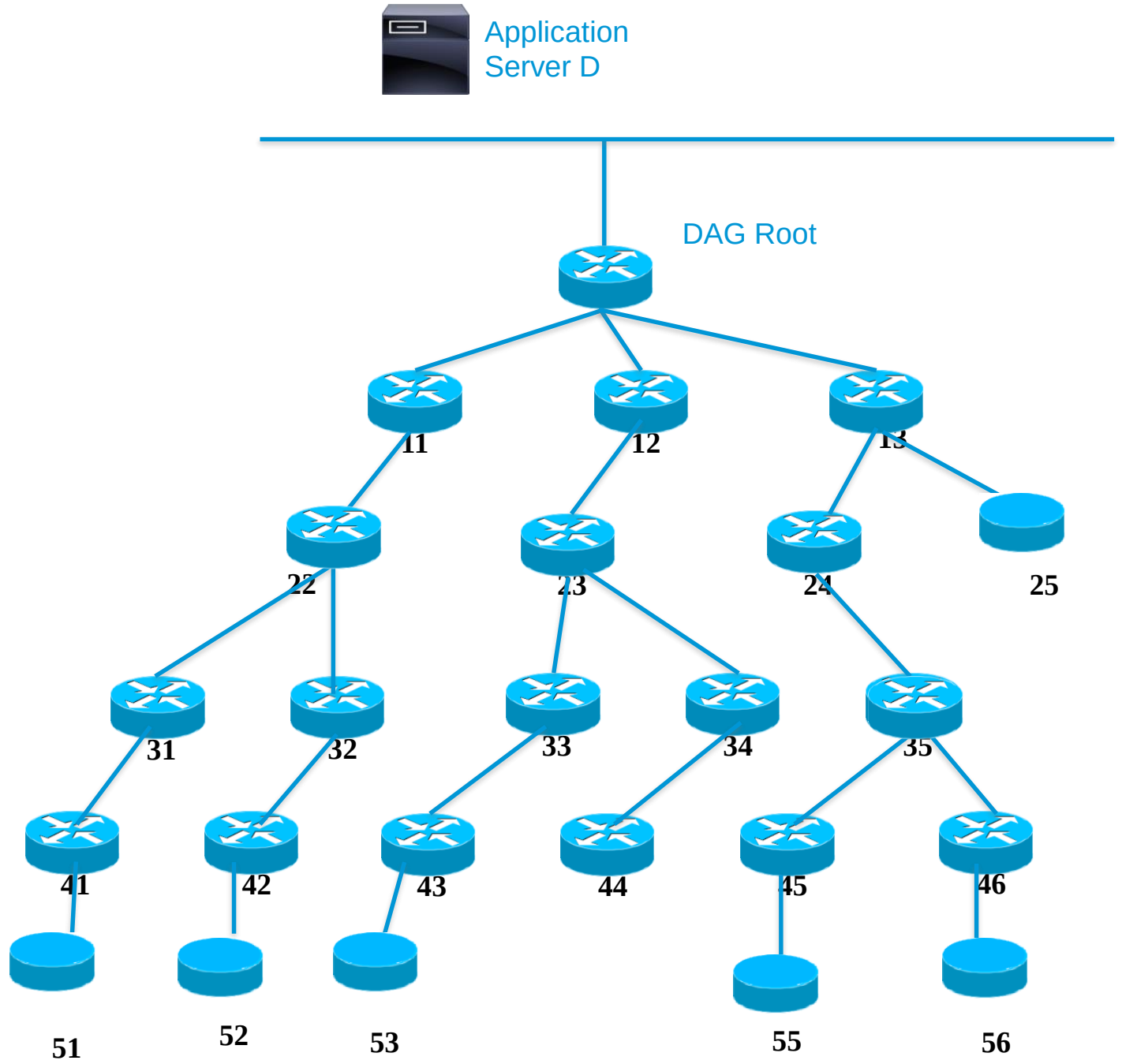
DAO-ACK unicast, self 41 as parent, final destination 53 as target

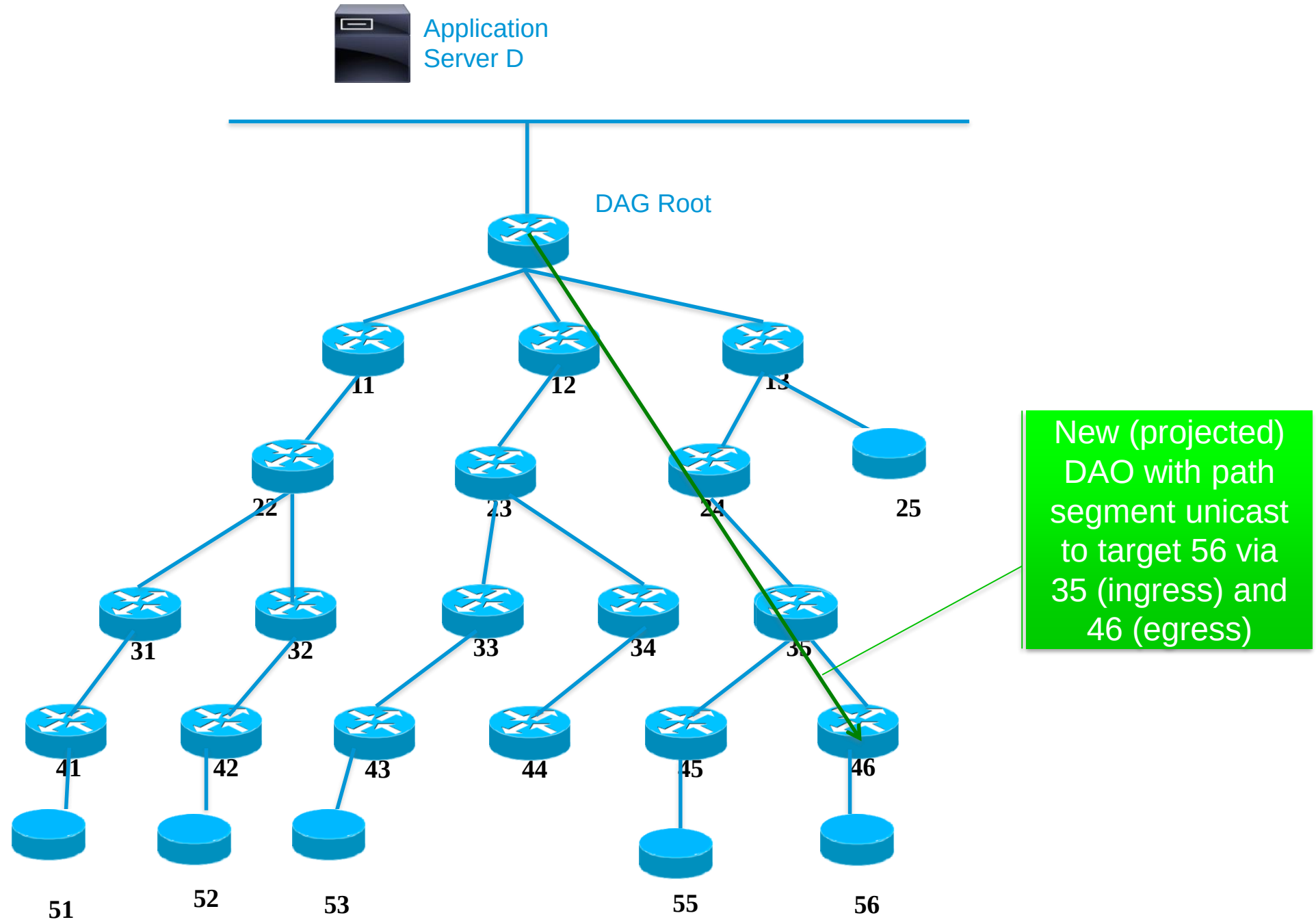


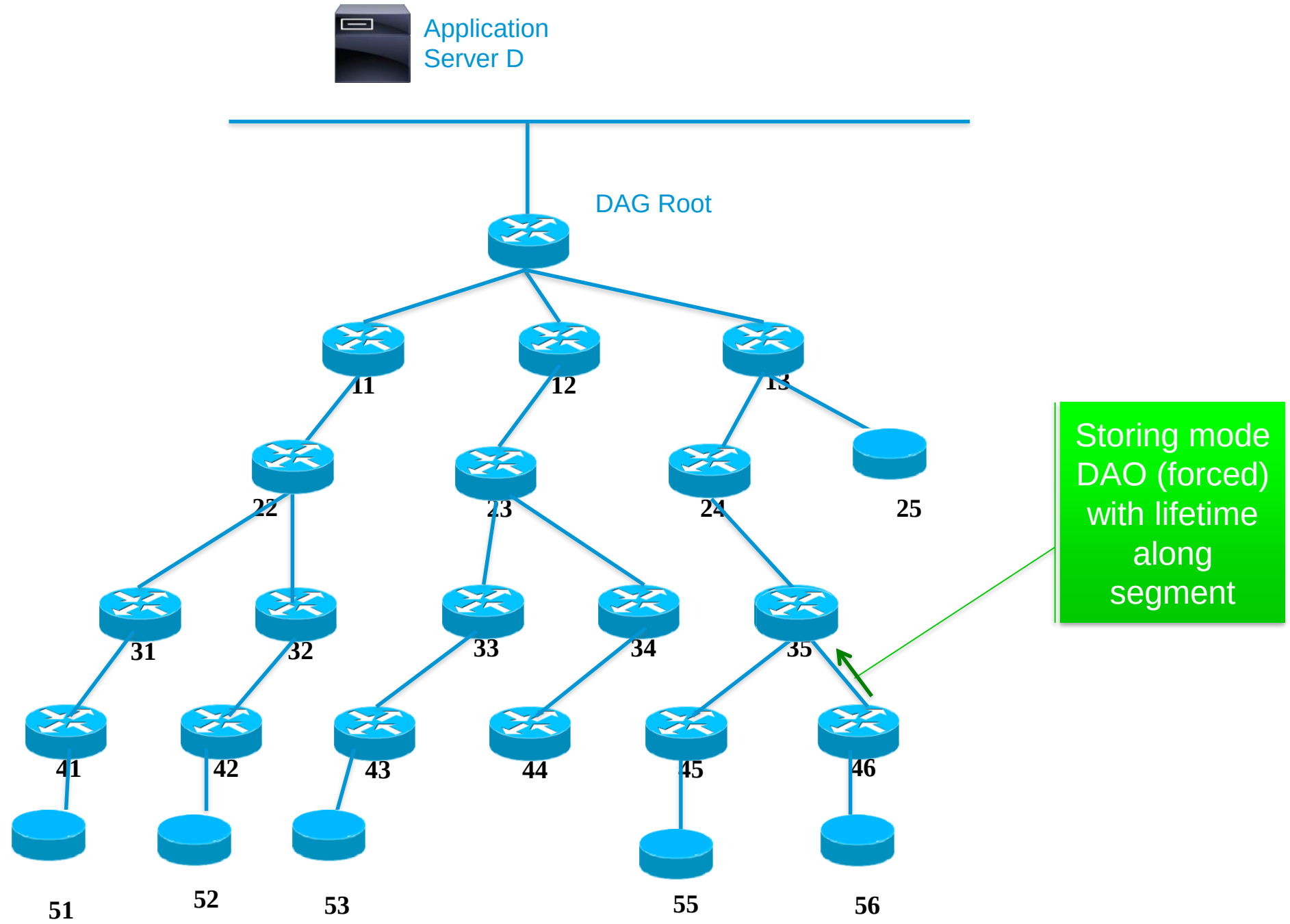
Optimized Path

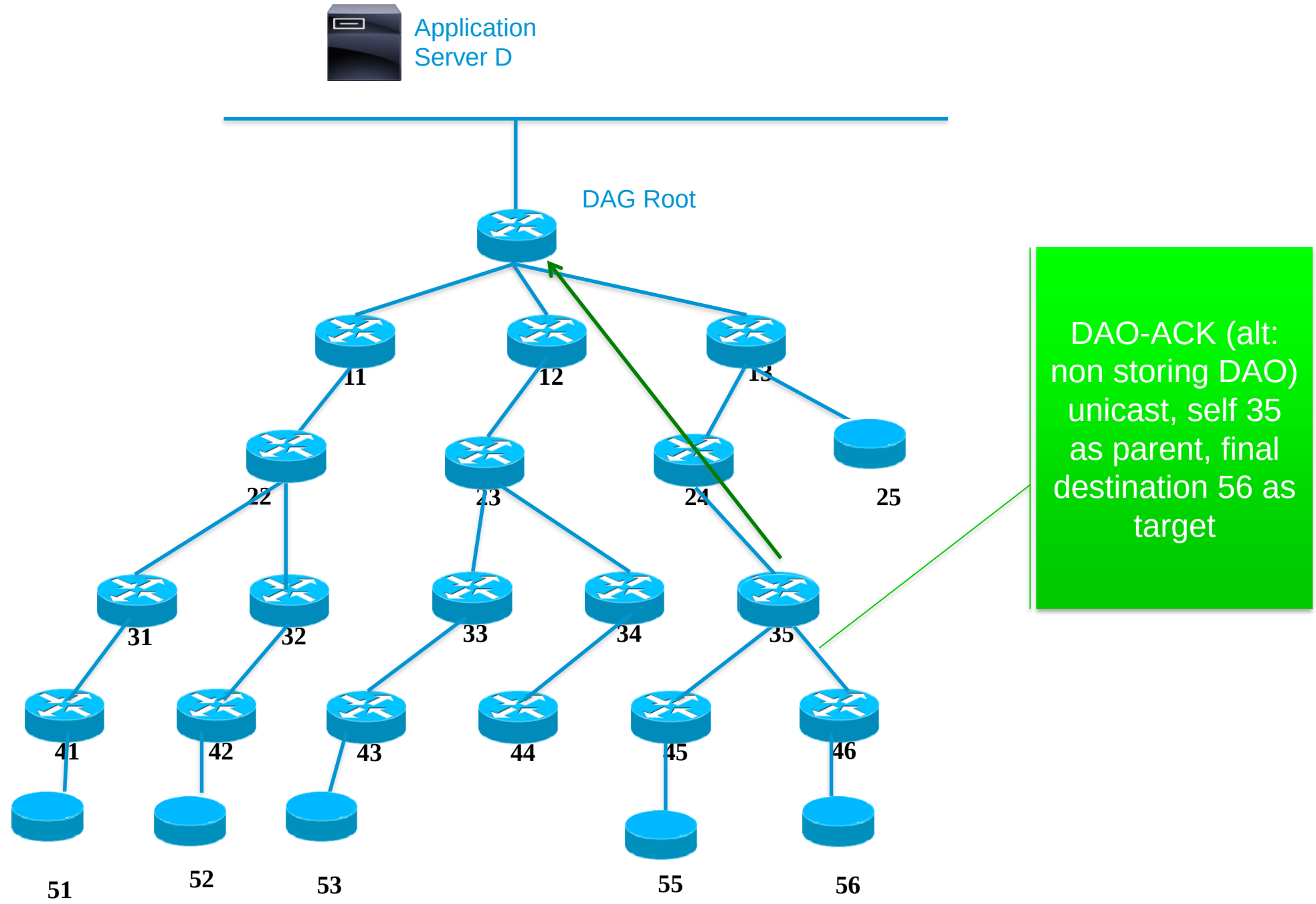


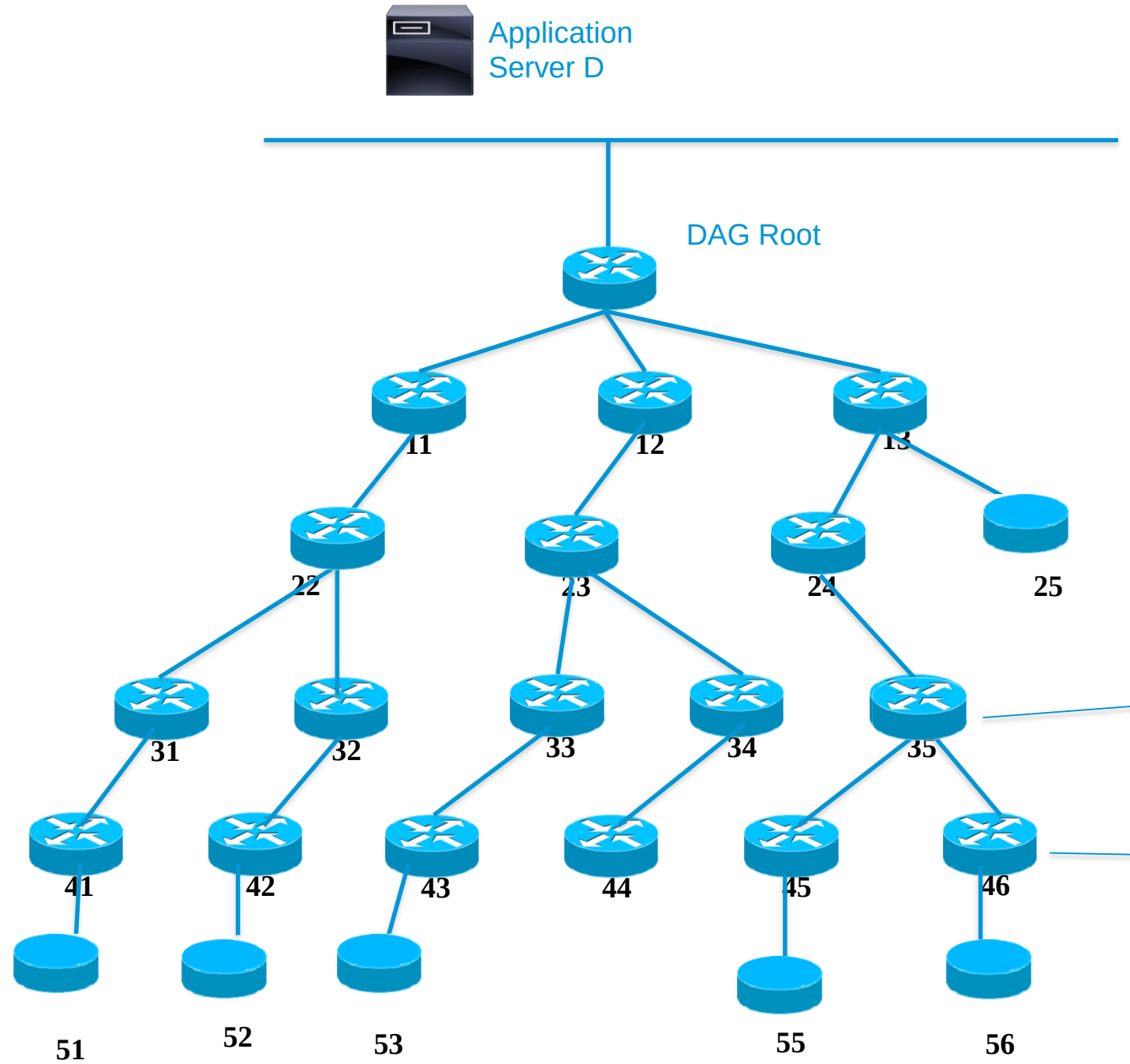
Existing non storing optimization


Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56


Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56
New (projected) DAO with path segment unicast to target 56 via 35 (ingress) and 46 (egress)


Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56
Storing mode
DAO (forced)
with lifetime
along
segment

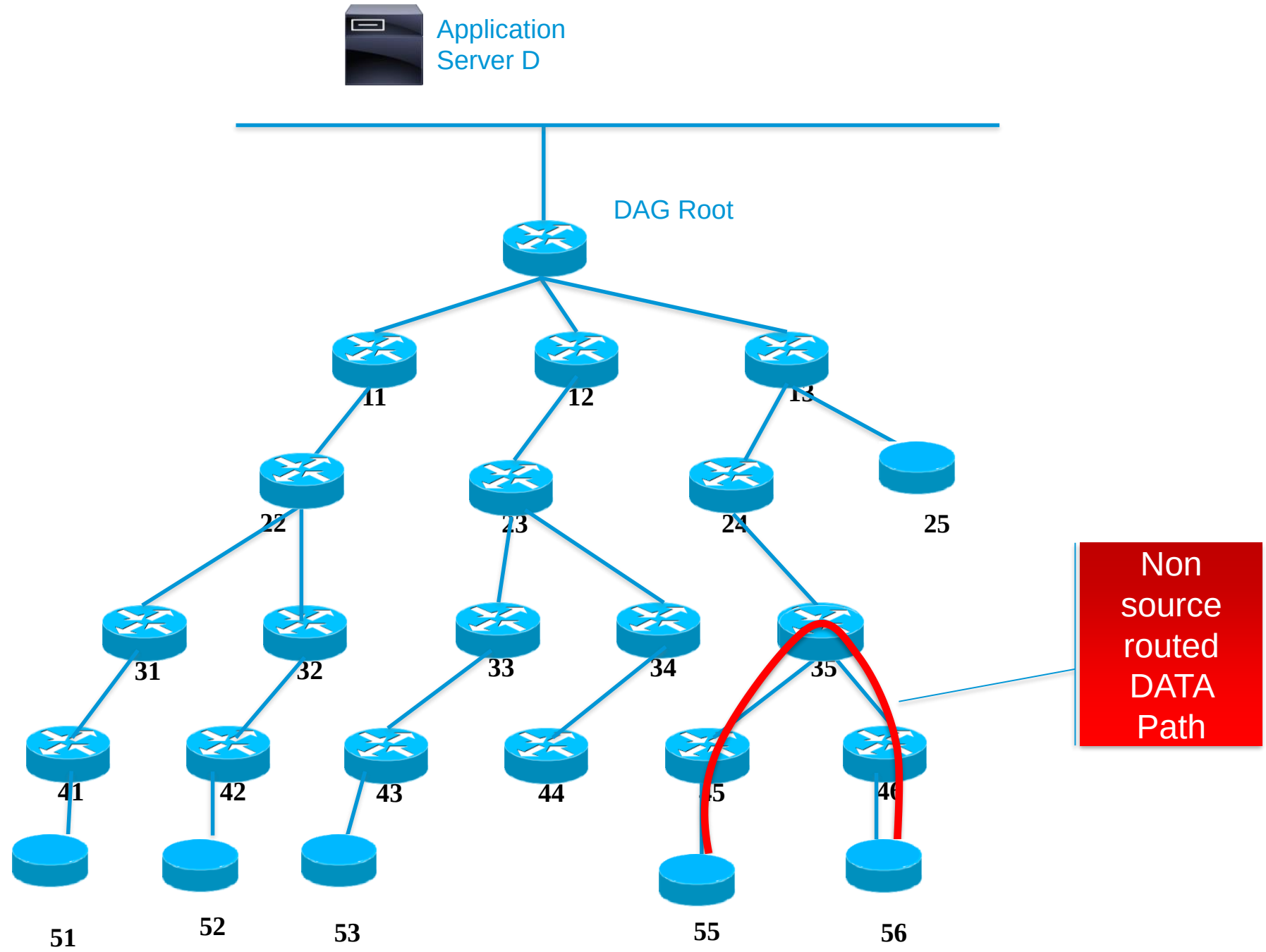

Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56
DAO-ACK (alt: non storing DAO) unicast, self 35 as parent, final destination 56 as target

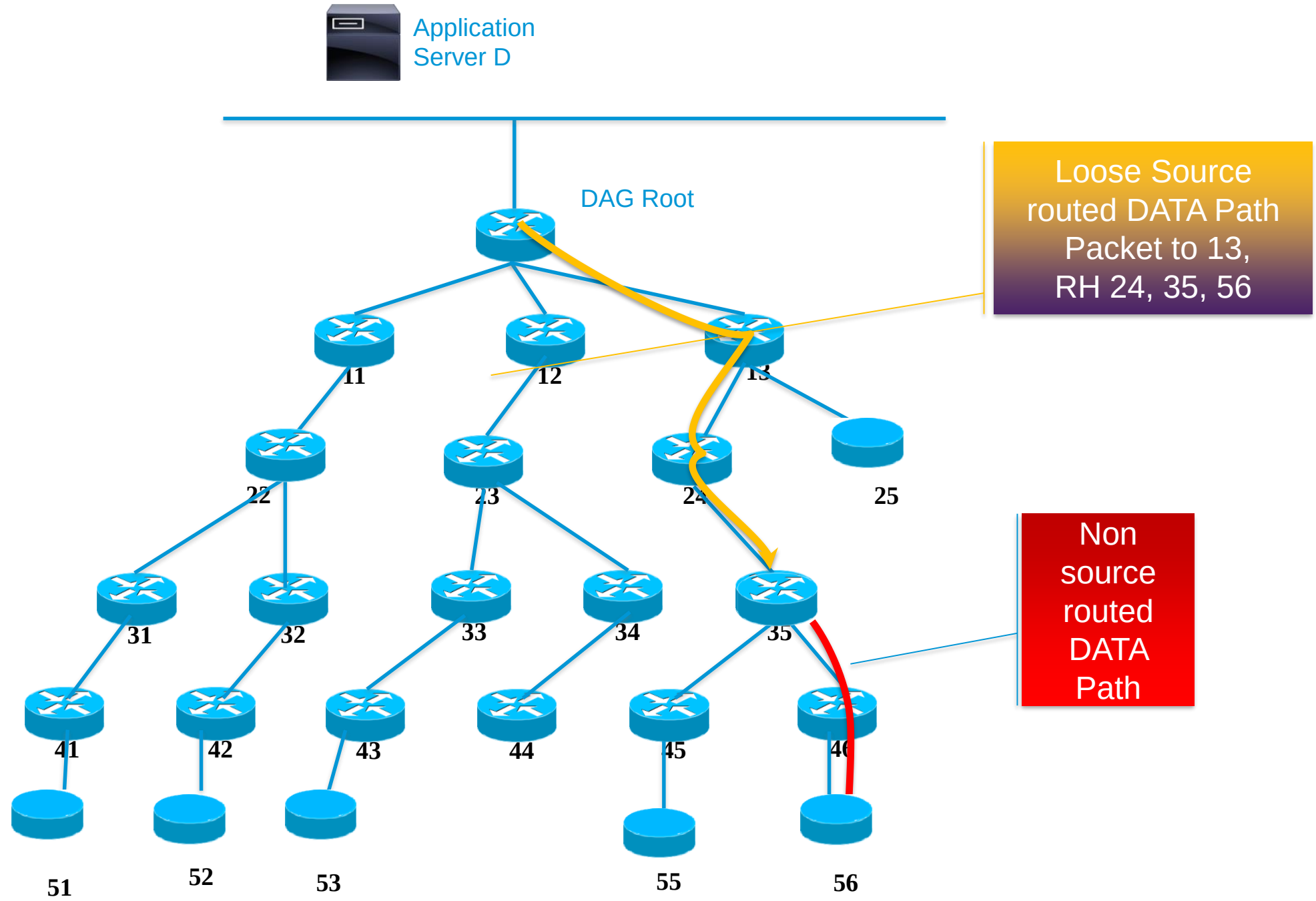


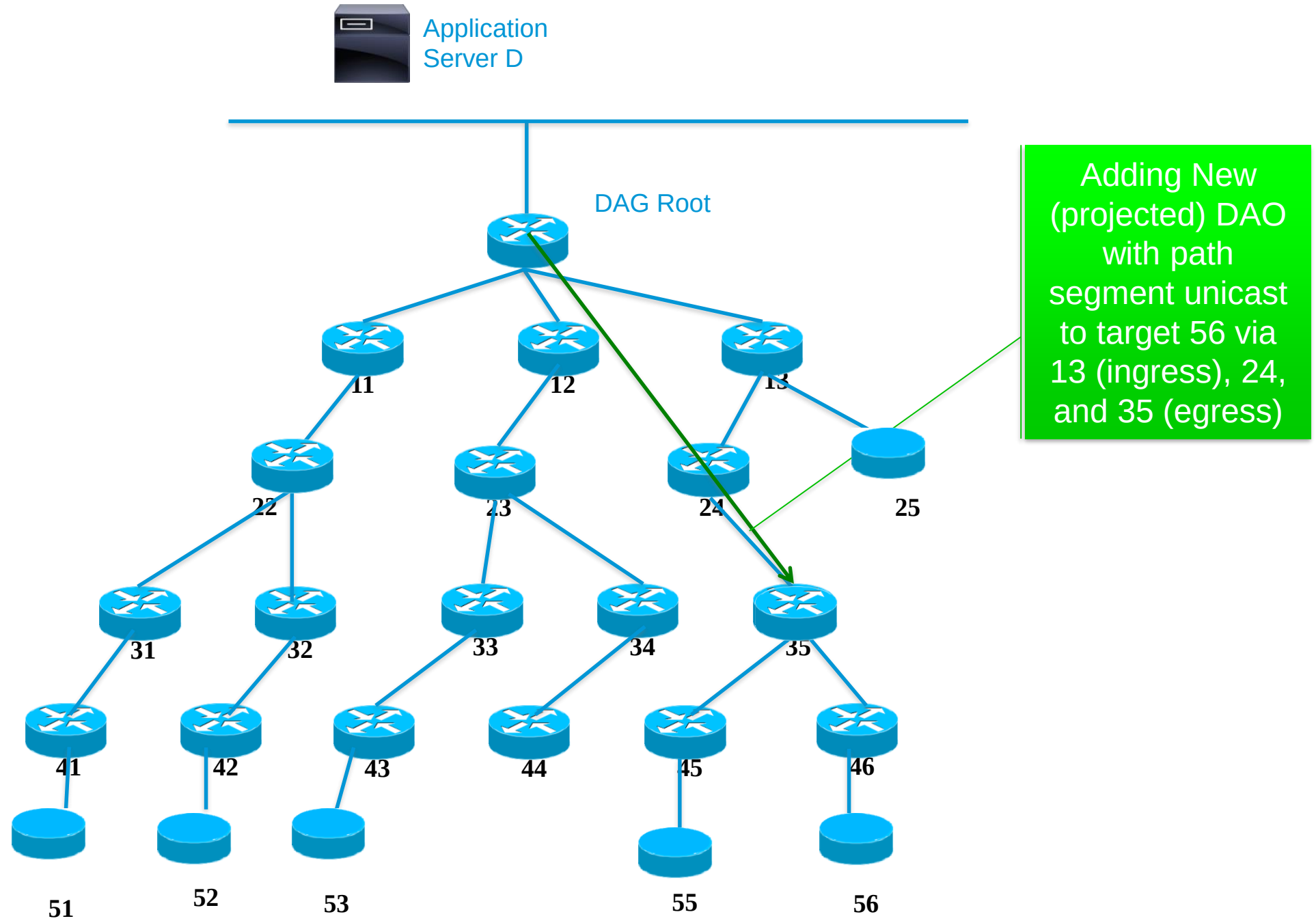
DAO from 46 installs a route to 56 in 35
(all nodes in projected route from ingress included to egress excluded)
=> egress should already have a route to target

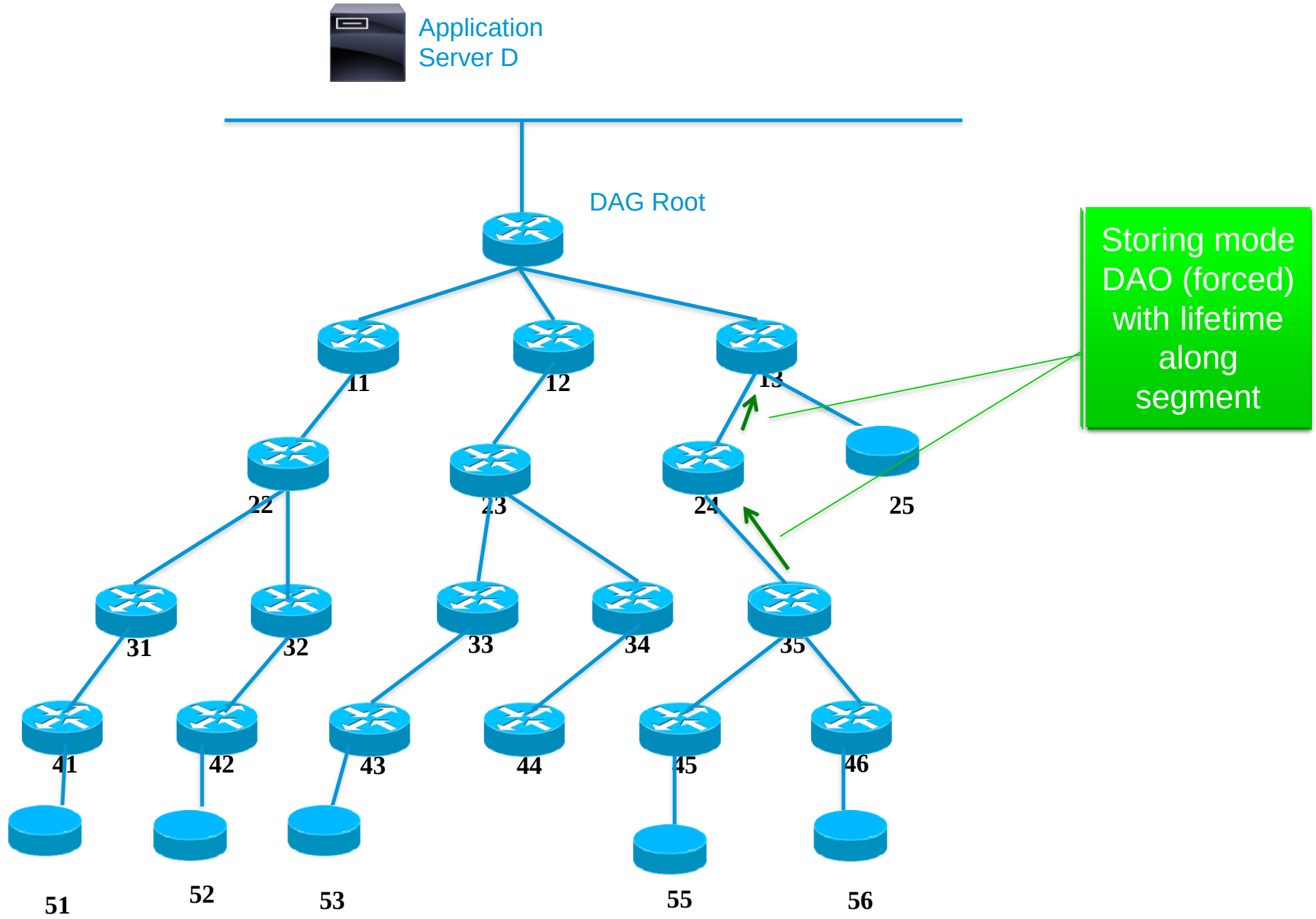
56 via 46

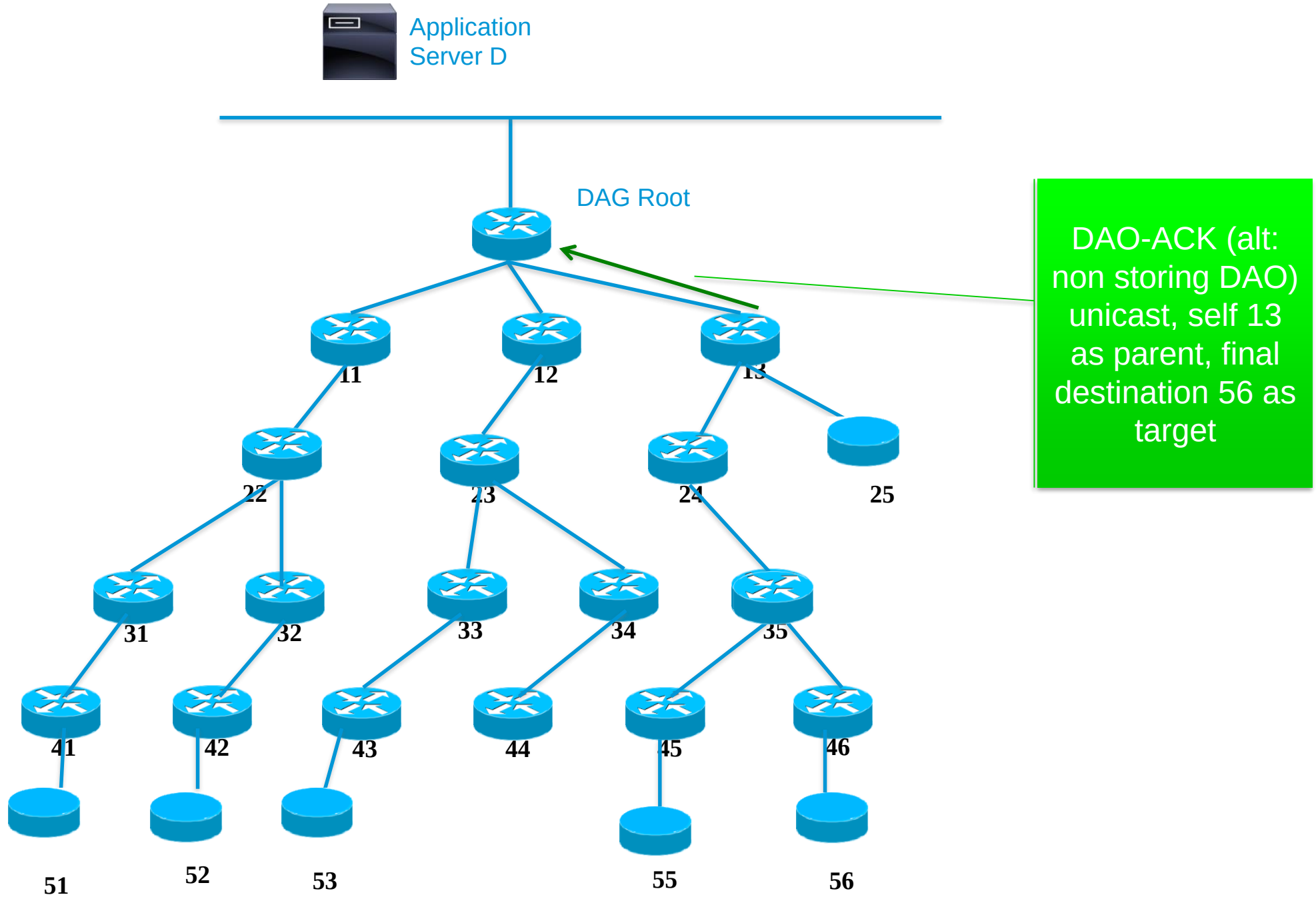
Preexisting connected route to 56

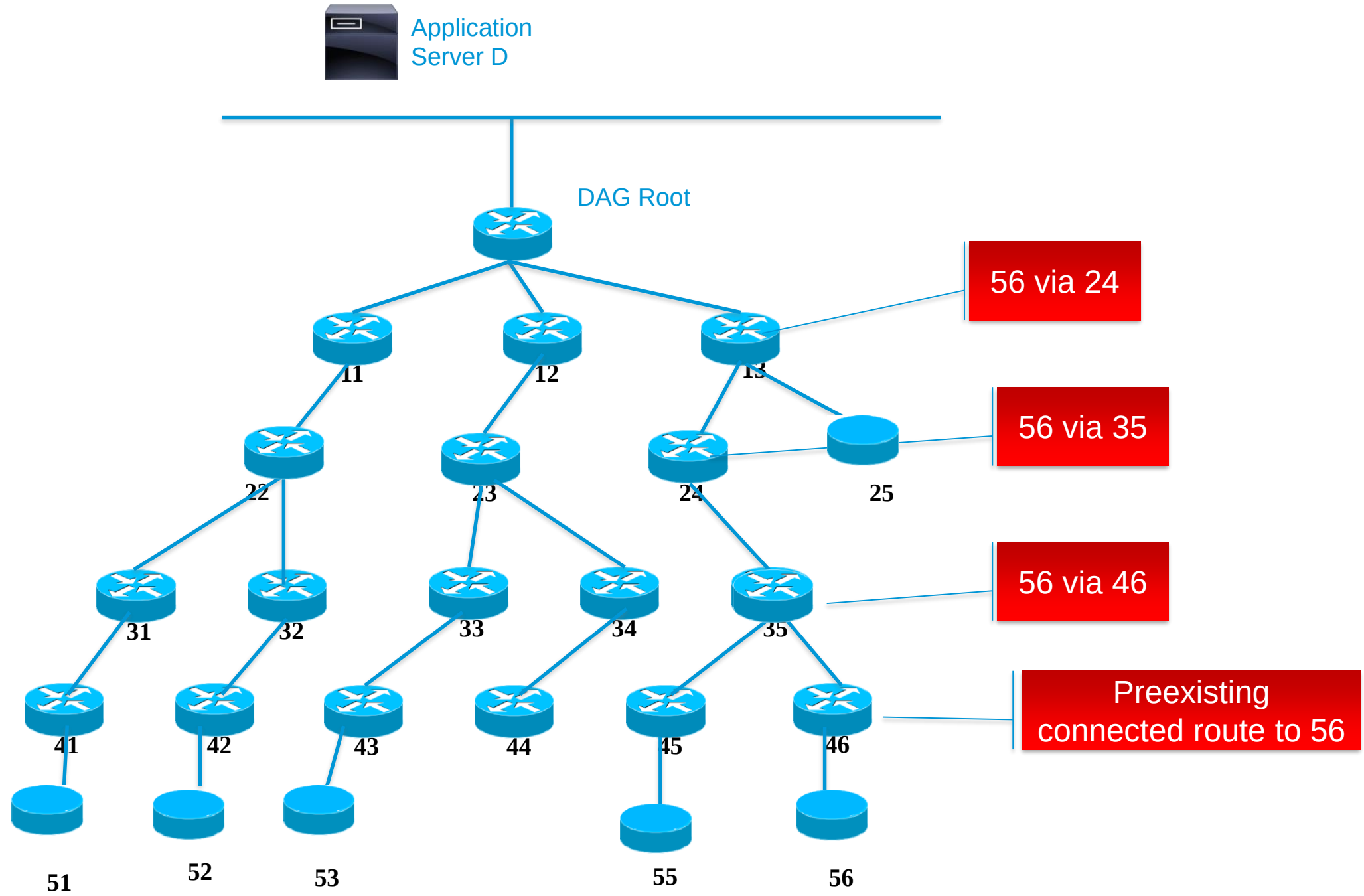

Application
Server D
DAG Root
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56
Non source routed DATA Path

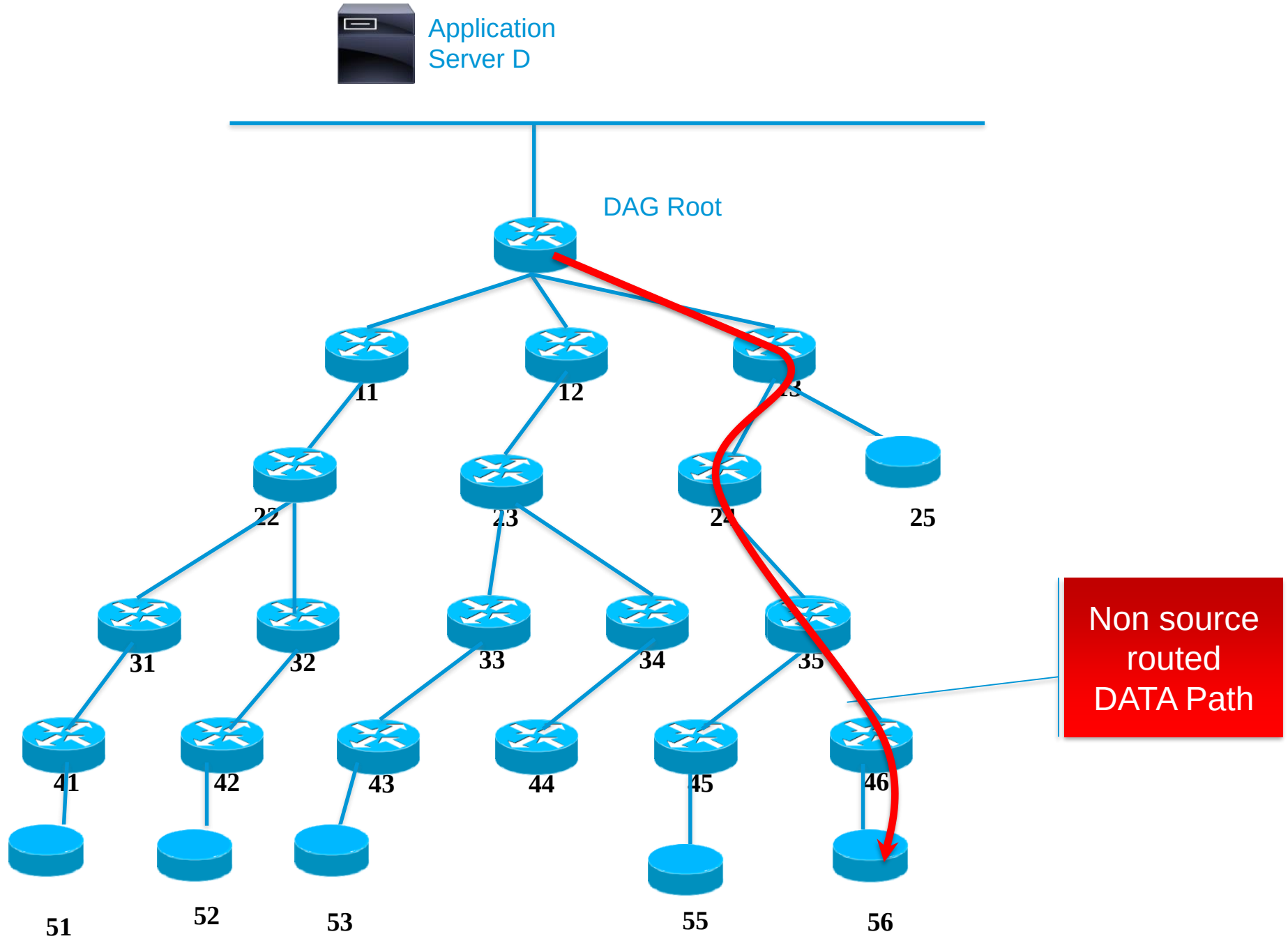

Application
Server D
DAG Root
Loose Source routed DATA Path Packet to 13, RH 24, 35, 56
Non source routed DATA Path
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56


Application
Server D
DAG Root
Adding New (projected) DAO with path segment unicast to target 56 via 13 (ingress), 24, and 35 (egress)
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

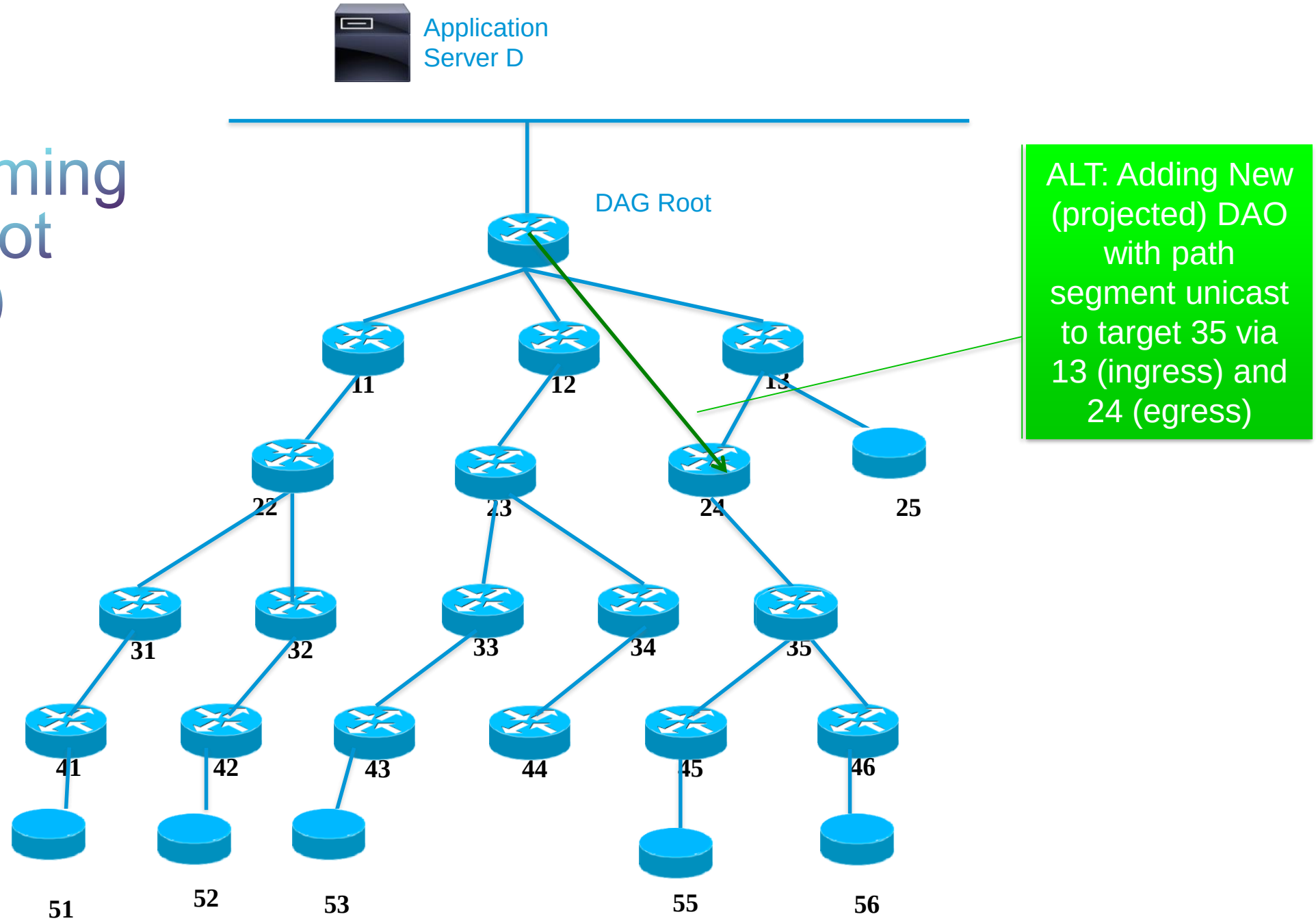

Application
Server D
DAG Root
Storing mode
DAO (forced)
with lifetime
along
segment
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

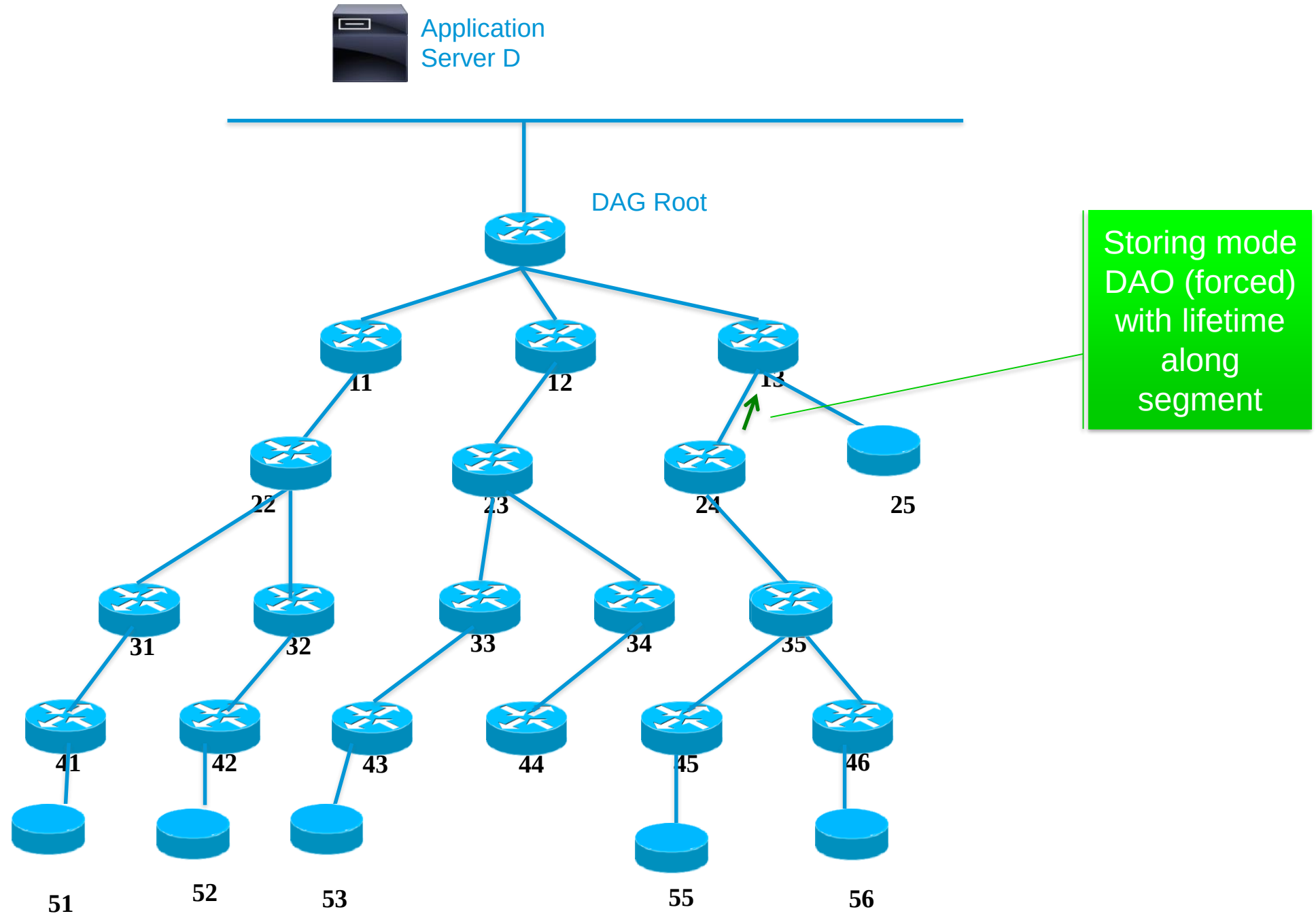

Application
Server D
DAG Root
DAO-ACK (alt: non storing DAO) unicast, self 13 as parent, final destination 56 as target
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

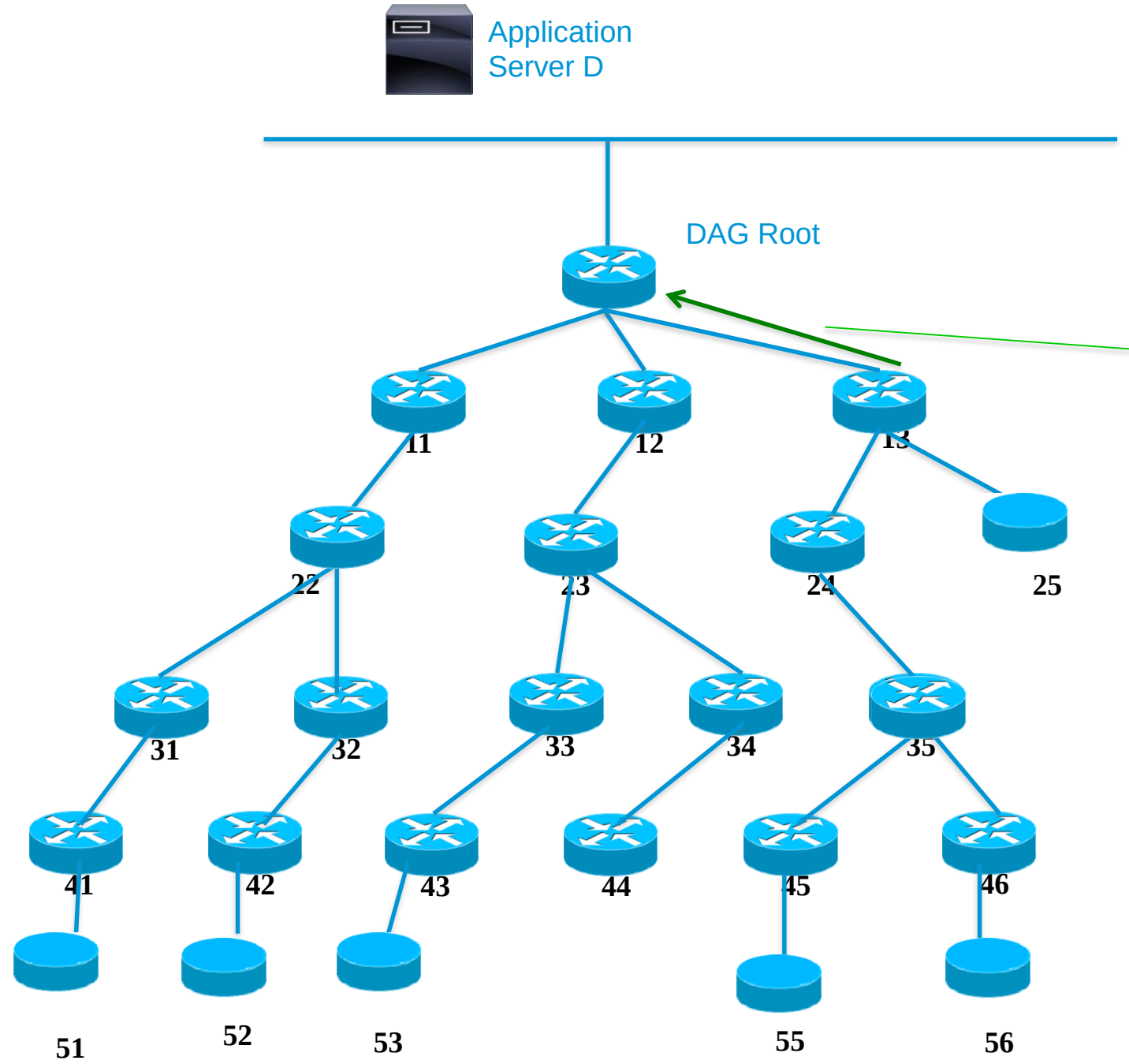

Application
Server D
DAG Root
56 via 24
56 via 35
56 via 46
Preexisting
connected route to 56
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56


Application
Server D
DAG Root
11
12
22
23
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56
Non source
routed
DATA Path

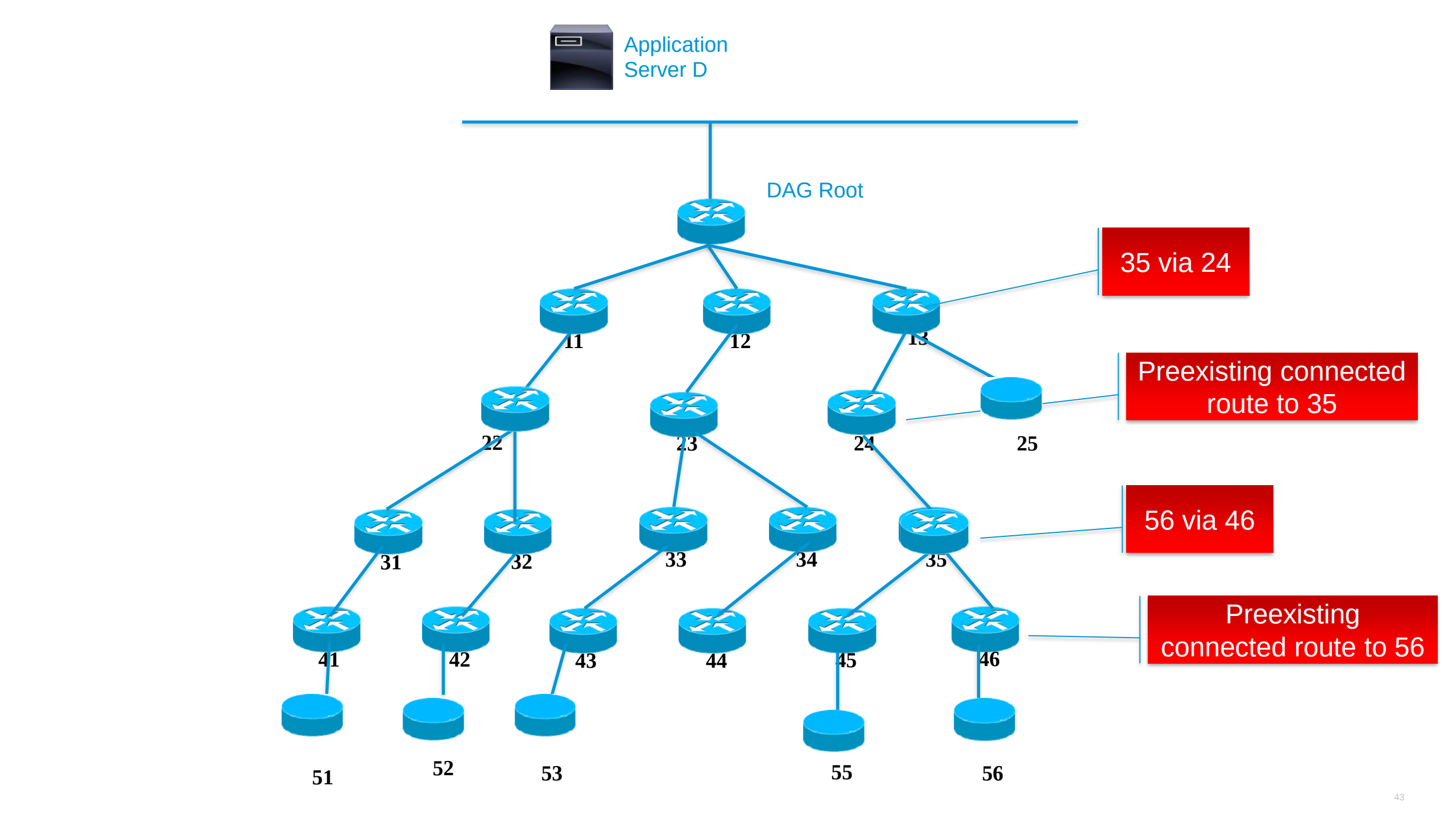
Alternate Programming By the root (Michael)

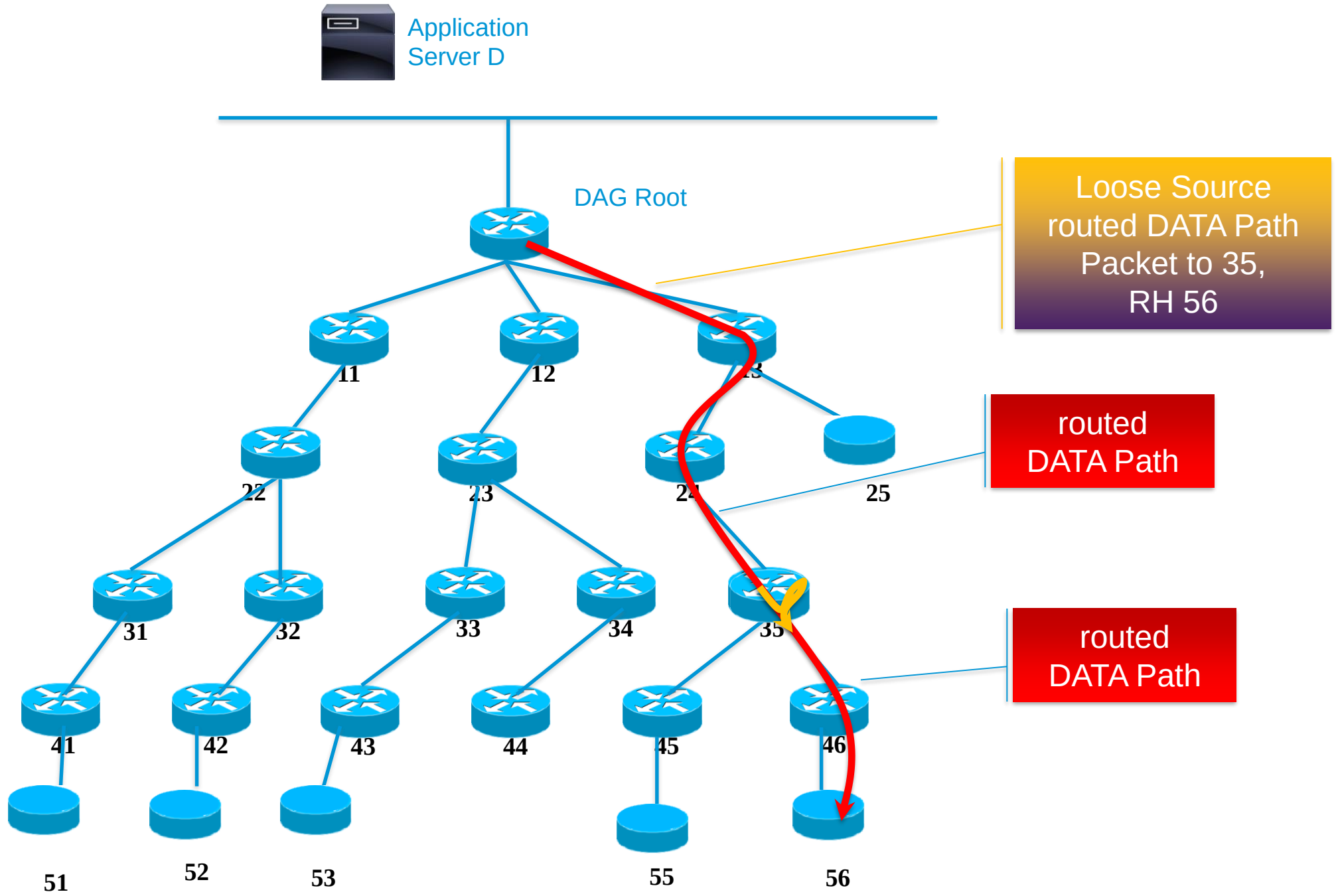



Application
Server D
DAG Root
Storing mode
DAO (forced)
with lifetime
along
segment
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56



DAO-ACK (alt: non storing DAO) unicast, self 13 as parent, final destination 56 as target


Application
Server D
DAG Root
35 via 24
Preexisting connected
route to 35
56 via 46
Preexisting
connected route to 56
11
12
13
22
23
24
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56


Application
Server D
DAG Root
Loose Source routed DATA Path Packet to 35, RH 56
routed DATA Path
routed DATA Path
11
12
22
23
25
31
32
33
34
35
41
42
43
44
45
46
51
52
53
55
56

Routing for RPL Leaves

draft-thubert-roll-unaware-leaves-05

Pascal Thubert

IETF 103

Bangkok

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 periodicallThis 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 Extension

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

IETF 103

Bangkok

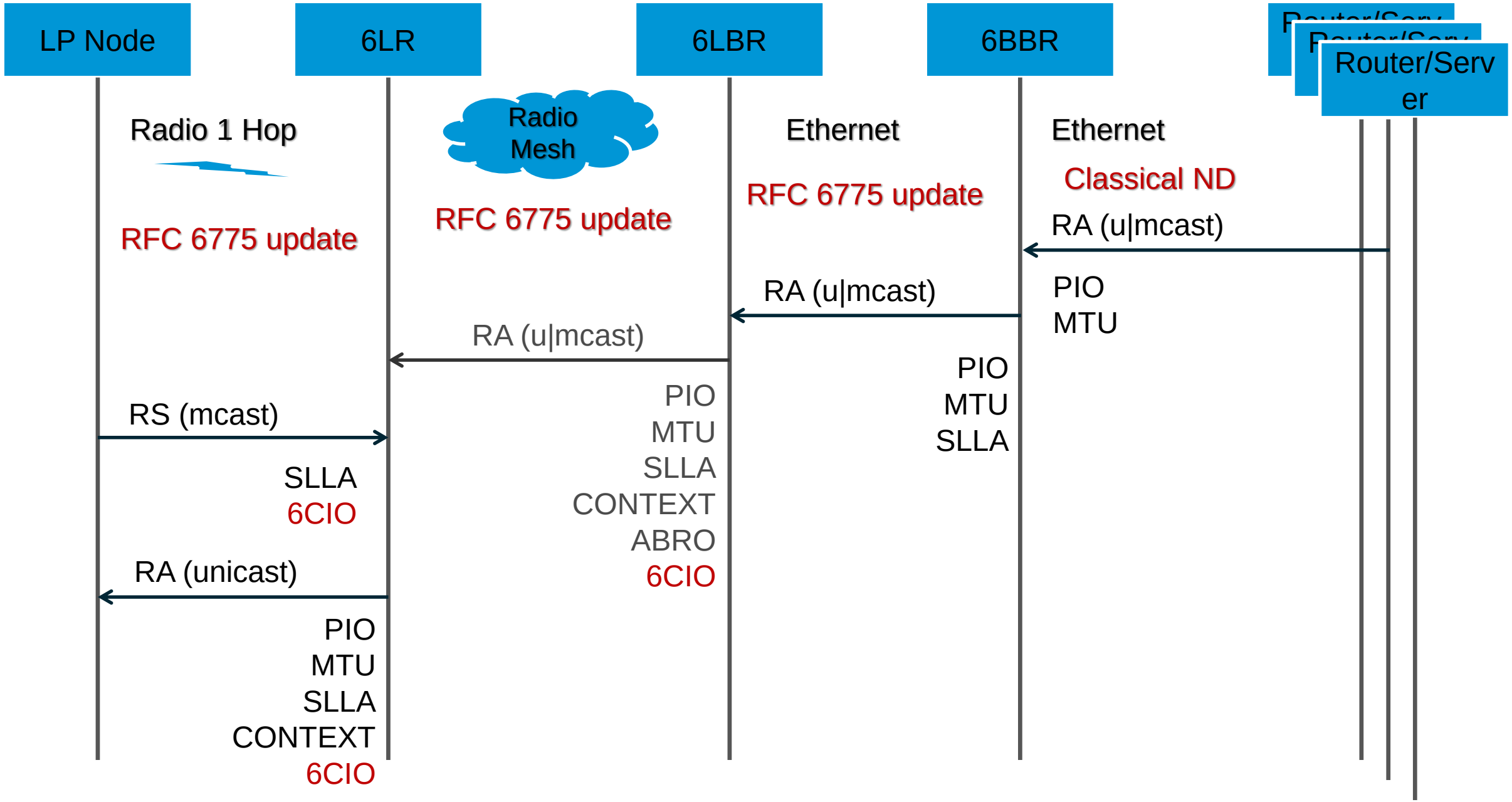
What are the 6LoWPAN ND extensions?

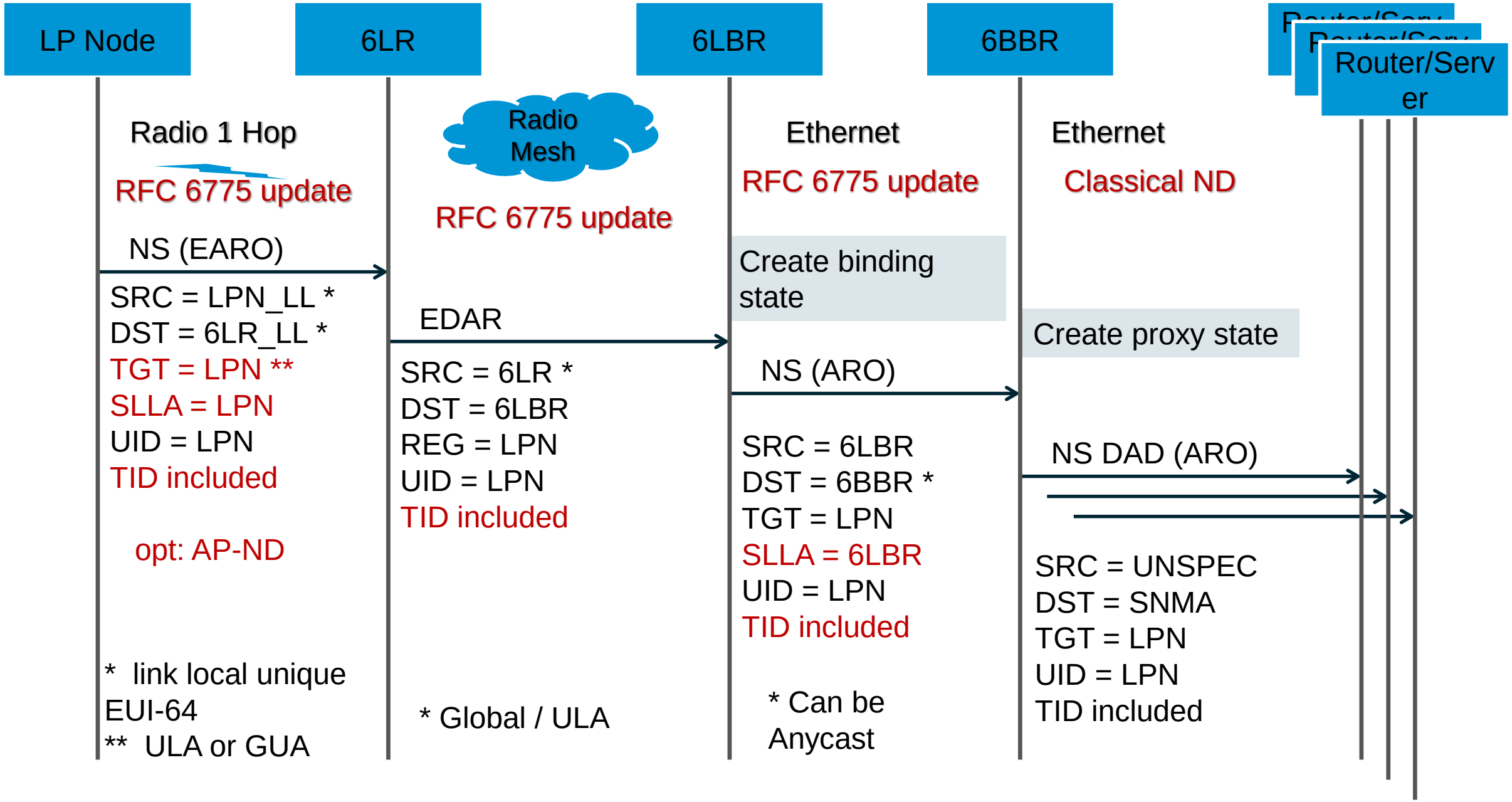
Provide for draft-thubert-6lo-rfc6775-update-reqs

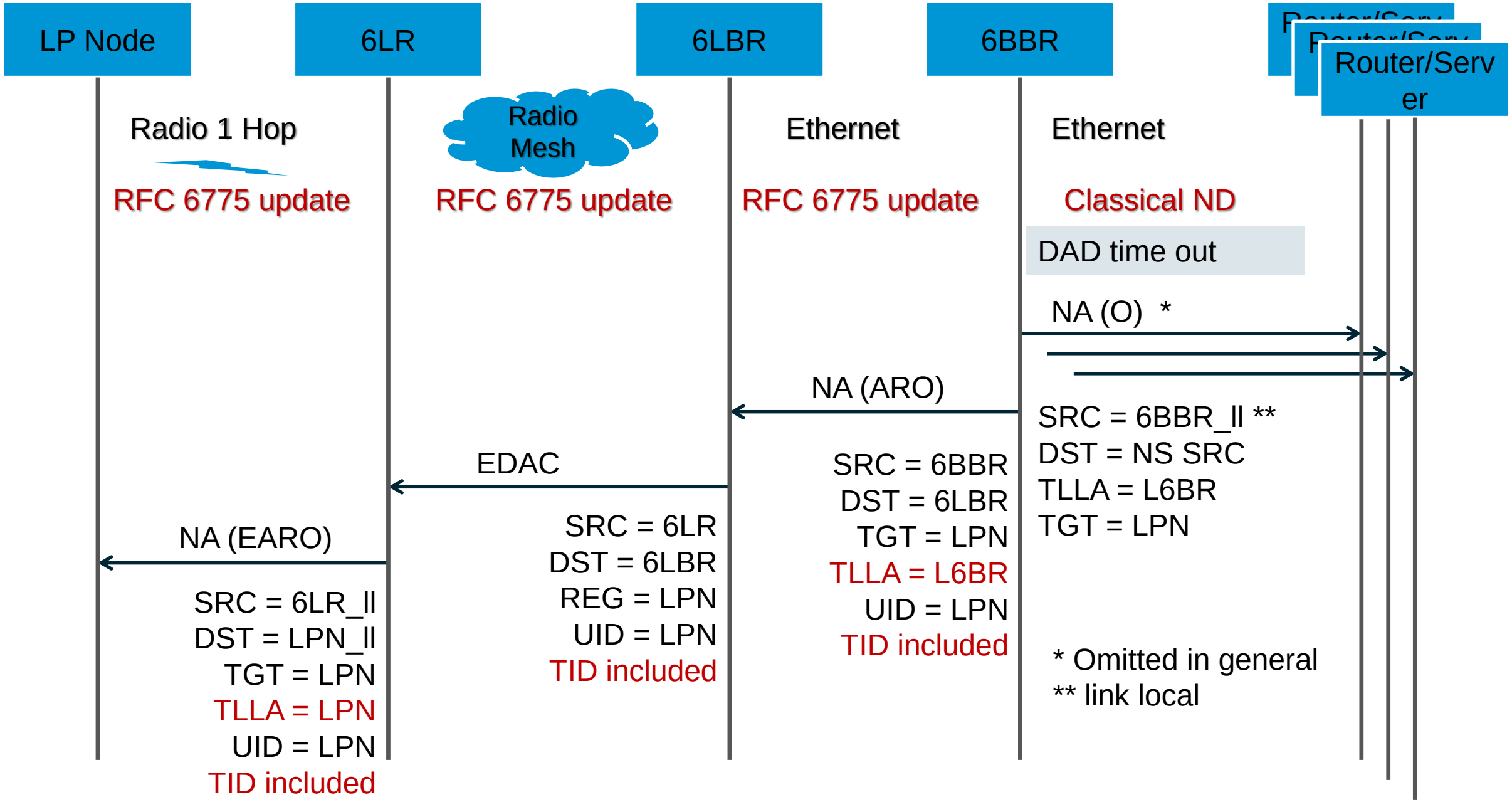
- draft-ietf-6lo-rfc6775-update
  - Simplifies the protocol (no DAR/DAC for LL, no secondary NC)
  - Enables proxy registration
- draft-ietf-6lo-ap-nd
  - Protects addresses against theft (Crypto ID in registration)
- draft-ietf-6lo-backbone-router
  - Federates 6lo meshes over a high speed backbone
  - ND proxy that mimics 802.11 association but at Layer 3

RFC 6775 Update

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


LP Node
6LR
6LBR
6BBR
Router/Server
Radio 1 Hop
RFC 6775 update
Radio Mesh
RFC 6775 update
Ethernet
RFC 6775 update
Ethernet
Classical ND
RA (u|mcast)
PIO
MTU
RA (u|mcast)
RA (u|mcast)
PIO
MTU
SLLA
PIO
MTU
SLLA
CONTEXT
ABRO
6CIO
RS (mcast)
SLLA
6CIO
RA (unicast)
PIO
MTU
SLLA
CONTEXT
6CIO


LP Node
6LR
6LBR
6BBR
Router/Server
Radio 1 Hop
RFC 6775 update
Radio Mesh
RFC 6775 update
Ethernet
RFC 6775 update
Ethernet
Classical ND
NS (EARO)
SRC = LPN_LL *
DST = 6LR_LL *
TGT = LPN **
SLLA = LPN
UID = LPN
TID included
opt: AP-ND
EDAR
SRC = 6LR *
DST = 6LBR
REG = LPN
UID = LPN
TID included
Create binding state
NS (ARO)
SRC = 6LBR
DST = 6BBR *
TGT = LPN
SLLA = 6LBR
UID = LPN
TID included
Create proxy state
NS DAD (ARO)
SRC = UNSPEC
DST = SNMA
TGT = LPN
UID = LPN
TID included
* link local unique EUI-64
** ULA or GUA
* Global / ULA
* Can be Anycast


LP Node
6LR
6LBR
6BBR
Router/Server
Radio 1 Hop
Radio Mesh
Ethernet
Ethernet
RFC 6775 update
RFC 6775 update
RFC 6775 update
Classical ND
DAD time out
NA (O) *
NA (ARO)
SRC = 6BBR_ll **
DST = NS SRC
TLLA = L6BR
TGT = LPN
EDAC
SRC = 6BBR
DST = 6LBR
TGT = LPN
TLLA = L6BR
UID = LPN
TID included
NA (EARO)
SRC = 6LR
DST = 6LBR
REG = LPN
UID = LPN
TID included
SRC = 6LR_ll
DST = LPN_ll
TGT = LPN
TLLA = LPN
UID = LPN
TID included
* Omitted in general
** link local

Current status

RFC 6775 Update

Draft-…-21

Past IESG review (based on -21)

- IANA steps

  https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-codes-type-157-code-suffix

- Done RFC Editor disambiguations

- RFC Editor state : RFC-EDITOR *

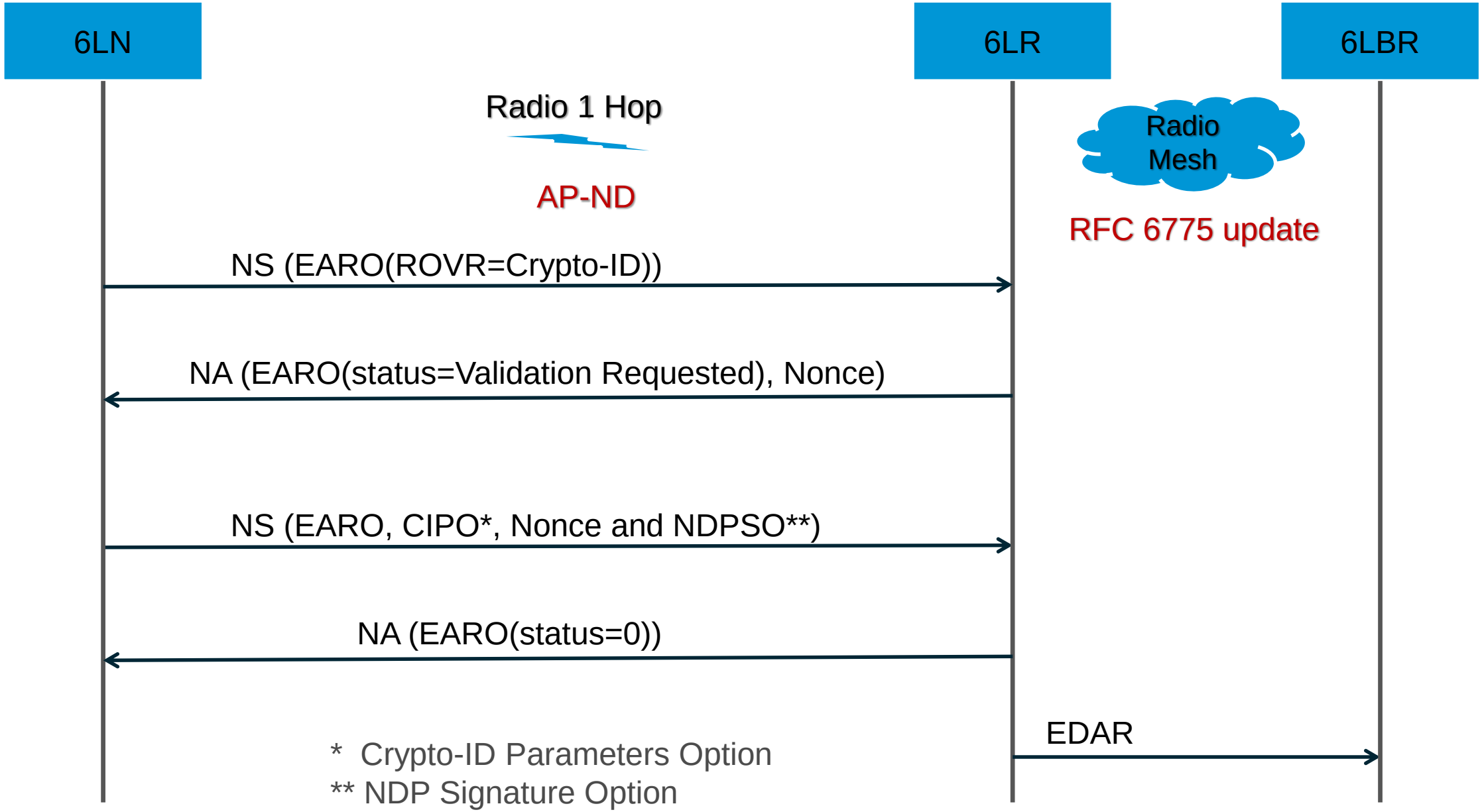
  * Awaiting final RFC Editor review before AUTH48

draft-ietf-6lo-ap-nd

P.Thubert, B. Sarikaya, M Sethi, R. Struik

Unmet expectations

- First come first Serve address registration
  - First registration for an address owns that address till it releases it
  - The network prevents hijacking
- Source address validation
  - Address must be topologically correct
  - Source of the packet owns the source address
- First Hop Security only?
  - Proxy ownership and routing advertisements not protected yet


6LN
6LR
6LBR
Radio 1 Hop
AP-ND
Radio Mesh
RFC 6775 update
NS (EARO(ROVR=Crypto-ID))
NA (EARO(status=Validation Requested), Nonce)
NS (EARO, CIPO*, Nonce and NDPSO**)
NA (EARO(status=0))
EDAR
* Crypto-ID Parameters Option
** NDP Signature Option

Recent changes

- Published -08
- René Struik joined as contributing author
- Updated the computation of the Crypto-ID
  - Crypto-Id in EARO is a truncated hash of the node's public-key
  - Digital signature (SHA-256/NIST P-256 or SHA-512/EdDSA) in NDPSO is executed on additional material (nonces, etc…, see updated section 6.2) for proof of ownership of the private key
  - Uses both nonces from the 6LN and 6LR
- Removed SHA-256 for EdCSA to comply with RFC 8032.

Security properties

- We made the size of the ROVR tunable so we can get high security. 64 bits seems inappropriate.
- At the moment a joining 6LN is challenged from the 6LR
  - The 6LBR MUST trust the 6LR
  - A rogue 6LR may pretend that it represents a 6LN that passed the challenge
  - Should we challenge all the way from the 6LBR?
  - Can the Crypto-ID be used in routing protocols, how?

Questions to the group

- Should we Extend the protection to the RPL domain?

draft-thubert-roll-unaware-leaves

P.Thubert

IETF 103

Bangkok

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 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.

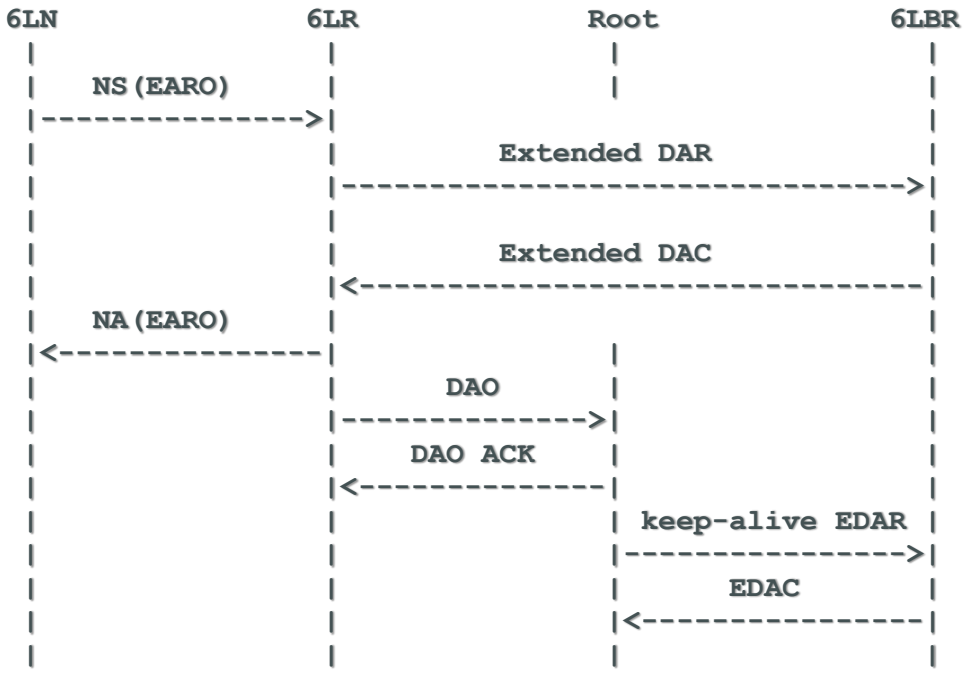
Mapping Fields from RPL DAO to NS(EARO) and EDA

- 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?

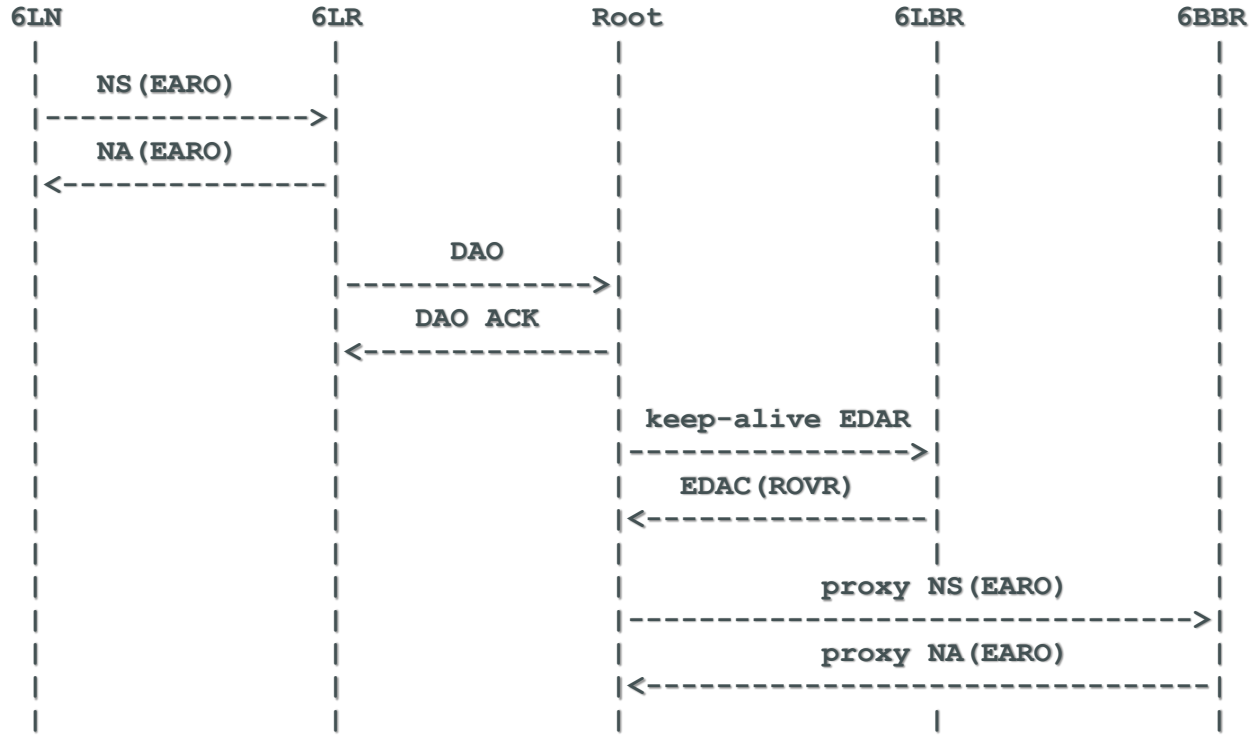
First registration

- 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