

P2MP Policy

draft-voyer-pim-sr-p2mp-policy

Authors:

Hooman Bidgoli, Nokia

Daniel Voyer, Bell Canada

Rishabh Parekh, Cisco

Jeffrey Zhang, Juniper

Presenter Hooman Bidgoli



I E T F[®]

Multicast Evolution

- There is a desire to simplify Next generation complex networks (i.e. 5G transport) from administration and protocol point of view.
- The controller provides an end to end view of the network and simplifies traffic engineering, slicing and monitoring of the end to end SLAs for each slice
- Protocols like SR simplify the underlay by removing the need of LDP/RSVP-TE protocols and use IGP/BGP to signal segments.
- **Multicast needs to follow suite**
- SR P2MP Policy removes legacy P2MP MPLS protocols like mLDP/RSVP-TE while providing traffic engineering via SR Policy attributes

Brief History

- Work started in Jun 2018 with draft-voyer-sr-p2mp-policy-00
- Presented in Prague IETF
- Last revision was draft-voyer-sr-p2mp-policy-03
- Addressed comments:
 - Split draft in two docs: Stateless Replication segment in SPRING
 - PCE based Tree-building in PIM WG

Relevant Drafts

[draft-voyer-pim-sr-p2mp-policy-00](#)

[draft-voyer-spring-sr-replication-segment-00](#)

[draft-hb-spring-sr-p2mp-policy-yang-01](#)

[draft-hsd-pce-sr-p2mp-policy-01](#)

[draft-hb-idr-sr-p2mp-policy-00](#)

SR P2MP Segment

- A Point-to-Multipoint (P2MP) segment connects a Root node to a set of Leaf nodes in segment routing domain.
- A P2MP segment contains Replication Segments, each providing forwarding instructions at Root, Transit Nodes and Leaf Nodes.
- It is identified via <ROOT, Tree-ID>
- PCC Initiated: Root and Leaves can be discovered via multicast procedures like NG-MVPN (RFC 6514, 6513) or PIM (Protocol Independent Multicast) on PCC and the relevant information send to the PCE
- PCE Initiated: Root and Leaves can be configure explicitly on the PCE or controller and programmed on the PCC

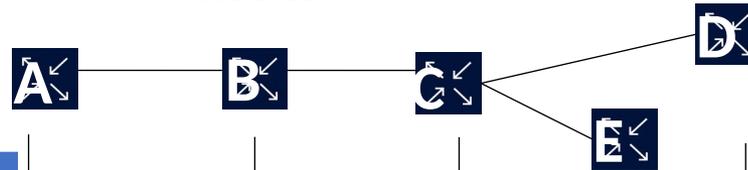
Replication Segment

- Is the forwarding instructions for the P2MP LSP
 - Label instructions
 - Next-Hop information
 - Fast Reroute instructions
- A Replication segment is defined via following
 - Root: The root of the P2MP segment that the replication segment is for;
 - Tree-ID: Tree that the replication segment is part of;
 - Instance-ID: Unique path-instance ID per <Root, Tree-ID>, it identifies a P2MP LSP
 - **Replication-SID: Segment ID for this Replication Segment.**
 - **Replicaiton-SIDs can't be stacked as each replication segment can be a egress or transit.**
- Two Replication Segments can be connected directly via adjacent nodes or they can be non-adjacent and connected via a SID List (Unicast)



SR P2MP Objects

Non-SR-P2MP nodes



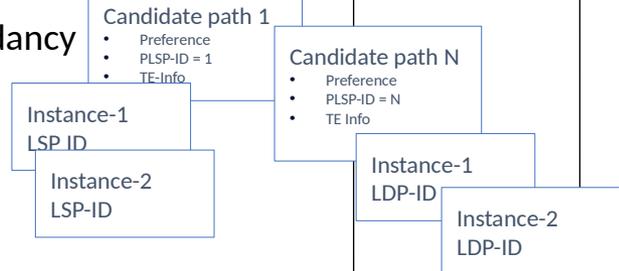
SID, Forwarding instruction for this segment

SR P2MP Policy

- ROOT Node
- Leaf Node
- Constrains
- Tree-ID

SR P2MP Policy

P2MP LSP Redundancy



End to End Optimization

Replication Policy

- Node-ID
- Tree-ID
- Root
- Instance ID
- Inc Rep SID
- Rep SID Action

Unicast SR Policy

Replication Policy

- Node-ID
- Tree-ID
- Root
- Instance ID
- Inc Rep SID
- Rep SID Action

Replication Policy

- Node-ID
- Tree-ID
- Root
- Instance ID
- Inc Rep SID
- Rep SID Action

Forwarding info
Sid-List
Fast Reroute

Forwarding Info

- Next-hop-group-id [nh-id] //array of nh
 - Next-hop-id <id>
 - Next-hop-add
 - Next-hop-int
 - Protect-nh <id>
 - Sid-list [list of outgoing labels]

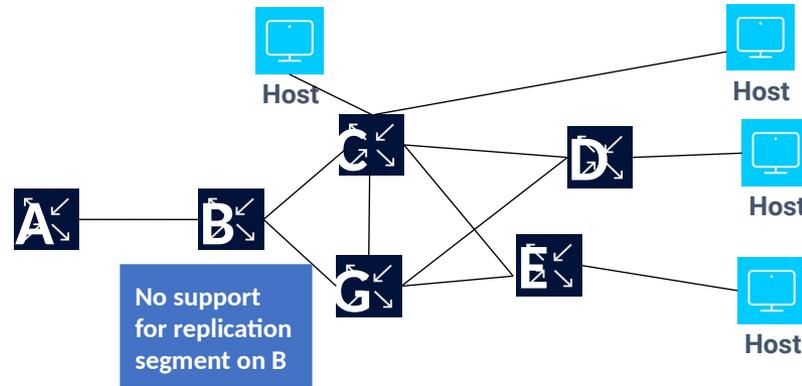
Forwarding Info

- Next-hop-group-id [nh-id] //array of nh
 - Next-hop-id <id>
 - Next-hop-add
 - Next-hop-int
 - Protect-nh <id>
 - Sid-list [list of outgoing labels]

Forwarding Info

- Next-hop-group-id [nh-id] //array of nh
 - Next-hop-id <id>
 - Next-hop-add
 - Next-hop-int
 - Protect-nh <id>
 - Sid-list [list of outgoing labels]

Example single candidate path



1. The primary path (candidate path 1) is A to C to LEAF D and LEAF E with C being a BUD node
2. B does not support Replication Segment

SR P2MP Policy

- ROOT Node=A
- Leaf Node=D,E
- Tree-ID=1

Candidate path 1

- Preference = 1000

Instance-1
LSP ID = 1

Replication Policy A

- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = B
 - Sid-list B,C <C is bottom of Stack>

Replication Policy C

- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID = C

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D
 - Sid-list <D>
- Next-hop-group-id 1
 - Next-hop-add = E
 - Sid-list <E>

Replication Policy D

- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID = D

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

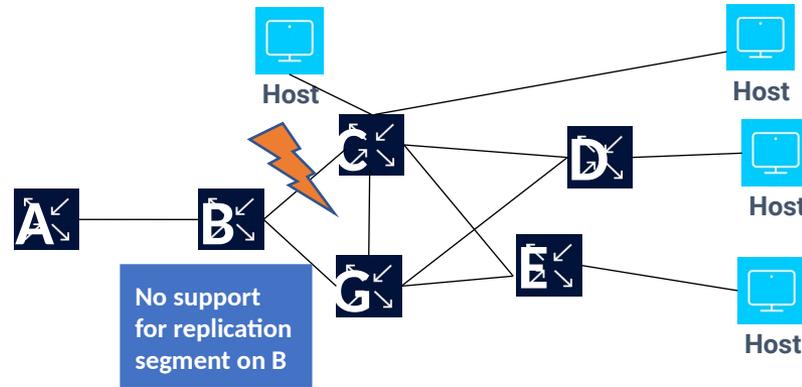
Replication Policy E

- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID = E

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

Example Global Optimization of Candidate path 1



1. The link between B and C is broken
2. Node B can do a FRR through B-G-C
3. The controller can do global optimize the candidate path 1 via G

SR P2MP Policy

- ROOT Node=A
- Leaf Node=D,E
- Tree-ID=1

Candidate path 1

- Preference = 1000

Instance-1
LSP ID = 1

Instance-1
LSP ID = 2

Replication Policy A

- Tree-ID =1
- Root = A
- Instance ID = 2
- Inc Rep SID

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = B
 - Sid-list B,G <C is bottom of Stack>

Replication Policy G

- Tree-ID =1
- Root = A
- Instance ID = 2
- Inc Rep SID = G

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D
 - Sid-list <D>
- Next-hop-group-id 1
 - Next-hop-add = E
 - Sid-list <E>

Replication Policy D

- Tree-ID =1
- Root = A
- Instance ID = 2
- Inc Rep SID = D

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

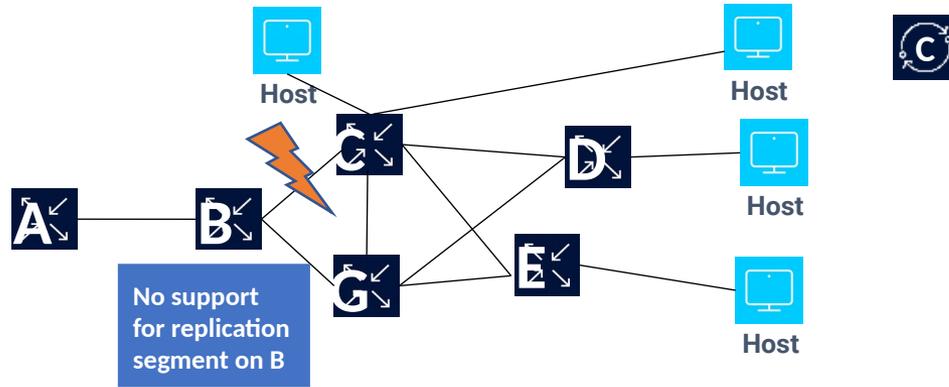
Replication Policy E

- Tree-ID =1
- Root = A
- Instance ID = 2
- Inc Rep SID = E

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

Example P2MP Tree Redundancy



SR P2MP Policy

- ROOT Node=A
- Leaf Node=D,E
- Tree-ID=1

Candidate path 1

- Preference = 1000

Instance-1
LSP ID = 1

Replication Policy A

- Tree-ID = 1
- Root = A
- Instance ID = 1
- Inc Rep SID

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = B
 - Sid-list B,C <C is bottom of Stack>

Replication Policy C

- Tree-ID = 1
- Root = A
- Instance ID = 1
- Inc Rep SID = C

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D
 - Sid-list <D>
- Next-hop-group-id 1
 - Next-hop-add = E
 - Sid-list <E>

Replication Policy D

- Tree-ID = 1
- Root = A
- Instance ID = 1
- Inc Rep SID = D

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D

Replication Policy E

- Tree-ID = 1
- Root = A
- Instance ID = 1
- Inc Rep SID = E

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

Candidate path 2

- Preference = 100

Instance-1
LSP ID = 3

Replication Policy A

- Tree-ID = 1
- Root = A
- **Instance ID = 3**
- Inc Rep SID

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = B
 - Sid-list B,G <C is bottom of Stack>

Replication Policy G

- Tree-ID = 1
- Root = A
- **Instance ID = 3**
- Inc Rep SID = G

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D
 - Sid-list <D>
- Next-hop-group-id 1
 - Next-hop-add = E
 - Sid-list <E>

Replication Policy D

- Tree-ID = 1
- Root = A
- **Instance ID = 2**
- Inc Rep SID = D

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = D

Replication Policy E

- Tree-ID = 1
- Root = A
- **Instance ID = 3**
- Inc Rep SID = E

Forwarding Info

- Next-hop-group-id 0
 - Next-hop-add = na

Next Steps

- Looking for adaptation in PIM WG