P2MP Policy

Draft-hb-idr-sr-p2mp-policy

Authors:

Hooman Bidgoli, Nokia
Daniel Voyer, Bell Canada
Andrew Stone, Nokia
Rishabh Parekh, Cisco
Serge Krier, Cisco
Arvind Venkateswaran, Cisco

Presenter Hooman Bidgoli

Update/Relevant Drafts

Multiple Vendors are in the mist of implementing this draft.

draft-spring-sr-replication-segment (adopted)

draft-ietf-pim-sr-p2mp-policy (adopted)

draft-hb-spring-sr-p2mp-policy-yang-01

draft-ietf-bess-mvpn-evpn-sr-p2mp-02 (adopted)

draft-hsd-pce-sr-p2mp-policy-03 (adoption call on going until Nov 25th)

draft-hb-idr-sr-p2mp-policy-04 (asking for adoption in this IETF, IPR call has been concluded)

draft-hb-pim-p2mp-policy-ping-001 (adoption call on going)

SR P2MP Policy

- A Point-to-Multipoint (P2MP) Policy connects a Root node to a set of Leaf nodes via Candidate Paths (CPs).
  - CPs have preferences, highest preference is the active CP
  - Each CP has two or more path instance for golobal optimization and MBB procedures

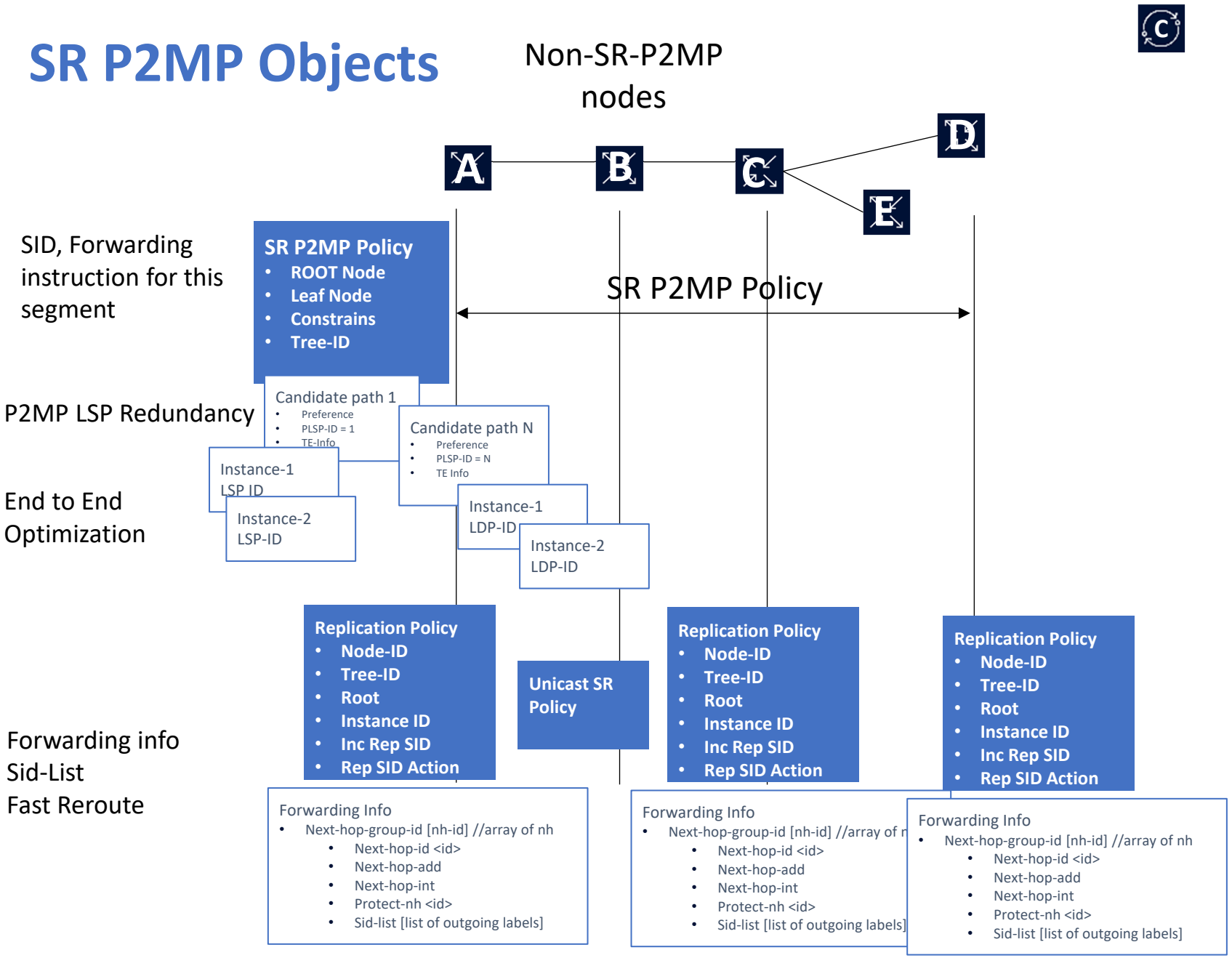
- A P2MP segment contains Replication Segments, each providing forwarding instructions and protection at Root, Transit Nodes and Leaf Nodes.

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

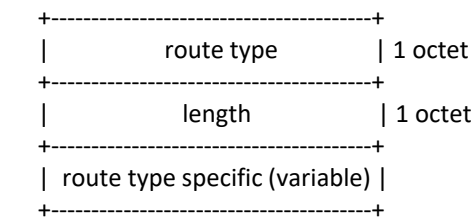
SR P2MP Objects



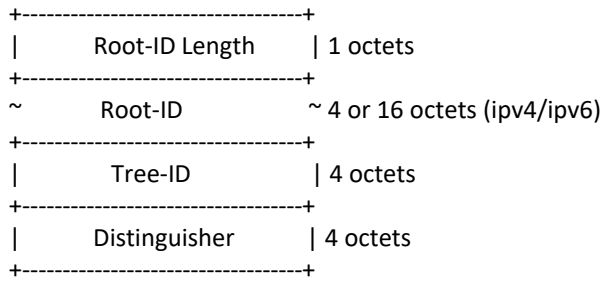
New BGP NLRI and Route Types

- New BGP NLRI, called the P2MP-POLICY NLRI
- A new SAFI is defined: the SR P2MP Policy SAFI, (Codepoint tbd assigned by IANA)
- Route Types
  - P2MP Policy route
  - Replication segment Binding SID route
  - Replication segment OIF route

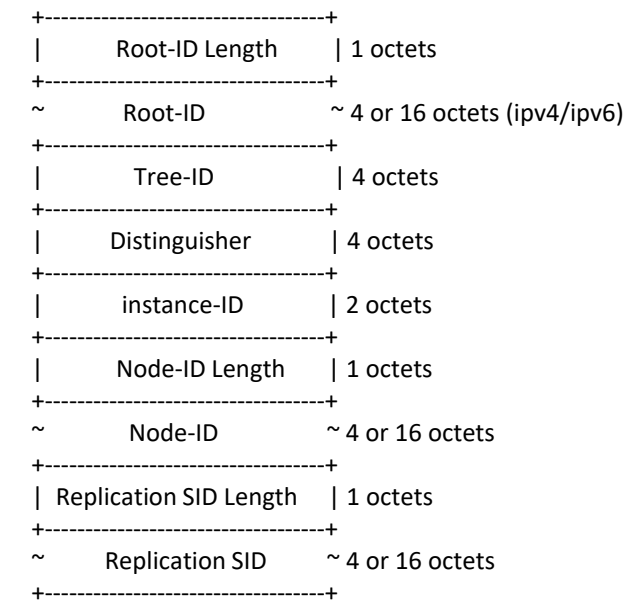
P2MP Policy NLRI



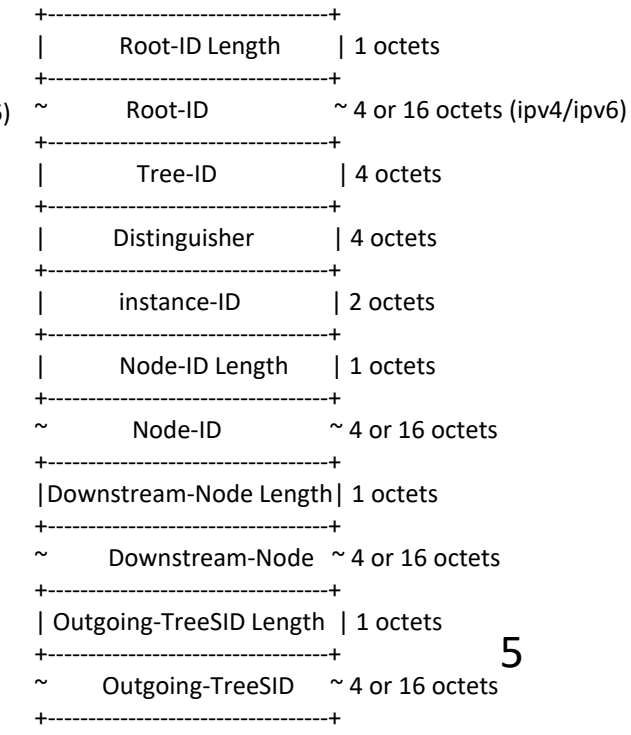
P2MP Policy route



Replication Segment Binding SID route



Replication segment OIF route



BGP SR P2MP Policy

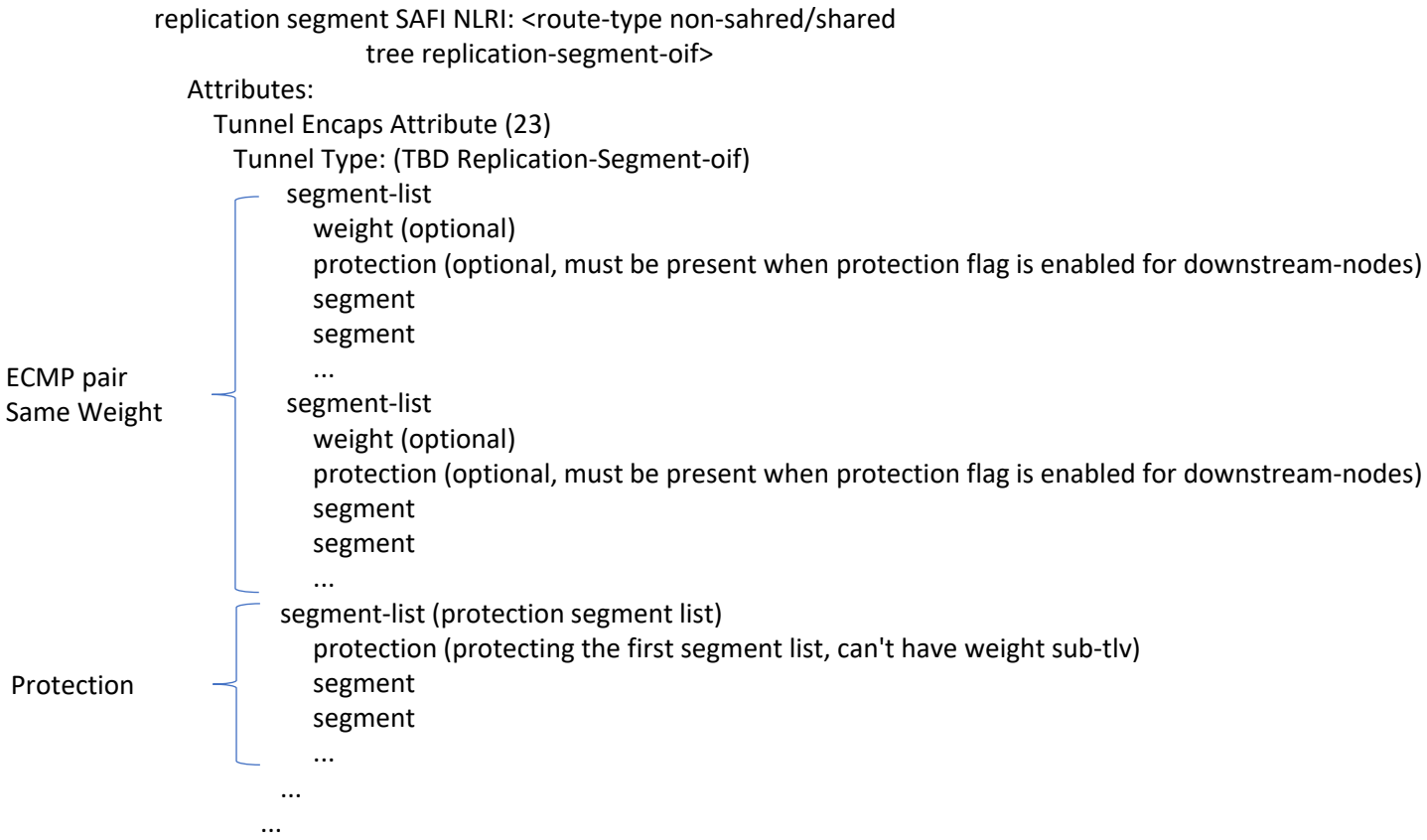
SR P2MP Policy SAFI NLRI: <route-type p2mp-policy>
Attributes:
Tunnel Encaps Attribute (23)
Tunnel Type: (TBD, P2MP-Policy)
Preference
Policy Name
Policy Candidate Path Name
leaf-list (optional)
remote-end point
remote-end point
...
path-instance
active-instance-id
instance-id
instance-id
...

Candidate Path and its preference

List of leaves discovered by Controller
It can optionally be downloaded to the node for debug info

Path Instances under a candidate path, used for global optimization

BGP SR P2MP Policy



- With this new model each OIF is downloaded individually
- Closer implementation to Segment-Routing-Policy
- ECMP is optional for each OIF via multiple segment-list
- Protection is optional for each OIF via protection sub-tlv

Next Steps

- Comments welcome
- IPR call was done
- Asking for adoption

Thank You!

SR P2MP YANG Model

```
+--rw p2mp-traffic-engiineering!
     +--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-nexthop-id?       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
```

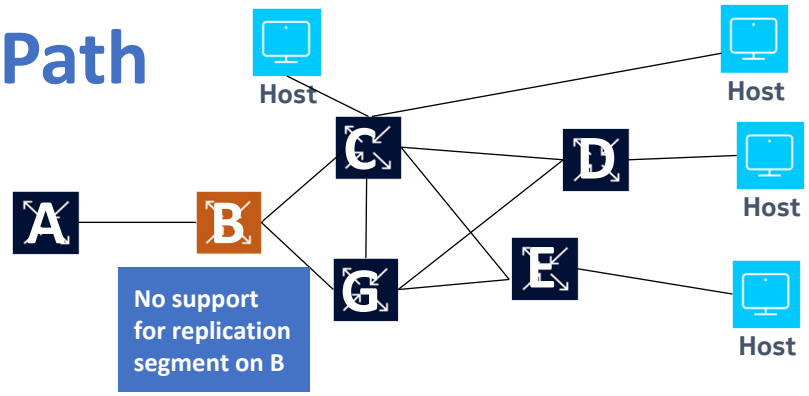
Replication Segment

- 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;
  - Node-ID: The node this Replication Segment belongs too.
  - 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.**
    - **There could be exceptions like using a shared replication segment for FRR**
- 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.
  - A tree has its own replication segment at its root.
  - 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.

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

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>
- Next-hop-group-id 2
  - Next-hop-add = 127.0.0.1 (Bud)

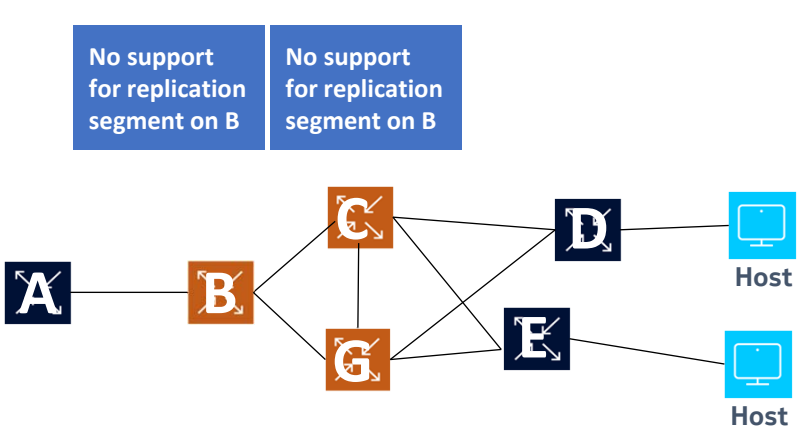
Replication Policy D
- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID = D

Forwarding Info
- Next-hop-group-id 0
  - Next-hop-add = 127.0.0.1

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 = 127.0.0.1

Example 2



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.

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,D <D is bottom of Stack>
- Next-hop-group-id 1
  - Next-hop-add = B
  - Sid-list B,G,E <E is bottom of Stack>

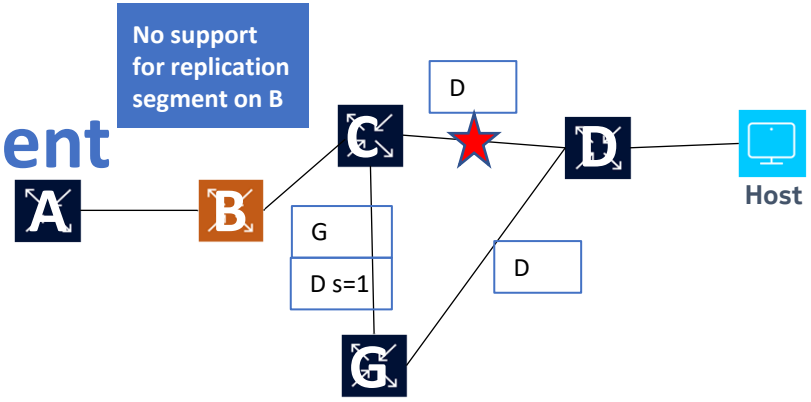
Replication Policy D
- Tree-ID =1
- Root = A
- Instance ID = 1
- Inc Rep SID = D

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 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.
4. G Pops bypass label (Implicit Null and forwards D).

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>
  - Prot NH 1
- Next-hop-group-id 1
  - Next-hop-add = G
  - Sid-list <G>

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 <impl-null>

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

Thank You!