Root initiated routing state in RPL

draft-ietf-roll-dao-projection

P. Thubert, Ed.; R.A. Jadhav, M. Gillmore

Pascal Thubert

Interim Aug. 2021

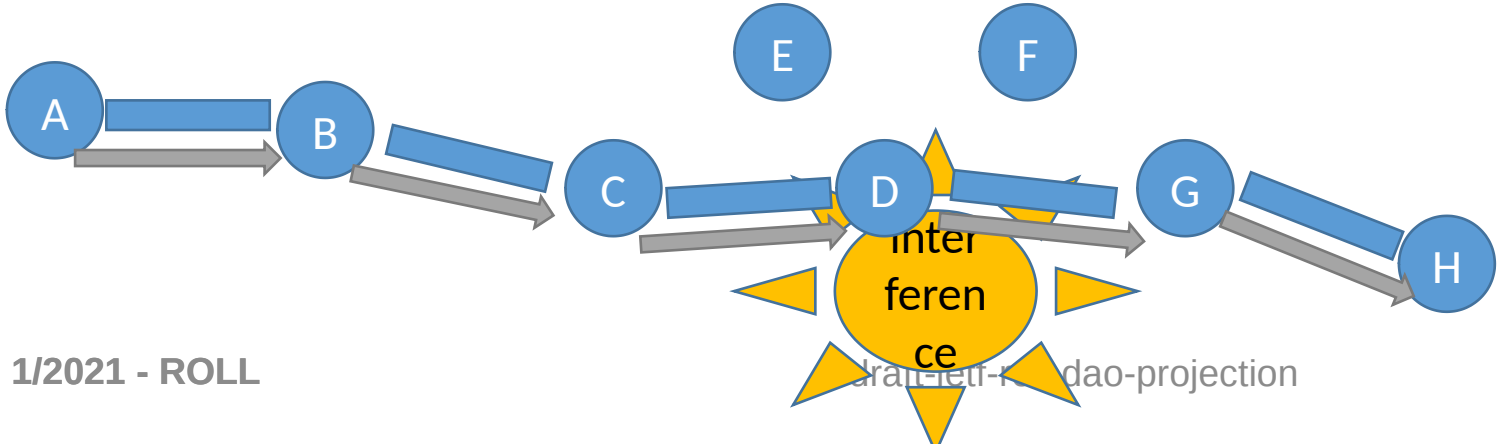
ROLL Virtual Meeting

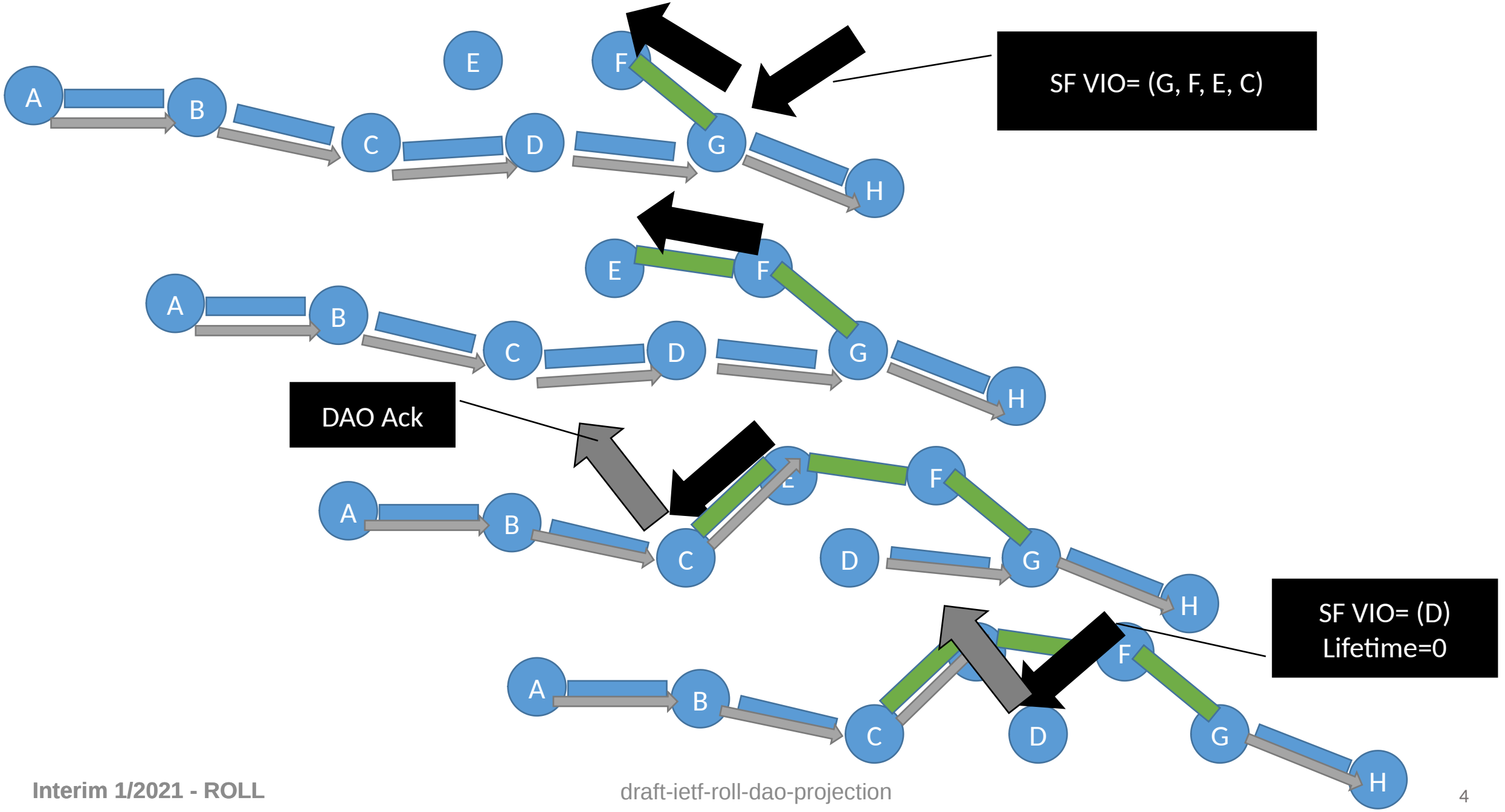
Status to the draft

- Published -17 to -19 since last interim
- Sibling address in SIO based on RFC 8505 discovery
- Simplification: PDR always from Track Ingress, which provides TrackID
- Clarification on maintenance: how to repair Track Segments (storing mode) and Legs (non-Storing e2e)
- Clarification: Policies to inject traffic on Tracks is OOS
- Reviews by S.V.R. Anand and Toerless (in progress)

On Maintenance (Anand's review)

- Anand's point on making the connectivity to the root more reliable.
- No-Path P-DAO indicated by lifetime of 0
- Can indicate a section of a Segment
- Updating can be lossless but possible misordering; e.g.:



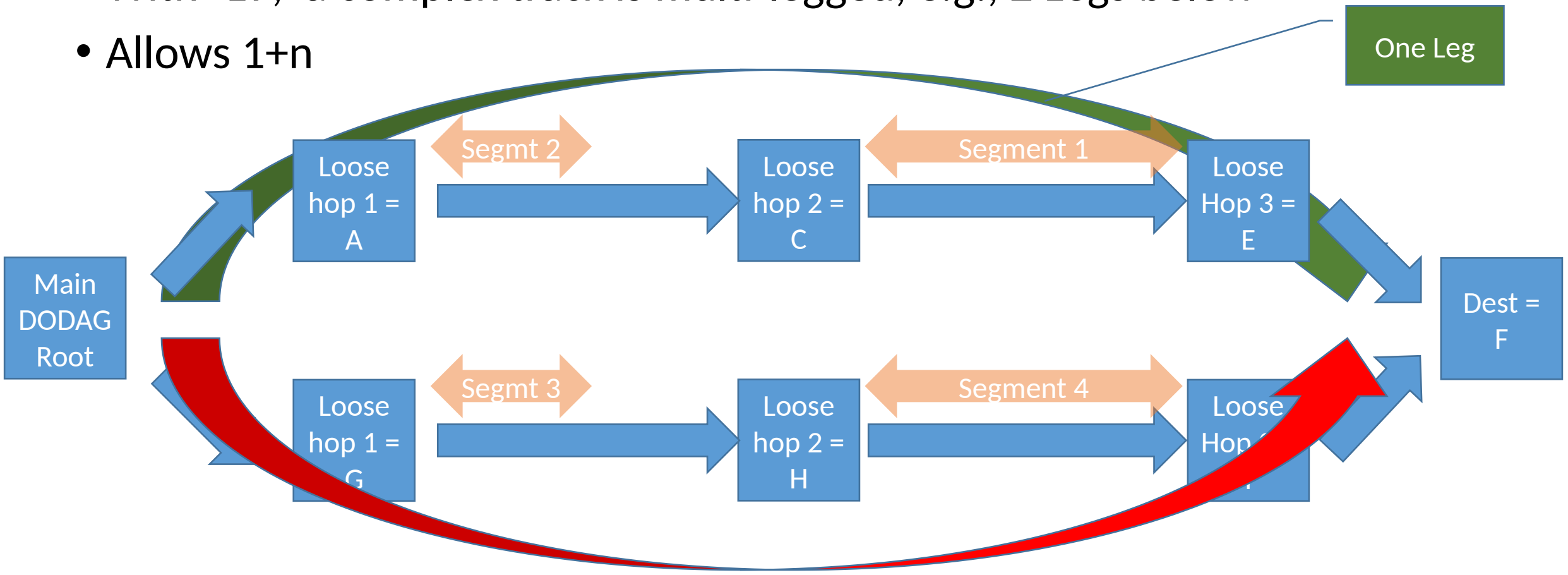

A
B
C
D
E
F
G
H
SF VIO= (G, F, E, C)
DAO Ack
SF VIO= (D)
Lifetime=0

On Track

- A Track is a RPL local instance installed by P DAO.
- Goal: to be equivalent to Non-Storing Mode main DODAG
- The spec only builds multi-legged Tracks (parallel or crossing)
  - Root is Track Ingress, Root address + DODAGID identify the Track
  - A Leg is signaled by a Non-Storing-Mode P-DAO message
  - Track Ingress encapsulates external packets (as in useofrplinfo)
  - Track Ingress places the SRH in the packet in source routed tracks
  - There cannot be non-storing segments (only Tracks within Tracks)
- Storing Mode P-DAO signals Segment of a Track or of main DODAG
  - Does not need re-encapsulation
  - Unless implicit Track => Do we support that ?

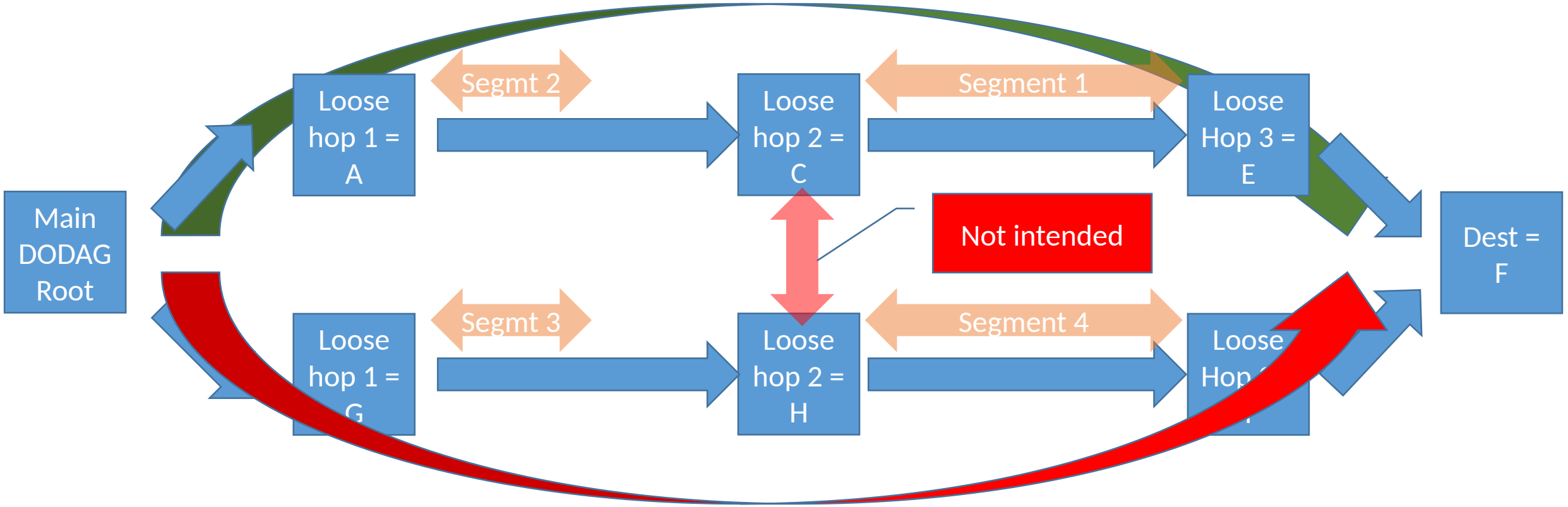
Discussion: building real DODAGs

- With -19, a complex track is multi-legged, e.g., 2 Legs below
- Allows 1+n



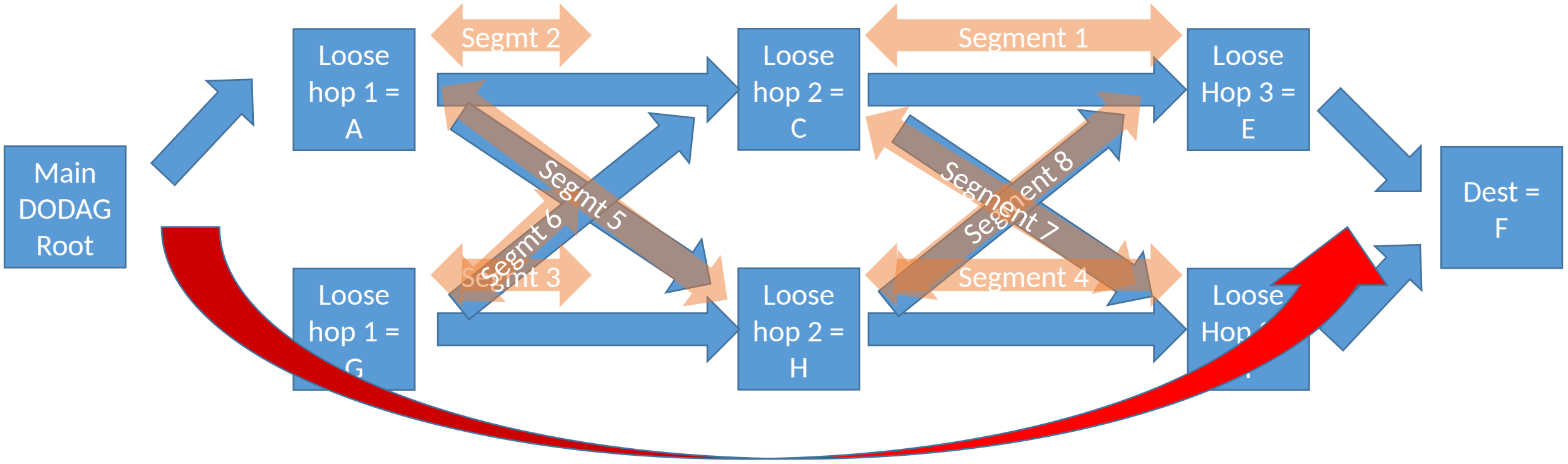
Discussion: building real DODAGs

- NO North-South Segment



Discussion: How to do this?

- Proposal: use RFC 6550 non-storing Target and Transit to indicate loose parent child relationship, many of them in one P-DAO



Proposal on the table: building real DODAGs

- Allow the root to pack 1*(1*target 1*transit) in P_DAO
- Provides the Ingress similar information to what a main Root has in non storing mode
- Ingress makes same computations as main Root and gets the instance topology as a DODAG.

Slides from previous meetings

Status to the draft (cont)

- RPI modified to indicate P-Route
- Extending RFC 6553 and RFC 8138

```
         0                   1                   2
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |1|0|E| Length  | 6LoRH Type 7  | RPLInstanceID |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- New P-RPI-6LoRH, both elective and non-elective forms

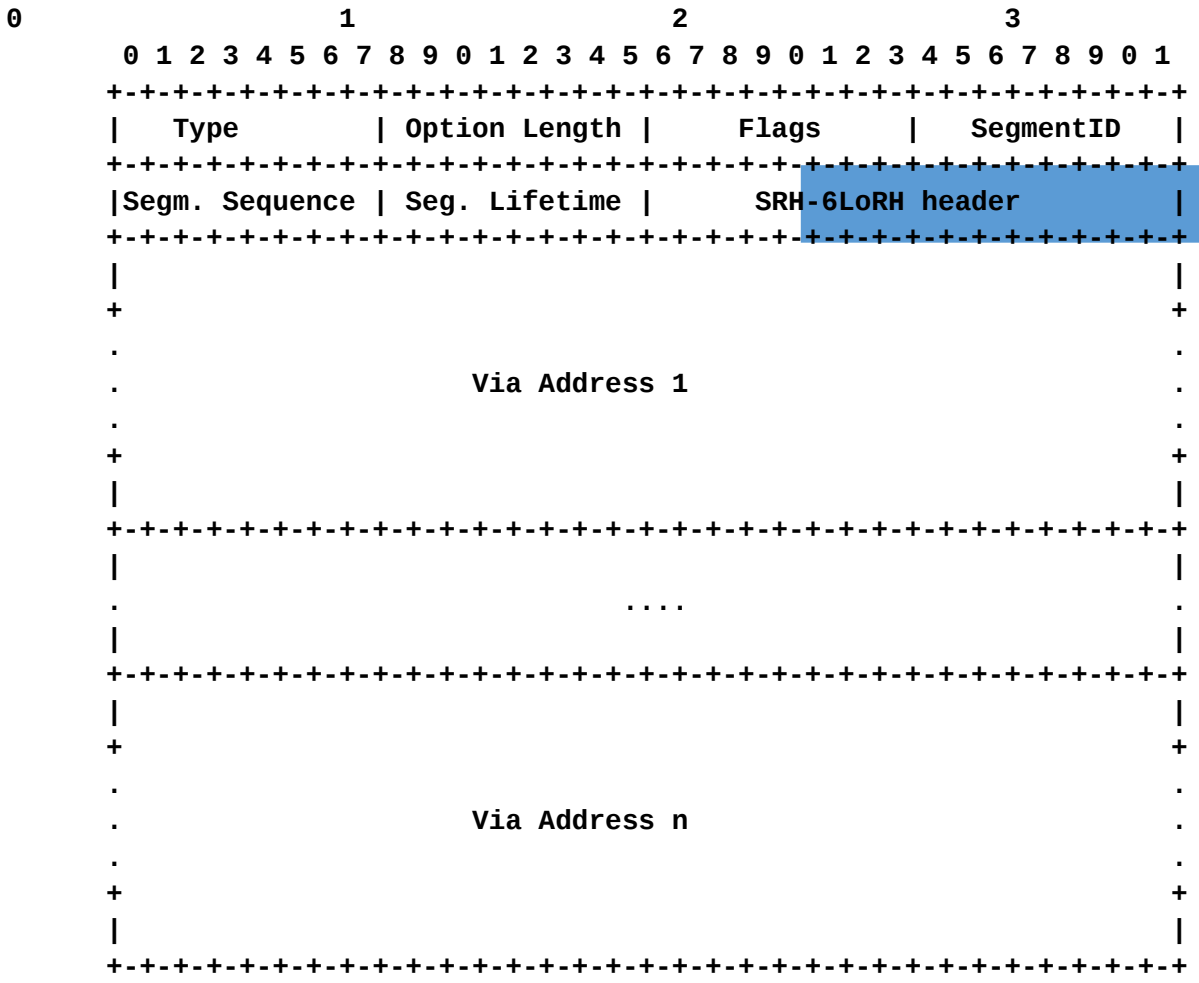
Encapsulation Rules

- Source of outer header MUST be Track Ingress- think DODAG Root
- RPL Instance ID in RPI MUST indicate TrackID (if not main DODAG)
- SR-VIO: Loose from Track Ingress, excluded, to Egress, included
  - Copied Verbatim in inserted SRH-6LoRH,
  - Requires encapsulation (can be recursive)
- SF-VIO: Strict from Segment Ingress to Egress, both included
  - No Encapsulation if Source and RPI both match Segment definition
  - A Segment is an Implicit Track if P-DAO Ingress == 1st SF-VIO entry
- TBD: matching rules, Flow Info option, when to tunnel?

P-DAO construction

- RPL Target Options can be factorized
- But there is one and only one VIO (SF-VIO or SR-VIO)
- So the Ack management is easier
- VIO sent to egress; SR-VIO sent to ingress
- Track ID is a RPL local instance ID
- Taken from the Track Egress Name Space

P-DAO Format



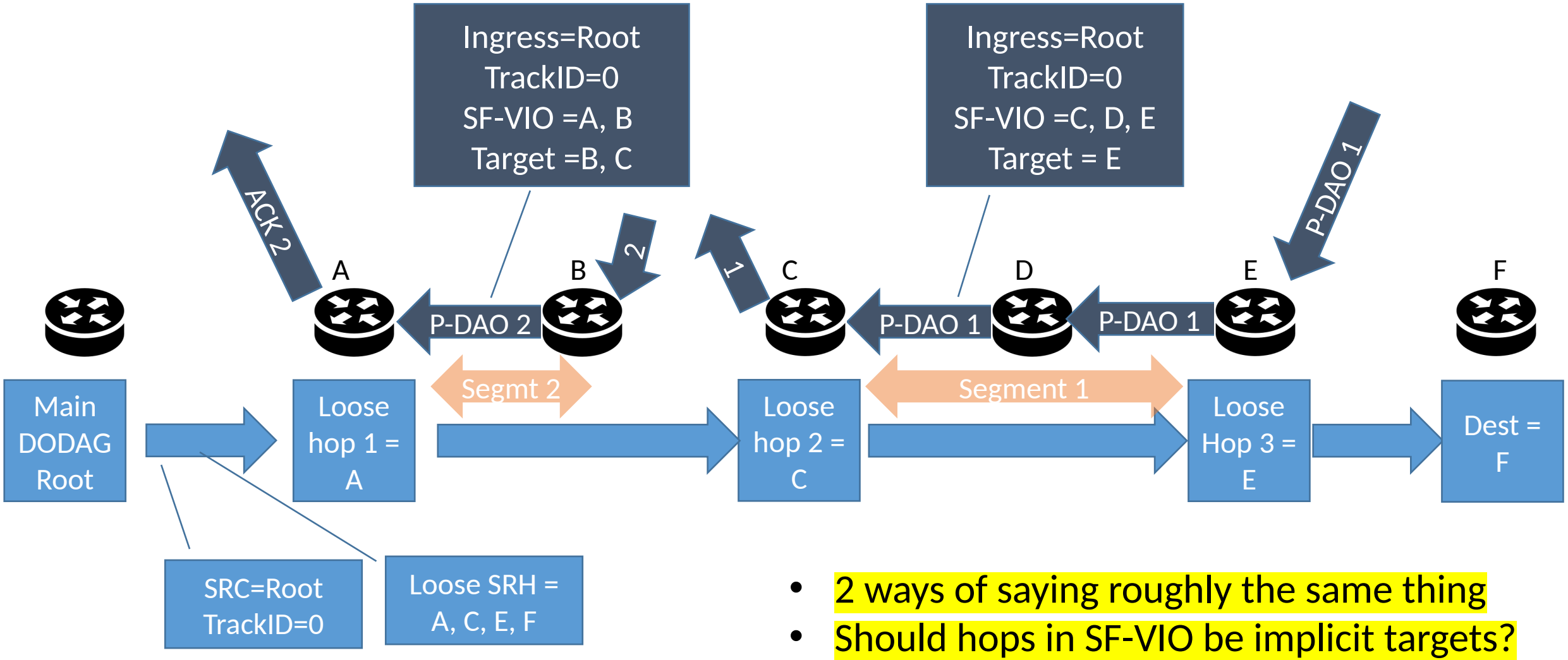
May be more than one in Non-storing Mode

Must be optimized in Non-storing Mode, to be used as is in packets

Topology awareness

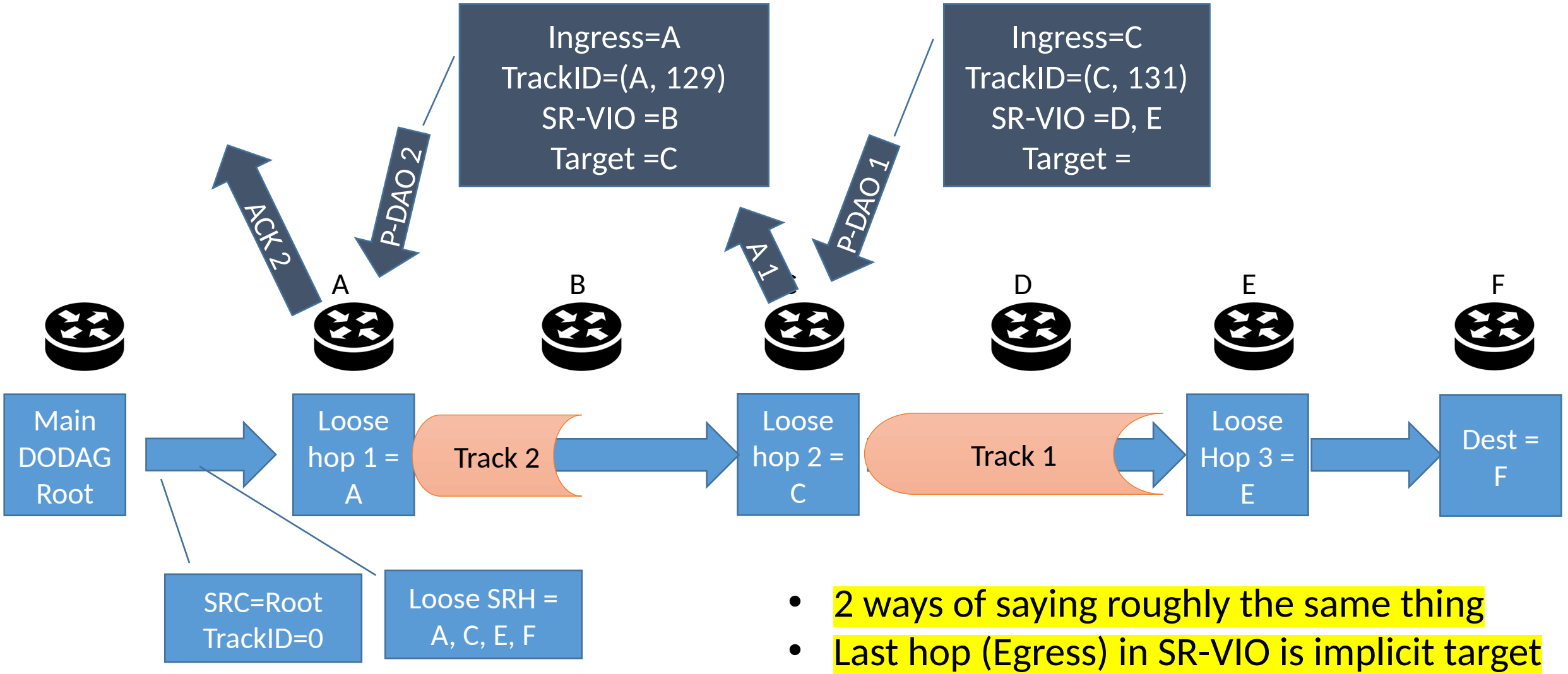
- Initially out of scope
- Now we have non storing mode + Sibling info option
  - Acronym conflict with RPL’s Solicited Information Option
- Needed for profiles >= 3
- Which sibling to advertise is still out of scope
  - Separate draft?

Profile 1:
Compress SRH in main DODAG with strict SM Segments



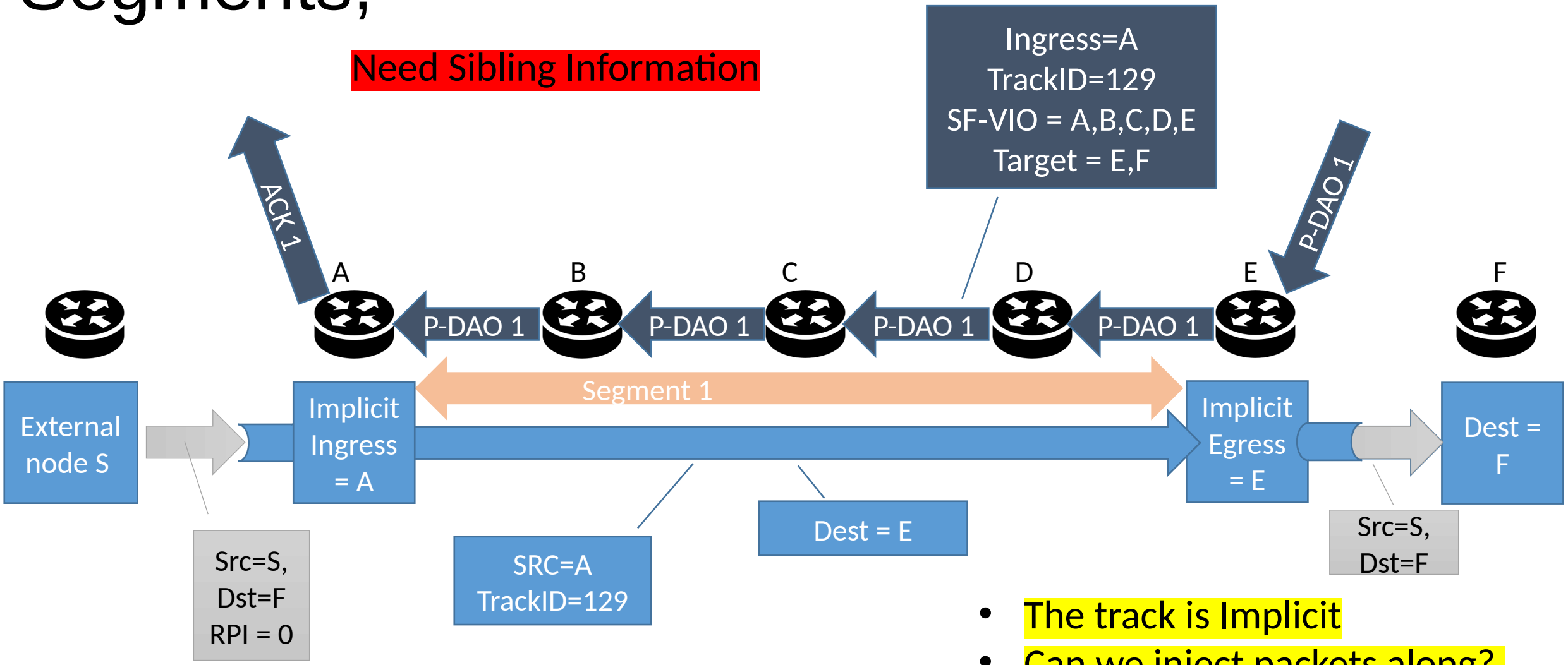
- 2 ways of saying roughly the same thing
- Should hops in SF-VIO be implicit targets?

Profile 2:
Compress SRH in main DODAG with Strict NSM Tracks



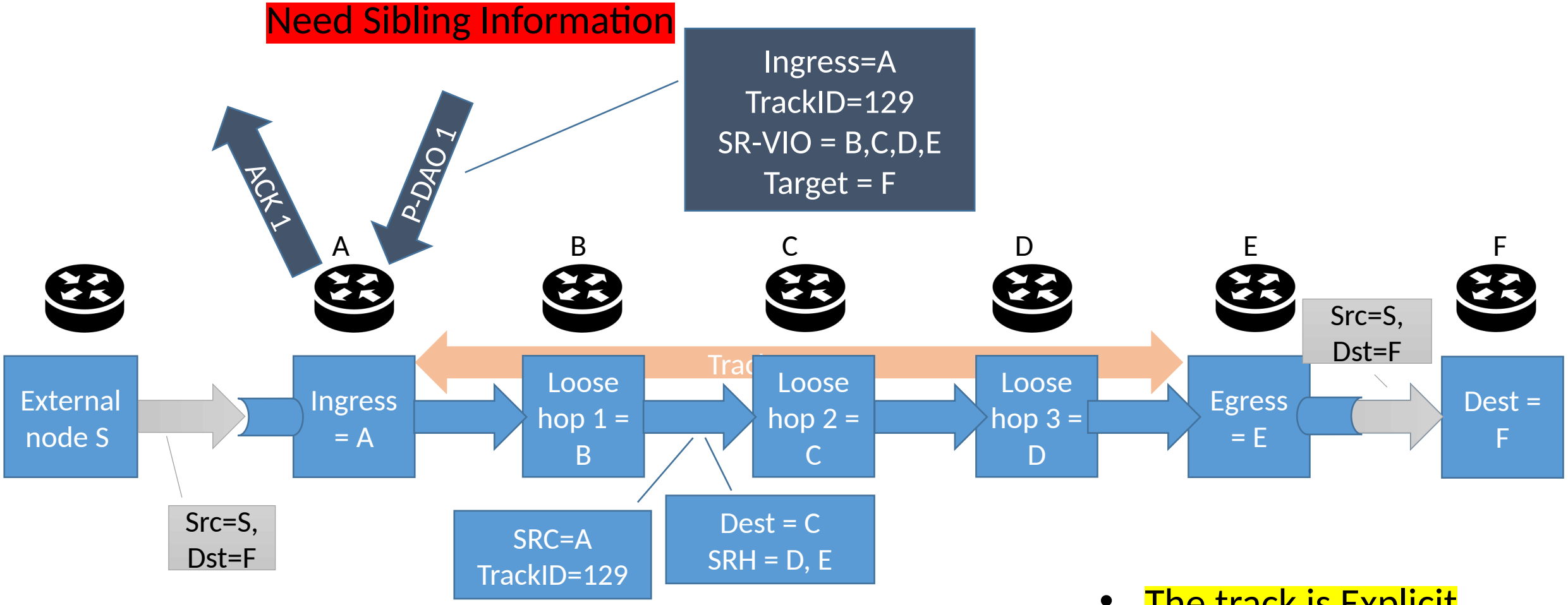
- 2 ways of saying roughly the same thing
- Last hop (Egress) in SR-VIO is implicit target

Profile 3: Implicit Track with Strict SM Segments,



- The track is Implicit
- Can we inject packets along?

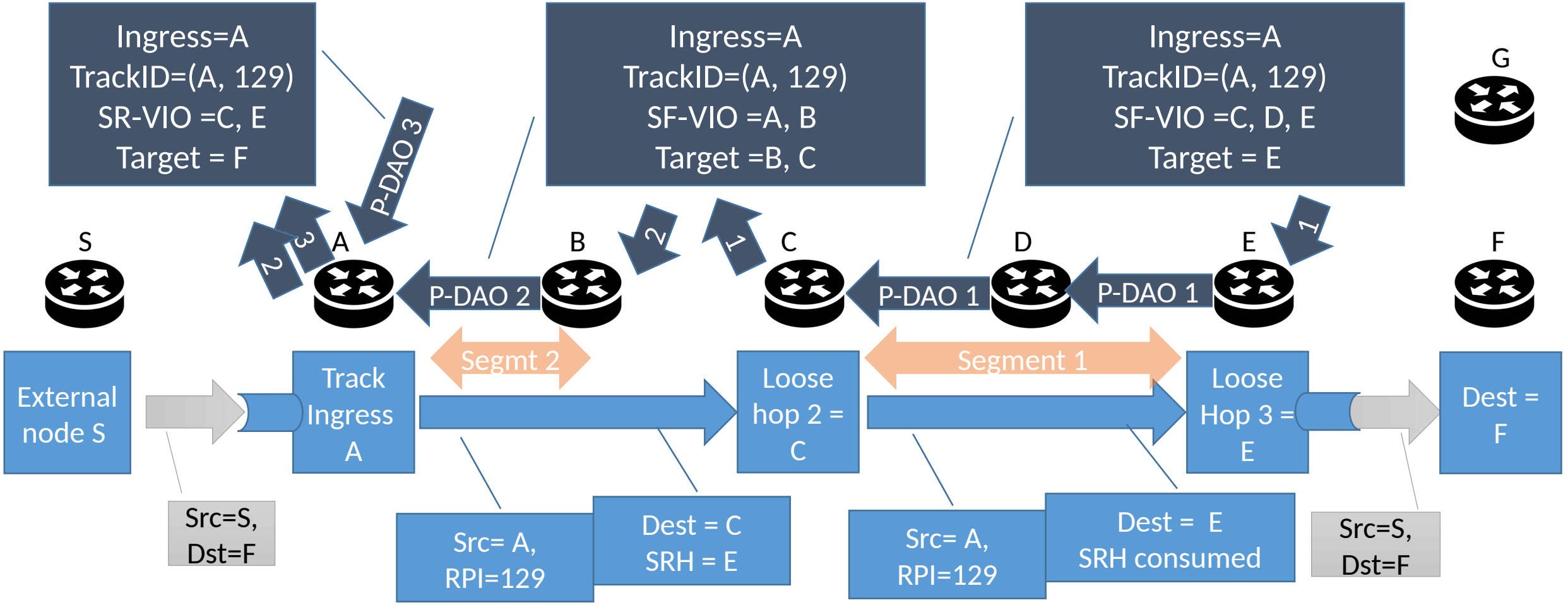
Profile 4: Strict NSM Explicit Track



- The track is Explicit
- Same encap as profile 2

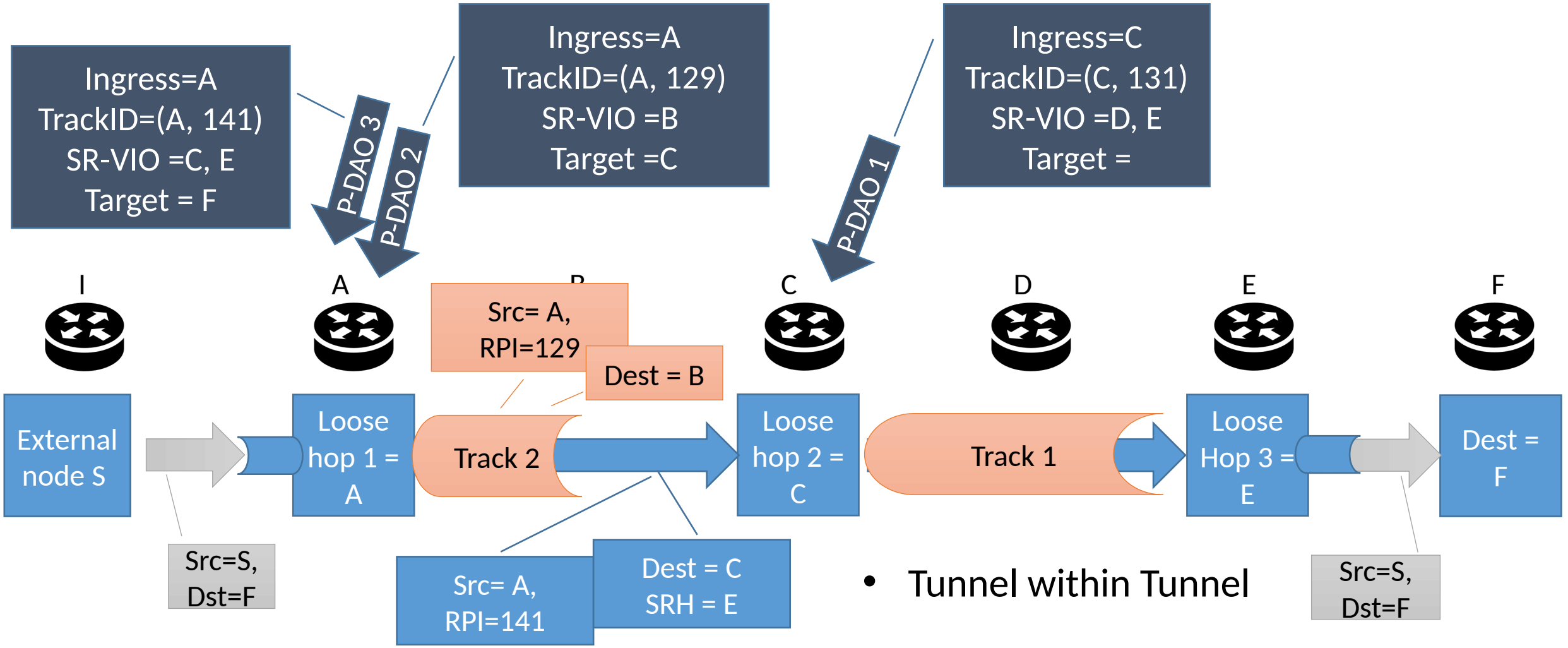
Profile 5: Compress SRH in Track with Strict SM Segments

Need Sibling Information



- Same as Profile 1, but for Track

Profile 6:
Compress SRH in Track with NSM Tracks (Recursive?)



- Tunnel within Tunnel

Huimin’s comments / suggestions

- Lifetime unit: ReqLifetime, Track lifetime, and Segment Lifetime are defined as 8 bits. And their lifetime Unit is obtained from the DODAG configuration option. It will lead to inflexibility as all tracks in the PAN use the same lifetime unit. We propose to define lifetime unit separately for each track (for example adding a 2-bit flag to indicate second, minute, hour, day). Details can be discussed later.
- Now the TrackID has the same meaning as Local RplInstanceID. How does a node judge whether the received message is a P-DAO message or Local RPL instance DAO message? Is it possible to define a flag in the P-DAO message?
- The P-DAO track/segment is single-directional. I suggest to add the possibility for creating bi-directional segments/tracks. We can add a flag in the PDR message to indicate the requested track is single-directional or bi-directional.
- I suggest to add a flow of message exchanges for “PDR, PDR-ACK, P-DAO, P-DAO ACK” in the draft.

Other to be done

- Loop avoidance
- Who sends PDR? If it was destination, then it could select the trackID from its name space
- ND (RFC 8505) to maintain sibling neighbor state
- Be very specific if Ingress and Egress are listed in VIOs
  - Ingress to indicate which source address to use
  - Egress to build the full SRH 6LoRH

BAckup