P2MP Policy
draft-hsd-pce-sr-p2mp-policy

Authors:
Hooman Bidgoli, Nokia
Daniel Voyer, Bell Canada
Ehsan Hemmati, Cisco
Saranya Rajarathinam, Nokia
Tarek Saad, Juniper
Siva Sivabalan, Ciena
Update/Relevant Drafts

Multiple Vendors are implementing/finished implementing this draft.

draft-spring-sr-replication-segment (adopted)
draft-ietf-pim-sr-p2mp-policy (adopted)
draft-hb-spring-sr-p2mp-policy-yang-01 (should we move it to PIM WG?)
draft-Parekh-bess-mvpn-evpn-sr-p2mp-00 (Next for adaptation)
draft-hsd-pce-sr-p2mp-policy-01 (Will ask for adaptation call for IETF 109)
draft-hb-idr-sr-p2mp-policy-00 (Will ask for adaptation call for IETF 110)
draft-hb-pim-p2mp-policy-ping-00 (New)
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.
SR P2MP Segment

- A Point-to-Multipoint (P2MP) segment connects a Root node to a set of Leaf nodes in segment routing domain.

- A Point-to-Multipoint Policy contains
  - Is identified via ROOT-ID and TREE-ID
  - A set of Leaves
  - Candidate paths used for P2MP Tree redundancy
  - Candidate paths contain Path-Instances used for Global Optimization

- 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 configured explicitly on the PCE or controller and programmed on the PCC
Replication Segment

• Is the forwarding instructions for the P2MP LSP
  • Label instructions
  • Fast Reroute instructions

• A Replication segment can be 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;
  • LSP-ID: LSP-ID is unique per <root and p2mp policy> OR
    • node-address
    • Replicatoin-id

  • **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)
Shared Replication Segment

- Shared Replication segment is defined via following

  - Two or more P2MP trees May share a replication segment.

  - Replication segment may be identified with Zero ROOT-ID, a unique Replication-ID (for the Tree-ID) and the Node-ID

  - As an example it can be used for Facility FRR when the by-pass tunnel is made of only Replication Segments to protect a nexthop. i.e. LFA or TI-LFA is not sued.
PCC Init Example

1. BGP
   MC AD Routes

2. Root: Learn about the leaves via the AD routes

3. Update Controller: RootID Leaves

4. Calculate the Tree and its Replication Segments

5. Program PCCs with Replication Segments and forwarding information, including SID-LISTS

ROOT None SR-P2MP nodes LEAVES
SR P2MP Objects

Non-SR-P2MP nodes

Head-end policy = PMSI

P2MP LSP Redundancy

End to End Optimization

Forwarding Info
Sid-List
Fast Reroute

SR P2MP Policy
- ROOT Node, key
- Leaf Node
- Tree-ID, key

Candidate path 1
- Preference
- PLSP-ID = 1
- TE-Info

Path-Instance-1
LSP ID (tree-1)

Path-Instance-2
LSP ID (tree-2)

Candidate path N
- Preference
- PLSP-ID = N
- TE Info

Path-Instance-1
LDP-ID

Path-Instance-2
LDP-ID

Replication segment
- Node-ID???
- Tree-ID
- Root
- Instance ID
- Inc Rep SID
- Rep SID Action

Unicast SR Policy

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]
SR P2MP YANG Model

```yang
+--rw p2mp-traffic-engineering!
  +--rw p2mp-policy* [root-address tree-id]
    | +--rw root-address inet:ip-address
    | +--rw tree-id          uint32
    | +--rw p2mp-policy-name? string
    | +--rw admin-state?     enumeration
    | +--ro oper-state?      enumeration
    +--rw leaf-list* [leaf-address]
    | +--rw leaf-address inet:ip-address
    | +--rw admin-state?     enumeration
    +--rw candidate-path* [protocol-id originator discriminator]
        +--rw protocol-id       enumeration
        +--rw originator       inet:ip-address
        +--rw discriminator     uint32
        +--rw candidate-path-name? string
        +--rw admin-state?     enumeration
        +--ro oper-state?      enumeration
        +--rw preference?       uint32
        +--rw constraints* [index]
        | +--rw index           uint32
        | +--rw attributes?     uint32
        | +--rw explicit-routing* [index]
        | | +--rw index           uint32
        | | +--rw attributes?     uint32
        | | +--rw path-instances* [index]
        | | | +--rw index           uint32
        | | | +--rw instance-id?
        | |     - ../../../replication-segment/replication-id
        | |     +--ro oper-state?     enumeration
        | +--rw replication-segment* [node-address replication-id]
          | +--rw replication-segment* [node-address replication-id]
          | | +--rw node-address inet:ipv4-address
          | | +--rw replication-id      uint32
          | | +--rw admin-state?     enumeration
          | | +--ro oper-state?      enumeration
          | | +--rw root-address?    inet:ipv4-address
          | | +--rw tree-id?         uint32
          | | +--rw instance-id?     uint32
          | +--rw replication-sid?  uint32
          | +--rw downstream-nodes* [downstream-index]
          | | +--rw downstream-index  uint32
          | | +--rw next-hop-address? inet:ip-address
          | | +--rw next-hop-interface-name? if:interface-ref
          | | +--rw protecting-next-hop? boolean
          | +--rw protect-next-hop? uint32
          | +--rw (label)?
          |     +--:(sid-list)
          |     | +--rw sid-list* [index]
          |     | | +--rw index           uint32
          |     | | +--rw sid-segment-type? uint32
          |     |     +--:(sr-policy)
          |     |     | +--rw sr-policy* [replication-sid]
          |     |     | | +--rw replication-sid     uint32
          |     |     | | +--rw sr-policy?     string
          |     |     |     +--:(rsvp-te)
          |     |     |     | +--rw rsvp-te* [replication-sid]
          |     |     |     | | +--rw replication-sid     uint32
          |     |     |     | | +--rw rsvp-te-tunnel-id? uint32
          | +--rw replication-segment* [node-address replication-id]
          |     +--rw replication-segment* [node-address replication-id]
```

...
Example 1
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
1. Ingress Replication from A to D and A to E
2. Root and Leaves need to support Replication Policy.
3. B, C, G don’t support P2MP Policy and are part of the unicast SR.
4. All SR resiliency functionality can be used in unicast SR domain.
Example 3
FRR via Shared Replication Segment

1. The primary path is A to C to LEAF D
2. Link between C and D is cut, FRR NextHop Protection via G
3. G can use a Shared RS to act as a facility bypass for multiple trees.

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 S ID = C

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>

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 G
- Tree-ID = 100
- Root = 0
- Instance ID = 1
- Inc Rep SID = G

Forwarding Info
- Next-hop-group-id 0
  - Next-hop-add = D
  - Sid-list <G>
Next Steps

• Asking for adaptation of this draft

Thank you!