

Root initiated routing state in RPL

draft-ietf-roll-dao-projection

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

Pascal Thubert

IETF 109

ROLL Virtual Meeting

Status to the draft

- Moved from 11 to 14 since last IETF
- Main DODAG must be non-storing Mode
 - To advertise the DODAG structure to the Root
 - Topology knowledge augmented with Sibling Info Option
 - VIA Option lists hops within one DODAG
- 1 P-DAO == 1 Segment == $n * \text{RTO (target)} + 1 \text{ RPO (Via)}$
- 1 Track == $p * \text{segments}$
- RFC 8138 compression of the address list in RPOs

VIA Info
Option as
a
multihop
Transit
Info

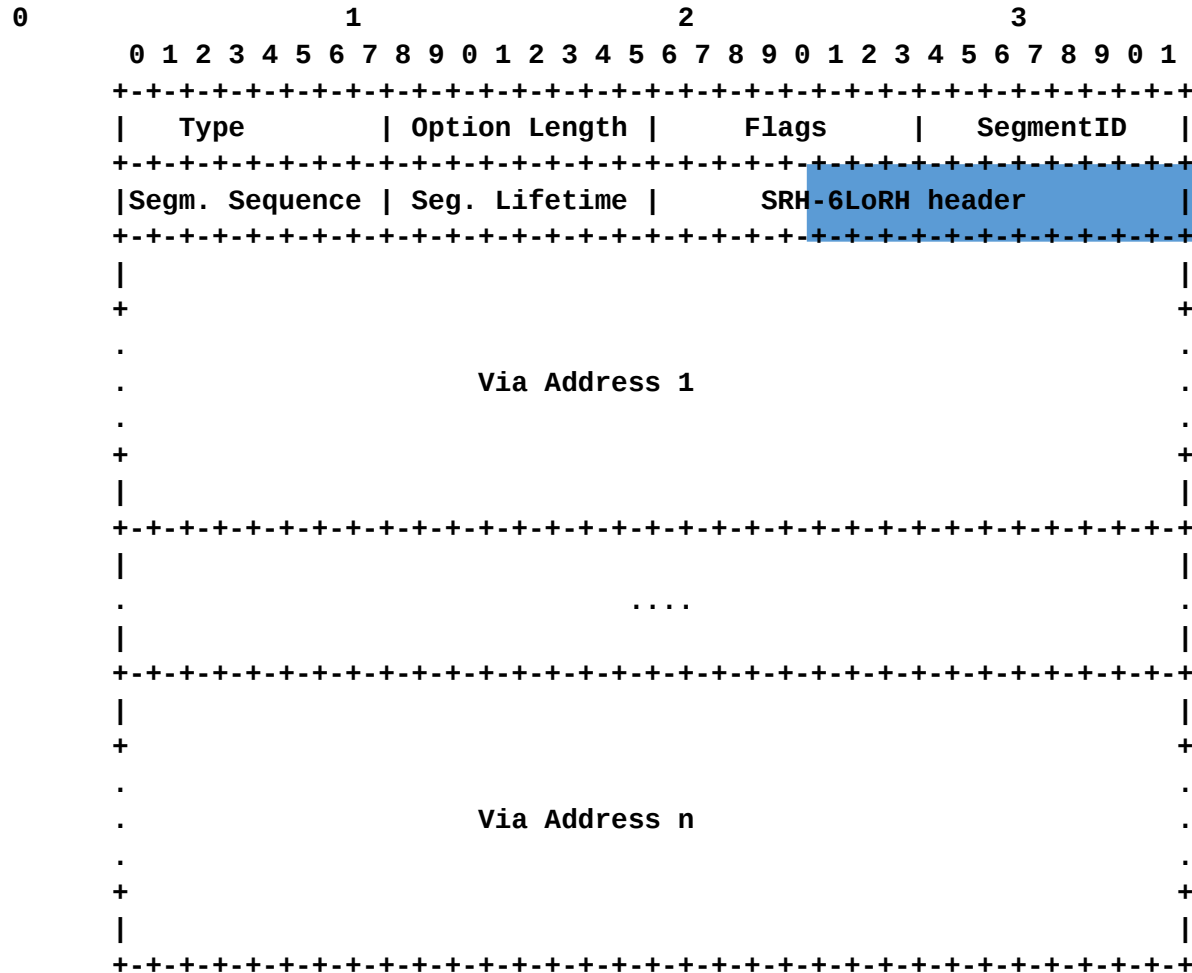
Topology awareness

- Initially out of scope
- Now we have non storing mode + Sibling info option
- Which sibling to advertise is still out of scope

P-DAO construction

- RPL Target Options can still be factorized
- But there is one and only one RPO (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 (Segment ID too?)
- Taken from the Track Egress Name Space

P-DAO Format



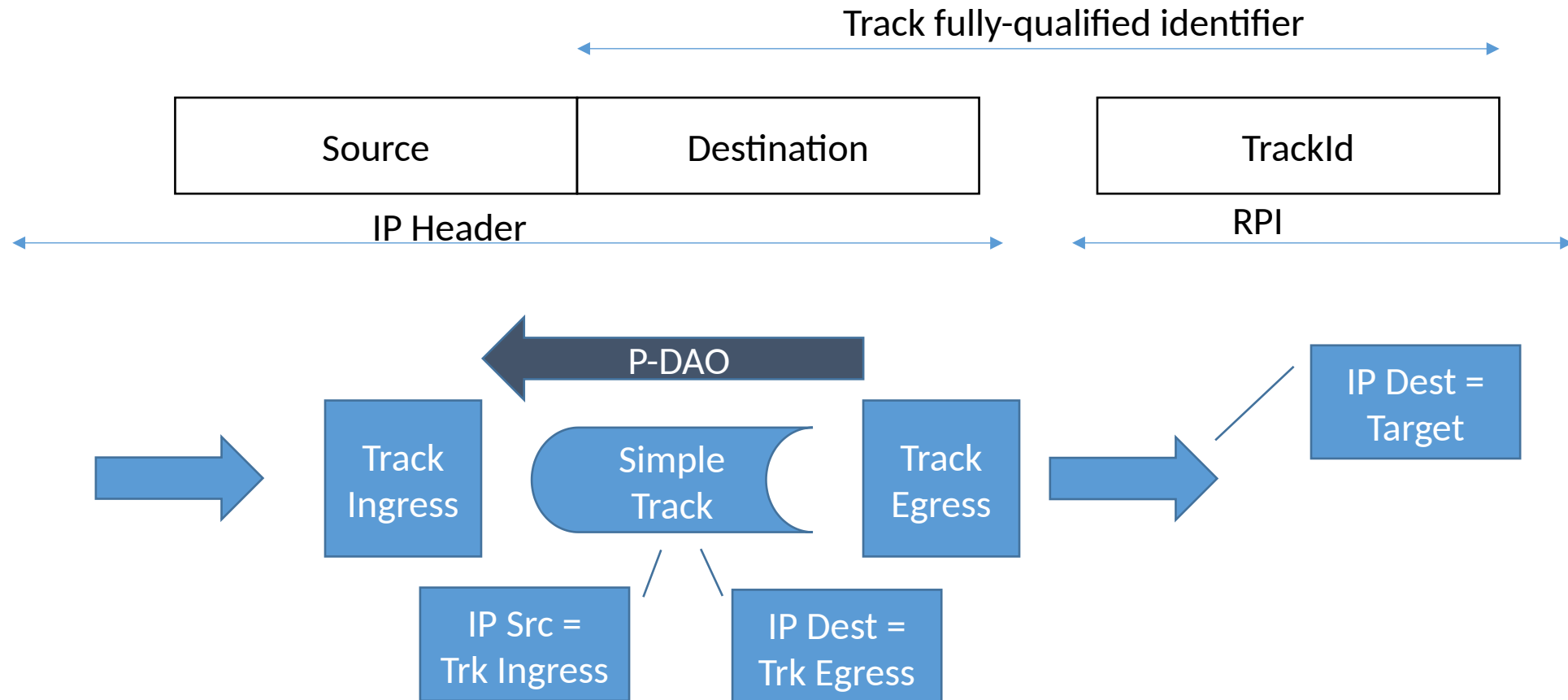
May be more than one in Non-storing Mode

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

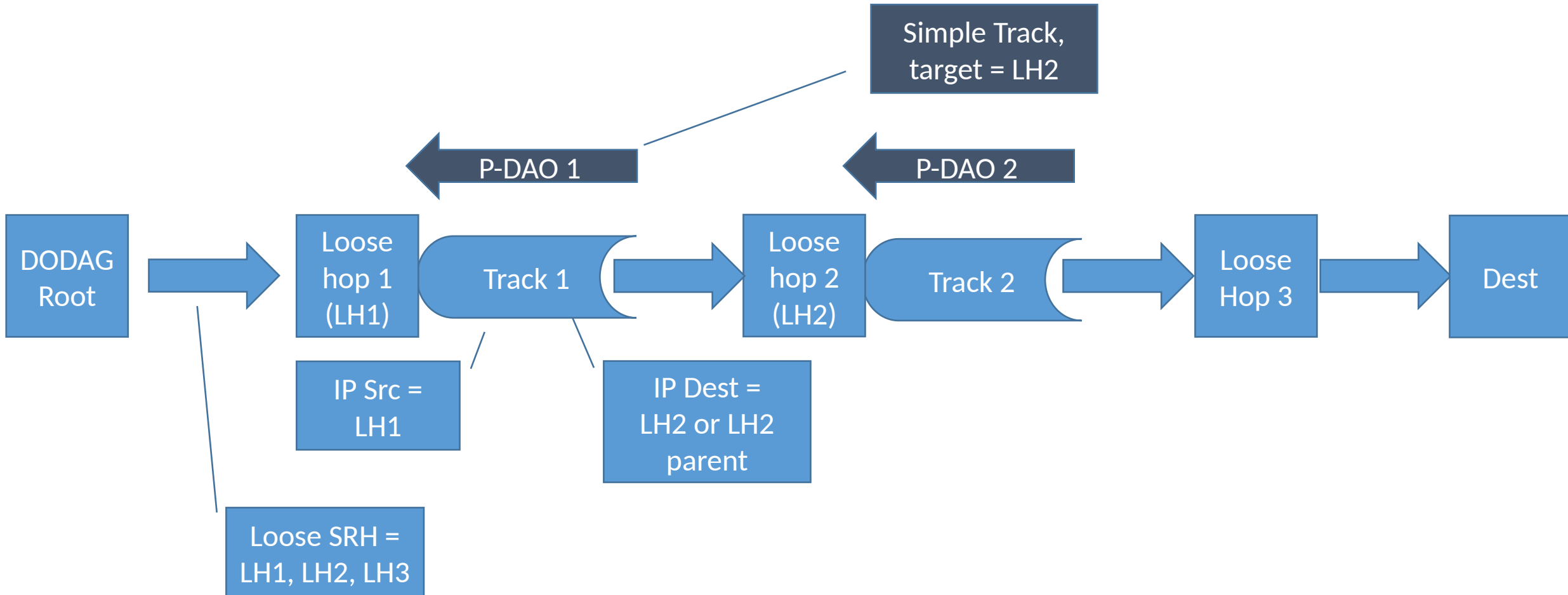
Encapsulation Rules

- Final destination of outer header MUST be Track Egress
- RPL Instance ID in RPI is TrackID
- Encapsulation needed if either
 - IP source != Track ingress or IP destination != Track egress
- Fine in Storing mode
 - but in non-storing how do we signal segments?
- As written RH is « inserted », 6LORH-SRH added in front
- Else we'll need to consider a segment as another encaps.

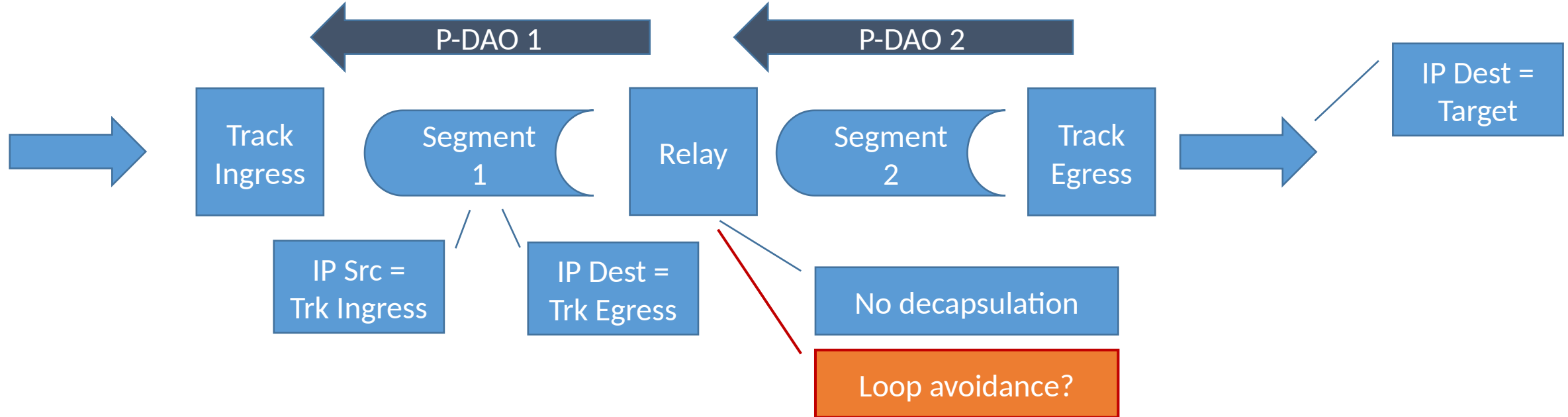
Encapsulation single segment, all MOPs



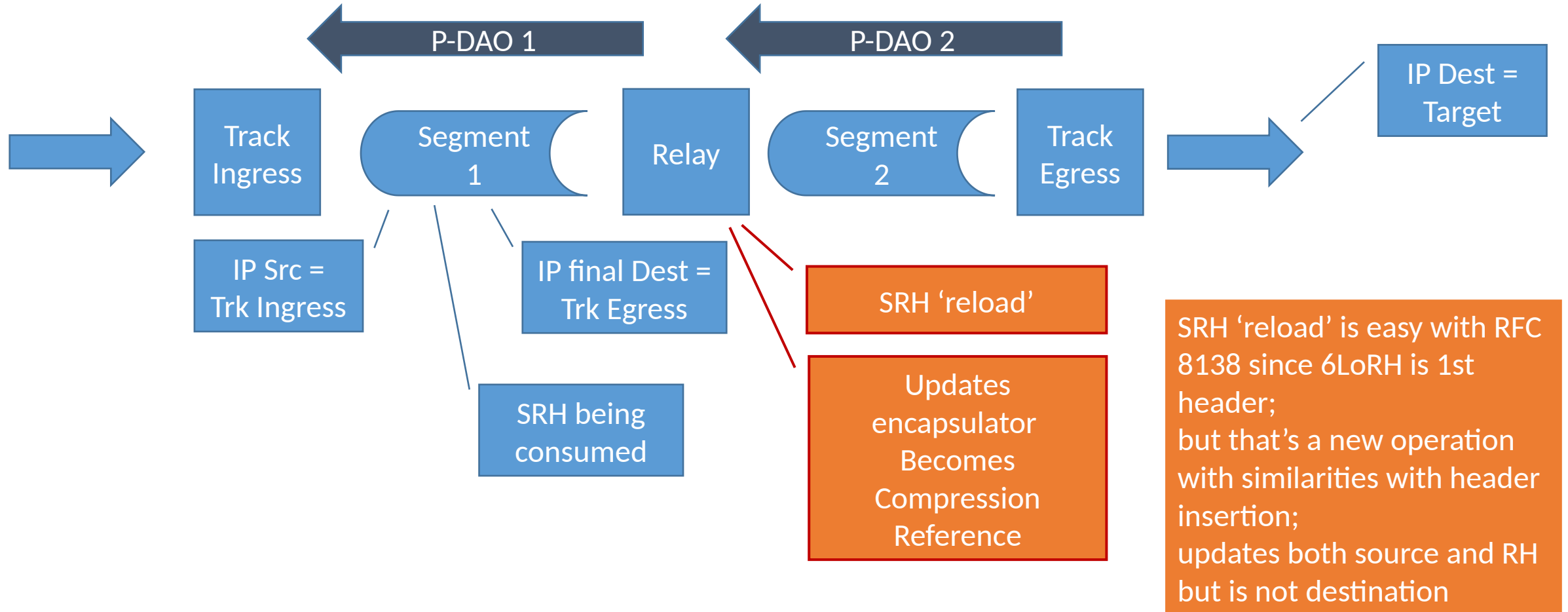
Loose non-storing mode



Encapsulation storing mode



Encapsulation non storing mode



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.

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