Segment Routing Tree Segment Identifier

Tree Segment Identifier (Tree-SID) is an SDN controller-based approach to build label switched multicast (LSM) Trees for efficient delivery of multicast traffic in an SR domain and without the need for multicast protocol running in the network. With Tree SID, trees are centrally computed and controlled by a path computation element (SR-PCE).

A Replication segment (as specified in IETF draft "SR Replication segment for Multi-point Service Delivery") is a type of segment which allows a node (Replication node) to replicate packets to a set of other nodes (Downstream nodes) in a Segment Routing Domain.

A Replication segment includes the following:

- Replication SID: The Segment Identifier of a Replication segment. This is an SR-MPLS label (Tree SID label).
- Downstream nodes: Set of nodes in Segment Routing domain to which a packet is replicated by the Replication segment.

A Point-to-Multipoint (P2MP) tree is formed by stitching Replication segments on the Root node, intermediate Replication nodes, and Leaf nodes. This is referred to as an SR P2MP Policy (as specified in IETF draft "Segment Routing Point-to-Multipoint Policy").

An SR P2MP policy works on existing MPLS data-plane and supports TE capabilities and single/multi routing domains. At each node of the tree, the forwarding state is represented by the same Replication segment (using a global Tree-SID specified from the SRLB range of labels).

An SR P2MP policy request contains the following:

- Policy name
- SID for the P2MP Tree (Tree-SID)
- Address of the root node
- Addresses of the leaf nodes
- Optimization objectives (TE, IGP, delay metric)
- Constraints (affinity)

The SR-PCE is responsible for the following:

1. Learning the network topology - *to be added*

2. Learning the Root and Leaves of a Tree - *describe dynamic and static Tree SIDs (16-17) - Tree SID Policy Types and Behaviors*
3. Computing the Tree
4. Allocating MPLS label for the Tree
5. Signaling Tree forwarding state to the routers
6. Re-optimizing Tree

Tree SID Policy Types and Behaviors

- Static P2MP Policies—can be configured in the following ways:
  - Tree SID parameters provided via Cisco Crosswork Optimization Engine (COE) UI
    - COE passes the policy configuration to the SR-PCE via REST API (no Tree-SID CLI at PCE). This method allows for SR-PCE High Availability (HA).

Note Refer to the *Traffic Engineering in Crosswork Optimization Engine* chapter in the Cisco Crosswork Optimization Engine documentation.

  - Tree SID parameters configured via Tree-SID CLI at the SR-PCE

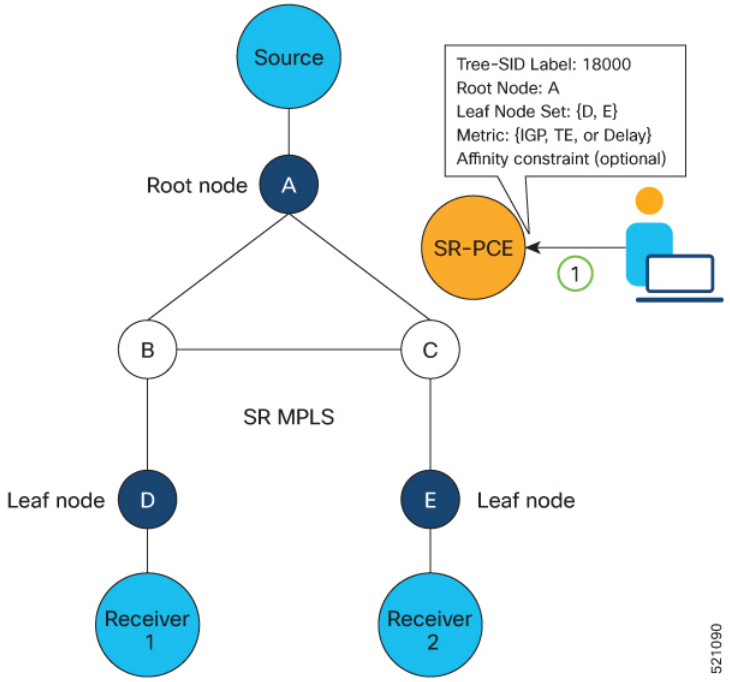
Caution With this method, SR-PCE HA is not supported. For this reason, this configuration method is not recommended.

- Dynamic P2MP Policies—can be configured in the following ways:
  - A BGP mVPN is configured in the network (PE nodes) – service configuration via CLI or Cisco NSO
    - As a result, BGP control plane is used for PE auto-discovery and customer multicast signaling.
  - Tree SID parameters are provided by mVPN PEs via PCEP to the PCE. This method allows for SR-PCE High Availability (HA).

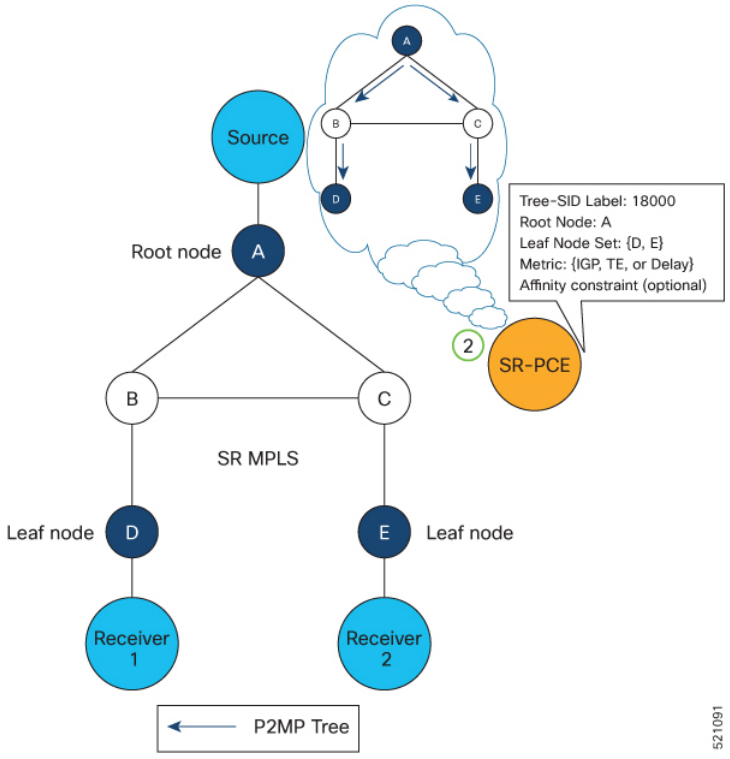
Tree SID Workflow Overview

This sections shows a basic workflow using a static Tree SID policy:

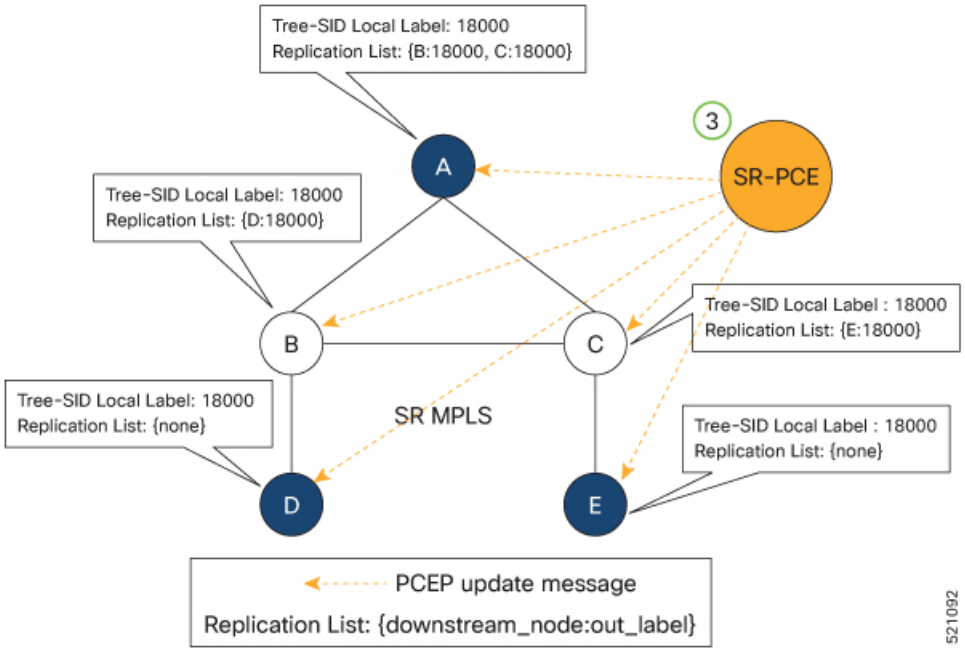
1. User creates a static Tree-SID policy, either via Crosswork Optimization Engine (preferred), or via CLI at the SR-PCE (not recommended).



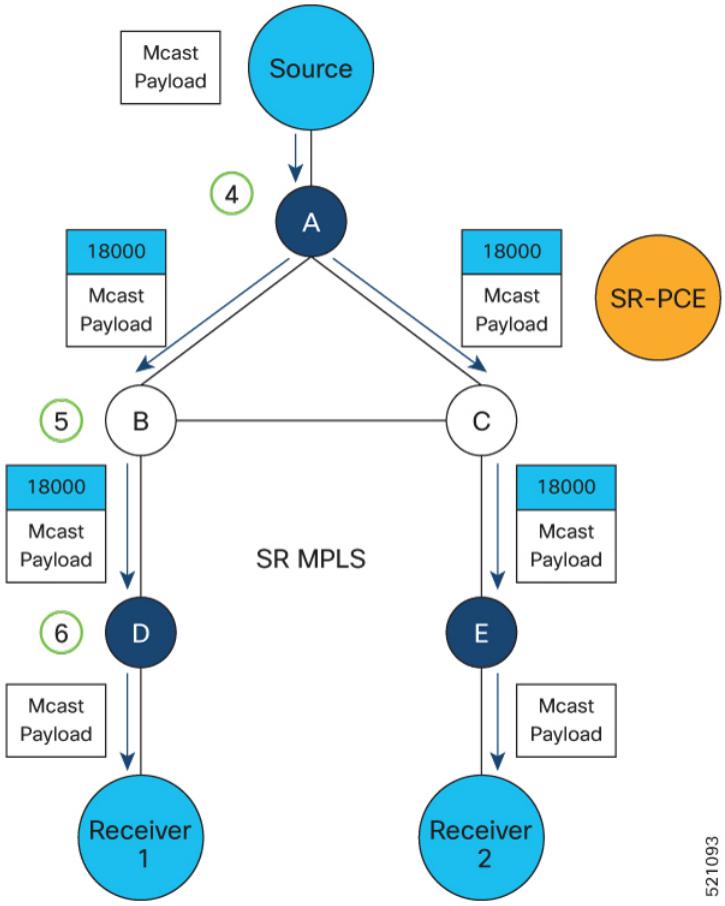
2. SR-PCE computes the P2MP Tree.



3. SR-PCE instantiates the Tree-SID state at each node in the tree.



4. The Root node encapsulates the multicast traffic, replicates it, and forwards it to the Transit nodes.
5. The Transit nodes replicate the multicast traffic and forward it to the Leaf nodes.
6. The Leaf nodes decapsulate the multicast traffic and forward it to the multicast receivers.



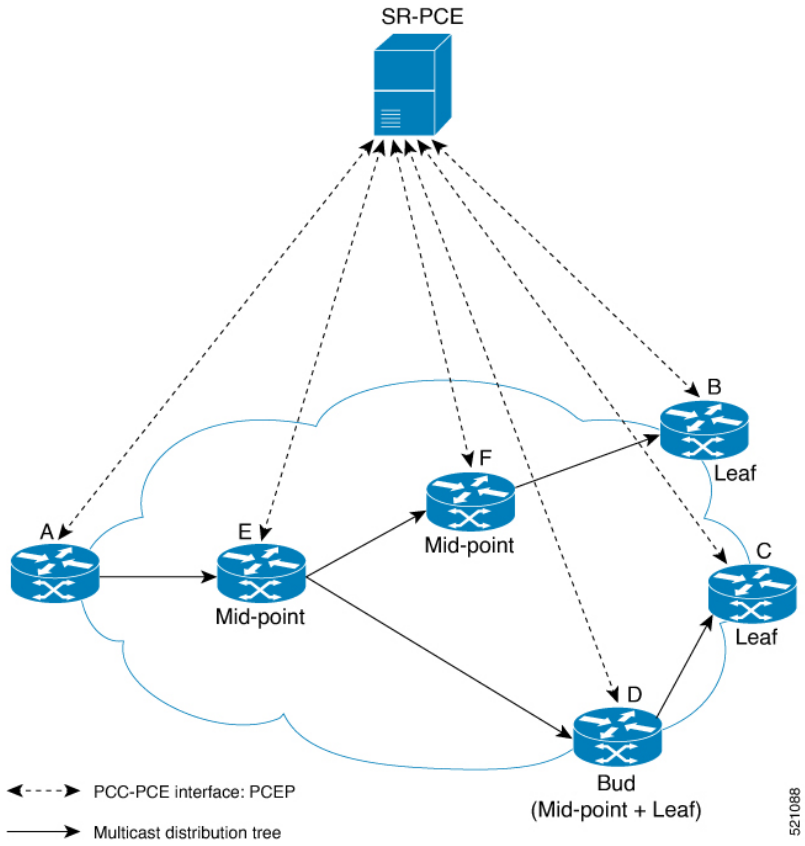
Usage Guidelines and Limitations

- SR-PCE High Availability (HA) is supported for dynamic P2MP policies and for static P2MP policies configured via Cisco Crosswork Optimization Engine (COE) UI.
- SR-PCE HA is not supported for static Tree-SID policy configured via Tree-SID CLI at the SR-PCE. Tree-SID can only be controlled by a single PCE. Configure only one PCE on each PCC in the Tree-SID path.

Bud Node Support

In a multicast distribution tree, a Bud node is a node that acts as a leaf (egress) node as well as a mid-point (transit) node toward the downstream sub-tree.

In the below multicast distribution tree topology with Root node {A} and Leaf nodes set {B, C, D}, node D is a Bud node. Similarly, if node E is later added to the Leaf set, it would also become a Bud node.



The tree computation algorithm on SR-PCE has been enhanced to detect a Bud node based on knowledge of the Leaf set, and to handle Leaf/Transit node transitions to Bud node. The role of the Bud node is also explicitly signaled in PCEP.

Configure Static Segment Routing Tree-SID via CLI at SR-PCE

Caution With this configuration method, SR-PCE HA is not supported. For this reason, this configuration method is not recommended.

To configure static Segment Routing Tree-SID for Point-to-Multipoint (P2MP) SR policies, complete the following configurations:

1. Configure Path Computation Element Protocol (PCEP) Path Computation Client (PCC) on all nodes involved in the Tree-SID path (root, mid-point, leaf)
2. Configure Affinity Maps on the SR-PCE
3. Configure P2MP SR Policy on SR-PCE
4. Configure Multicast on the Root and Leaf Nodes

Configure PCEP PCC on All Nodes in Tree-SID Path

Configure all nodes involved in the Tree-SID path (root, mid-point, leaf) as PCEP PCC. For detailed PCEP PCC configuration information, see Configure the Head-End Router as PCEP PCC.

Configure Affinity Maps on the SR-PCE

Use the **affinity bit-map** *COLOR bit-position* command in PCE SR-TE sub-mode to define affinity maps. The bit-position range is from 0 to 255.

```
Router# configure
Router(config)# pce
Router(config-pce)# segment-routing traffic-eng
Router(config-pce-sr-te)# affinity bit-map RED 23
Router(config-pce-sr-te)# affinity bit-map BLUE 24
Router(config-pce-sr-te)# affinity bit-map CROSS 25
Router(config-pce-sr-te)#
```

Configure P2MP SR Policy on SR-PCE

Configure the end-point name and addresses, Tree-SID label, and constraints for the P2MP policy.

Use the **endpoint-set** *NAME* command in SR-PCE P2MP sub-mode to enter the name of the end-point set and to define the set of end-point addresses.

```
Router(config-pce-sr-te)# p2mp
Router(config-pce-sr-te-p2mp)# endpoint-set BAR
Router(config-pce-p2mp-ep-set)# ipv4 10.1.1.2
Router(config-pce-p2mp-ep-set)# ipv4 10.1.1.3
Router(config-pce-p2mp-ep-set)# ipv4 10.1.1.4
Router(config-pce-p2mp-ep-set)# exit
Router(config-pce-sr-te-p2mp)#
```

Use the **policy** *policy* command to configure the P2MP policy name and enter P2MP Policy sub-mode. Configure the source address, endpoint-set color, Tree-SID label, affinity constraints, and metric type.

```
Router(config-pce-sr-te-p2mp)# policy FOO
Router(config-pce-p2mp-policy)# source ipv4 10.1.1.6
Router(config-pce-p2mp-policy)# color 10 endpoint-set BAR
Router(config-pce-p2mp-policy)# treesid mpls 15200
Router(config-pce-p2mp-policy)# candidate-paths
Router(config-pce-p2mp-policy-path)# constraints
Router(config-pce-p2mp-path-const)# affinity
Router(config-pce-p2mp-path-affinity)# exclude BLUE
Router(config-pce-p2mp-path-affinity)# exit
Router(config-pce-p2mp-path-const)# exit
Router(config-pce-p2mp-policy-path)# preference 100
Router(config-pce-p2mp-policy-path-preference)# dynamic
Router(config-pce-p2mp-path-info)# metric type te
```

```
Router(config-pce-p2mp-path-info)# root
Router(config)#
```

Configure Multicast on the Root and Leaf Nodes

On the root node of the SR P2MP segment, use the **router pim** command to enter Protocol Independent Multicast (PIM) configuration mode to statically steer multicast flows into an SR P2MP policy.

Note Enter this configuration only on an SR P2MP segment. Multicast traffic cannot be steered into a P2P policy.

```
Router(config)# router pim
Router(config-pim)# vrf name
Router(config-pim-name)# address-family ipv4
Router(config-pim-name-ipv4)# sr-p2mp-policy FOO
Router(config-pim-name-ipv4-srp2mp)# static-group 235.1.1.5 10.1.1.6
Router(config-pim-name-ipv4-srp2mp)# root
Router(config)#
```

On the root and leaf nodes of the SR P2MP tree, use the **mdt static segment-routing** command to configure the multicast distribution tree (MDT) core as Tree-SID from the multicast VRF configuration submode.

```
Router(config)# multicast-routing
Router(config-mcast)# vrf TEST
Router(config-mcast-TEST)# address-family ipv4
Router(config-mcast-TEST-ipv4)# mdt static segment-routing
```

On the leaf nodes of an SR P2MP segment, use the **static sr-policy** *p2mp-policy* command to configure the static SR P2MP Policy from the multicast VRF configuration submode to statically decapsulate multicast flows.

```
Router(config)# multicast-routing
Router(config-mcast)# vrf TEST
Router(config-mcast-TEST)# address-family ipv4
Router(config-mcast-TEST-ipv4)# static sr-policy FOO
```

Running Config

The following example shows how to configure the end point addresses and P2MP SR policy with affinity constraints on SR-PCE.

```
pce
 segment-routing
  traffic-eng
   affinity bit-map
    RED 23
    BLUE 24
    CROSS 25
   !
   p2mp
    endpoint-set BAR
     ipv4 10.1.1.2
     ipv4 10.1.1.3
     ipv4 10.1.1.4
    !
```

```
    policy FOO
     source ipv4 10.1.1.6
     color 10 endpoint-set BAR
     treesid mpls 15200
     candidate-paths
      preference 100
       dynamic
        metric
         type te
        !
       !
      !
      constraints
       affinity
        exclude
         BLUE
        !
       !
      !
     !
    !
   !
  !
 !
!
```

The following example shows how to statically decapsulate multicast flows on the leaf nodes.

```
multicast-routing
 vrf TEST
  address-family ipv4
   static sr-policy FOO
  !
 !
!
```

The following example shows to configure the multicast distribution tree (MDT) core as Tree-SID on the root and leaf nodes.

```
multicast-routing
 vrf TEST
  address-family ipv4
   mdt static segment-routing
  !
 !
!
```

The following example shows how to steer traffic to the SR P2MP policy on the root node.

```
router pim
 vrf TEST
  address-family ipv4
   sr-p2mp-policy FOO
    static-group 232.1.1.5 10.1.1.6
   !
  !
 !
!
```

Multicast VPN: Dynamic Tree-SID MVPN (with TI-LFA)

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Multicast VPN: Dynamic Tree-SID MVPN (with TI-LFA)	Release 7.3.1	With this feature, you can use SR and MVPN for optimally transporting IP VPN multicast traffic over the SP network, using SR-PCE as a controller. With SR's minimal source router configuration requirement, its ability to implement policies with specific optimization objectives and constraints, protect against network failures using TI-LFA FRR mechanism, and use SR-PCE to dynamically generate optimal multicast trees (including when topology changes occur in the multicast tree), the SR-enabled SP network can transport IP multicast traffic efficiently.

Prerequisites for Multicast VPN: Tree-SID MVPN With TI-LFA

- The underlay OSPF/IS-IS network is configured, and OSPF/IS-IS adjacency is formed between routers, across the network.
- BGP is configured for the network, and BGP adjacency is formed between routers. BGP MVPN configuration information is provided in this feature document.
- To understand the benefits, know-how, and configuration of SR and SR-TE policies, see About Segment Routing and Configure SR-TE Policies.

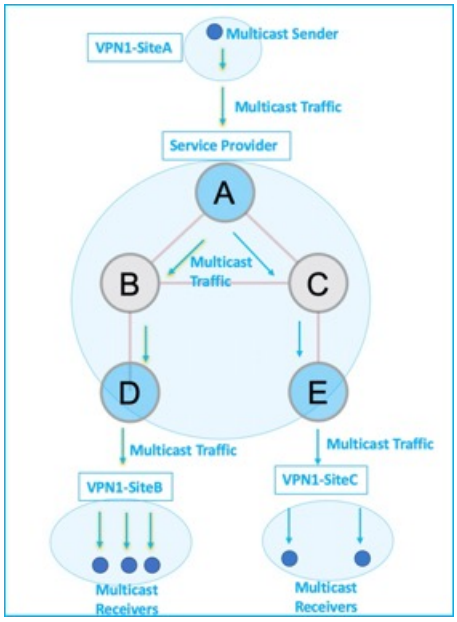
Information About Multicast VPN: Tree-SID MVPN With TI-LFA

Typically, a customer's IP VPN is spread across VPN sites. IP VPN customer traffic is sent from one site to another over a VPN Service Provider (SP) network.

When IP multicast traffic within a (BGP/MPLS) IP VPN is transported over an SP network (say, from **VPN1-Site-A** to **VPN1-Site-B**, as shown in the image), the SP network requires protocols and procedures to optimally transport multicast traffic from a multicast sender in Site-A to multicast receivers in Site-B.

This use case explains how to enable SR multicast for an SP network, and efficiently transport IP VPN multicast traffic (sent from **VPN1-Site-A** and) received at PE router A, through to PE routers D and E, towards receivers in sites **VPN1-Site-B** and **VPN1-Site-C**.

Figure 1: IP VPN Multicast Traffic Flow Over An SP Network



To enable the *Multicast VPN: Tree-SID MVPN With TI-LFA* feature, the following protocols and software applications are used.

OSPF/IS-IS - The underlay network is created with OSPF/IS-IS routing protocol, and reachability is established across the network. See *Configure Segment Routing for IS-IS Protocol*or *Configure Segment Routing for OSPF Protocol* chapter for details.

BGP Multicast VPN (MVPN) – The PE routers (A, D, and E) are IP VPN end-points for IP multicast traffic arriving at the SP network (at PE router A) and exiting the SP network (at PE routers D and E). So, BGP MVPN is enabled on the PE routers. NSO is used to configure BGP MVPN on the PE routers.

BGP Auto-Discovery (AD) - To enable distributed VPN end-point discovery and C-multicast flow mapping and signalling, BGP AD function is configured on the PE routers. A BGP Auto-Discovery route contains multicast router (loopback IP address) and tree identity (segment ID) information. It carries the information in the Provider Multicast Service Interface (PMSI) Tunnel Attribute (PTA).

C-multicast states are signaled using BGP.

SR - To transport IP multicast traffic between the VPN end-points (PE routers A, D, and E), Provider (or P-) tunnels are used. In a P-tunnel, the PE devices are the tunnel end-points. P-tunnels can be generated using different technologies (RSVP-TE, P2MP LSPs, PIM trees, mLDP P2MP LSPs, and mLDP MP2MP LSPs). In this use case, Segment Routing (SR) is used for its benefits that were noted earlier.

With SR and SR-PCE, a Tree-SID Point-to-Multipoint (P2MP) segment is used to create P-Tunnels for MVPN. You can specify SR policy optimization objectives (such as *metrics*) and constraints (such as *affinity*) in an SR policy and send it to the SR-PCE controller, so that it can dynamically create SR multicast trees for traffic flow.

SR-PCE - This is a controller which, based on the provided SR policy information, computes optimal paths for a multicast tree, and deploys the tree forwarding state on the multicast routers. When a topology change occurs, SR-PCE automatically computes a new, optimal multicast tree, and deploys the new tree forwarding state on the multicast routers.

TI-LFA - In SR-TE, Topology-Independent Loop-Free Alternate (TI-LFA) fast reroute (FRR) function is used to reduce link and node failure reaction time. When the primary next-hop (router link) fails, a pre-computed alternate next hop is used to send traffic. TI-LFA FRR is used when transporting IP VPN multicast traffic.

Overview of Multicast VPN: Tree-SID MVPN With TI-LFA

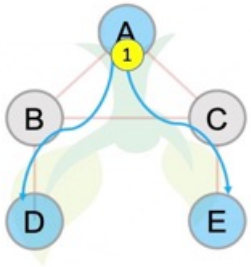
The following sections provide an overview of Tree-SID MVPN and TI-LFA. The topology remains the same, with PE routers A, D, and E acting as VPN end-points for carrying IP VPN multicast traffic.

Tree-SID MVPN Overview

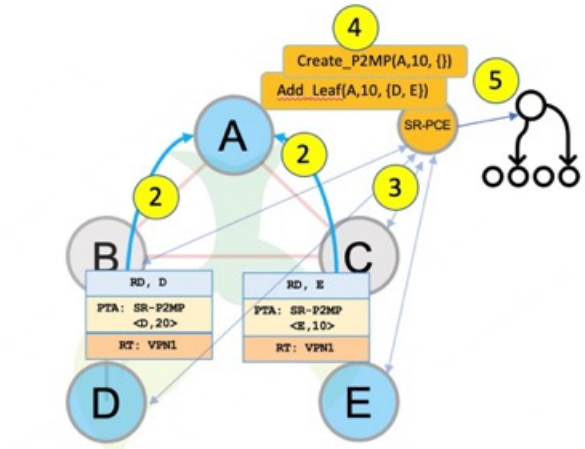
1. For SR, A is designated as the SR head-end router, and D and E are designated as the SR end-points.

   For multicast traffic, A is the root of the SR multicast tree, and D and E are leaf routers of the tree. B and C are the other multicast routers. The objective is to send the IP multicast traffic arriving at A to D and E, as needed

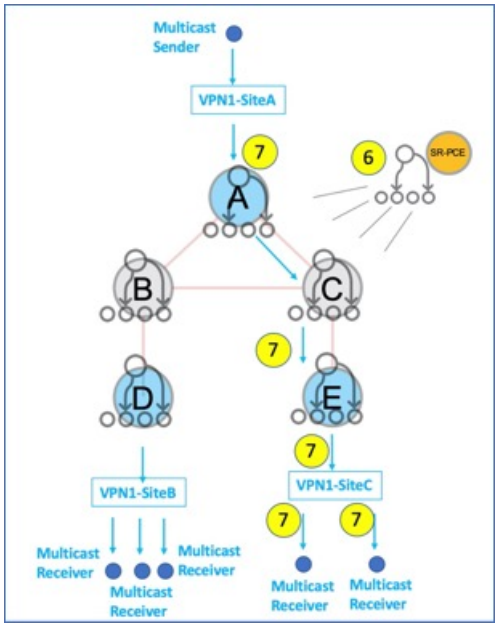
   Figure 2: Multicast Tree



2. A discovers leaf routers' information through BGP MVPN.
3. Path Computation Element Protocol (PCEP) is used for the SR multicast policy communication between A and the SR-PCE server, and communication between PE routers and the SR-PCE server.
4. When the head-end router SR policy is created on A, and PCEP configurations are enabled on the SR-PCE server and all multicast routers, SR-PCE receives the SR policy and leaf router identity information from A.
5. Based on the policy information it receives, including TE objectives and constraints, SR-PCE builds multicast distribution trees in the underlay for efficient VPN traffic delivery.



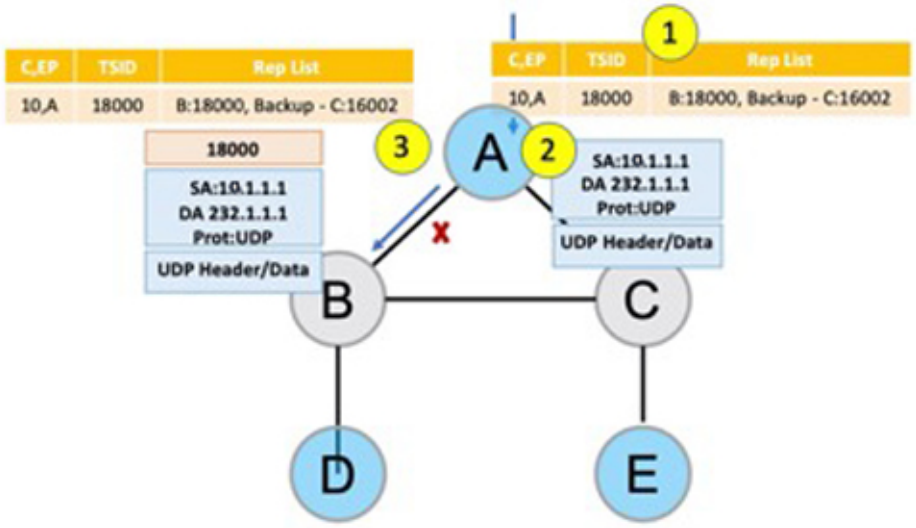
6. SR-PCE assigns an SID for the SR multicast tree policy, and deploys the multicast tree forwarding state on the multicast routers.
7. When IP multicast traffic is sent from VPN1-SiteA to PE router A, it steers it into the SR policy, and sends it towards D and E, which forward it to multicast traffic receivers in the sites VPN1-SiteB and VPN1-SiteC.
8. When a leaf/multicast router is added or removed, PE router A updates the SR multicast policy and sends it to SR-PCE. SR-PCE computes new multicast routes, and deploys the multicast tree forwarding state information on the multicast routers.



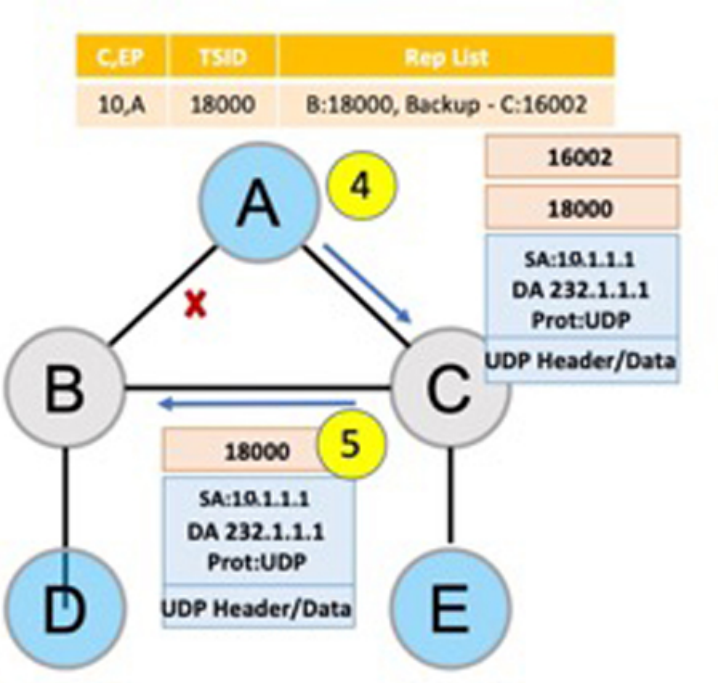
TI-LFA FRR Overview

High-level TI-LFA FRR function is depicted in these steps:

1. Tree-SID FRR state information.
   - The link from A to B is protected.
   - SID 16002 is the node SID of B.
   - A programs a backup path to B, through C.
2. IP multicast traffic arrives at A which steers the flow onto the tree.
3. A encapsulates and replicates to B, but the link to B is down.



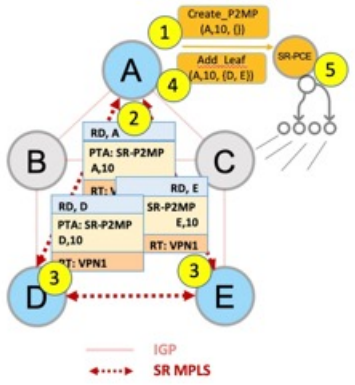
4. A sends the traffic on the backup path, to C.
5. C sends the traffic to B where normal traffic processing resumes.



SR Multicast Tree Types

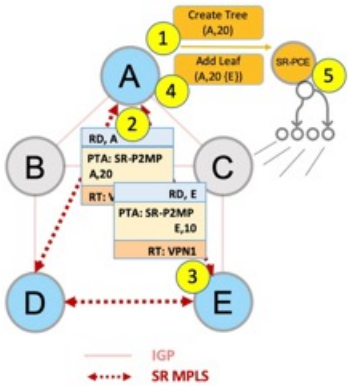
This is an overview of the types of SR multicast trees you can configure, depending on your requirement. You can create a full mesh, on-demand, or optimal multicast tree for IP VPN multicast flow in the SP network.

Figure 3: Full Mesh Multicast Tree



1. A assigns Tree-ID 10 and invokes a Create an SR multicast tree request by sending the multicast router and tree ID information (A, 10) towards SR-PCE.
2. A announces BGP AD Inclusive PMSI (I-PMSI) route with the PTA (A, 10). Inclusive PMSI - Traffic that is multicast by a PE router on an I-PMSI is received by all other PEs in the MVPN. I-PMSIs are generated by Inclusive P-tunnels .
3. A discovers VPN endpoints D and E from their BGP AD Type I-PMSI route messages.
4. A invokes an Add SR multicast leaf router request (for D and E) to SR-PCE.
5. SR-PCE computes and generates the multicast tree forwarding state information on all the routers that are part of the tree.

Figure 4: On-Demand SR Multicast Tree

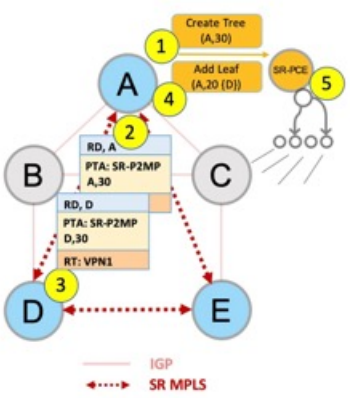


1. A assigns Tree-ID 20 and invokes a Create an SR multicast tree request by sending the multicast router and tree ID information (A, 20) towards SR-PCE.
2. A announces BGP AD Selective PMSI (or S-PMSI) route with PTA (A, 20). A sets the leaf-info-required to discover endpoint interest set.

   Selective PMSI - Traffic multicast by a PE on an S-PMSI is received by some PEs in the MVPN. S-PMSIs are generated by Selective P-tunnels.

3. E has a receiver behind it, and announces a BGP-AD leaf route towards A. A discovers service endpoint E for the on-demand tree.
4. A invokes an Add SR multicast leaf router request (for E) to SR-PCE.
5. SR-PCE computes and generates the multicast tree information for all the routers that are part of the tree.

Figure 5: Optimal Multicast Tree



1. A decides to optimize a flow and assigns Tree-ID 30 and invokes a Create an SR multicast tree request by sending the multicast router and tree ID information (A, 30) towards SR-PCE.
2. A announces BGP AD I-PMSI route with PTA (A,30). A sets the leaf-info-required to discover endpoint interest set.
3. D has a receiver behind it, and announces a BGP-AD leaf route towards A. A discovers service endpoint D for optimized flow.
4. A invokes an Add SR multicast leaf router request (for D) to SR-PCE.
5. SR-PCE computes and generates the multicast tree information for all the routers that are part of the tree.

Configurations

Head End Router Configuration (Router A) - The following configuration is specific to the head end router.

Configure TE Constraints and Optimization Parameters

```
Router# configure terminal
Router(config)# segment-routing traffic-engineering
```

An affinity bit-map is created so that it can be applied to a link or interface.

```
Router(config-sr-te)# affinity-map name 10 bit-position 24
Router(config-sr-te)# commit
```

An affinity (or relationship) is created between the SR policy path and the link color so that SR-TE computes a path that includes or excludes links, as specified. The head-end router automatically follows the actions defined in the ODN template (for color 10) upon the arrival of VPN routes with a BGP color extended community that matches color 10.

```
Router(config)# segment-routing traffic-engineering
Router(config-sr-te)# on-demand color 10 dynamic
Router(config-sr-te-color-dyn)# affinity include-all name red
```

```
Router(config-sr-te-color-dyn)# affinity include-any name blue
Router(config-sr-te-color-dyn)# affinity exclude-any name green
Router(config-sr-te-color-dyn)# metric type te
Router(config-sr-te-color-dyn)# commit
```

The SR policy configuration on the head-end router A will be sent to the SR-PCE server, after a connection is established between A and SR-PCE.

Multicast Router Configuration

Configure PCEP Client on Multicast Routers

Associate each multicast router as a client of the SR-PCE server. The **pce address ipv4** command specifies the SR-PCE server's IP address.

```
Router# configure terminal
Router(config)# segment-routing traffic-engineering
Router(config-sr-te)# pcc pce address ipv4 3.3.3.3
Router(config-pcc-pce)# commit
```

SR PCE Server Configuration

Configure Label Range for Multicast Trees

Configure the label range to be used for transporting IP multicast traffic in SP network.

```
Router(config)# pce segment-routing traffic-eng p2mp label-range min 30000 max 60000
Router(config)# commit
```

Configure FRR

The following configurations enable FRR for all SR multicast (P2MP) trees, including dynamic and static implementations.

The **lfa** keyword enables LFA FRR on the PCE server.

```
Router(config)# pce segment-routing traffic-eng p2mp fast-reroute lfa
Router(config)# commit
```

Alternatively, you can configure FRR for each individual tree using the following configuration. The **lfa** keyword under a specific multicast policy (**tree1** in this example) enables LFA FRR function for the specified SR multicast P2MP tree.

For dynamic trees, L-flag in LSP Attributes PCEP object controls FRR on a tree.

```
Router(config)# pce
Router(config-pce)# address ipv4 192.168.0.5
Router(config-pce)# segment-routing traffic-eng p2mp policy tree1 fast-reroute lfa
Router(config-pce)# commit
```

You can create FRR node sets using the **frr-node-set from ipv4** *address* and **frr-node-set to ipv4** *address* commands to specify the *from* and *to* paths on a multicast router that requires FRR protection. In this configuration, the PCE server is configured to manage the FRR function for traffic from 192.168.0.3 sent towards 192.168.0.4 and 192.168.0.5.

```
Router(config)# pce
Router(config-pce)# address ipv4 192.168.0.5
Router(config-pce)# segment-routing traffic-eng
Router(config-pce-sr-te)# p2mp
Router(config-pce-sr-te-p2mp)# frr-node-set from ipv4 192.168.0.3
Router(config-pce-sr-te-p2mp)# frr-node-set to ipv4 192.168.0.4
Router(config-pce-sr-te-p2mp)# frr-node-set to ipv4 192.168.0.5
Router(config-pce-sr-te-p2mp)# commit
```

Disable ECMP load splitting

To disable ECMP load splitting of different trees on the SR-PCE server, configure the **multipath-disable** command.

```
Router(config)# pce segment-routing traffic-eng p2mp multipath-disable
Router(config)# commit
```

Multicast Routing Configuration On PE Routers

The following MVPN configurations are required for VPN end-points, the 3 PE routers.

Configure Default MDT SR P2MP MVPN Profile

In this configuration, an MDT profile of the type *default* is created, and the SR multicast policy with color 10 will be used to send IP multicast traffic, as per the constraints and optimizations of the policy, through the multicast tree.

You can also specify the FRR LFA function with the **mdt default segment-routing mpls fast-reroute lfa** command.

```
Router(config)# multicast-routing vrf cust1
Router(config-mcast-cust1)# address-family ipv4
Router(config-mcast-cust1-ipv4)# mdt default segment-routing mpls color 10
Router(config-mcast-cust1-ipv4)# commit
```

Configure Partitioned MDT SR P2MP MVPN Profile

In this configuration, an MDT profile of the type *partitioned* is created, and the SR multicast policy with color 10 will be used to send IP multicast traffic, as per the constraints and optimizations of the policy, through the multicast tree.

You can also specify the FRR LFA function with the **mdt partitioned segment-routing mpls fast-reroute lfa** command.

```
Router(config)# multicast-routing vrf cust1
Router(config-mcast-cust1)# address-family ipv4
Router(config-mcast-cust1-ipv4)# mdt partitioned segment-routing mpls color 10
Router(config-mcast-cust1-ipv4)# commit
```

The following Data MVPN configuration is required at the Ingress PE (router A) where the multicast flows need to be steered onto the *data* MDT for SR multicast traffic flow.

Note - *Data* MDT can be configured for *Default* and *Partitioned* profiles.

Configure Data MDT for SR P2MP MVPN

In this configuration, an MDT profile of the type *data* is created, and the SR multicast policy with color 10 will be used to send IP multicast traffic, as per the constraints and optimizations of the policy, through the multicast tree.

- You can enable the FRR LFA function with the **mdt data segment-routing mpls fast-reroute lfa** command. This enables LFA FRR for SR multicast trees created for all data MDT profiles.
- As an alternative to the color keyword, you can specify a route policy in the **route-policy** command, and define the route policy separately (as mentioned in the next configuration).
- The **threshold** command specifies the threshold above which a multicast flow is switched onto the data MDT. The **immediate-switch** keyword enables an immediate switch of a multicast flow to the data MDT, without waiting for threshold limit to be crossed.
- The **customer-route-acl** keyword specifies an ACL to enable specific multicast flows to be put on to the data MDT.

- **color** and **fast-reroute lfa** keywords are mutually exclusive with the **route-policy** configuration. The objective is to apply constraints (through **color**) or FRR (through LFA protection) to either all data MDTs, or apply them selectively per data MDT, using the **set on-demand-color** and **set fast-reroute lfa** options in the route policy (configured in the **mdt data** configuration).

```
Router(config)# multicast-routing vrf cust1
Router(config-mcast-cust1)# address-family ipv4
Router(config-mcast-cust1-ipv4)# mdt data segment-routing mpls 2 color 10
Router(config-mcast-cust1-ipv4)# commit
```

Route Policy Example

The route policy designates multicast flow-to-SR multicast policy mapping, with different colors.

- With this configuration, IP multicast flows for the 232.0.0.1 multicast group are steered into the SR multicast policy created with the on-demand color 10, while flows for 232.0.0.2 are steered into the policy created with color 20.
- The *data* MDT SR multicast tree created for the 232.0.0.2 multicast group is enabled with FRR LFA protection.
- Route policies can also be used to match other parameters, such as source address.

```
Router(config)# route-policy TSID-DATA
Router(config-rpl)# if destination in (232.0.0.1) then
Router(config-rpl-if)# set on-demand-color 10
Router(config-rpl-if)# pass
Router(config-rpl-if)# elseif destination in (232.0.0.2) then
Router(config-rpl-elseif)# set on-demand-color 20
Router(config-rpl-elseif)# set fast-reroute lfa
Router(config-rpl-elseif)# pass
Router(config-rpl-elseif)# endif
Router(config-rpl)# end-policy
Router(config)# commit
```

Configure MVPN BGP Auto-Discovery for SR P2MP

The following configuration is required on all PE routers, and is mandatory for *default* MDT, *partitioned* MDT, and *data* MDT.

Configure the BGP Auto-Discovery function for transporting IP multicast traffic.

```
Router(config)# multicast-routing vrf cust1
Router(config-mcast-cust1)# address-family ipv4
Router(config-mcast-cust1-ipv4)# bgp auto-discovery segment-routing
Router(config-mcast-cust1-ipv4-bgp-ad)# commit
```

Verification

View MVPN Context Information - You can view MVPN VRF context information with these commands.

View Default MDT Configuration

This command displays SR multicast tree information, including the MDT details (of *default* type, etc), and customer VRF information (route target, route distinguisher, etc).

```
Router# show mvpn vrf vpn1 context

MVPN context information for VRF vpn1 (0x9541cf0)

RD: 1:10 (Valid, IID 0x1), VPN-ID: 0:0
Import Route-targets : 2
```

```
  RT:192.168.0.4:0, BGP-AD
  RT:192.168.0.4:17, BGP-AD
BGP Auto-Discovery Enabled (I-PMSI added)
SR P2MP Core-tree data:
  MDT Name: TRmdtvpn1, Handle: 0x4150, idb: 0x956fc30
  MTU: 1376, MaxAggr: 255, SW_Int: 30, AN_Int: 60
  RPF-ID: 3, C:0, O:1, D:0, CP:0
  Static Type : - / -
  Def MDT ID: 524289 (0x93993f0), added: 1, HLI: 0x80001, Cfg: 1/0
  Part MDT ID: 0 (0x0), added: 0, HLI: 0x00000, Cfg: 0/0
  Ctrl Trees : 0/0/0, Ctrl ID: 0 (0x0), Ctrl HLI: 0x00000
```

View Partitioned MDT Configuration

This command displays SR multicast tree information, including the MDT details (of *partitioned* type, etc), and customer VRF information (route target, route distinguisher, etc).

```
Router# show mvpn vrf vpn1 context

MVPN context information for VRF vpn1 (0x9541cf0)

RD: 1:10 (Valid, IID 0x1), VPN-ID: 0:0
Import Route-targets : 2
  RT:192.168.0.4:0, BGP-AD
  RT:192.168.0.4:17, BGP-AD
BGP Auto-Discovery Enabled (I-PMSI added) , MS-PMSI sent
SR P2MP Core-tree data:
  MDT Name: TRmdtvpn1, Handle: 0x4210, idb: 0x956fc30
  MTU: 1376, MaxAggr: 255, SW_Int: 30, AN_Int: 60
  RPF-ID: 1, C:0, O:1, D:0, CP:0
  Static Type : - / -
  Def MDT ID: 0 (0x0), added: 0, HLI: 0x00000, Cfg: 0/0
  Part MDT ID: 524292 (0x9399318), added: 1, HLI: 0x80004, Cfg: 1/0
  Ctrl Trees : 0/0/0, Ctrl ID: 0 (0x0), Ctrl HLI: 0x00000
```

View Partitioned MDT Ingress PE Configuration

This command displays SR multicast tree information on the PE router that receives the multicast traffic on the SP network. The information includes PE router details, MDT details, Tree-SID details, and the specified customer VRF information.

```
Router# show mvpn vrf vpn1 pe

MVPN Provider Edge Router information

VRF : vpn1

PE Address : 192.168.0.3 (0x9570240)
  RD: 0:0:0 (null), RIB_HLI 0, RPF-ID 13, Remote RPF-ID 0, State: 0, S-PMSI: 2
  PPMP_LABEL: 0, MS_PMSI_HLI: 0x00000, Bidir_PMSI_HLI: 0x00000, MLDP-added: [RD 0, ID 0,
Bidir ID 0, Remote Bidir ID 0], Counts(SHR/SRC/DM/DEF-MD): 0, 0, 0, 0, Bidir: GRE RP Count
 0, MPLS RP Count 0RSVP-TE added: [Leg 0, Ctrl Leg 0, Part tail 0 Def Tail 0, IR added:
[Def Leg 0, Ctrl Leg 0, Part Leg 0, Part tail 0, Part IR Tail Label 0
  Tree-SID Added: [Def/Part Leaf 1, Def Egress 0, Part Egress 0, Ctrl Leaf 0]
  bgp_i_pmsi: 1,0/0 , bgp_ms_pmsi/Leaf-ad: 1/1, bgp_bidir_pmsi: 0, remote_bgp_bidir_pmsi:
0, PMSIs: I 0x9570378, 0x0, MS 0x94e29d0, Bidir Local: 0x0, Remote: 0x0, BSR/Leaf-ad 0x0/0,
 Autorp-disc/Leaf-ad 0x0/0, Autorp-ann/Leaf-ad 0x0/0
 IIDs: I/6: 0x1/0x0, B/R: 0x0/0x0, MS: 0x1, B/A/A: 0x0/0x0/0x0

  Bidir RPF-ID: 14, Remote Bidir RPF-ID: 0
  I-PMSI:   Unknown/None (0x9570378)
  I-PMSI rem:  (0x0)
  MS-PMSI:   Tree-SID [524290, 192.168.0.3] (0x94e29d0)
  Bidir-PMSI:  (0x0)
  Remote Bidir-PMSI:  (0x0)
```

```
  BSR-PMSI:  (0x0)
  A-Disc-PMSI:  (0x0)
  A-Ann-PMSI:  (0x0)
  RIB Dependency List: 0x0
  Bidir RIB Dependency List: 0x0
    Sources: 0, RPs: 0, Bidir RPs: 0
```

View Partitioned MDT Egress PE Configuration

This command displays SR multicast tree information on the MVPN egress PE router that sends multicast traffic from the SP network towards multicast receivers in the destination sites. The information includes PE router, Tree-SID, MDT, and the specified customer VRF details.

```
Router# show mvpn vrf vpn1 pe

MVPN Provider Edge Router information

PE Address : 192.168.0.4 (0x9fa38f8)
  RD: 1:10 (valid), RIB_HLI 0, RPF-ID 15, Remote RPF-ID 0, State: 1, S-PMSI: 2
  PPMP_LABEL: 0, MS_PMSI_HLI: 0x00000, Bidir_PMSI_HLI: 0x00000, MLDP-added: [RD 0, ID 0,
Bidir ID 0, Remote Bidir ID 0], Counts(SHR/SRC/DM/DEF-MD): 1, 1, 0, 0, Bidir: GRE RP Count
 0, MPLS RP Count 0RSVP-TE added: [Leg 0, Ctrl Leg 0, Part tail 0 Def Tail 0, IR added:
[Def Leg 0, Ctrl Leg 0, Part Leg 0, Part tail 0, Part IR Tail Label 0
  Tree-SID Added: [Def/Part Leaf 0, Def Egress 0, Part Egress 1, Ctrl Leaf 0]
  bgp_i_pmsi: 1,0/0 , bgp_ms_pmsi/Leaf-ad: 1/0, bgp_bidir_pmsi: 0, remote_bgp_bidir_pmsi:
0, PMSIs: I 0x9f77388, 0x0, MS 0x9fa2f98, Bidir Local: 0x0, Remote: 0x0, BSR/Leaf-ad 0x0/0,
 Autorp-disc/Leaf-ad 0x0/0, Autorp-ann/Leaf-ad 0x0/0
 IIDs: I/6: 0x1/0x0, B/R: 0x0/0x0, MS: 0x1, B/A/A: 0x0/0x0/0x0

  Bidir RPF-ID: 16, Remote Bidir RPF-ID: 0
  I-PMSI:   Unknown/None (0x9f77388)
  I-PMSI rem:  (0x0)
  MS-PMSI:   Tree-SID [524292, 192.168.0.4] (0x9fa2f98)
  Bidir-PMSI:  (0x0)
  Remote Bidir-PMSI:  (0x0)
  BSR-PMSI:  (0x0)
  A-Disc-PMSI:  (0x0)
  A-Ann-PMSI:  (0x0)
  RIB Dependency List: 0x9f81370
  Bidir RIB Dependency List: 0x0
    Sources: 1, RPs: 1, Bidir RPs: 0
```

View Data MDT Information

The commands in this section displays SR multicast tree information for *data* MDTs. The information includes cache, router-local, and remote MDT information.

View Data MDT Cache Information

```
Router# show pim vrf vpn1 mdt cache
Core Source      Cust (Source, Group)               Core Data        Expires
192.168.0.3      (26.3.233.1, 232.0.0.1)            [tree-id 524292]  never

192.168.0.4      (27.3.233.6, 232.0.0.1)            [tree-id 524290]  never

   Leaf AD:  192.168.0.3
```

View Local MDTs Information

```
Router# show pim vrf vpn1 mdt sr-p2mp local

Tree                          MDT            Cache DIP Local VRF Routes  On-demand
Identifier                    Source         Count     Entry Using Cache Color
[tree-id 524290 (0x80002)]    192.168.0.4    1     N   Y     1           10
   Tree-SID Leaf:  192.168.0.3
```

View Remote MDTs Information

```
Router # show pim vrf vpn1 mdt sr-p2mp remote

Tree                            MDT             Cache DIP Local VRF Routes  On-demand
Identifier                      Source          Count     Entry Using Cache Color
[tree-id 524290 (0x80002)]      192.168.0.4     1     N   N     1           0
```

View MRIB MPLS Forwarding Information

This command displays labels used for transporting IP multicast traffic, on a specified router.

```
Router# show mrib mpls forwarding

LSP information (XTC) :
  LSM-ID: 0x00000, Role: Head, Head LSM-ID: 0x80002
    Incoming Label       : (18000)
    Transported Protocol : <unknown>
    Explicit Null        : None
    IP lookup            : disabled

    Outsegment Info #1 [H/Push, Recursive]:
      OutLabel: 18000, NH: 192.168.0.3, Sel IF: GigabitEthernet0/2/0/0

LSP information (XTC) :
  LSM-ID: 0x00000, Role: Tail, Peek
  RPF-ID: 0x00011, Assoc-TIDs: 0xe0000011/0x0, MDT: TRmdtvpn1
    Incoming Label       : 18001
    Transported Protocol : <unknown>
    Explicit Null        : None
    IP lookup            : enabled

    Outsegment Info #1 [T/Pop]:
      No info.
```

SR-PCE Show Commands

View Tree Information On PCE Server

This command displays SR multicast tree information on the SR-PCE server.

```
Router# show pce lsp p2mp

Tree: sr_p2mp_root_192.168.0.1_tree_id_524290
 Label:    18000     Operational: up  Admin: up
 Metric Type: TE
 Transition count: 3
 Uptime: 00:00:03 (since Fri Jan 24 14:57:51 PST 2020)
 Source: 192.168.0.1
 Destinations: 192.168.0.4
 Nodes:
  Node[0]: 192.168.0.2 (rtrM)
   Role: Transit
   Hops:
    Incoming: 18000 CC-ID: 4
    Outgoing: 18000 CC-ID: 4 (17.17.17.4) [rtrR]
  Node[1]: 192.168.0.1 (rtrL1)
   Role: Ingress
   Hops:
    Incoming: 18000 CC-ID: 5
    Outgoing: 18000 CC-ID: 5 (12.12.12.2) [rtrM]
  Node[2]: 192.168.0.4 (rtrR)
   Role: Egress
   Hops:
    Incoming: 18000 CC-ID: 6
```

For dynamic SR multicast trees created for MVPN, the **show** command has filters to view root multicast router and Tree-ID information. When the root router is specified, all multicast trees from that root are displayed. When root and Tree-ID are specified, only the specified tree information is displayed.

```
Router# show pce lsp p2mp root ipv4 10.1.1.1 524289

Tree: sr_p2mp_root_10.1.1.1_tree_id_524289, Root: 10.1.1.1 ID: 524289
 Label:    20000     Operational: up  Admin: up
 PCC: 10.1.1.1

 Local LFA FRR: Disabled
 Metric Type: TE
 Transition count: 11
 Uptime: 00:03:37 (since Mon May 11 12:53:33 PDT 2020)
 Destinations: 10.1.1.3, 10.1.1.4, 10.1.1.5
 Nodes:
  Node[0]: 10.1.1.1 (root1)
   Role: Ingress
   Hops:
    Incoming: 20000 CC-ID: 26
    Outgoing: 20000 CC-ID: 26 (192.168.114.4) [mid-4]
    Outgoing: 20000 CC-ID: 26 (192.168.112.2) [mid-2]
  Node[1]: 10.1.1.4 (mid-4)
   Role: Egress
   Hops:
    Incoming: 20000 CC-ID: 27
  Node[2]: 10.1.1.2 (mid-2)
   Role: Transit
   Hops:
    Incoming: 20000 CC-ID: 28
    Outgoing: 20000 CC-ID: 28 (192.168.123.3) [leaf-3]
    Outgoing: 20000 CC-ID: 28 (192.168.125.5) [leaf-5]
  Node[3]: 10.1.1.3 (leaf-3)
   Role: Egress
   Hops:
    Incoming: 20000 CC-ID: 29
  Node[4]: 10.1.1.5 (leaf-5)
   Role: Egress
   Hops:
    Incoming: 20000 CC-ID: 30
```

The following output shows that LFA FRR is enabled on the hop from rtrR to rtrM. Unlike typical multicast replication where the address displayed is the remote address on the link to a downstream router, the IP address 192.168.0.3 (displayed with an exclamation mark) is the router-ID of the downstream router rtrM. The output also displays the LFA FRR state for the multicast tree.

```
Router# show pce lsp p2mp

Tree: sr_p2mp_root_192.168.0.4_tree_id_524290
 Label:    18000     Operational: up  Admin: up
 LFA FRR: Enabled
 Metric Type: TE
 Transition count: 1
 Uptime: 3d19h (since Thu Feb 13 13:43:40 PST 2020)
 Source: 192.168.0.4
 Destinations: 192.168.0.1, 192.168.0.2
 Nodes:
  Node[0]: 192.168.0.3 (rtrM)
   Role: Transit
   Hops:
    Incoming: 18000 CC-ID: 1
    Outgoing: 18000 CC-ID: 1 (12.12.12.1) [rtrL1]
    Outgoing: 18000 CC-ID: 1 (15.15.15.2) [rtrL2]
  Node[1]: 192.168.0.4 (rtrR)
```

```
   Role: Ingress
   Hops:
    Incoming: 18000 CC-ID: 2
    Outgoing: 18000 CC-ID: 2 (192.168.0.3!) [rtrM]
  Node[2]: 192.168.0.1 (rtrL1)
   Role: Egress
   Hops:
    Incoming: 18000 CC-ID: 3
  Node[3]: 192.168.0.2 (rtrL2)
   Role: Egress
   Hops:
    Incoming: 18000 CC-ID: 4
```

Multicast Tree Information on Routers

```
Router# show segment-routing traffic-eng p2mp policy

SR-TE P2MP policy database:
---------------------
! - Replications with Fast Re-route

Policy: sr_p2mp_root_192.168.0.1_tree_id_524290  LSM-ID: 0x2
 Role: Leaf
 Replication:
  Incoming label: 18001 CC-ID: 6

Policy: sr_p2mp_root_192.168.0.4_tree_id_524290  LSM-ID: 0x80002 (PCC-initiated)
 Color: 0
 LFA FRR: Disabled
 Role: Root
 Replication:
  Incoming label: 18000 CC-ID: 2
  Interface: None [192.168.0.3!]  Outgoing label: 18000 CC-ID: 2
 Endpoints:
  192.168.0.1, 192.168.0.2
```

For SR multicast policies originated locally on the router (root router of a dynamic MVPN multicast policy) additional policy information is displayed. The information includes color, end points, and whether LFA FRR is requested by the local application. When the SR-PCE server enables LFA FRR on a specific hop, the outgoing information shows the address of the next router with an exclamation mark and None is displayed for the outgoing interface.

For dynamic SR multicast trees created for MVPN, the **show** command has filters for displaying root multicast router and Tree-ID information. When the root router is specified, all multicast trees for that root are displayed. When root and Tree-ID are specified, only the specified tree information is displayed.

```
Router# show segment-routing traffic-eng p2mp policy root ipv4 1.1$

SR-TE P2MP policy database:
---------------------
! - Replications with Fast Re-route, * - Stale dynamic policies/endpoints

Policy: sr_p2mp_root_10.1.1.1_tree_id_524289  LSM-ID: 0x691
 Root: 10.1.1.1, ID: 524289
 Role: Transit
 Replication:
  Incoming label: 20000 CC-ID: 28
  Interface: Bundle-Ether23 [192.168.123.3]  Outgoing label: 20000 CC-ID: 28
  Interface: Bundle-Ether25 [192.168.125.5]  Outgoing label: 20000 CC-ID: 28

Policy: sr_p2mp_root_10.1.1.1_tree_id_524290  LSM-ID: 0x692
 Root: 10.1.1.1, ID: 524290
 Role: Transit
 Replication:
```

```
Incoming label: 19999 CC-ID: 28
Interface: Bundle-Ether23 [192.168.123.3]  Outgoing label: 19999 CC-ID: 28
Interface: Bundle-Ether25 [192.168.125.5]  Outgoing label: 19999 CC-ID: 28
```

Multicast: SR-PCE High Availability (HA) Support for Dynamic Tree-SID (mVPN)

Table 2: Feature History Table

Feature Name	Release	Description
High Availability Support for Dynamic Tree-SID (Multicast VPN)	Release 7.8.1	We have introduced more resilience for building multicast VPN (mVPN) dynamic tree-SIDs by providing High Availability (HA) for the Segment Routing Path Computation Element (SR-PCE). This HA is made possible by adding another SR-PCE to the network. As a result, there's a noncompute or standby PCE for the mVPN dynamic policies. The root Path Computation Element Client (PCC) elects the active SR-PCE. If an active PCE failure occurs, the root PCC delegates the compute role for the mVPN dynamic Tree-SID to the standby SR-PCE.

Segment Routing Point-to-Multipoint policy (SR-P2MP) in Tree-SID is the solution for carrying multicast traffic in the Segment Routing Domain but it works in the presence of just one SR-PCE in the network. However, the SR-PCE HA feature supports the mVPN dynamic Tree-SID with more than one SR-PCE to manage the network.

For example, when PCE1 is unavailable due to a system failure or reboot, PCE2 uses the PCReport packet information sent by PCC and assumes the role of PCE1 ensuring the following:

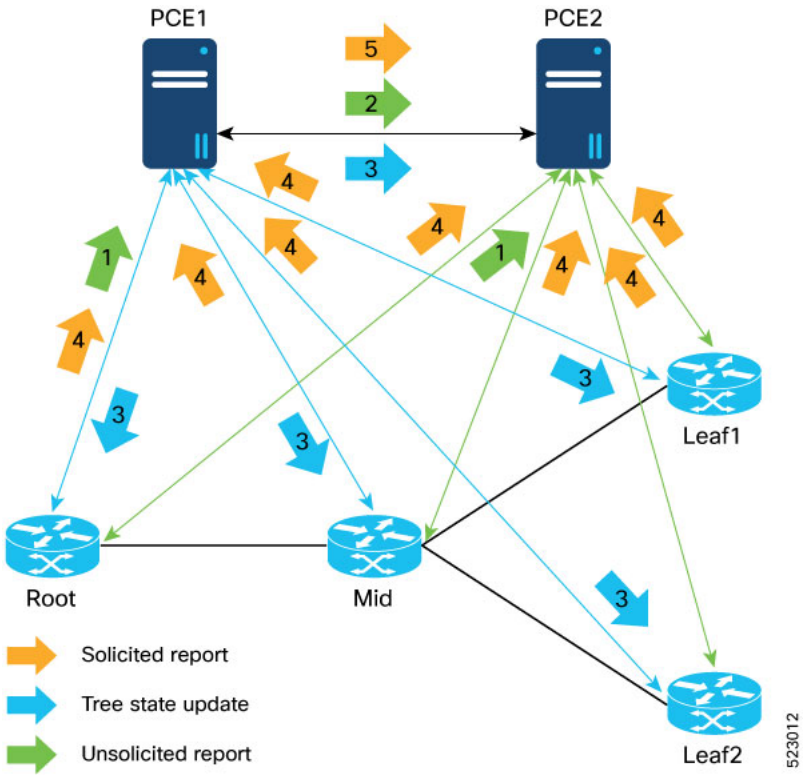
- Avoid failure of the cluster with no or minimal data loss.
- PCE2 is a hot-standby PCE that detects the failure as they occur ensuring high availability of the cluster always.
- Recovery of the network occurs with minimal or with no data loss.

Network Handling

Understanding how each PCE operates in different states helps in configuring the SR-PCE HA and ensuring steady operability without any data loss. Following sections describe each state:

- **Steady State**

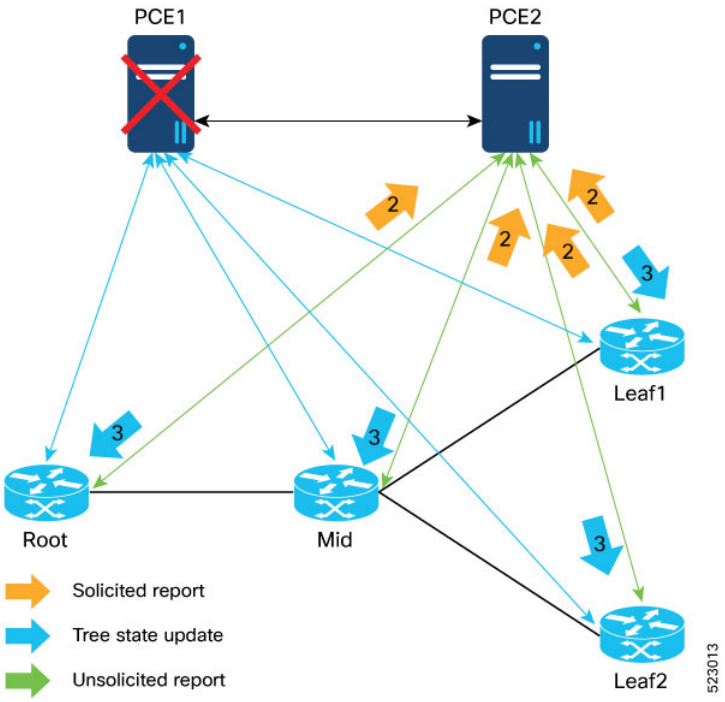
  In steady state, the following events occur:



1. Root request SR-P2MP tree creation:
   - Delegates to PCE1
   - Sends the PCReport to PCE1 with **D-bit** set to **1**
   - Sends the PCReport to PCE2 with **D-bit** set to **0**
2. PCE1 forwards the PCReport to PCE2.
3. PCE1 acts as the compute PCE:
   - Sends PCInitiate to the Mid and Leaf nodes
   - Sends PCUpdate to Root
   - Syncs Tree State for all the nodes with state-sync PCE2
4. All PCCs respond with PCReport:
   - With **D-bit** set to **1** to delegated “Creator” PCE (PCE1)
   - With **D-bit** set to **0** to the other PCE2
5. PCE1 forwards all the reports with D-bit set from PCCs to PCE2.

- **PCE Failure**

  When PCE fails, the following events occur:

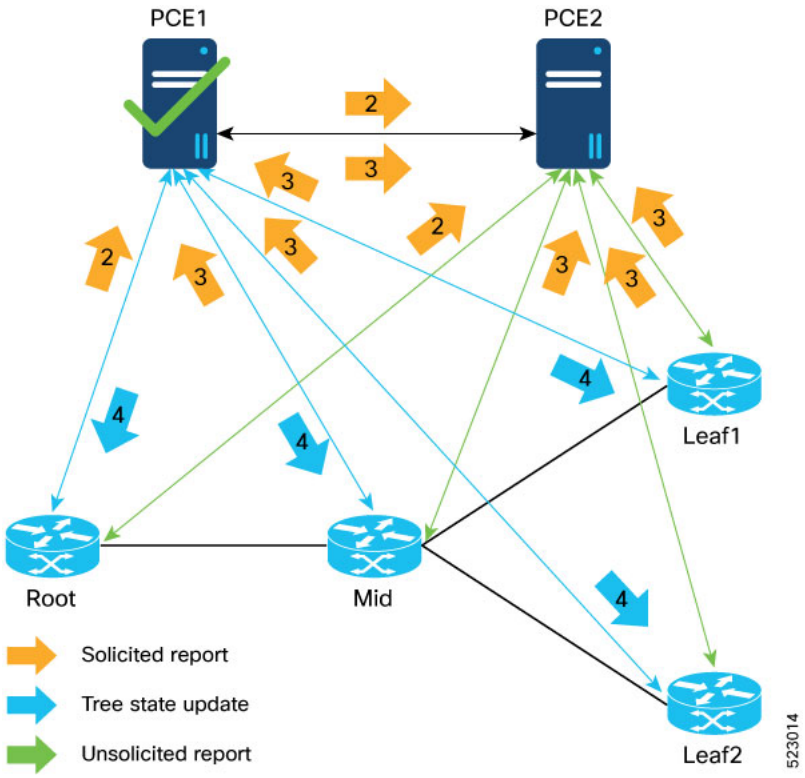


1. PCE1 fails.
2. Root re-delegate to PCE2 immediately and sends the PCReport with **D-bit** Set to **1**
    - With the PCC-centric approach, Mid and Leaf nodes PCCs also re-delegate to PCE2 immediately and sends the PCReport with **D-bit** set to **1**
3. PCE2 becomes the Compute PCE, recomputes Tree-SID and sends the update to PCCs.

- **PCE Restore**

  Image

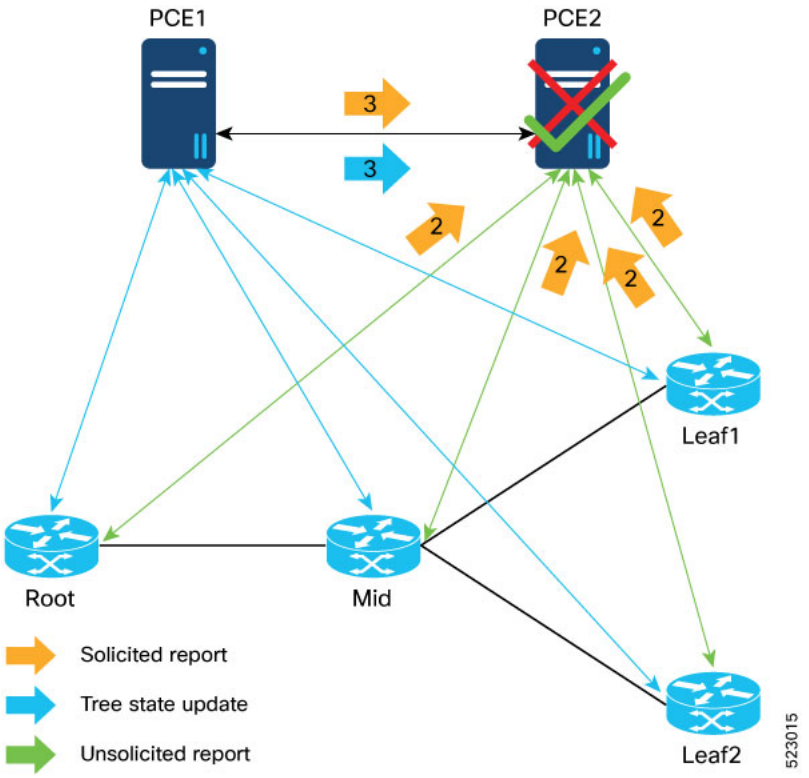
  When PCE restores, the following events occur:



1. PCE1 is restored.
2. Root redelegates to PCE1 after the delegation timer expires:
   - Sends the PCReport to PCE1 with **D-bit** set to **1**
   - Sends the PCReport to PCE2 with **D-bit** set to **0**
3. With the PCC-Centric approach, Mid and Leaf nodes PCCs also re-delegate to PCE1 after the Delegation timer expires.
   - Sends the PCReport to PCE2 with **D-bit** set to **0**
4. PCE1 becomes the compute PCE and recomputes the Tree-SID.
   - Sends PCUpdate to PCCs participating in Tree-SID creation

- **Redundant or Backup PCE Down or Up Event**

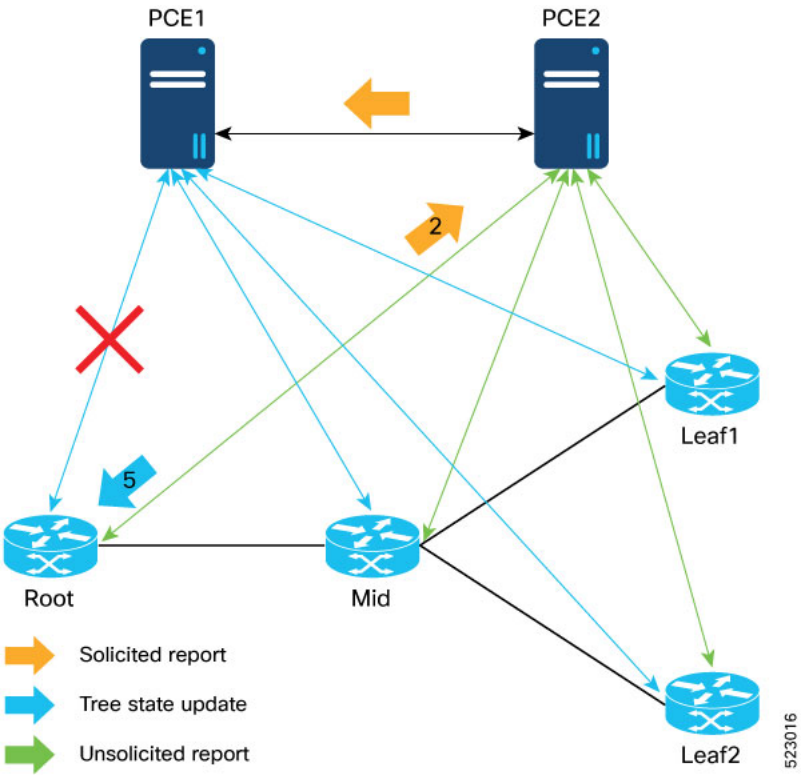
  When the redundant or the backup PCE is down or up, the following events occur:



1. When the backup PCE fails or is down, it is a "No-Operation" event from the Tree's point of view.
2. When the backup PCE is back up then all the PCCs resend the PCReports with **D-bit** set to **0**.
3. PCE1 syncs all the delegated reports and locally computed Tree states with PCE2.

- **PCC Initiated PCEP Session with the Root is Down**

  When the PCC initiated PCEP session with the Root is down, the following events occur:

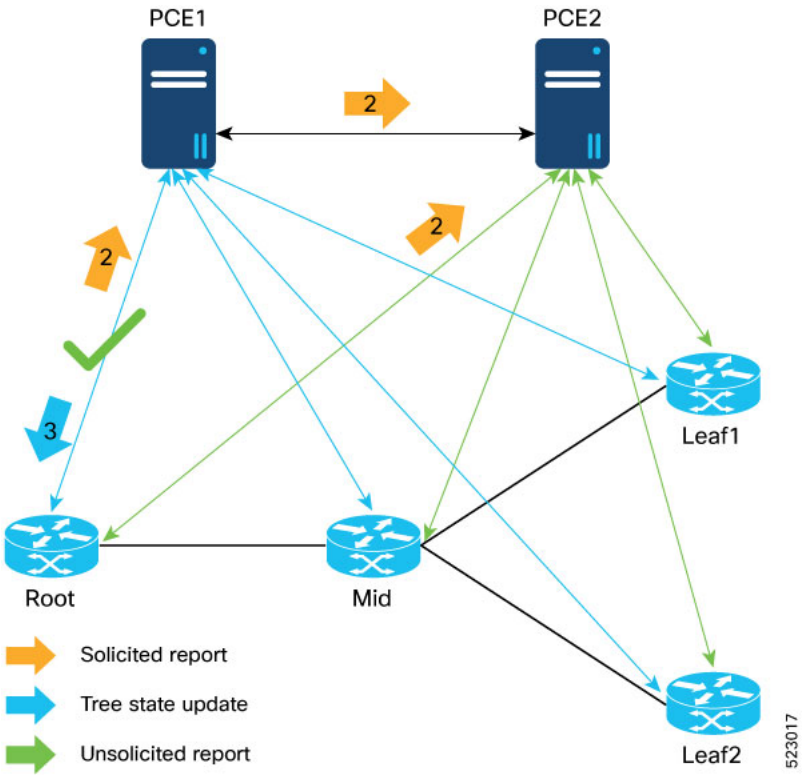


1. PCEP session from the Root to PCE1 is down but the Mid and Leaf nodes continue to have the session with PCE1.
2. Root redelegates to PCE2 immediately and sends the PCReport with **D-bit** set to **1**.
3. PCE2 takes the responsibility of the compute PCE, recomputes Tree, and sends the PCUpdate.

Note Mid or leaf nodes are still delegated to PCE1, any updates to these nodes are done through PCE1.

- **PCC Initiated PCEP Session with the Root is Restored**

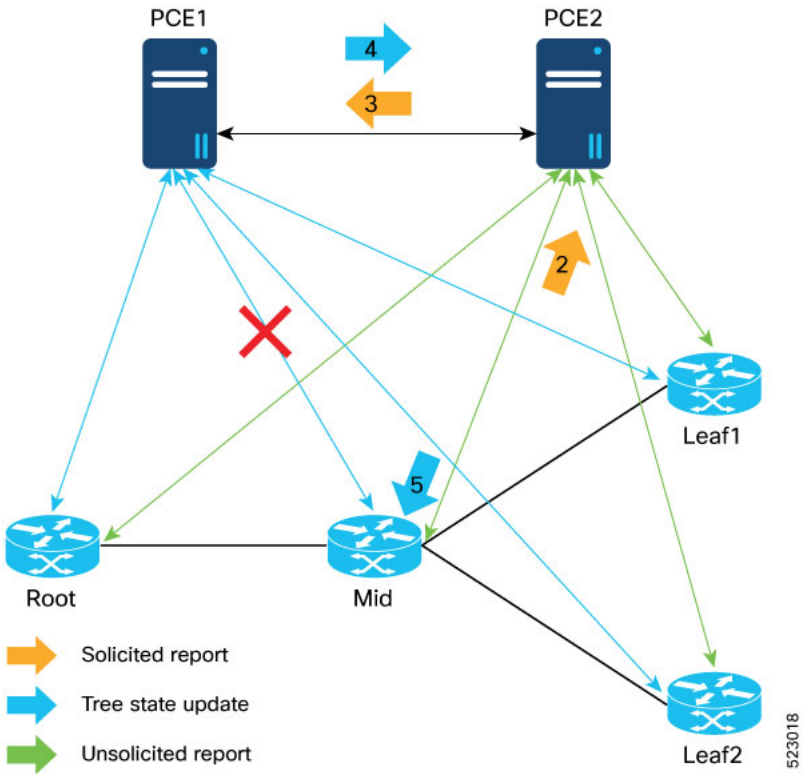
  When the PCC initiated PCEP session with the Root is back up, the following events occur:



1. PCEP session from the Root to PCE1 is restored.
2. Root redelegates to PCE1 immediately after the delegation timer expires:
   - Sends the PCReport to PCE1 with **D-bit** set to **1**
   - Sends the PCReport to PCE2 with **D-bit** set to **0**
3. PCE1 reclaims the responsibility of the compute PCE, recomputes Tree, and sends the PCUpdate.

- **PCC Initiated - PCEP session with Mid or Leaf node is Down**

  When the PCC initiated PCEP session with the Mid or Leaf node is down, the following events occur:



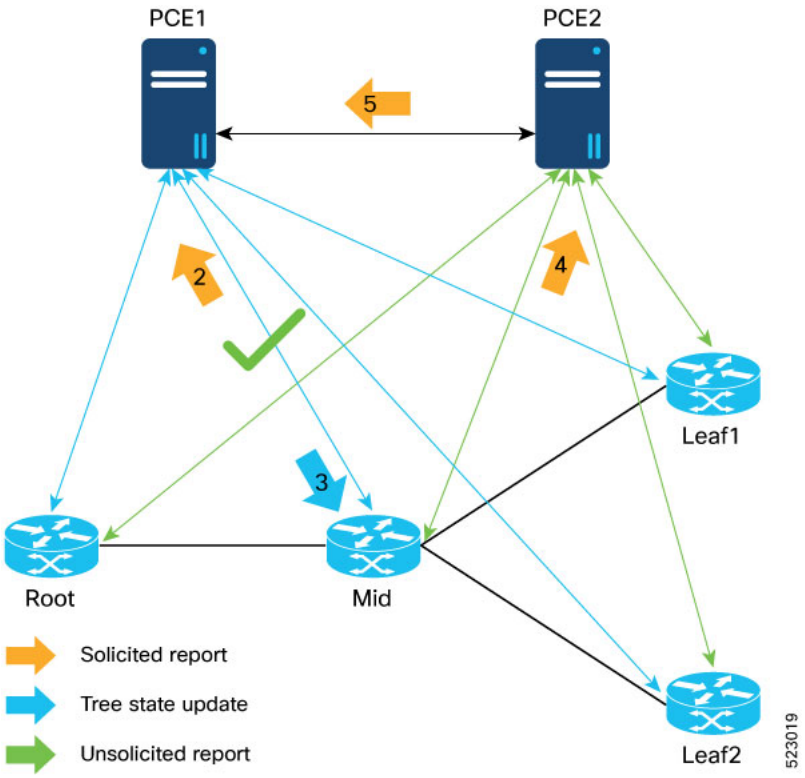
1. PCEP session from Mid to PCE1 is down but the Root and Leaf node still has session to PCE1
   - PCE1 is still responsible for Tree compute.
   - PCE1 holds LSP state for Mid for sometime to allow redelegation and Report from PCE2 (60 seconds Tree-SID Peer Down Timer on PCE).
2. With PCC-Centric approach, Mid redelegates immediately to PCE2 and sends a PCReport with **D-bit** set to **1**.
3. PCE2 forwards PCReport to PCE1.
4. PCE1 sends PCUpdate to PCE2 for Mid immediately.
5. PCE2 always forwards the PCUpdate to Mid (PCC) if the PCE2 still has delegation of LSP from the PCC.

Note A node, which does not have PCEP session is considered for P2PM path compute for new Tree.

In this example, a new Tree from the Root to the same set of Leaf nodes cannot be brought up because PCE1 does not have a PCEP session to the Mid node.

- **PCC Initiated - PCEP session with Mid or Leaf node is Restored**

  When the PCC initiated PCEP session with the Mid or Leaf node is restored, the following events occur:



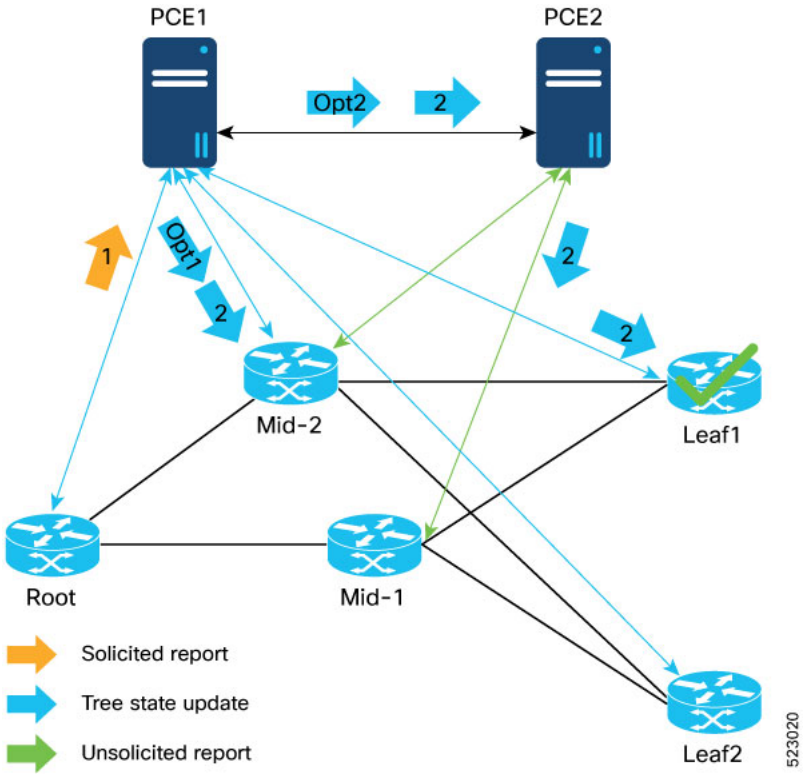
1. PCEP session from Mid to PCE1 is restored.
   - Root and Leaf node still have the session to PCE1.
   - PCE1 is still responsible for path compute.
2. With PCC-centric approach, Mid node redelegates to PCE1 after the delegation timer expires.
   - Sends a PCReport with **D-bit** set to **1**.
3. PCE1 sends PCUpdate to Mid node.
4. Mid node sends a PCReport to PCE2 with **D-bit** set to **0**.
5. PCE2 withdraws "Interest" from PCE1.

Note Even after the session between PCE1 and Mid node is restored, PCE1 keeps pushing updates to PCE2 until the "interest" from LSP is withdrawn. The PCE to which the LSP is delegated push the update down to PCC.

It is possible that both OCEs have the **D-bit** set for the LSP momentarily. In such a case, both PCEs will push down the Updates. The PCC accepts the update from the PCE to which it has delegated to.

- **PCC Initiated - Existing Tree Change with Split PCEP Session**

When there is a PCC Initiated - Existing Tree Change with Split PCEP Session, the following events occur:

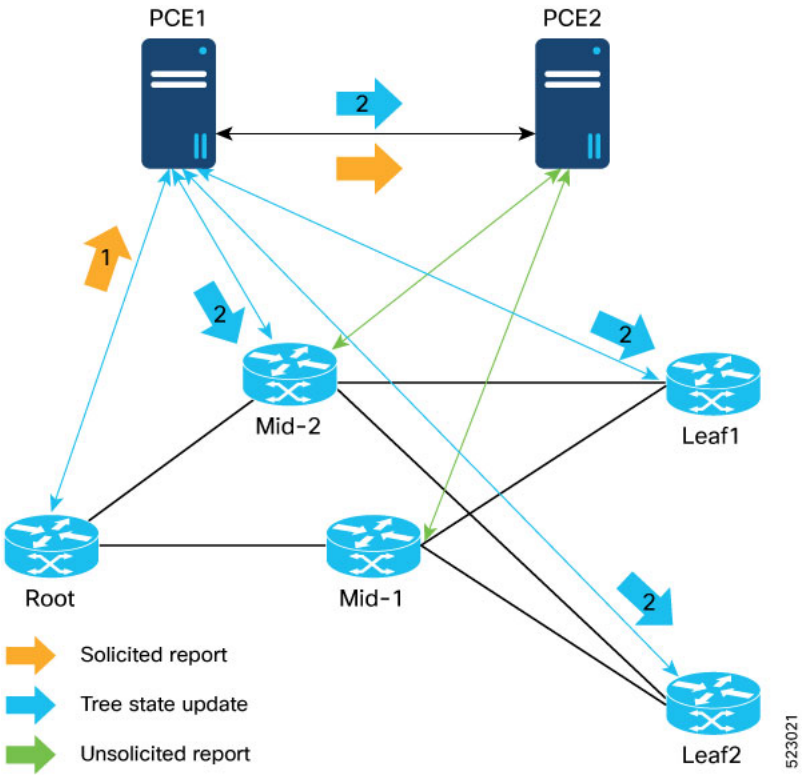


Root and Leaf node have PCEP session with PCE1 and the Mid-1 node has PCEP session restored and hence delegated to PCE2. The new mid node (Mid2) is introduced with PCEP sessions to both PCEs.

1. Root updates Tree to add Leaf1
    - Sends PCReport to PCE1 with **D-bit**set
2. PCE1 computes Tree
    - **Option 1:** PCE1 will not consider Mid-1 node because it does not have a direct PCEP session to it.
        - Sends PCInitiate to Mid-2 and Leaf1 node PCCs
        - **Cons:** This may not be the best path to Endpoint. The path to a given Endpoint may be different if it gets deleted and re-added.
    - **Option 2 (Preferred):** PCE1 will consider reachability through Mid-1 node because it knows Mid-1 node is part of the Tree and delegated to PCE2 (Over State-Sync channel)
        - Sends PCUpdate to Mid-1 node through PCE2

- **PCC Initiated - New Tree with Split PCEP Session**

  When there is a PCC Initiated - New Tree Change with Split PCEP Session, the following events occur:



Root, Mid-2, and Leaf nodes have PCEP session with PCE1 and the Mid-1 node has PCEP session UP with PCE2.

1. Root creates a new Tree to Leaf1 and Leaf2
   - Sends PCReport to PCE1 with **D-bit** set
2. PCE1 computes Tree
   - PCE1 will not consider Mid-1 node because it does not have a direct PCEP session to it
   - Sends PCInitiate to Mid-2 and Leaf node PCCs

Note Con: Another (Existing) tree to the same Endpoints may be programmed through Mid-1 node.

Limitations and Guidelines

Thi section lists the limitation and guidelines:

- The IOS-XR forwarding devices and the SR-PCE must be upgraded to IOSXR 7.8.1 release to enable this feature in the network.
- No support for PCE HA support for CLI-configured static TreeSID.

Configuration Steps

Captured below are the configuration steps required to set up the TreeSID PCE HA feature.

1. **Configuration on forwarding device**

   This section guides you to configure the forwarding device.

   a. **PCE configuration on a PCC**

   The following configuration is required on the PCC routers to configure the PCE's that will provide the redundancy.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# pcc
Router(config-pcc)# pce address ipv4 2.2.2.1
Router(config-pcc)# precedence 200
Router(config-pcc)# pce-group pce-ha-group
Router(config-sr-te-pce)# pce address ipv4 4.4.4.2
Router(config-pce)# precedence 0
Router(config-pce)# pce-group pce-ha-group
```

Note To elect a Compute PCE, you must set the precedence. In the example above there are 2 PCEs configured with 200 and 0. The PCE with a lower precedence takes up the compute role.

   b. **Recommended steps for costing in a new SR-PCE in the network**

   If the network needs to be upgraded with a new SR-PCE, the operators can add the new SR-PCE as a more preferred PCE in the above configuration on the root of the SR-P2MP tree. Doing so results in the PCC re-delegating all the SR-P2MP LSPs to the newly configured SR-PCE. This delegation allows the new SR-PCE to assume the role of computation for the SR-P2MP trees. The older SR-PCE can be costed out of the network. Following these steps will allow the transition from one SR-PCE to another.

   c. **TreeSID with PCE groups**

   - **Associating a PCE with a PCE group:** A PCE can be associated to a PCE group using the below configuration. It must be noted that a PCE can only be associated to one PCE group

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# pcc
Router(config-pcc-pce)# pce address ipv4 192.168.0.5
Router(config-pcc-pce)# pce-group test
```

   - **Associating a PCE group with on-demand color:** With a PCE now associated with a PCE group, the PCE group can be associated with an on-demand color (which you later associate with a P2MP policy) using the following configuration.

Note Note: While the same PCE group can be used across many on-demand colors, there can only be one PCE group associated with one on-demand color configuration.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te-color)on-demand color 10
Router(config-sr-te-color)pce-group test
```

- **Associating a dynamic MVPN policy with color:** A dynamic SR P2MP policy is associated with an on-demand color, which provides an abstraction to the constraints or metrics that the policy must satisfy. The same structure will be used to associate a PCE group to the dynamic SR P2MP policy.

```
Default TreeSID
Router(config)# vrf vpn1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# bgp auto-discovery segment-routing
Router(config-vrf-af)# mdt default segment-routing mpls color 10
Router(config-vrf-af)# mdt data segment-routing mpls 10 route-policy treesid
threshold 0

Router(config)# route-policy treesid
Router(config)# set on-demand-color 20
Router(config)# end-policy

Partitioned TreeSID
Router(config)# vrf vpn2
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# bgp auto-discovery segment-routing
Router(config-vrf-af)# mdt partitioned segment-routing mpls color 10
```

2. Configuration on SR-PCE

Note **PCE state-sync configuration:** The following configuration must be done on all PCEs participating in PCE state-sync. Configuring it on only one PCE will only enable that PCE to sync state uni-directionally.

```
Router(config)# pce state-sync ipv4 192.168.0.6
```

Running Configuration

Run the show command to review the running configuration:

```
pce
====
 address ipv4 192.168.0.5
 api
  user admin
   password encrypted 094D4A04100B464058
  !
  authentication basic
```

```
 !
 state-sync ipv4 192.168.0.6
 segment-routing
  traffic-eng
   p2mp
    timers reoptimization 60
    frr-node-set from
     ipv4 192.168.0.2
    !
    frr-node-set to
     ipv4 192.168.0.1
     ipv4 192.168.0.3
    !
    label-range min 18000 max 19000
    multipath-disable
   !
   affinity bit-map
    red 1
    blue 3
    black 31
    green 2
   !
  !
 !
!
PCC
====
segment-routing
 traffic-eng
  pcc
   pce address ipv4 192.168.0.5
   !
   pce address ipv4 192.168.0.6
   !
   redundancy pcc-centric
   timers initiated state 60
   timers initiated orphan 30
  !
 !
!
```

Verification

Run the following show commands to verify if the PCE compute role is set:

- This is an example of the command run on a Compute SR-PCE:

```
Router# Show pce lsp ipv4 p2mp
Tree: sr_p2mp_root_192.168.0.4_tree_id_524289, Root: 192.168.0.4 ID: 524289
 PCC: 192.168.0.4
 Label:    19000
 Operational: up  Admin: up  Compute: Yes
 Local LFA FRR: Disabled
 Metric Type: IGP
 Transition count: 1
 Uptime: 00:21:37 (since Fri Mar 11 18:36:06 PST 2022)
 Destinations: 192.168.0.1, 192.168.0.3
 Nodes:
  Node[0]: 192.168.0.2+ (rtrM)
   Delegation: PCC
   PLSP-ID: 1
   Role: Transit
```

```
    State Changes: 0x2 (New Hops)
    Endpoints: 192.168.0.3 192.168.0.1
    Hops:
     Incoming: 19000 CC-ID: 1
     Outgoing: 19000 CC-ID: 1 (13.13.13.3) [rtrL2:192.168.0.3]
       Endpoints: 192.168.0.3
     Outgoing: 19000 CC-ID: 1 (10.10.10.1) [rtrL1:192.168.0.1]
       Endpoints: 192.168.0.1
  Node[1]: 192.168.0.4 (rtrR)
    Delegation: PCC
    Locally computed
    PLSP-ID: 1
    Role: Ingress
    Endpoints: 192.168.0.3 192.168.0.1
    Hops:
     Incoming: 19000 CC-ID: 2
     Outgoing: 19000 CC-ID: 2 (16.16.16.2) [rtrM:192.168.0.2]
       Endpoints: 192.168.0.3 192.168.0.1
  Node[2]: 192.168.0.3 (rtrL2)
    Delegation: PCC
    Locally computed
    PLSP-ID: 1
    Role: Egress
    Hops:
     Incoming: 19000 CC-ID: 4
Node[3]: 192.168.0.1+ (rtrL1)
    Delegation: PCC
    Locally computed
    PLSP-ID: 2
    Role: Egress
    State Changes: 0x7 (New Node,New Hops,Role Change)
    Hops:
     Incoming: 19000 CC-ID: 5


Event history (latest first):
   Time                    Event
  Mar 11 18:57:12.522    Received report from all nodes awaiting report, state: Pruning
 stale legs on non-root nodes
  Mar 11 18:57:12.522    No nodes awaiting report, state: Pruning stale legs on non-root
 nodes
   Mar 11 18:57:12.522     Received report from all nodes awaiting report, state:
Programming the root node
   Mar 11 18:57:12.522     No nodes awaiting report, state: Programming the root node
   Mar 11 18:57:12.522     Received report from all nodes awaiting report, state:
Programming non-root nodes
   Mar 11 18:57:12.512     Node 192.168.0.1 delegated by PCC
   Mar 11 18:57:12.473     Path computation returned a different result, signaling new
path
  Mar 11 18:57:12.473    Received report from all nodes awaiting report, state: Pruning
 stale legs on non-root nodes
   Mar 11 18:57:12.472     TreeSID Leaf set changed
   Mar 11 18:51:10.146     Node 192.168.0.1 undelegated by PCC
   Mar 11 18:51:10.080     Received report from all nodes awaiting report, state:
Programming the root node
   Mar 11 18:51:10.080     No nodes awaiting report, state: Programming the root node
   Mar 11 18:51:10.080     Received report from all nodes awaiting report, state:
Programming non-root nodes
   Mar 11 18:51:10.080     No nodes awaiting report, state: Programming non-root nodes
   Mar 11 18:51:10.080     Path computation returned a different result, signaling new
path
   Mar 11 18:51:10.080     Received report from all nodes awaiting report, state: None
   Mar 11 18:51:10.080     TreeSID Leaf set changed
  Mar 11 18:36:06.813    Received report from all nodes awaiting report, state: Pruning
```

```
 stale legs on non-root nodes
  Mar 11 18:36:06.813     No nodes awaiting report, state: Pruning stale legs on non-root
 nodes
  Mar 11 18:36:06.813     Received report from all nodes awaiting report, state:
Programming the root node
  Mar 11 18:36:06.813     No nodes awaiting report, state: Programming the root node
  Mar 11 18:36:06.813     Received report from all nodes awaiting report, state:
Programming non-root nodes
  Mar 11 18:36:06.552     Node 192.168.0.3 delegated by PCC
  Mar 11 18:36:06.534     Path computation returned a different result, signaling new
path
  Mar 11 18:36:06.534     Received report from all nodes awaiting report, state: Pruning
 stale legs on non-root nodes
  Mar 11 18:36:06.534     No nodes awaiting report, state: Pruning stale legs on non-root
 nodes
  Mar 11 18:36:06.534     Operational state changed to up (0 transitions)
  Mar 11 18:36:06.534     Received report from all nodes awaiting report, state:
Programming the root node
  Mar 11 18:36:06.323     Node 192.168.0.4 delegated by PCC
  Mar 11 18:36:06.323     TreeSID Leaf set changed
  Mar 11 18:36:06.316     Received report from all nodes awaiting report, state:
Programming non-root nodes
  Mar 11 18:36:06.316     Node 192.168.0.1 delegated by PCC
  Mar 11 18:36:06.298     Node 192.168.0.2 delegated by PCC
  Mar 11 18:36:06.249     PCE compute role set
  Mar 11 18:36:06.249     TreeSID metric type changed to 0
  Mar 11 18:36:06.249     TreeSID Leaf set changed
  Mar 11 18:36:06.249     TreeSID created
```

- This is an example of the command run on a Non-Compute SR-PCE:

```
Router# Show pce lsp ipv4 p2mp
Tree: sr_p2mp_root_192.168.0.4_tree_id_524289, Root: 192.168.0.4 ID: 524289
 PCC: 192.168.0.4
 Label:    19000
 Operational: standby  Admin: up  Compute: No
 Local LFA FRR: Enabled
 Metric Type: IGP
 Transition count: 0
 Destinations: 192.168.0.1, 192.168.0.3
 Nodes:
  Node[0]: 192.168.0.4+ (rtrR)
   Delegation: PCE 192.168.0.5
   PLSP-ID: 1
   Role: None
   State Changes: 0x3 (New Node,New Hops)
   Hops:
    Incoming: 19000 CC-ID: 2
    Outgoing: 19000 CC-ID: 2 (16.16.16.2)
  Node[1]: 192.168.0.2+ (rtrM)
   Delegation: PCE 192.168.0.5
   PLSP-ID: 1
   Role: None
   State Changes: 0x3 (New Node,New Hops)
   Hops:
    Incoming: 19000 CC-ID: 1
    Outgoing: 19000 CC-ID: 1 (13.13.13.3)
    Outgoing: 19000 CC-ID: 1 (10.10.10.1)
  Node[2]: 192.168.0.3+ (rtrL2)
   Delegation: PCE 192.168.0.5
   PLSP-ID: 1
   Role: None
   State Changes: 0x3 (New Node,New Hops)
   Hops:
```

```
    Incoming: 19000 CC-ID: 4
  Node[3]: 192.168.0.1+ (rtrL1)
   Delegation: PCE 192.168.0.5
   PLSP-ID: 2
   Role: None
   State Changes: 0x3 (New Node,New Hops)
   Hops:
    Incoming: 19000 CC-ID: 5

 Event history (latest first):
  Time                    Event
  Mar 11 18:57:12.688     Node 192.168.0.1 delegated by PCE 192.168.0.5
  Mar 11 18:57:12.485     TreeSID Leaf set changed
  Mar 11 18:51:10.082     TreeSID Leaf set changed
  Mar 11 18:36:06.713     Node 192.168.0.3 delegated by PCE 192.168.0.5
  Mar 11 18:36:06.499     TreeSID Leaf set changed
  Mar 11 18:36:06.499     Node 192.168.0.1 delegated by PCE 192.168.0.5
  Mar 11 18:36:06.499     Node 192.168.0.2 delegated by PCE 192.168.0.5
  Mar 11 18:36:06.291     Node 192.168.0.4 delegated by PCE 192.168.0.5
  Mar 11 18:36:06.291     TreeSID metric type changed to 0
  Mar 11 18:36:06.291     TreeSID Leaf set changed
  Mar 11 18:36:06.291     TreeSID created
```

- This is an example of the show command to verify the policy information in PCC:

```
Router# show segment-routing traffic-eng p2mp
Policy: sr_p2mp_root_192.168.0.4_tree_id_524290  LSM-ID: 0x40002
 Root: 192.168.0.4, ID: 524290
 PCE stale timer: Running  Start: Dec 31 16:22:06.847
 PCE Group: NULL
 PCC info:
     Symbolic name: sr_p2mp_root_192.168.0.4_tree_id_524290
     Creator PCE: 192.168.0.5
     Delegator PCE: 192.168.0.5
     PLSP-ID: 2
     Is orphan: no
     State timer:
       Running: no
 Delegated Connection: 192.168.0.5
 Creator Connnection: 192.168.0.5
 Role: Transit
 Tree label: Unlabelled
 Head LSM-ID label: Unlabelled
 Replication:
  Event history (latest first):
   Time                    Event
   Jan 25 15:57:20.848     Updated delegator addr: 192.168.0.5 in pcc_info
  Jan 25 15:39:34.019     Forwarding updated: LBL RW ADD LBL: 18999 TBL-ID: 0xe0000000
flags: 0x0 LSM-ID: 0x40002 || OUTINFO ADD LBL: 18999 -> 18999 IFH: 0x0 addr: 192.168.0.1
|| LMRIB FLUSH
   Jan 25 15:39:34.019     TreeSID created
```

Multicast: SR-PCE High Availability Support for Static Tree-SID

Table 3: Feature History Table

Feature Name	Release	Feature Description
Multicast: SR-PCE High Availability Support for Static Tree-SID	Release 7.9.1	This feature provides High Availability (HA) capability for static Tree-SID by using more than one Segment Routing Path Computation Elements (SR-PCE) in a multicast network. The feature helps in computing reliable paths for large and complex networks.

For carrying multicast traffic in the Segment Routing domain, the current available solution is the Segment Routing Point-to-Multipoint policy Tree (SR-P2MP) also known as the Tree-SID. In the Tree-SID implementation, one type is the Static Tree-SID and the other is the Dynamic Tree-SID, which is explained in the previous section.

The Tree-SID categorization is based on the way the tree root, mid, and the leaf switches are learnt in the network. In the Static Tree-SID, the trees are created by the network operator. The network operator statically defines the SR-P2MP policy (root and the leaf set) on the Path Computation Element (PCE). The SID value is also statically defined by the network operator.

The SR-PCE HA support for Static Tree-SID feature enables multiple SR-PCE instances to work together to provide reliable path computation for the network. In case of a disruption or failure of the primary SR-PCE, it provides a failover mechanism, ensuring that the multicast traffic is not disrupted.

This feature is useful in large and complex networks where reliable path computation function is critical.

Compute PCE Selection

The SR-PCE high availability (HA) support requires the network to have one SR-PCE assuming the role of a compute per SR-P2MP tree. This SR-PCE, referred to as the compute PCE, assumes the responsibility of computing the SR-P2MP tree and instantiating the tree on the participating Path Computation Element Clients (PCC).

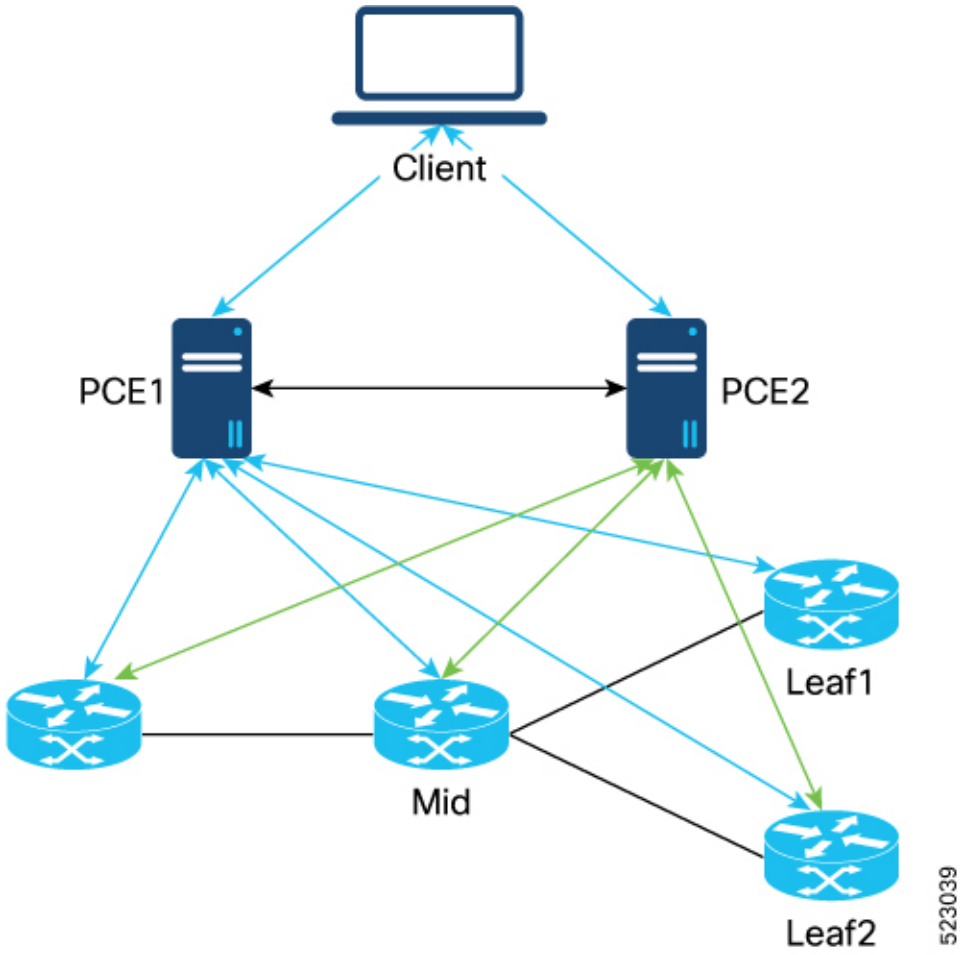
The SR-PCE computes an SR-P2MP tree, based on its current view of the topology from the Root to the interested endpoints. When there is more than one SR-PCE available to manage a network, ensure that only one SR-PCE takes the role of computing SR-P2MP tree.

The SR-PCEs that manage the SR-P2MP trees, should be in sync with each other with respect to the state of the SR-P2MP tree. The SR-PCEs exchange and synchronize the delegated SR-P2MP LSP state with each other over the PCEP state-sync channel. Keeping the state-sync peers in a hot-standby state allows for a smooth transition if the computing PCE goes down. The SR-PCE taking over the compute maintains the current forwarding state on the PCCs until it recomputes the tree.

In the event of a network failure, the compute PCE gets the information about which SR-PCE the PCC is delegated to, from the PCReports.

Rest-Initiated Static TreeSID

The following figure shows an example of an SR network topology managed by two PCEs.



In this topology, all nodes participating in the SR-P2MP forwarding have a PCEP session with all SR-PCEs responsible for managing the tree. Additionally, the SR-PCEs also establish and maintain a PCEP session among themselves. This PCEP session between the PCEs is referred to as the PCE state-sync channel.

For REST initiated TreeSID, when a Crosswork Optimization Engine (COE) client sends a policy creation request, SR-PCE which receives the request acts as a compute PCE, creates a tree, and synchronizes the request with the peer PCE through the PCEP-sync channel.

The SR-PCEs should have a consistent view of the topology to allow for hitless switchover in the event of PCEP session failures between the PCC and the SR-PCEs.

For client-initiated TreeSID, the tree is created after receiving a request from Client at the SR-PCE. This install request contains information about the root, interested endpoints, and the constraints to be satisfied for the tree.

When a request is received on SR-PCE, that PCE is set as a compute PCE and the create request is forwarded to the peer PCE through the PCEP sync channel.

PCE State-Sync Session

The PCE state-sync channel is a PCEP session established between two SR-PCEs to synchronize the LSP state. This state-sync channel is used for the following purposes:

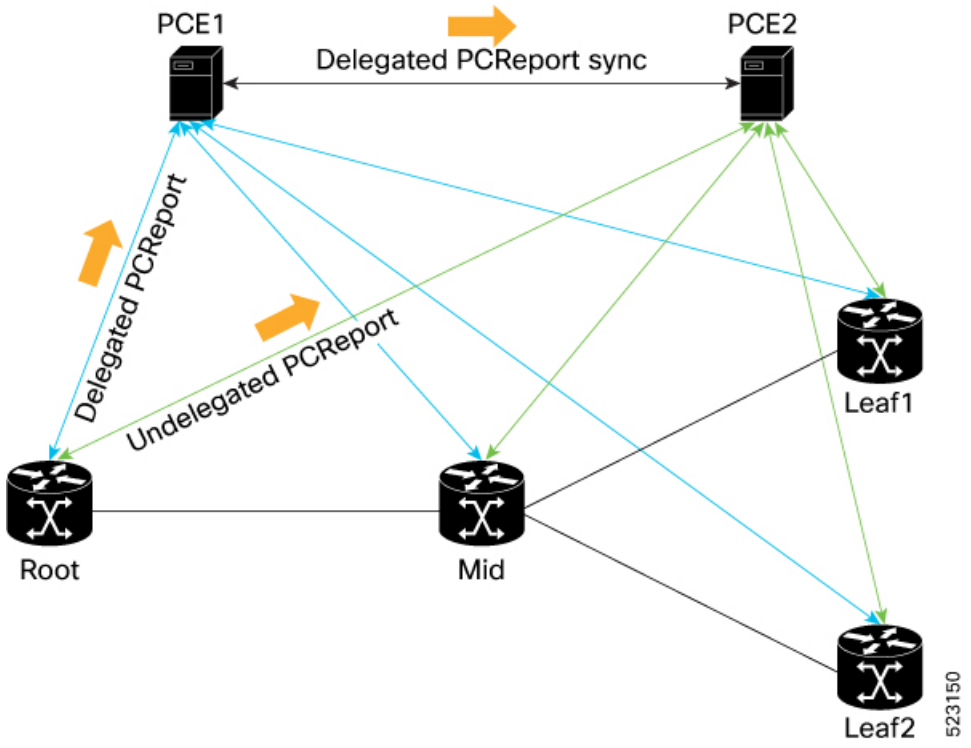
- Synchronizing delegated PCReport messages
- Proxy-forwarding P2MP Tree updates to PCCs through PCInitiate and PCUpdate messages

You should configure the state-sync channel on all SR-PCEs participating in PCE state-sync. TreeSID PCE HA is supported only over IPv4 PCEP state-sync sessions.

When the state-sync session is configured, the SR-PCE establishes a PCEP session with the peer mentioned in the configuration and performs an initial sync for SR-P2MP tree state. This process involves synchronizing all delegated SR-P2MP LSPs with the peer PCE.

PCReport State-Sync

A delegated PCReport shows that the LSP is delegated to a particular SR-PCE. For TreeSID PCE-HA, all such delegated reports are forwarded over the state-sync channel. The PCReport is forwarded to let the state-sync SR-PCEs know which SR-PCE a PCC has delegated the LSP to.

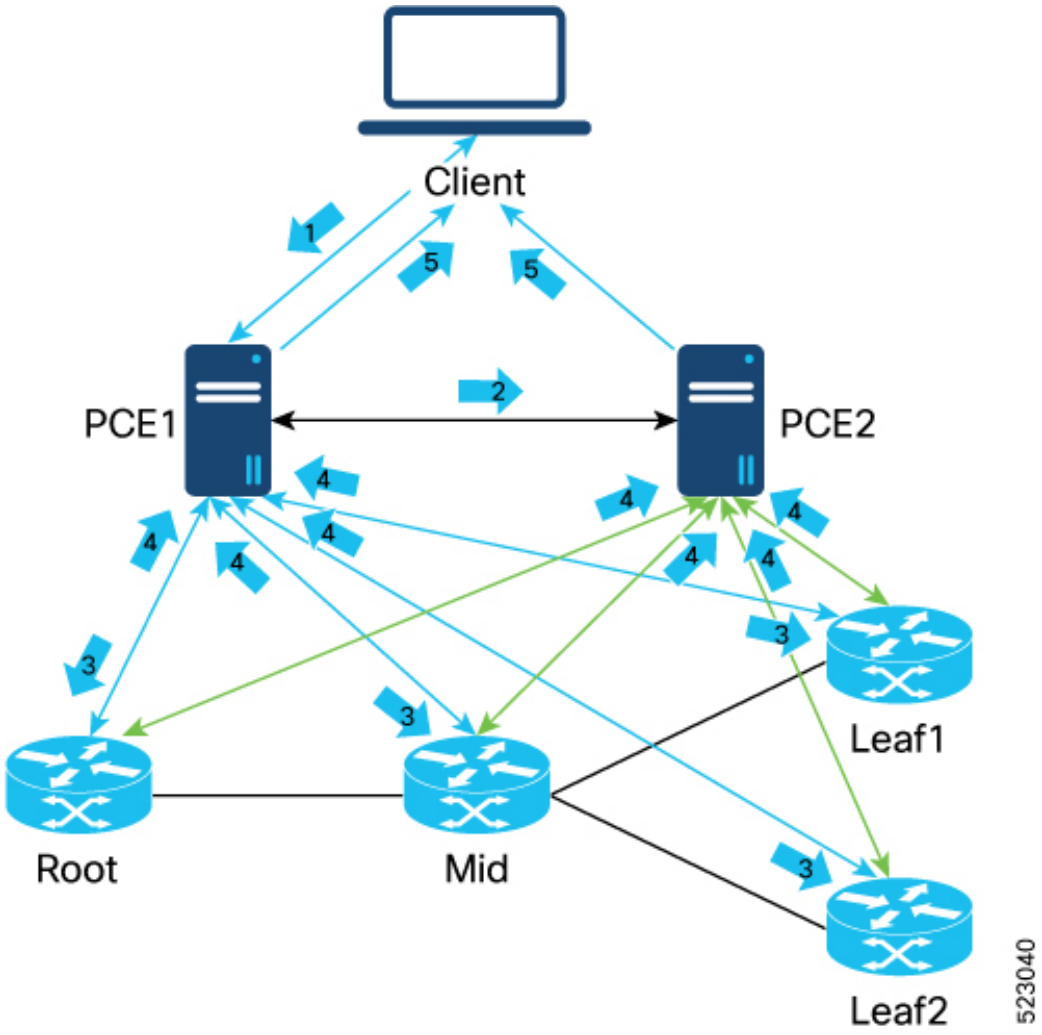


Network Handling

Understanding how each PCE operates in different states helps in configuring the SR-PCE HA and ensuring steady operability without any data loss. Following sections describe each state:

Steady State

In Steady state, the following events occur:

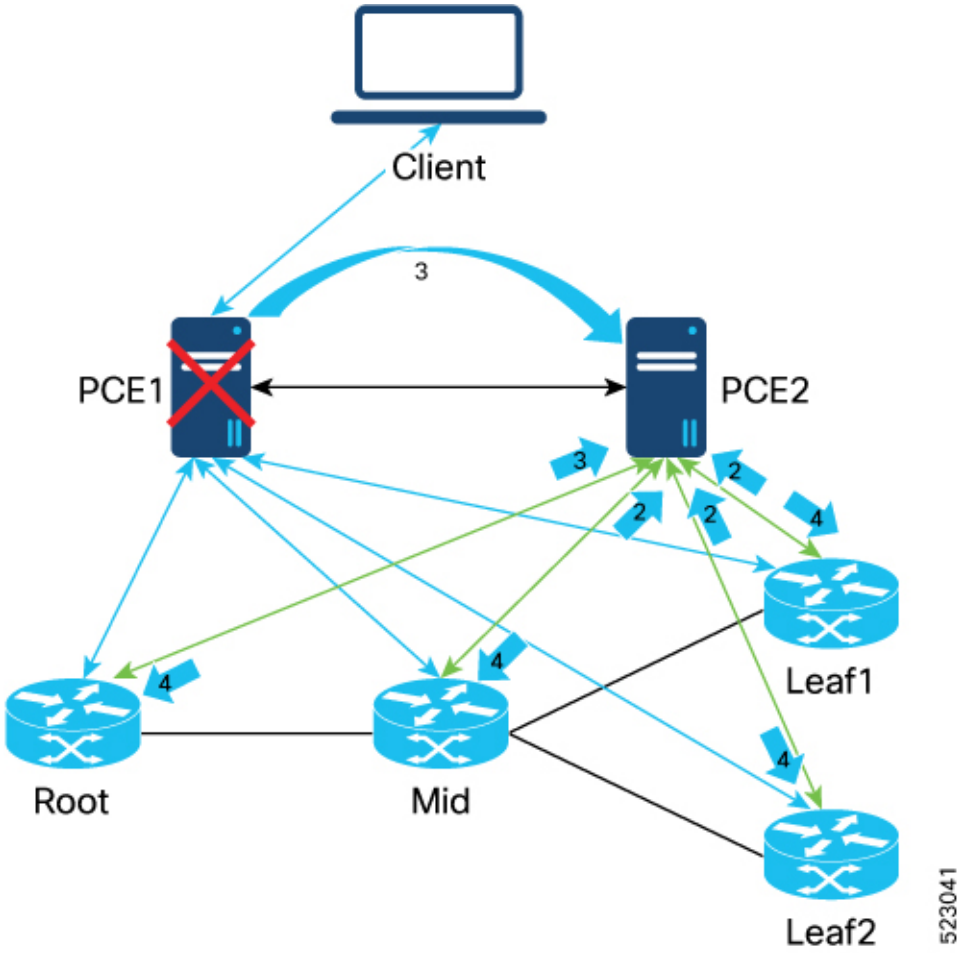


1. Client requests SR-P2MP tree creation.
2. Policy information is synchronized over the PCEP channel with sibling PCE.
3. PCE1 acts as the primary PCE, and computes the Tree.
   - PCE1 sends PCInitiate to the Root, Mid, and the Leaf nodes.
4. All PCCs respond with a PCReport.
   a. With D-bit set to 1 to the delegated creator PCE (PCE1).
   b. With D-bit set to 0 to other PCE (PCE2).
5. Both PCEs send the REST update notification to the Client.
   - This update helps the Client to know which PCE is the LSP delegated to.

- This information also helps in deciding which is the PCE a Tree Update/Delete request must be sent to.

PCE Failure

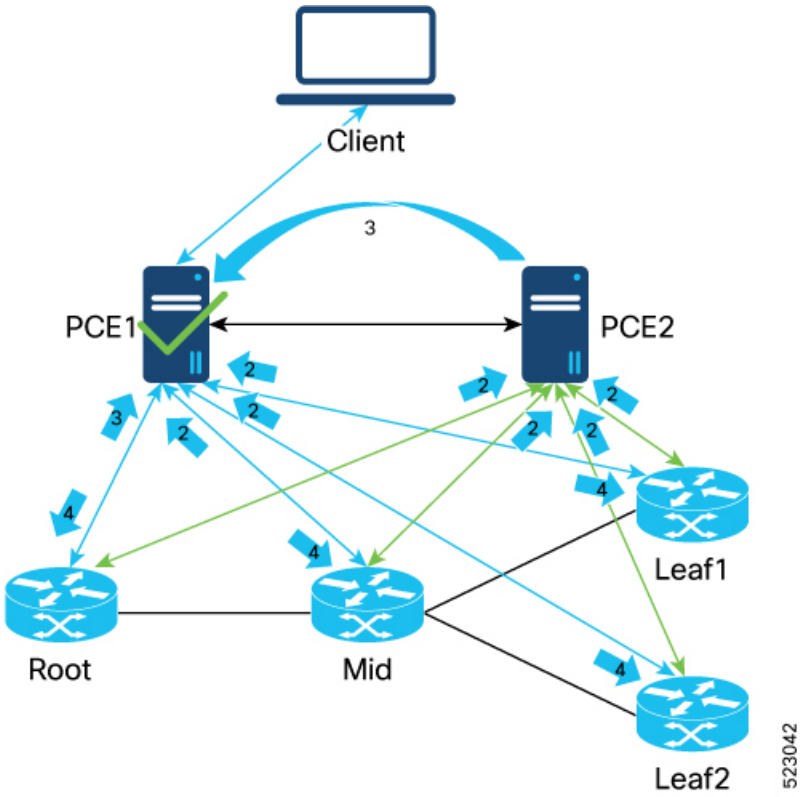
When PCE fails, the following events occur:



1. PCE1 fails.
2. All Tree nodes re-delegate to PCE2 immediately and sends PCReport with the D-bit set to 1.
3. PCE2 becomes the primary Compute PCE when it receives the re-delegated LSP from the Root, recomputes Tree-SID, and sends the update to PCCs.

PCE Restore

When PCE restores, the following events occur:



1. PCE1 is restored.

   PCE1 is the 'creator' and hence the preferred PCE.

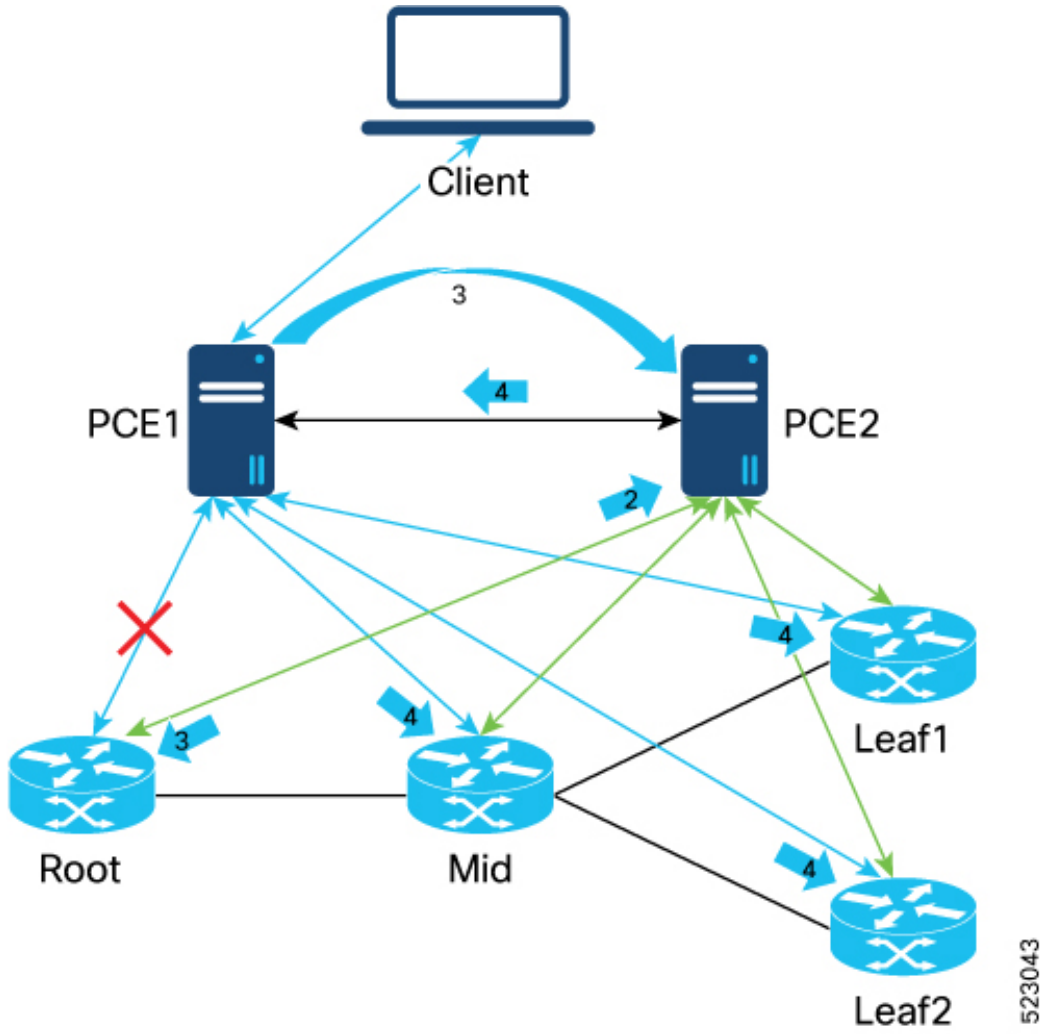
2. All Tree nodes redelegate to PCE1 after the delegation timer expires.
   - Sends PCReport with D-bit set to 1
   - Sends PCReport to PCE2 with D-bit set to 0

3. PCE1 becomes the primary Compute PCE when it receives the re-delegated LSP from the Root, recomputes Tree-SID, and sends update to PCCs.

Note Different PCCs can be delegated to different PCEs for a brief period. Any tree-update during this period is pushed to the sibling PCE over the state-sync channel to be forwarded to the delegated PCCs.

PCC Initiated PCEP Session with the Root is Down

When the PCC initiated PCEP session with the Root is down, the following events occur:

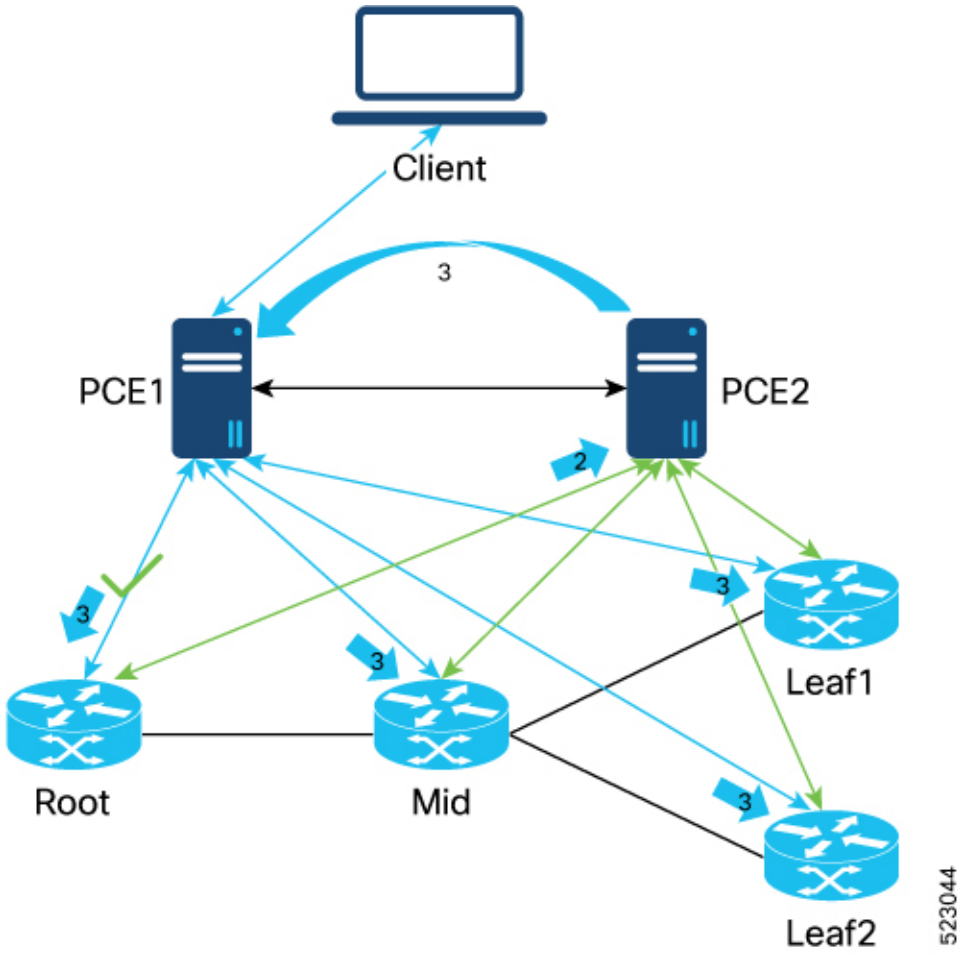


1. PCEP session between the Root and the PCE1 is down.
2. However, the Mid and Leaf nodes continue to have the session with PCE1.
3. Root redelegates to PCE2 immediately and sends the PCReport with D-bit set to 1.
4. PCE2 becomes the primary-Compute PCE recomputes Tree, and sends update to PCCs.

Note Updates to the Mid and Leaf PCCs are sent through the sibling PCE.

PCC Initiated PCEP Session with the Root is Restored

When the PCC initiated PCEP session with the Root is back up, the following events occur:

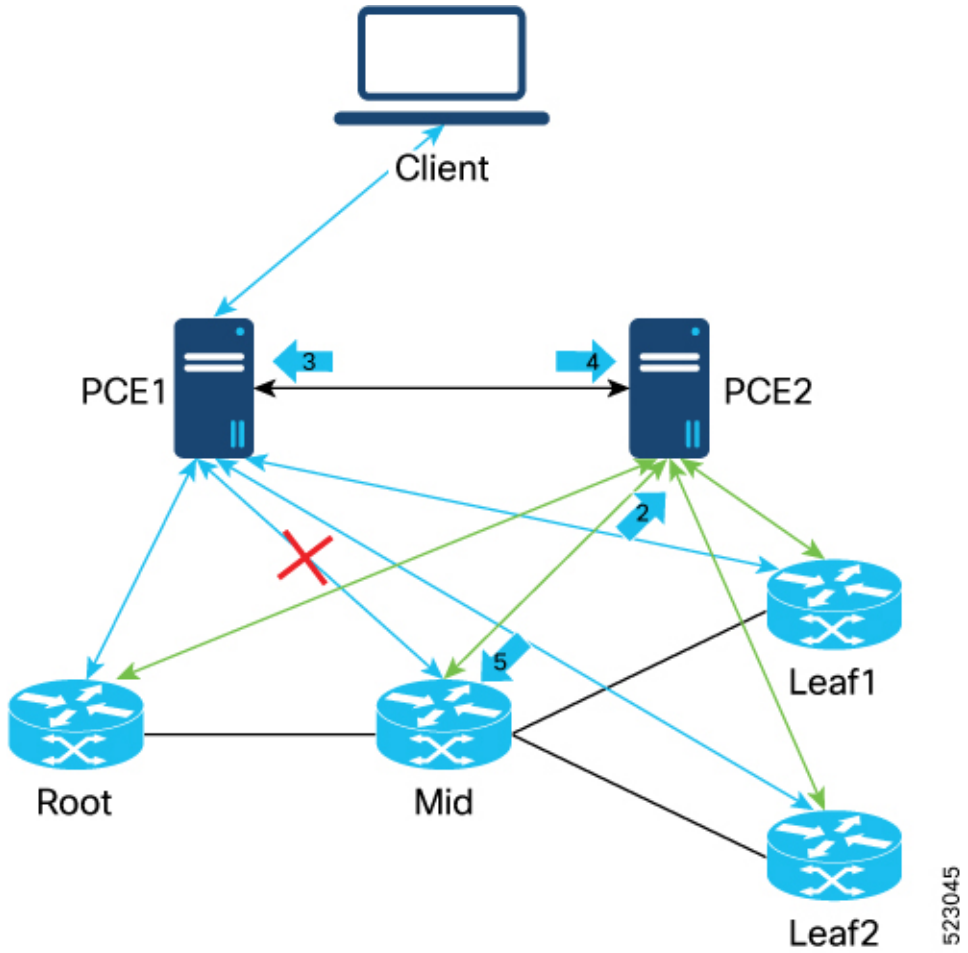


1. PCEP session between the Root and the PCE1 is restored.
2. Root redelegates to PCE1 immediately after the delegation timer expires.
   - Sends PCReport with D-bit set to 1
   - Sends PCReport to PCE2 with D-bit set to 0

PCE1 becomes the primary-compute PCE, recomputes Tree-SID, and sends the PCUpdate.

PCC Initiated - PCEP session with Mid or Leaf node is Down

When the PCC initiated PCEP session with the Mid or Leaf node is down, the following events occur:



1. PCEP session between Mid to PCE1 is down.

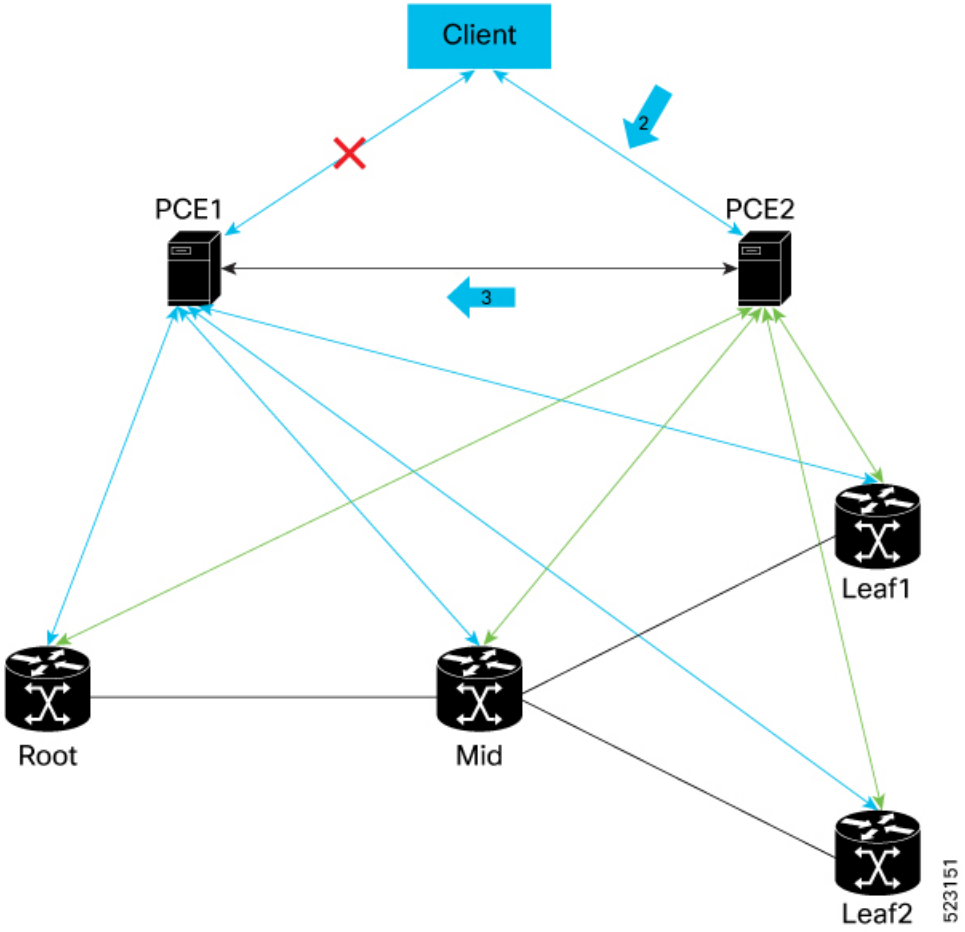
   However, the Root and Leaf node have PCEP sessions to PCE1.
2. Mid re-delegates LSP to PCE2 immediately and sends PCReport with D-bit set to 1.
3. PCReport is forwarded to PCE1 over the sibling channel (PCE2).

   PCE1 is still the primary PCE.
4. Sends PCUpdate to the sibling PCE2 and the PCE2 forwards PCUpdate down to the Mid PCC.
5. Mid PCC responds with PCReport which is forwarded to the PCE1.

Note Need the ability to send PCInitiate with R-flag to sibling PCE to remove state from a PCC.

Session Between Client and PCE is Down

When the session between the PCE and the Client is down, the following events occur:



- Session between PCE1 and Client is down.

  PCE delegation change does not happen. Client has information about PCE2 through which it can reach the PCCs. All update, create, and delete requests are sent to PCE2.

- PCReport is forwarded to the PCE1 over the sibling channel.

  However, the PCE1 is still the Compute PCE.

Limitations and Guidelines

- PCE HA for CLI-configured static Tree-SID is not supported.
- TreeSID PCE HA is supported only over IPv4 PCEP state-sync sessions.

Configuration on Forwarding Router

The following configurations are required to set up the TreeSID PCE HA feature.

PCE Configuration on a PCC

The following PCE configuration on the PCC routers provides redundancy.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# pcc
Router(config-pcc)# pce address ipv4 2.2.2.1
Router(config-pcc)# precedence 200
Router(config-pcc)# pce-group pce-ha-group
Router(config-sr-te-pce)# pce address ipv4 4.4.4.2
Router(config-pce)# precedence 0
Router(config-pce)# pce-group pce-ha-group
```

Note To choose a Compute PCE, you must set the precedence. In this example, two PCEs are configured with 200 and 0. The PCE with a lower precedence picks the compute role.

Adding a New SR-PCE in the Network

If required, the operator can add a new SR-PCE as a more preferred PCE in the above configuration on the root of the SR-P2MP tree. When the operator adds the new SR-PCE, the following events occur:

1. The PCC re-delegates all SR-P2MP LSPs to the new configured SR-PCE.
2. The new SR-PCE takes over the role of computation for the SR-P2MP trees.
3. The operator can remove the older SR-PCE from the network.

Associating a PCE with a PCE Group

A PCE can be associated to only one PCE group. Use the following configuration to associate a PCE with a PCE group:

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# pcc
Router(config-pcc-pce)# pce address ipv4 192.168.0.5
Router(config-pcc-pce)# pce-group test
```

Associating a PCE Group with on-demand-color

You can associate only one PCE group with one on-demand-color configuration. However, the same PCE group can be used across many on-demand-colors.

After associating a PCE with a PCE group, you can associate the PCE group to an on-demand-color, which later is associated with a P2MP policy. Use the following configuration:

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te-color)on-demand color 10
Router(config-sr-te-color)pce-group test
```

PCE state-Sync Configuration on SR-PCE

Configure all PCEs in the network for PCE state-sync. Configuring it on only one PCE enables only that PCE to sync state unidirectionally.

```
Router(config)# pce state-sync ipv4 192.0.2.6
```

Verifying the PCE Configurations

Run the following show commands to verify if the PCE compute role is set. The following is an example of the command run on a Compute SR-PCE:

```
Router# show pce lsp ipv4 p2mp
Tree: sr_p2mp_root_192.168.0.4_tree_id_524289, Root: 192.168.0.4 ID: 524289
 PCC: 192.168.0.4
 Label:    19000
 Operational: up  Admin: up
 Compute: Yes
 Local LFA FRR: Disabled
 Metric Type: IGP
 Transition count: 1
 Uptime: 00:21:37 (since Fri Mar 11 18:36:06 PST 2022)
 Destinations: 192.168.0.1, 192.168.0.3
 Nodes:
  Node[0]: 192.168.0.2+ (rtrM)
   Delegation: PCC
   PLSP-ID: 1
   Role: Transit
   State Changes: 0x2 (New Hops)
   Endpoints: 192.168.0.3 192.168.0.1
   Hops:
    Incoming: 19000 CC-ID: 1
    Outgoing: 19000 CC-ID: 1 (13.13.13.3) [rtrL2:192.168.0.3]
      Endpoints: 192.168.0.3
    Outgoing: 19000 CC-ID: 1 (10.10.10.1) [rtrL1:192.168.0.1]
      Endpoints: 192.168.0.1
  Node[1]: 192.168.0.4 (rtrR)
   Delegation: PCC
   Locally computed
   PLSP-ID: 1
   Role: Ingress
   Endpoints: 192.168.0.3 192.168.0.1
   Hops:
    Incoming: 19000 CC-ID: 2
    Outgoing: 19000 CC-ID: 2 (16.16.16.2) [rtrM:192.168.0.2]
      Endpoints: 192.168.0.3 192.168.0.1
  Node[2]: 192.168.0.3 (rtrL2)
   Delegation: PCC
   Locally computed
   PLSP-ID: 1
   Role: Egress
   Hops:
    Incoming: 19000 CC-ID: 4
  Node[3]: 192.168.0.1+ (rtrL1)
   Delegation: PCC
   Locally computed
   PLSP-ID: 2
   Role: Egress
   State Changes: 0x7 (New Node,New Hops,Role Change)
   Hops:
    Incoming: 19000 CC-ID: 5


Event history (latest first):
  Time                    Event
  Mar 11 18:57:12.522     Received report from all nodes awaiting report, state: Pruning
stale legs on non-root nodes
  Mar 11 18:57:12.522     No nodes awaiting report, state: Pruning stale legs on non-root
nodes
  Mar 11 18:57:12.522    Received report from all nodes awaiting report, state: Programming
```

```
 the root node
  Mar 11 18:57:12.522     No nodes awaiting report, state: Programming the root node
  Mar 11 18:57:12.522    Received report from all nodes awaiting report, state: Programming
 non-root nodes
  Mar 11 18:57:12.512     Node 192.0.6.1 delegated by PCC
  Mar 11 18:57:12.473     Path computation returned a different result, signaling new path
  Mar 11 18:57:12.473     Received report from all nodes awaiting report, state: Pruning
stale legs on non-root nodes
  Mar 11 18:57:12.472     TreeSID Leaf set changed
  Mar 11 18:51:10.146     Node 192.168.0.1 undelegated by PCC
  Mar 11 18:51:10.080    Received report from all nodes awaiting report, state: Programming
 the root node
  Mar 11 18:51:10.080     No nodes awaiting report, state: Programming the root node
  Mar 11 18:51:10.080    Received report from all nodes awaiting report, state: Programming
 non-root nodes
  Mar 11 18:51:10.080     No nodes awaiting report, state: Programming non-root nodes
  Mar 11 18:51:10.080     Path computation returned a different result, signaling new path
  Mar 11 18:51:10.080     Received report from all nodes awaiting report, state: None
  Mar 11 18:51:10.080     TreeSID Leaf set changed
  Mar 11 18:36:06.813     Received report from all nodes awaiting report, state: Pruning
stale legs on non-root nodes
  Mar 11 18:36:06.813     No nodes awaiting report, state: Pruning stale legs on non-root
nodes
  Mar 11 18:36:06.813    Received report from all nodes awaiting report, state: Programming
 the root node
  Mar 11 18:36:06.813     No nodes awaiting report, state: Programming the root node
  Mar 11 18:36:06.813    Received report from all nodes awaiting report, state: Programming
 non-root nodes
  Mar 11 18:36:06.552     Node 192.168.0.3 delegated by PCC
  Mar 11 18:36:06.534     Path computation returned a different result, signaling new path
  Mar 11 18:36:06.534     Received report from all nodes awaiting report, state: Pruning
stale legs on non-root nodes
  Mar 11 18:36:06.534     No nodes awaiting report, state: Pruning stale legs on non-root
nodes
  Mar 11 18:36:06.534     Operational state changed to up (0 transitions)
  Mar 11 18:36:06.534    Received report from all nodes awaiting report, state: Programming
 the root node
  Mar 11 18:36:06.323     Node 192.168.0.4 delegated by PCC
  Mar 11 18:36:06.323     TreeSID Leaf set changed
  Mar 11 18:36:06.316    Received report from all nodes awaiting report, state: Programming
 non-root nodes
  Mar 11 18:36:06.316     Node 192.168.0.1 delegated by PCC
  Mar 11 18:36:06.298     Node 192.168.0.2 delegated by PCC
  Mar 11 18:36:06.249     PCE compute role set
  Mar 11 18:36:06.249     TreeSID metric type changed to 0
  Mar 11 18:36:06.249     TreeSID Leaf set changed
  Mar 11 18:36:06.249     TreeSID created
```

On a non-compute SR-PCE

```
The following is an example of the command run on a non-compute SR-PCE:
Tree: sr_p2mp_root_192.168.0.4_tree_id_524289, Root: 192.168.0.4 ID: 524289
 PCC: 192.168.0.4
 Label:    19000
 Operational: standby  Admin: up  Compute: No
 Local LFA FRR: Enabled
 Metric Type: IGP
 Transition count: 0
 Destinations: 192.168.0.1, 192.168.0.3
 Nodes:
  Node[0]: 192.168.0.4+ (rtrR)
   Delegation: PCE 192.168.0.5
   PLSP-ID: 1
   Role: None
   State Changes: 0x3 (New Node,New Hops)
```

```
  Hops:
   Incoming: 19000 CC-ID: 2
   Outgoing: 19000 CC-ID: 2 (16.16.16.2)
 Node[1]: 192.168.0.2+ (rtrM)
  Delegation: PCE 192.168.0.5
  PLSP-ID: 1
  Role: None
  State Changes: 0x3 (New Node,New Hops)
  Hops:
   Incoming: 19000 CC-ID: 1
   Outgoing: 19000 CC-ID: 1 (13.13.13.3)
   Outgoing: 19000 CC-ID: 1 (10.10.10.1)
 Node[2]: 192.168.0.3+ (rtrL2)
  Delegation: PCE 192.168.0.5
  PLSP-ID: 1
  Role: None
  State Changes: 0x3 (New Node,New Hops)
  Hops:
   Incoming: 19000 CC-ID: 4
 Node[3]: 192.168.0.1+ (rtrL1)
  Delegation: PCE 192.168.0.5
  PLSP-ID: 2
  Role: None
  State Changes: 0x3 (New Node,New Hops)
  Hops:
   Incoming: 19000 CC-ID: 5

Event history (latest first):
 Time                    Event
 Mar 11 18:57:12.688     Node 192.168.0.1 delegated by PCE 192.168.0.5
 Mar 11 18:57:12.485     TreeSID Leaf set changed
 Mar 11 18:51:10.082     TreeSID Leaf set changed
 Mar 11 18:36:06.713     Node 192.168.0.3 delegated by PCE 192.168.0.5
 Mar 11 18:36:06.499     TreeSID Leaf set changed
 Mar 11 18:36:06.499     Node 192.168.0.1 delegated by PCE 192.168.0.5
 Mar 11 18:36:06.499     Node 192.168.0.2 delegated by PCE 192.168.0.5
 Mar 11 18:36:06.291     Node 192.168.0.4 delegated by PCE 192.168.0.5
 Mar 11 18:36:06.291     TreeSID metric type changed to 0
 Mar 11 18:36:06.291     TreeSID Leaf set changed
 Mar 11 18:36:06.291     TreeSID created
```