Root initiated routing state in RPL

draft-ietf-roll-dao-projection

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

Pascal Thubert

IETF 110

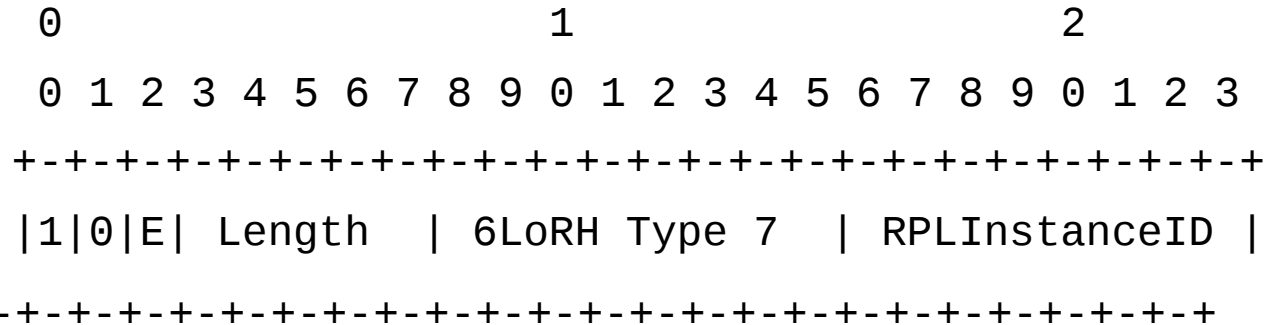ROLL Virtual Meeting

Status to the draft

See the VIA Info Option as a multihop Transit Info Option

- Published -15 and -16 since last IETF

- Non-Storing Mode SRH may be loose

- Main DODAG MUST be Non-Storing Mode

- Track <=> Non-Storing Mode main DODAG:
  - Root is Track Ingress,
  - Signaled by one or more Non-Storing-Mode P-DAO messages
  - 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 withing 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 ?

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

# 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

# Encapsulation and signaling

- Several Profiles to simplify implementations
- All with the same model based on
  - non-storing mode for the loose tracks : segment routing
  - Storing mode to signal the segments that fill the loose hops
- Tracks are local instances
- useofrplinfo applies for encapsulation of external targets

# Encapsulation Details

- 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?

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

Ingress=Root
TrackID=0
SF-VIO =A, B
Target =B, C

Ingress=Root
TrackID=0
SF-VIO =C, D, E
Target = E

ACK 2

P-DAO 1

2

1

A    B    C    D    E    F

P-DAO 2    P-DAO 1    P-DAO 1

Main DODAG Root

Segmt 2

Segment 1

Loose hop 1 = A

Loose hop 2 = C

Loose Hop 3 = E

Dest = F

SRC=Root
TrackID=0

Loose SRH = A, C, E, F

- 2 ways of saying roughly the same thing
- Should hops in SF-VIO be implicit targets?
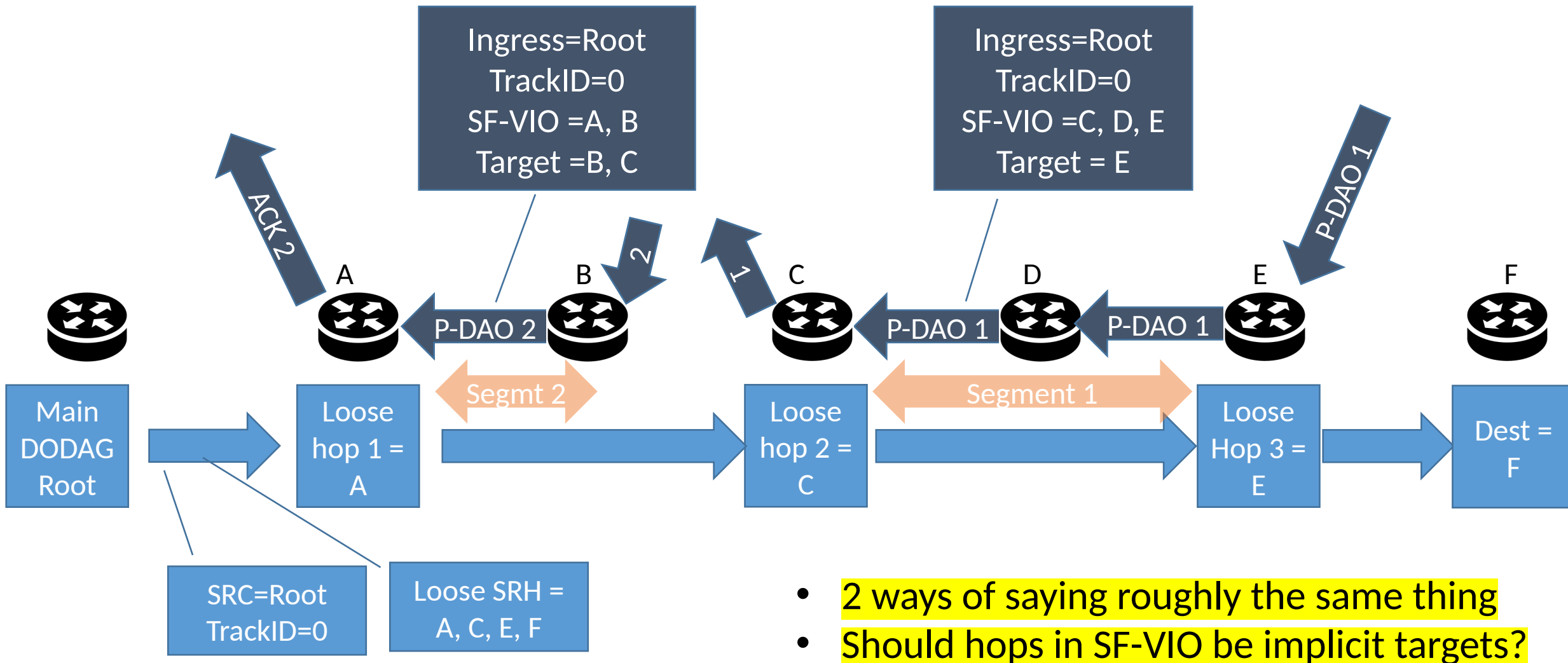
# Profile 2:
# Compress SRH in main DODAG with Strict NSM Tracks

Ingress=A
TrackID=(A, 129)
SR-VIO =B
Target =C

Ingress=C
TrackID=(C, 131)
SR-VIO =D, E
Target =

ACK 2

P-DAO 2

A 1

P-DAO 1

A          B          C          D          E          F

| Main DODAG Root | Loose hop 1 = A | Track 2 | Loose hop 2 = C | Track 1 | Loose Hop 3 = E | Dest = F |

SRC=Root
TrackID=0

Loose SRH = A, C, E, F
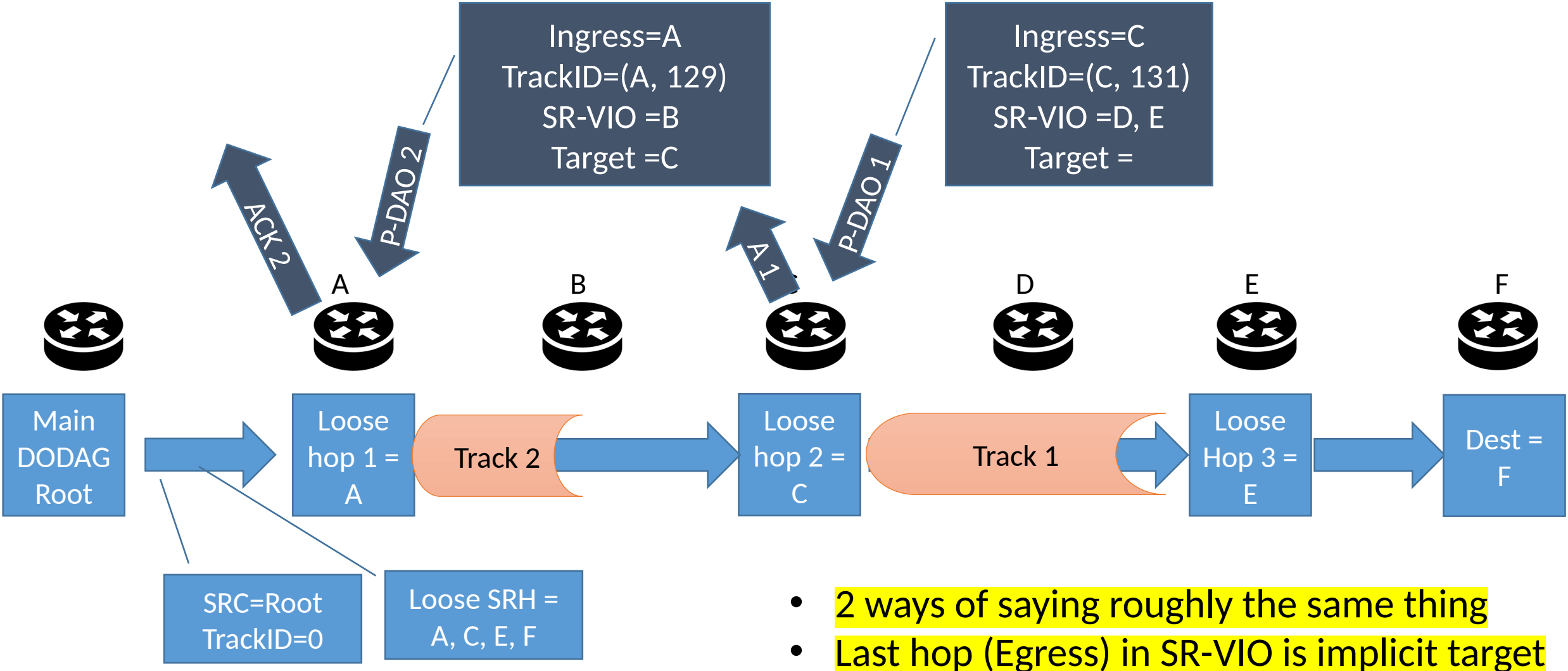
- **2 ways of saying roughly the same thing**
- **Last hop (Egress) in SR-VIO is implicit target**

# Profile 3: Implicit Track with Strict SM Segments,

Need Sibling Information

Ingress=A
TrackID=129
SF-VIO = A,B,C,D,E
Target = E,F

ACK 1

P-DAO 1

A    B    C    D    E    F

P-DAO 1   P-DAO 1   P-DAO 1   P-DAO 1

Segment 1

External node S

Implicit Ingress = A

Implicit Egress = E

Dest = F

Src=S, Dst=F RPI = 0

SRC=A TrackID=129

Dest = E

Src=S, Dst=F
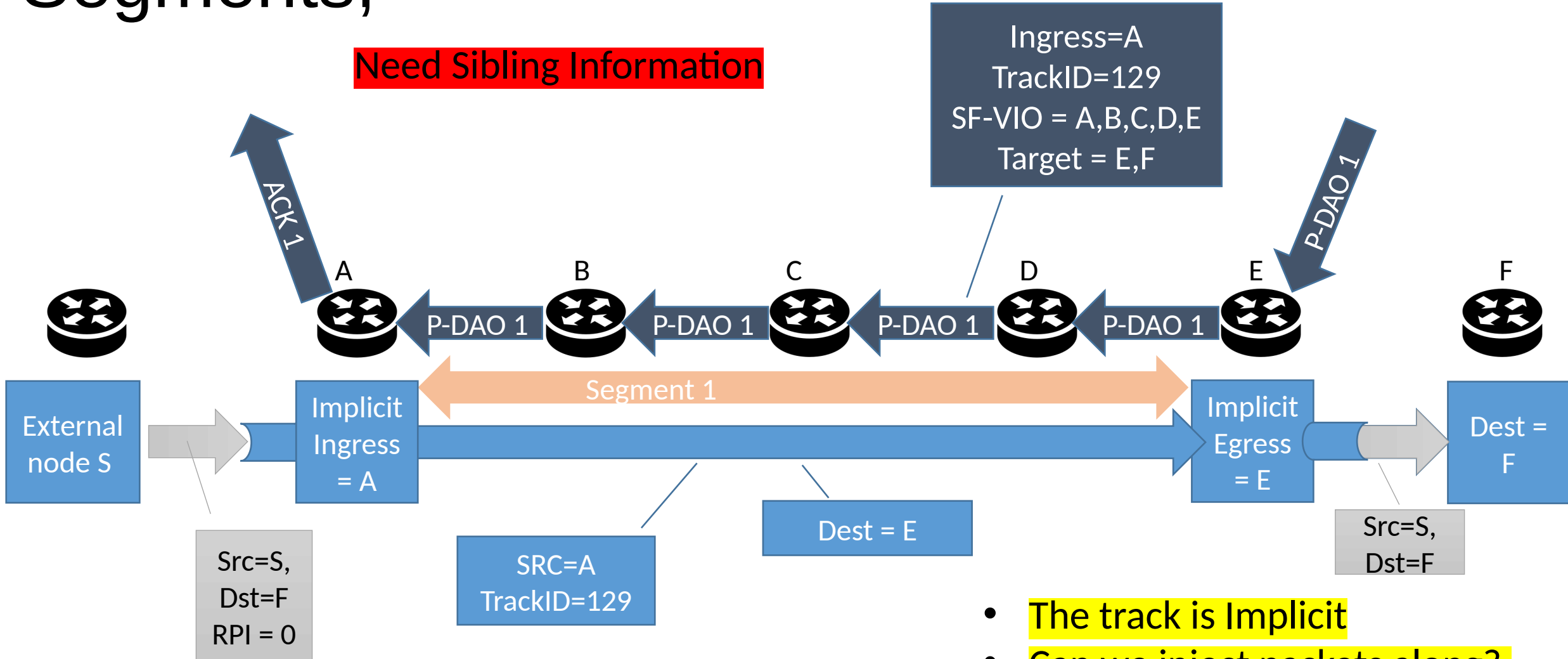
- The track is Implicit
- Can we inject packets along?

# Profile 4: Strict NSM Explicit Track

Need Sibling Information

Ingress=A
TrackID=129
SR-VIO = B,C,D,E
Target = F

ACK 1

P-DAO 1

A    B    C    D    E    F

External node S → Ingress = A → Loose hop 1 = B → Loose hop 2 = C → Loose hop 3 = D → Egress = E → Dest = F

Track

Src=S, Dst=F

Src=S, Dst=F

SRC=A
TrackID=129

Dest = C
SRH = D, E
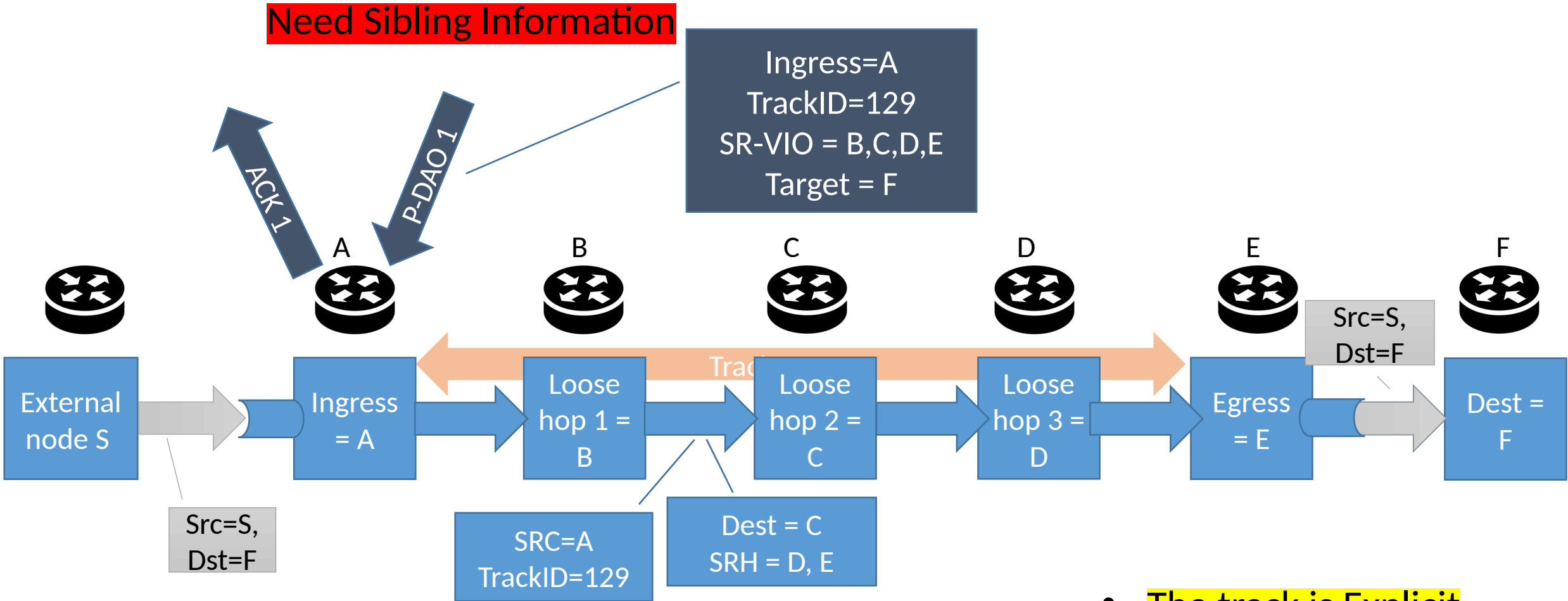
- The track is Explicit
- Same encap as profile 2

# Profile 5:
# Compress SRH in Track with Strict SM Segments

Ingress=A
TrackID=(A, 129)
SR-VIO =C, E
Target = F

P-DAO 3

Ingress=A
TrackID=(A, 129)
SF-VIO =A, B
Target =B, C

Ingress=A
TrackID=(A, 129)
SF-VIO =C, D, E
Target = E

G

3
2
2
1
1

S

A          B          C          D          E          F

P-DAO 2          P-DAO 1          P-DAO 1

Segmt 2          Segment 1

External node S

Track Ingress A

Loose hop 2 = C

Loose Hop 3 = E

Dest = F

Src=S, Dst=F

Src= A, RPI=129

Dest = C SRH = E

Src= A, RPI=129

Dest = E SRH consumed

Src=S, Dst=F
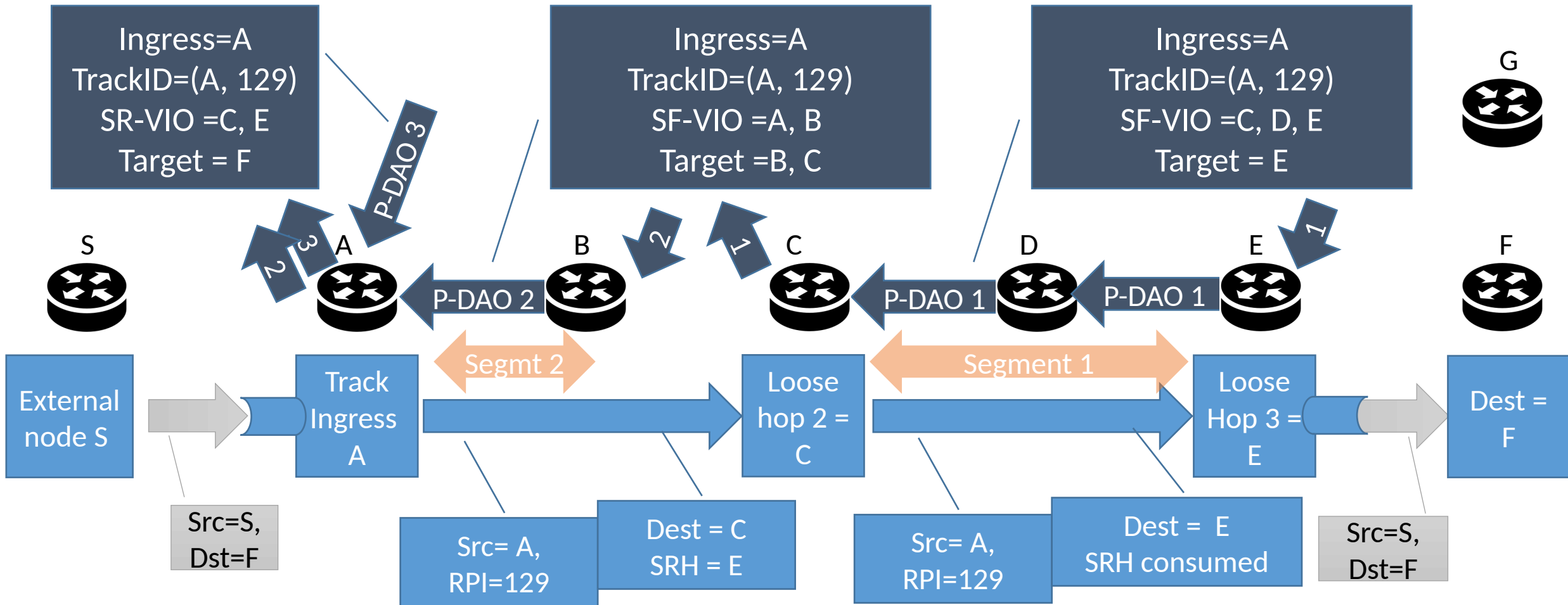
- Same as Profile 1, but for Track

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

Ingress=A
TrackID=(A, 141)
SR-VIO =C, E
Target = F

Ingress=A
TrackID=(A, 129)
SR-VIO =B
Target =C

Ingress=C
TrackID=(C, 131)
SR-VIO =D, E
Target =

P-DAO 3

P-DAO 2

P-DAO 1

I  A  B  C  D  E  F

External node S

Loose hop 1 = A

Src= A, RPI=129

Dest = B

Track 2

Loose hop 2 = C

Track 1

Loose Hop 3 = E

Dest = F

Src=S, Dst=F

Src= A, RPI=141

Dest = C
SRH = E

Src=S, Dst=F

- Tunnel within Tunnel
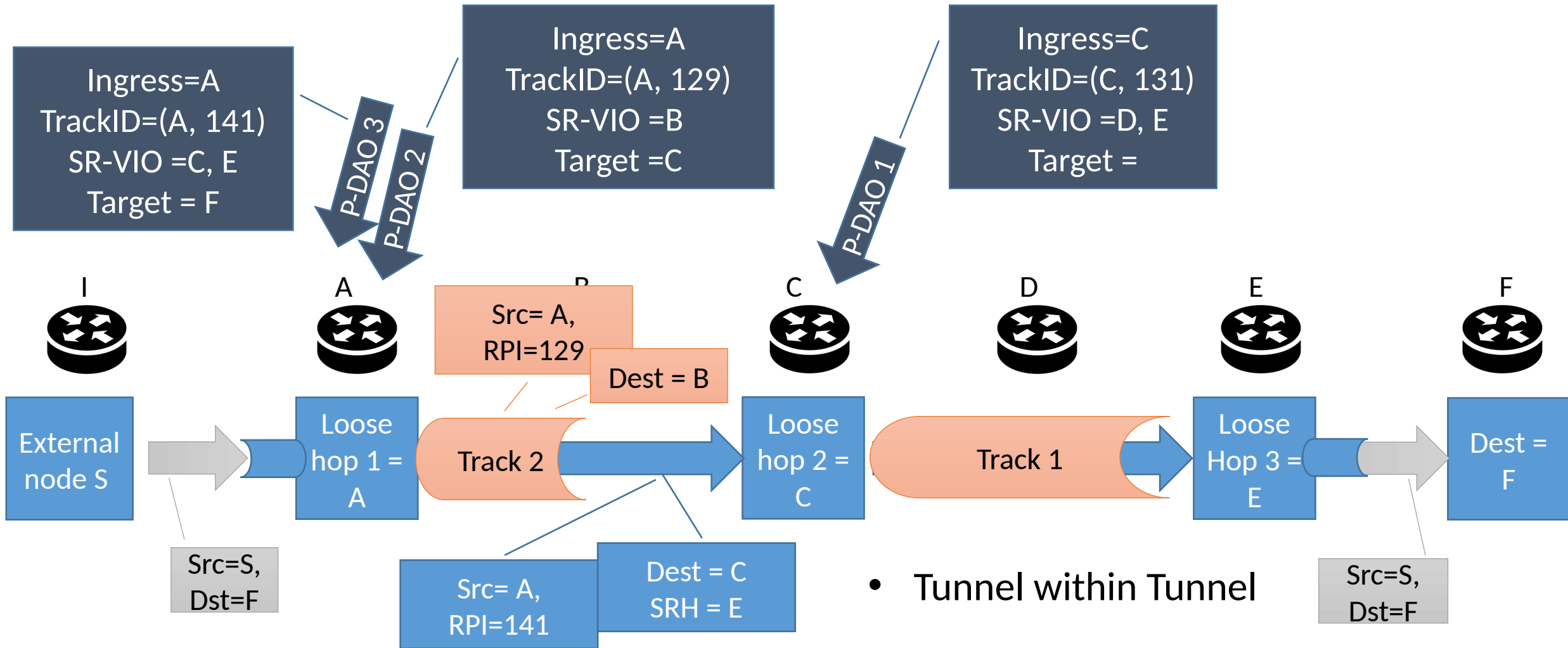
# Topology awareness

- 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?

# 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