Root initiated routing state in RPL

draft-ietf-roll-dao-projection

Pascal Thubert, Rahul Arvind Jadhav, Michael Richardson

IETF 115, London

Presenter: Pascal Thubert
DAO Projection (Centralized RPL)

- Root connected-to or acting-as controller
  - Uses topological info from main DODAG
  - New Sibling Information Option (and P-DAO request)
  - Uses Projected DAO to install paths in the network

- Builds **Segments** to compress SHR
  - Compresses selected long paths in main DODAG
  - Uses Storing Mode Projected DAO to install strict (serial) paths

- Builds new DODAGs called **Tracks**
  - Enables optimized P2P (east – west) routing
  - Uses Non-Storing Mode Projected DAO to install loose (dotted-line) graphs
  - Leveraging Segments to complete the graph
The RPL Track: A DODAG rooted at Ingress

Non-Storing mode
P-DAO for L1
Targets = \{Ti\}

P-DAO Ack

P-DAO Ack

Storing mode
P-DAO for S1
Targets = \{E\}

Relay A

Fwd node F

Another Track

Fwd node G

Relay B

Egress E

Target Tn

Ingress I

Root

Fwd node H

East

Packet flow

West

Targets
\{Tx\}

L1 = I->A->E to \{Ti\},  L2 = I->B->E to \{Ti\},  L3 = I->A->B->E to \{Ti\}

Legs

Segments
S1 = A=>F=>G to E,  S2 = I=>H to B

SubTracks

Any Set \subset\{L1, L2, L3\} but \{\}

draft-ietf-roll-dao-projection
Draft Status

• WGLC at -26
• All known issues addressed at current ( -29 )
• Ready for publication
-28 (as promised at IETF 114) refine on WGLC issues:

- Clarify that each instance implies a RIB
- Multi-topology routing loop avoidance rules:
  - neighbor> indirect via common neighbor > Segment > Track
  - Partial order between RPL instances to allow jumping
- Crossing Segments discussion
- Clarifying mcast DAO exposes neighbors in SIOs
Status of the draft (cont.)

-> 29 Clean up:
  - Remove duplicated text in intro
  - Lower case “main” in “main DODAG”
Next

• Publication request?
DAO Projection

Backup Slides
DAO Projection (Centralized RPL)

- Root connected-to or acting-as controller
  - Uses topological info from main DODAG
  - New Sibling Information Option (and P-DAO request)
  - Uses Projected DAO to install paths in the network

- Builds **Segments** to compress SHR
  - Compresses selected long paths in main DODAG
  - Uses Storing Mode Projected DAO to install strict (serial) paths

- Builds new DODAGs called **Tracks**
  - Enables optimized P2P (east – west) routing
  - Uses Non-Storing Mode Projected DAO to install loose (dotted-line) graphs
  - Leveraging Segments to complete the graph
DAG Root

Controller

Src: X
Dest: 56
Stuff

Src: Root
Dest: 56
RPI
Via: 13
Via: 24
Via: 35
Via: 46
Src: X
Dest: 56
Stuff

Stuff

Via: 13
Via: 24
Via: 35
Via: 46
Src: Root
Dest: 56
RPI

draft-ietf-roll-dao-projection
Projected-DAO to target 56 with path segment via 24 (ingress), 35, and then 46 (egress)
Storing mode DAO to 56 upwards segment (24, 35, 46)
Storing mode DAO to 56 upwards segment (24, 35, 46)
DAO from 46 installs a route via 35 to 56 in 24

DAO from 46 installs a route via 46 to 56 in 35

Preexisting connected route to 56
Controller

DAG Root

Non source routed DATA Path
Loose Source routed DATA Path
Packet to 13, RH 24, 56

Non source routed DATA Path
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
New Via Information Option Format

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type | Option Length | Flags | P-RouteID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Segm. Sequence | Seg. Lifetime | Segm. Sequence | Seg. Lifetime |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | Via Address 1 (compressed by RFC 8138) | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | . | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | Via Address n (compressed by RFC 8138) | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | . | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | Additional SRH-6LoRH Header(s) | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| . | . | . | . |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SRH-6LoRH head |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|May be more than one in Non-storing Mode |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SRH-6LoRH head |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Must be optimized in Non-storing Mode, to be used as is in packets |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SRH-6LoRH head |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SRH-6LoRH head |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
New Sibling Information Option Format

In DAO and mcast DAO; mcast DAO allows indirect forwarding
The RPL Track: A local DODAG rooted at Ingress

- **Segments**: S1 = A=>F=>G to E, S2 = I=>H to B
- **Targets**: S1 = E, S2 = {Ti}
- **SubTracks**: Any Set \( \subset \) {L1, L2, L3} but \{ \}
- **Root**: Ingress I
- **P-DAO Ack**: P-DAO for L1, Targets = \{Ti\}
- **Another Track**: Fwd node F
- **Fwd node H**: L1 = I->A->E to \{Ti\}
- **L2 = I->B->E to \{Ti\}
- **L3 = I->A->B->E to \{Ti\}
- **Non-Storing mode**: P-DAO for L1, Targets = \{Ti\}
- **Storing mode**: P-DAO for S1, Targets = \{E\}
- **East** → **West**: Packet flow
- **Targets**: S1 = E, S2 = {Ti}
- **Legs**: L1 = I->A->E to \{Ti\}, L2 = I->B->E to \{Ti\}, L3 = I->A->B->E to \{Ti\}
- **Storing mode**: P-DAO for S1, Targets = \{E\}
- **Non-Storing mode**: P-DAO for L1, Targets = \{Ti\}
- **Fwd node G**: Egress E
- **Fwd node F**: Relay A
- **Relay B**: Fwd node H
- **Target Tn**: East

[Diagram showing packet flow and track details]
Some rules

• Track is set up by installing Legs and Segment
  • with the same Track ID
• Non-Storing Mode P-DAO signals a Leg
• Storing Mode P-DAO signals a Segment
• Storing Mode P-DAO enables loose hops
  • in Non-Storing main DODAG (typically TrackId is Global instance ID)
  • in Tracks (typically TrackId is Local instance ID to track Ingress)
• Track Egress is implicit Target in Non-Storing Mode
• Leg hop is either a Segment of this Track or another Track
Complex track

- A complex track is multi-legged, e.g., 2 Legs below
- Allows 1+n
RPL vs RAW

- RPL has no North-South Segment
Inter Leg

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

- **Loose hop 1 = A**
  - SRC=Root TrackID=0
  - Loose SRH = A, C, E, F

- **Loose hop 2 = C**
  - Segmt 2

- **Loose hop 3 = E**
  - Dest = F

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

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

- **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 =

- 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?

Need Sibling Information
Profile 4: Strict NSM Explicit Track

Need Sibling Information

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

• 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

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

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

Need Sibling Information

• Same as Profile 1, but for Track
Profile 6: Compress SRH in Track with NSM Tracks (Recursive?)

External node S

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 =

Loose hop 1 = A
Dest = B

Dest = C

Track 2

Track 1

Loose hop 2 = C

Loose Hop 3 = E

Dest = F

Tunnel within Tunnel

Src=S, Dst=F

Src=A, RPI=129
Dest = C

Src=A, RPI=141
SRH = E

Src=S, Dst=F

Src=S, Dst=F