Network Working Group H. Bidgoli, Ed.
Internet-Draft Nokia
Intended status: Standards Track V. Voyer
Expires: 27 August 2026 Cisco Systems
A. Budhiraja
Cisco System
R. Parekh
Arrcus
S. Sivabalan
Ciena
23 February 2026

PCEP extensions for SR P2MP Policy
draft-ietf-pce-sr-p2mp-policy-14

Abstract

Segment Routing (SR) Point-to-Multipoint (P2MP) Policies are a set of policies that enable architecture for P2MP service delivery. This document specifies extensions to the Path Computation Element Communication Protocol (PCEP) that allow a stateful PCE to compute and initiate P2MP paths from a Root to a set of Leaf nodes.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Requirements Language 6
4. Overview of PCEP Operation in SR P2MP Network 6
4.1. Related and Inherited Documents 6
4.2. Overview of SR P2MP Policy Objects 7
4.2.1. SR P2MP Policy and Candidate Path Identification . . 9
4.2.2. Replication Segment Identification 9
4.2.3. PCECC Use in Replication Segment 9
4.3. PCEP Prodecures 10
4.3.1. PCE-Init Procedure 10
4.3.2. PCC-Init Procedure 11
4.3.3. Common Procedures 11
4.3.3.1. Replicatoin Segment Instantiation 11
4.3.3.2. New Candidate Path Creation 12
4.3.3.3. Adding new Leaf nodes 12
4.3.3.4. Conveying active state 13
4.3.4. Global Optimization of the Candidate Path 13
4.3.5. local optimizatoin 14
4.3.6. Fast Reroute 14
4.3.7. Connecting Replication Segment via Segment List . . . 15
4.3.8. Tree Deletion 15
4.3.8.1. PCC Initiated 16
4.3.8.2. PCE Initiated 16
4.4. PCEP Messages 16
4.4.1. SR P2MP Policy 17
4.4.2. Replication Segment 19
4.4.3. Considerations 22
4.4.3.1. Path Attribute Object 23
4.4.3.2. CCI Object 23
5. Object Format 23
5.1. Open Message Extension 23
5.2. PCE Capabliity SubTLV 24
5.3. Association Type Capability 24
5.4. Symbolic Name TLV 25
5.5. SR P2MP Policy 25
5.5.1. P2MP Policy Association Group 25

5.5.1.1. Extended Association ID TLV 26
5.5.2. P2MP-END-POINTS Object 26
5.6. P2MP Policy and Replication Segment Identifier Object and TLV 28
5.6.1. Extension of the LSP Object, SR-P2MP-LSPID-TLV . . . 28
5.7. Replication Segment 31
5.7.1. Message format 31
5.7.2. CCI Object 32
5.7.3. SR-ERO Rules 33
6. IANA Consideration 33
6.1. PCEP P2MP Association type 33
6.2. Endpoint Type 34
6.3. PCEP TLV Type Indicators 34
6.4. New CCI Object Type 35
7. Security Considerations 35
8. Implementation Status 36
8.1. Cisco Implementation 36
9. Manageability Considerations 36
9.1. Operatoin Consideration 36
9.2. Liveness Detection and Monitoring 37
9.3. Verify Correct Operations 37
9.4. Requirements On Other Protocols 37
9.5. Impact On Network Operations 37
10. Acknowledgments 38
11. Contributors 38
12. References 38
12.1. Normative References 38
12.2. Informative References 40
Appendix A. Packet Examples 40
A.1. Report for Leaf Add 40
A.2. SR P2MP Policy Candidate Path Init 41
A.3. Replication segment PCE Initiated on Transit and LEaves 45
Appendix B. Example Workflows 47
Authors' Addresses 52

1. Introduction

The document [draft-ietf-pim-sr-p2mp-policy] defines a variant of the SR Policy called SR Point-to-Multipoint (P2MP) Policy for creation of P2MP trees.

A SR P2MP Policy is constructed using one or more Replication segments ([RFC9524]) from a Root node to a set of Leaf nodes, optionally through a set of intermediate transit nodes that perform replication.

A SR P2MP Policy is installed only on the Root node of a P2MP tree and consists of one or more Candidate Paths (CP). Each CP is made up of P2MP Tree Instances (PTI), and each PTI is constructed in the network via Replication segments.

A Replication segment [RFC9524], corresponds to the forwarding state of a P2MP segment on a particular node and provides forwarding instructions for the segment.

As per [draft-ietf-pim-sr-p2mp-policy] a P2MP service can be realized by two types of a P2MP Trees, Ingress Replication or a P2MP tree.

For Ingress Replication [RFC7988]. Dataplane packet replication only occurs on the Root, which forwards individual copies of traffic to each leaf directly over a segment routed path. The corresponding SR P2MP Policy consists of Replication segments only for the Root node and the Leaf nodes.

A P2MP service delivery can be more efficient using a P2MP Tree. In this case, corresponding SR P2MP policy consists of Replication segments on the Root, Leaf, and Transit nodes which exist in the topology between the Root and Leaf nodes. The Root and Transit nodes perform dataplane packet replication along the tree as a packet traverses from the root towards each leaf.

The PCE discovers the root and the leaves via different methods. As an example, the leaves and the root can be explicitly configured on the PCE or PCC can update the PCE with the identity of the root and the leaves when it discovers them via multicast protocols like MP-BGP and MVPN procedures [RFC6513] or PIM. Once the controller is informed of the Root node and Leaf nodes, it can calculate the SR P2MP Policy and any of the required Replication segments from the Root node to the Leaf nodes. The computation may include traffic engineering criteria and any additional Service Level Agreements (SLAs) that is used to construct the tree.

This document defines PCEP objects, TLVs and the procedures to instantiate a P2MP Policy and Replication Segments.

Only Stateful PCE is within scope of this document and Stateless PCE only is out of scope.

2. Terminology

The readers of this document should familiarize themselves with the following documents and sections for terminology and details implementation of the SR P2MP Policy.

[RFC9524] section 1.1 defines terms specific to SR Replication Segment and also explains the Node terminology in a multicast domain, including the Root Node, Leaf Node and a Bud Node. [draft-ietf-pim-sr-p2mp-policy] section 2, defines terms and concepts specific to SR P2MP Policy including the CP and the PTI.

In addition, below section defines terms used frequently in this document. Refer to Terminology sections of [RFC9256], and [RFC9524] for other terms used in Segment Routing.

* Unicast Segment Routing (SR): Standard Segment Routing constructs, capabilities and behavior which is not multicast or replication aware.

* Replication segment: A segment in SR domain that replicates packets. See [RFC9524], for details.

* Replication node: A node in SR domain which replicates packets based on Replication segment.

* Downstream nodes: A Replication segment replicates packets to a set of nodes. These nodes are Downstream nodes.

* Replication state: State held for a Replication segment at a Replication node. It is conceptually a list of replication branches to Downstream nodes. The list can be empty.

* Replication SID: Data plane identifier of a Replication segment. This is a SR-MPLS label or SRv6 Segment Identifier (SID).

* Point-to-Multipoint Service: A service that has one ingress node and one or more egress nodes. A packet is delivered to all the egress nodes

* Transit node: alternative name for Replication node.

* Root node: An ingress node of a P2MP service.

* Leaf node: An egress node of a P2MP service.

* Bud node: A node that is both a Replication node and a Leaf node.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Overview of PCEP Operation in SR P2MP Network

After discovering the Root node and Leaf nodes, the PCE programs the PCCs with relevant information needed to create a P2MP Tree.

A SR P2MP policy is a variant of SR policy as such it inherits similar concept as draft [RFC9862]. A SR P2MP Policy is composed of a collection of CPs. CPs are computed by the PCE and can be used for P2MP Tree redundancy. Only a single CP may be active at each time. Each CP can be globally optimized, therefore it consists of multiple PTIs. A PTI can be considered as a P2MP LSP. If a CP needs to be globally optimized two PTIs can be programmed from the root to a set of leaves and via make before break procedures the CP can switch from the original PTI to the new optimized PTI. The forwarding states of these PTIs are build via replication segments. Each PTI initiated on the root has its own set of replication segments on the Root, Transit and Leaf nodes.

A replication segment is set of forwarding instructions on a specific node. Each instruction may be a PUSH or SWAP operation before forwarding out of an interface, or a POP action on bud and leaf nodes.

PCE MAY also calculate and download additional information for the replication segments, such as protections next-hops for link protection.

4.1. Related and Inherited Documents

* [RFC8231] The bases for a stateful PCE, and reuses the following objects or a variant of them

  - <SRP Object>

  - <LSP Object>

  - A variation of the LSP identifier TLV is defined in this draft, to support P2MP LSP Identifier

* [RFC8306] P2MP capabilities advertisement.

* [RFC9862] Candidate Paths for SR P2MP Policy is used for Tree Redundancy. As an example, a SR P2MP Policy can have multiple Candidate Paths. Each protecting the primary Candidate Path. The active CP is chosen via the preference of the Candidate Path.

* [RFC3209] Defines the instance-ID, instance-ID is used for global optimization of a CP within a SR P2MP policy. Each CP can have two PTIs. These PTIs are equivalent to sub-lsps (instance-IDs). To switch between PTIs procedures like Make Before Break (MBB) and global optimization can be used. Instance-ID is equivalent to LSP ID.

* [RFC9256] Segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.

* [RFC8306] P2MP End Point objects, used for the PCC to update the PCE with discovered Leaves.

* [RFC9050] for programming and identifying the Replication Segment. A new PCE CC Capability sub Tlv is introduced to indicated the support to handle PCE CC based label download for SR P2MP.

* [RFC9862]defines CCI object-type for SR-MPLS. This document redefines a new version of the SR-MPLS CCI object-type.

* [draft-ietf-pce-multipath] Forwarding instruction for a P2MP LSP is defined by a set of SR-ERO sub-objects in the ERO object, ERO-ATTRIBUTES object and MULTIPATH-BACKUP TLV as defined in this draft.

* [RFC8664] SR-ERO Sub Object used in the multipath.

4.2. Overview of SR P2MP Policy Objects

* SR P2MP Policy

  - Is only relevant on the Root of the P2MP tree and is a policy on PCE. It is downloaded only on the root node and is identified via <Root, Tree-ID> It contains the following information:

    o Root node of the P2MP Segment

    o Set of Leaf nodes of the P2MP Segment

    o Tree-ID, which is a unique identifier of the P2MP tree on the Root o A set of CPs belonging to the policy

    o Optional Constraints used to build these CPs

* Candidate Path (CP):

  - Is used for P2MP Tree redundancy where the Candidate Path with the highest preference is the active path.

  - Each CP can contain two P2MP Tree Instance (PTI) for global optimization procedures (i.e. make before break)

  - Contains information regarding originator, discriminator, preference, P2MP Tree Instances

* Tree Instance:

  - Is used for global optimization of the CP. Two PTIs can be present under a Candidate Path but only a single PTI is active at a time.

  - A Tree Instance is identified via <Root, Tree-ID, Instance-ID>

* Replication Segment:

  - Is the forwarding information needed on each replication node for building the forwarding path for each PTI of the CP.

  - Explained further in upcoming sections, there are 2 ways to identify the replication segment, depending on the type of replication segment (shared replication segment or non-shared replication segment)

    o It is identified via Tree-ID and Root and PTI for non-shared replication segment.

    o It is identified via Node-ID, Replication-ID, for shared replication segment. As per [draft-ietf-pim-sr-p2mp-policy] a shared replication segment is not associated to a single tree and can be used for constructing by-pass tunnels for local protection on a Replication node.

    o Contains forwarding instructions, in the form of a list of outgoing segments each of which may be a segment list or a single replication segment with next-hop information.

    o On the forwarding plane the Replication Segment is identified via the incoming Replication SID.

- o Is established on every node that may replicate (e.g., Root, Transit, Bud) or receive (e.g., Leaf) the packet in an SR P2MP tree.

4.2.1. SR P2MP Policy and Candidate Path Identification

A SR P2MP Policy and its CP can be identified on the root via the P2MP LSP Object. This Object is a variation of the LSP ID Object defined in [RFC8231] and is as follow:

* PLSP-ID: [RFC8231], is assigned by PCC and is unique per Candidate Path. It is constant for the lifetime of a PCEP session. Each additional Candidate Path is assigned a new PLSP-ID by PCC. Multiple Candidate Paths can co-exist but only one is active.

* Root: is equivalent to the first node on the SR P2MP Policy path.

* Tree-ID: an identifier that is unique in context of the Root node. This value may be assigned manually on the Root node or advertised via the Provider Tunnel Association(PTA) in a multicast BGP Auto-Discovery(AD) route.

* Instance-ID: serves as the PTI identifier and is assigned by the PCE. A CP can have up to two PTIs for global optimization. Instance-IDs are unique within the SR P2MP Policy and are assigned by the PCE per PTI within that Policy. While different P2MP policies may use the same Instance-ID for their PTI, each PTI within a CP of an SR P2MP Policy should use the same Instance-ID across the tree on its Root, Transit, and Leaf nodes when programming its replication segments on the PCC.

4.2.2. Replication Segment Identification

The key to identify a replication segment is also a P2MP LSP Object. With varying encoding rules for the SR-P2MP-LSPID-TLV which will be explained in later sections.

4.2.3. PCECC Use in Replication Segment

PCECC and a variant of CCI object is used in Replication Segment to identify a cross connect. A cross connect is an incoming SID and set of outgoing interfaces and their corresponding SID or SID List. The CCI objects contain the incoming SID and the outgoing interfaces which are presented via the ERO objects, which each may contain a segment list.

4.3. PCEP Prodecures

A SR P2MP policy on the Root can be either PCE-initiated or PCC-initiated, depending on how the Root and Leaf nodes are discovered. A PCE-initiated SR P2MP policy is configured directly on the PCE, while a PCC-initiated SR P2MP policy may be triggered by the PCC, sourced from local configuration or discovered with multicast protocols such as MVPN (see [RFC6513]), PIM, IGMP etc. In both cases, PCE-initiated mechanisms are used to install Replication segments on Transit, Bud and Leaf nodes.

Note: Algorithms and techniques to compute the P2MP tree replicating nodes are out of scope of this document.

4.3.1. PCE-Init Procedure

* PCE is informed through its API or configuration mechanism to instantiate a SR P2MP Policy. PCE is configured with the Root and a set of Leaf nodes.

* PCE computes a P2MP tree from the Root node to all Leaf nodes which satisfy the configured constraints.

* PCE sends a PCInitiate message to the Root. The PCInitiate message is configured with a unique Instance-ID for the PTI within the CP of the SR P2MP Policy. The PCInitiate message sent to the root MUST set the Tree-ID to 0. The endpoint-object MAY optionally be included in the PCInitiate message for providing list of Leaf nodes to the PCC for informational purposes.

* In response to the PCInitiate message, the PCC will assign the PLSP-ID and Tree-ID for the CP. The PCC uses the Instance-ID defined in the PCInitiate message for the PTI contained within the CP. The PCC sends a PCRpt back to PCE containing the PLSP-ID, Tree-ID and Instance-ID. PCC MAY optionally include additional Leaf nodes that were also discovered by multicast procedures.

* PCE will send a PCInitiate message to the Root, Transit and the Leaf nodes to download the Replication Segment information. These messages will utilize the CCI object to identify the p2mp cross connect and encode the forwarding instruction information.

* PCE then sends a PCUpdate to the Root node indicating the association information (CP) and implicitly binds the CP to the installed CCI information.

4.3.2. PCC-Init Procedure

* Root node PCC discovers the leaf nodes via MVPN procedures or other mechanism

* Root sends a PCRpt message for SR P2MP policy to PCE including the Root, Tree-ID, PLSP-ID, symbolic-path-name, association object and any leaf nodes discovered. In addition:

* - Since the Instance-ID is set by the PCE, the root will set the Instance-ID to value to 0 in the RCRpt message.

  - For the association object, root will fill the association type according to the association type defined in this document. The association ID SHOULD be set to value 1 similar to [RFC9862]. The association source is set to the Root PCC Address.

* PCE receives the PCRpt and generates a unique Instance-ID for this PTI of the CP and compute the P2MP Policy and its replication segments.

  - PCE will send a PCInitiate message to the Root, Transit and the Leaf nodes to download the Replication Segment information. These messages will utilize the CCI object to encode the forwarding instruction information.

  - PCE then sends a PCUpdate to the Root node indicating the association information (CP) and implicitly binds the CP to the installed CCI information.

4.3.3. Common Procedures

The following procedures are the same for PCE-init or PCC-init SR P2MP Policy.

4.3.3.1. Replicatoin Segment Instantiation

* PCE distributes the Replication segments for each CP’s PTI to all Transit, Bud and Leaf nodes in the SR P2MP Tree using a PCInitiate message.

* - PLSP-ID: value MUST be set to zero and will be assigned by PCC.

  - Symbolic path name: generated by PCE, MUST be unique for the each Candidate Path on PCC.

  - Root: the same value of the SR P2MP Policy.

   - Tree-ID: it is RECOMMENDED the Tree-ID be set to the same Tree-ID defined on the Root node (which was assigned by the Root node PCC).

   - Instance-ID: set to the same value of path-instance on the Root.

* The PCInitiate message includes the EROs and utilizes the CCI object to encode the Replication segment forwarding instruction information.

* In response to the PCInitiate message, each Transit, Bud and Leaf node PCC generates a local PLSP-ID and sends a PCRpt to PCE informing PCE of the Replication segment state.

4.3.3.2. New Candidate Path Creation

* Any new CP for the SR P2MP Policy is downloaded by PCE to the Root by sending a PCInitiate message.

   - PLSP-ID: value MUST be set to zero and will be assigned by PCC.

   - Symbolic path name: generated by PCE, MUST be unique for each CP on PCC.

   - Root: the same value of the SR P2MP Policy.

   - Tree-ID: the same value of the SR P2MP Policy.

   - Instance-ID: value MUST be set to zero. The PCC generates the PTI’s Instance-ID of the CP.

* The PCE distributes the necessary Replication segment for the CP and its PTIs to the Root, Leaf nodes and the Transit nodes as described in section "Replication Segment Instantiation".

* Any update to the CP or Replication segments is performed via the PCUpd message. Association object MUST be present for CP PCUpdate and PCRpt message. CCI object MUST be present in the Replication segment updates.

4.3.3.3. Adding new Leaf nodes

* New Leaf nodes may be locally configured or discovered via multicast protocol procedures. New Replication segments may be instantiated or existing one updated to reach new Leaf nodes.

- If the new Leaf nodes reside on routers that are part of the existing SR P2MP Tree, then PCUpd is sent from PCE to necessary PCCs (Leaf, Bud or Root nodes) with the existing PLSP-ID, Instance-ID, Tree-ID and CC-ID.

- If the new Leaf nodes are residing on routers that are not part of the existing SR P2MP Tree, then a PCInitiate message is sent from PCE to each new Leaf and any necessary Transit nodes following section "Replication Segment Instantiation".

4.3.3.4. Conveying active state

The active CP is indicated by the PCC through the operational bits(Up/Active) of the LSP object in the PCRpt message.

A CP is made active based on the preference of the path. If the Root is programed with multiple CPs from different sources, as an example PCE and CLI, based on its tie-breaking rules, if it selects the CLI path, it will send a report to PCE for the PCE path indicating the status of label-download and sets operational bit of the LSP object to UP and Not Active. At any instance, only one PTI is permitted be active.

4.3.4. Global Optimization of the Candidate Path

When a P2MP LSP needs to be optimized for any reason (i.e. it is taking a FRR tunnel or new routers are added to the network) a global optimization of the CP is possible.

Each CP MAY contain two PTIs. The current unoptimized PTI is the active instance and its replication segments are forwarding the multicast packets from the root to the leaves. However the second optimized PTI will be setup with its own unique Replication Segments throughout the network, from the Root to the leaf nodes. These two PTIs MAY co-exist. The two PTIs are uniquely identified by their Instance-ID in the SR-P2MP-INSTANCE-ID-TLV (defined in this document).

After the optimized PTI has been downloaded successfully, PCC MUST send two reports, reporting UP of the new PTI the new LSP-ID, and a second reporting the tear down of the old PTI with the R bit of the LSP Object SET with the old Instance-ID in the SR-P2MP-INSTANCE-ID-TLV. This MBB procedure will move the multicast PDUs to the optimized Path-Instance.

The transit and leaf nodes SHOULD be able to accept traffic from both PTIs to minimize the traffic outage by the Make Before Break process.

It is recommended for the optimized PTI to be setup up by PCE from the leaf nodes to transit nodes and finally the root nodes to ensure the entire PTI’s path is programmed before the MBB procedure is initiated.

4.3.5. local optimizatoin

When one of the PCCs involved in the LSP lacks the capability to support more than one PTI, the possibility of achieving global make before break (MBB) is not feasible. However, with knowledge of the PCCs’ advertised capabilities, the PCE can detect this limitation and instead opt for local re-optimization of the CP. In such cases, the PCE can compute the optimized LSP by send the PCUpd message using the existing PTI and its Instance-ID for CP, specifically targeting the PCCs where the optimized LSP triggers a change in forwarding state.

4.3.6. Fast Reroute

This document identifies Facility Fast Reroute (FRR) procedures. Only Link Protection procedures are defined. How the Facility Path is computed and instantiated is outside of scope of this document.

```
      R
     | |
      T
      |
     ---
    |   |
    L1 L2
```

Figure 1: Redundant directly connected interfaces

As an example, in figure 1 both R and T are configured with replication segments. There are two interface between R and T. One can be used as primary and second as a bypass in case the primary interface is down. There can be two methods to protect the primary interface:

* The two replication segments on R and T can take advantage of unicast SR to connect to each other. In this case the LFA of unicast SR can be utilize to protect the primary interface between R and T.

* The replication segment provides protection nexthop, the protection nexthop can be programmed to take the alternate interface between R and T to protect the primary interface.

If the network already has implemented unicast SR then it might be advantageous to use LFA for SR to protect the link between R and T but if the network has not enabled unicast SR then the second option of replication segment protection nexthop is the preferred method.

```
        R---F1
        |   |
        T---F2
        |
         ---
        |   |
        L1 L2
```

Figure 2: Protection through alternative network path

As a second example, in Figure 2 R and T connected directly and via network F1..F2. In this example as per example 1, unicast SR can be used to connect the two Replication segments, including using SR LFA or R-LFA or TI-LFA to protect the direct link between R-T via F1. If unicast SR is not available within the network, the PCE optionally can establish a shared replication point on F1 and F2 and protect all path-instances that are traversing R-T via this shared Replication segment.

In addition, Penultimate Hop Popping (PHP) and implicit null label on the bypass path can be implemented to reduce the PCE programming on the merge point (MP) PCC.

4.3.7. Connecting Replication Segment via Segment List

There could be many nodes between two Replication Segments that do not support SR P2MP Policy or Replication Segment. It is possible to connect two non-adjacent Replication segments via a SR Policy or similar technology using a SID list and rely on IGP forwarding. The SID list can be comprised of any IGP supported segment types (ex: Binding, Adjacency, Node). This information is encoded via the SR-ERO sub-objects and ERO-attributes objects. The last segment in an encoding SID list MUST be a Replication segment.

4.3.8. Tree Deletion

The SR P2MP Policy and its Replication segment can be deleted by the PCC or by the PCE. To delete the SR P2MP Policy all the CP associated to that P2MP policy need to be deleted. The last CP that is being deleted, will delete the SR P2MP Policy Instance as well on the PCE or PCC.

To delete the CPs there are two methods:

1. The CPs can be deleted by deleting all the PTIs associated with them and the last PTI that is deleted will trigger the CP to be deleted.

2. The CP can be deleted entirely and this will delete all the associated PTIs as well.

4.3.8.1. PCC Initiated

For PCC to delete a CP, Root sends a PCRpt message with the R bit of the LSP object set and all the fields of the SR-P2MP-LSPID-TLV set to 0 (indicating to remove all state and PTIs associated with this SR P2MP Policy). The R bit in the LSP Object as defined in [RFC8231], refers to the removal of the LSP as identified by the SR-P2MP-INSTANCE-ID-TLV (defined in this document). An all zero (SR-P2MP-LSPID-TLV defines to remove all the state of the corresponding PLSP-ID.The PCE in response sends a PCInitiate message with R bit in the SRP object set to all nodes along the path to indicate deletion of the entries. In this case the Instance-ID can be set 0 with the R bit set to indicate removing the entire CP and all its PTIs.

For PCC to delete a PTI, Root send a PCRpt message with the R bit of the LSP object set and all the fields of the SR-P2MP-LSPID-TLV set to 0 but the Instance-ID value (indicating to remove the specific PTI associated with this P2MP tunnel). The PCE in response sends a PCInitiate message with R bit in the SRP object SET to all nodes along the path to indicate deletion of the entries. Note in this case the Instance-ID has to be set accordingly with the R bit set to indicate removing the specific PTI. This is useful for the global optimization case where after downloading the optimize PTI and ensuring the PTI is operational the PCC removes the old PTI.

4.3.8.2. PCE Initiated

When PCE is deleting a CP or a PTI it should delete all the replication segments of that CP or PTI as well before it moves to the next CP or PTI.

4.4. PCEP Messages

The objects and TLV’s defined in this document can be included in PCRpt, PCInitiate and PCUpd messages. The inclusion of the Objects and TLVs differ between SR P2MP Policy and Replication segment.

4.4.1. SR P2MP Policy

The format of the PCRpt message is updated as follows:

```
<PCRpt Message> ::= <Common Header>
                    <state-report-list>
```

Where:

```
 <state-report-list> ::=
      <state-report>[<state-report-list>]

 <state-report> ::= [<SRP>]
                     <LSP>
                    [<association-list>]
                    [<end-point-list>]
```

where:

```
 <SRP> is defined in RFC8231

 <LSP> is defined in [RFC8231] section 5.5.1

 <association-list> ::=
      <ASSOCIATION> [<association-list>]

 <end-point-list> ::= [<P2MP-END-POINTS>]
```

Where:

```
 <ASSOCIATION>
        is described in this doc

 <P2MP-END-POINTS>
        is described in this doc
```

The format of the PCUpd message is updated as follows:

```
<PCUpd Message> ::= <Common Header>
                    <update-request-list>

Where:

 <update-request-list> ::= <update-request>
                           [<update-request-list>]

 <update-request> ::= <SRP>
                      <LSP>
                     [<end-point-list>]

Where:

 <SRP> is defined in [RFC8231]

 <LSP> is defined in [RFC8231]section 5.5.1

 <end-point-list> ::= [<P2MP-END-POINTS>]

Where:

 <P2MP-END-POINTS> is described in this doc
```

The format of the PCInitiate message is updated as follows:

```
<PCInitiate Message> ::= <Common Header>
                         <PCE-initiated-lsp-list>

Where:

 <PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>
                              [<PCE-initiated-lsp-list>]

 <PCE-initiated-lsp-request> ::=
                       <PCE-initiated-lsp-instantiation>|
                       <PCE-initiated-lsp-deletion>)

 <PCE-initiated-lsp-instantiation> ::= <SRP>
                                       <LSP>
                                       <association-list>
                                       <end-point-list>
Where:

 <SRP> is defined in RFC8231

 <LSP> is defined in [RFC8231] in section 5.5.1

 <association-list> ::=
             <ASSOCIATION> [<association-list>]

 <end-point-list> ::= [<P2MP-END-POINTS>]

Where:

 <ASSOCIATION> is described in this doc

 <P2MP-END-POINTS> is described in in this doc
```

4.4.2. Replication Segment

The Replication Segment can be constructed via the following PCRpt, PCUpd and PCInitiate messages

NOTE:

* Replication segment does not use a association object

The format of the PCRpt message is updated as follows:

```
   <PCRpt Message> ::= <Common Header>
                       <state-report-list>

   Where:

    <state-report-list> ::=
            <state-report>[<state-report-list>]

    <state-report> ::= [<SRP>]
                        <LSP>
                        <CCI>
                        <intended-path>

   Where:

    <SRP> is defined in [RFC8231]

    <LSP> is defined in [RFC8231] section 5.5.1

    <CCI> is defined in this doc

    <intended-path> ::=
           ((<PATH-ATTRIB><ERO>)[<intended-path>])

   Where:

    <PATH-ATTRIB-ERO> is defined in [draft-ietf-pce-multipath]

    <ERO> is defined in [RFC8664]
```

The format of the PCUpd message is updated as follows:

```
<PCUpd Message> ::= <Common Header>
                    <update-request-list>

Where:

 <update-request-list> ::= <update-request>
                           [<update-request-list>]

 <update-request> ::= <SRP>
                      <LSP>
                      <CCI>
                      <intended-path>

Where:

 <SRP> is defined in [RFC8231]

 <LSP> is defined in [RFC8231] section 5.5.1

 <CCI> is defined in this doc

 <intended-path> ::=
       ((<PATH-ATTRIB><ERO>)[<intended-path>])

Where:

 <PATH-ATTRIB-ERO> is defined in [draft-ietf-pce-multipath]

 <ERO> is defined in [RFC8664]
```

The format of the PCInitiate message is updated as follows:

```
   <PCInitiate Message> ::= <Common Header>
                            <PCE-initiated-lsp-list>

   Where:

    <PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>
                                 [<PCE-initiated-lsp-list>]

    <PCE-initiated-lsp-request> ::=
                           (<PCE-initiated-lsp-instantiation>|
                            <PCE-initiated-lsp-deletion>)

    <PCE-initiated-lsp-instantiation> ::= <SRP>
                                          <LSP>
                                          [<CCI>]
                                          [<intended-path>]
   Where:

    <SRP> is defined in RFC8231

    <LSP> is defined in [RFC8231] section 5.5.1

    <CCI> is defined in section 4.4.3.3

    <intended-path> ::=
               ((<PATH-ATTRIB><ERO>)[<intended-path>])

   Where:

    <PATH-ATTRIB-ERO> is defined in [draft-ietf-pce-multipath]

    <ERO> is defined in [RFC8664]

    <PCE-initiated-lsp-deletion> is as per [RFC8281].
    <attribute-list> is as per [RFC8281].
```

4.4.3. Considerations

A SR P2MP Policy supports add, remove, and full replace of Leaf nodes in a message, described further in section 5.5.2 and with an example in section 8.1. However, a CP and Replication Segment MUST NOT carry delta information. Every PCRpt, PCInitiate and PCUpd message MUST contain the full list of the Leaf nodes, and Segment forwarding information that is needed to build the CP and its Replication segments. This is necessary to ensure that PCE or any new PCE is in sync with the PCC.

4.4.3.1. Path Attribute Object

This document uses [draft-ietf-pce-multipath] to identify each out-going interface in the Replication Segment. In addition each out-going interface can be protected by a backup path. The Path Attributes Object is used to provide the relation between the primary path and its backup path as per draft [draft-ietf-pce-multipath].

Note: Multipath weight TLV MUST NOT be used and SHOULD be ignored when revived. Composite Candidate Path TLV SHOULD NOT be present and SHOULD be ignored if present.

When a replication segment is being updated or new out-going interfaces are added to a specific replication segment, the PCRpt, PCInitiate and PCUpd messages sent via PCEP, maintains the previous ERO Path IDs and generates new Path IDs for new instructions. The PATH IDs are maintained for each specific forwarding instructions until the instructions are deleted. For example: When the first leaf is added, the PCE will update with Path ID 1 to the PCC. When the second leaf is added, according to the path calculated, PCE might just append the existing instruction Path ID 1 with a new Path ID 2 to construct the new PCUpd message.

4.4.3.2. CCI Object

The Central Control Instruction (CCI) Object is used to identify the entire cross connect of Explicit Route Object (ERO) which consist of incoming Replication SID and the set of outgoing Interfaces and their corresponding SIDs and/or SID lists. Any modification to this instruction should use this CCI ID to identify the cross connect uniquely. Leaf nodes and their corresponding Path IDs can be removed from the cross connect identified via the CCI. The CC-ID is assigned by the PCE.

The CCI Object used by the PCE to specify the controller instructions is defined in [RFC9050].
[draft-ietf-pce-pcep-extension-pce-controller-sr] defines CCI object-type for SR-MPLS. This document redefines a new version of the SR-MPLS CCI object-type for SR P2MP Policy in upcoming sections.

5. Object Format

5.1. Open Message Extension

A PCEP speaker indicates its ability to support SR P2MP Policy and Replication Segment during the PCEP initialization phase. Speakers which support SR P2MP Policy and Replication segment object MUST include SR-P2MP-POLICY-CAPABILITY TLV in the OPEN Object.

This draft defines a new SR-P2MP capability TLV with type TBD

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type=TBD         |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Number of Instances      |       number of replication   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Flags            |           reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD

Length: 4

Number of Instances 16 bits - Number of instances the advertising PCEP speaker supports. This is meaningful for PCEs. PCEs can determine the least number of instances that could be created for a SR P2MP policy.

Number of replication 16 bits - number of out going interfaces that the system is capable of having per multicast state.

Flags 16 bits - No Flags currently defined

Upon the receipt of an Open message, the receiving PCEP peer MUST determine whether the suggested PCEP session characteristics (leaf-types) are acceptable. If the suggested leaf-types are not acceptable to the receiving peer, it MUST send an PCEP Error message (PCErr) with Error-Type=2 (Capability not supported) and error-value X (new error type assigned by IANA incompatible SR P2MP leaf types) (See Section 5.5.2 for leaf-types).

5.2. PCE Capabliity SubTLV

If a PCEP speaker advertises SR P2MP Policy Capability then it MUST include the PST = 1 in the PATH-SETUP-TYPE-CAPABILITY TLV as per [RFC8664]

5.3. Association Type Capability

A Assoc-Type-List TLV as per [RFC8697] section 3.4 should be send via PCEP open object with following association type

Association Type Value	Association Name	Reference
TBD1	P2MP SR Policy Association	This document

OP-CONF-Assoc-RANGE (Operator-configured Association Range)should not be set for this association type and must be ignored.

5.4. Symbolic Name TLV

This document reuses symbolic path name from [RFC8231] section 7.3.2. For SR P2MP Policy a symbolic path is unique on the PCC. It is RECOMMENDED for the symbolic path name to be Root, Tree-ID and CP Discriminator values.

5.5. SR P2MP Policy

5.5.1. P2MP Policy Association Group

Two ASSOCIATION object types for IPv4 and IPv6 are defined in [RFC8697]. The ASSOCIATION object includes "Association type" indicating the type of the association group. This document adds a new Association type. Association type = TBD1 "SR P2MP Policy Association Type" for SR Policy Association Group (P2MP SRPAG).

For PCC-initiated SR P2MP Policy, the ASSOCIATION object is present in the first PCRpt message that is sent by the PCC to PCE to indicate the existence of the SR P2MP Policy and its CP. This first PCRpt message does not have a corresponding PCUpdate message but it does include the Association object accordingly.

The Association Source SHOULD be set to the root address of the P2MP tree.

In par with [RFC9862] section 4.2, P2MP policy reuses the four TLVs used in the SRPA object. P2MP policy also redefines the extended association ID TLV:

1. SRPOLICY-POL-NAME TLV: (optional) encodes SR P2MP Policy Name

2. SRPOLICY-CPATH-ID TLV: (mandatory) encodes SR P2MP Policy Candidate Path Identifier

3. SRPOLICY-CPATH-NAME TLV: (optional) encodes SR P2MP Policy Candidate Path name.

4. SRPOLICY-CPATH-PREFRENCE TLV: (optional) encodes SR P2MP Policy Candidate Path preference value.

5. In addition to above the extended association ID TLV has been modified to address the SR P2MP Policy

5.5.1.1. Extended Association ID TLV

In par with [RFC9862] the Extended Association ID TLV MUST be included and it MUST be in the following format for the SR P2MP Policy

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 31            |            Length = 4         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            TREE-ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 4 byte

Tree-ID: identifies the P2MP Tree which the Replication segment is part of

5.5.2. P2MP-END-POINTS Object

In order for the Root node to indicate operations of its Leaf nodes (Add/Remove/Replace-all), the PCRpt and PcInititiate messages are extended to include P2MP End Point <P2MP End-points> Object which is defined in [RFC8306]

The absence of the P2MP-END-POINTS Object means that there is no change in the leaf endpoint of the policy and the PCEP speaker MUST NOT consider the absence of the object as an indication of removal of the endpoints.

IPV4-P2MP END-POINTS:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Leaf type                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            ...                                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPV6-P2MP END-POINTS:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Leaf type                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Source IPv6 address (16 bytes)                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            ...                                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Leaf Types (derived from [RFC8306] section 3.3.2) :

1. New leaves to add (leaf type = 1)

2. Old leaves to remove (leaf type = 2)

3. the entire pce leaf list is overwritten and replaced with the new leaf list (leaf type = 5)

Note a PCE speaking node MUST NOT combine leaf type 1 and 2 with leaf type 5.

Note a PCE speaking node SHOULD NOT have the same node present in the leaf type 1 and 2 if both leaf types are present.

A given P2MP END-POINTS object gathers the Leaf nodes of a given type. A SR P2MP PCRpt MAY mix different types of Leaf nodes by including several P2MP END-POINTS objects. The END-POINTS object body has a variable length. These are multiples of 4 bytes for IPv4, multiples of 16 bytes, plus 4 bytes, for IPv6.

5.6. P2MP Policy and Replication Segment Identifier Object and TLV

Both SR P2MP Policy and Replication Segment are identified via the LSP object and more precisely via the SR-P2MP-LSPID-TLV

The SR P2MP Policy uses the PLSP-ID to identify the CP and the Instance-ID to identify a PTI within the CP.

A Replication Segment uses the SR-P2MP-LSPID-TLV to identify and correlate a Replication Segment to a P2MP Policy

As it was noted previously, for the Root the SR P2MP Policy and the Replication Segment is downloaded via the same PCUpd message.

5.6.1. Extension of the LSP Object, SR-P2MP-LSPID-TLV

The LSP Object is defined in Section 7.3 of [RFC8231]. It specifies the PLSP-ID to uniquely identify an LSP that is constant for the life time of a PCEP session. Similarly, for a P2MP Policy, the PLSP-ID identify a Candidate Path uniquely within the SR P2MP policy.

The LSP Object MUST include the new SR-P2MP-INSTANCE-ID-TLV (IPV4/IpV6) defined in this document below. This is a variation to the P2MP object defined in [draft-ietf-pce-stateful-pce-p2mp]

IPV4-SR-P2MP-INSTANCE-ID-TLV:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=TBD            |           Length=10           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Root                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Tree-ID                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Instance-ID           | Reserved      |  Flags    |R|A|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6-SR-P2MP-INSTANCE-ID-TLV :

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=TBD            |           Length=22           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                  Root                                         |
+                          (16 octets)                          +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Tree-ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Instance-ID           | Reserved      |  Flags    |R|A|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The type (16-bit) of the TLV is TBD (need allocation by IANA).

Root: Source Router IP Address

Tree-ID: Unique Identifier of this P2MP LSP on the Root.

Instance-ID : Identifier of PTI, Contains 32 Bit instance ID. Instance-id 0 is reserved.

Reserved: 8 bits reserved for future use.

Flags: 8 bits, A - Activate the Instance-ID, R - Remove the Instance-ID

At any given time, only one PTI MUST be active. Activating one PTI entails deactivating the other PTI, with the condition that the active PTI Instance-ID MUST have a non-zero value.

The (A) flag is meaningful for Root PCC and PCE. PCE MUST be setting (A) flag in the PCUpd containing SR-P2MP-INSTANCE-ID TLV for activating the PTI. The decision regarding when to set the (A) flag can be made locally on the PCE. E.g., this decision can be based on factors such as receiving PCRpt messages from all PCCs for the new PTI or utilizing a timer-based approach to ensure that the data plane is completely configured on all PCCs. It’s important to note that determining the appropriate timing for activating the new PTI is not within the scope of this document. After the activation of the SR P2MP Policy any PCUpd MUST include the (A) flag in the P2MP-Instance TLV.

Root PCC MUST set the (A) flag in the PCRpt as a response to receiving a PCUpd message with the (A) flag set.

If a PCE receives a PCRpt with the (A) flag set in response to a PCUpd message that did not have the (A) flag set, then PCE MUST treat this as an error. In such a case, PCE MUST send an PCEP Error message (PCErr) with Error-Type=10 (Reception of an Invalid Object) and error-value (X) (Invalid active instance).

For Transit or Leaf PCCs, receipt of a PCUpd message with the (A) flag MUST be treated as an error. Transit or Leaf PCCs MUST send an PCEP Error message (PCErr) with Error-Type=19 (Invalid Operation) and error-value (X) (Attempted activating instance on Transit or leaf PCC).

Figure 3 presents an example exchange of SR-P2MP-LSPID-TLV.

```
                  +-+-+                     +-+-+
                  |PCC|                     |PCE|
                  +-+-+                     +-+-+
                    |                         |
1) LSP state Report | -------- PCRpt ------> |
   With PLSP-ID and |  (SRP,                  |
   Instance-ID      |   LSP (SR-P2MP-LSPID),  |
                    |   P2MP-END-POINT)       |
                    |                         |
                    | <------- PCUpd -------  |2) PCUpd message sent
                    |  (SRP,                  |   to the PCC with
                    |   LSP (SR-P2MP-LSPID),  |   Path info and activating
                    |   Association,          |   instance.
                    |   P2MP SR Pol. ID TLV,  |
                    |   CPATH_ID TLV,         |
                    |   P2MP-END-POINT,       |
                    |   CCI, PATH_ATTRIB,     |
                    |   SR-ERO)               |
                    |                         |
                    |                         |
3) LSP State Report |---- PCRpt message ---->|
  (echoing Instance |                         |
    Active)         |                         |
                    |                         |
```

5.7. Replication Segment

As per [RFC9524] a replication segment has a next-hop-group which MAY contain a single outgoing replication SID or a list of SIDs (sr-policy-sid-list) In either case there needs to be a replication SID identifying the replication state on a downstream replication node. Two replication segments can be directly connected or connected via a unicast SR domain.

5.7.1. Message format

The format of a Replication Segment message encoding is similar to SR P2MP Policy. However, the SR P2MP Policy contains the association object and the Replication Segment message does not contain the association object. In addition, the Replication Segment carries the CCI object to identify a P2MP cross connect. The Replication Segment is distributed individually to the Root, Transit and Leaf nodes without the SR P2MP Policy. The Replication Segment uses SR-P2MP-LSPID-TLV as its identifier. The TLV is coded differently for shared and non-shared case

5.7.2. CCI Object

The CCI Object as defined in [RFC9050] is used to identify a forwarding instruction in the Replication Segment. A forwarding instruction is incoming SID and a set of outgoing branches.

The CCI Object can be include in Reports, initiate and Update messages for Replication Segments.

This document redefines a new version of the SR-MPLS CCI object-type [draft-ietf-pce-pcep-extension-pce-controller-sr] for P2MP Policy. The label in the CCI Object is the incoming SID. The outgoing SIDs are defined by the ERO Objects.

CCI Object-Type is 3 for SR P2MP Policy.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            CC-ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MT-ID    |    Algorithm  | role  |     flags         |V|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       SID/Label/Index                         |
+---------------------------------------------------------------+
|                                                               |
//                         Optional TLV                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flags:

The V and the L flags are defined as per [draft-ietf-pce-pcep-extension-pce-controller-sr]

The node action and role of ingress, transit, leaf or bud, is indicated via the 4 bit roles field

* Head, role type = 1

* Transit, role type = 2

* Leaf, role type = 3

* Bud, role type = 4

5.7.3. SR-ERO Rules

Forwarding information of a Replication Segment can be configured and steered via many different mechanisms. RFC [RFC8664] defines the NAI types.

As an example a replication SID can be steered via:

1. Replication SID steered with an IPv4/IPv6 directly connected nexthop (RFC 8664 NAI type 3, 4, 6 (adjacency))

    * In this case there will two SR-ERO in the ERO Object, with the Replication SID SR-ERO at the bottom and the IPv4/IPv6 SR-ERO on the top.

2. Replication SID steered with an IPv4/IPv6 loopback address that reside on the directly connected router. (NAI type 1..2 (node))

    * In this case there will two SR-ERO in the ERO Object, with the Replication SID SR-ERO at the bottom and the IPv4/IPv6 SR-ERO on the top.

3. Replication SID steered with unnumbered IPv4/IPv6 directly connected Interface (NAI type 5 (adjacency unnumbered)

4. Replication SID steered via a SR adjacency or node SID

    * In this case even a sid-list can be used to traffic engineer the path between two Replication Segment

    * The Replication SID SR-ERO is at the bottom while the segments describing the path are on top in order.

6. IANA Consideration

6.1. PCEP P2MP Association type

This draft defines a new Association type for SR P2MP Policy. IANA is requested to allocate a new value from the existing IANA Registry "ASSOCIATION TYPE Field" within the "Path Computation Element Protocol (PCEP) Numbers" registry group.

Type	Name	Reference
9	SR P2MP Policy Association	This document

6.2. Endpoint Type

[RFC8306] specified the P2MP END-POINTS object but did not create a registry for the 32-bit Leaf type field. This document establishes the registry and populates it with values from [RFC8306] and adds a new Leaf type. IANA is requested to create a new "Endpoint Leaf Types" registry with the allocation policy as IETF Review [RFC8126]. This new registry contains the following values:

Value	Description	Reference
0	Reserved	This document
1	New leaves to add	RFC 8306
2	Old leaves to remove	RFC 8306
3	Old leaves whose path can be modified/reoptimized	RFC 8306
4	Old leaves whose path must be left unchanged	RFC 8306
5	All old leaves overwritten and replaced with the new	This document

To keep it consistent with the Generalized Endpoint Types [RFC8779], this draft defines a new Endpoint Type in the "Generalized Endpoint Types" registry as follows:

Value	Type	Reference
5	Point-to-Multipoint with leaf type 5	This document

The Authors are requesting value 5 for this new endpoint type.

6.3. PCEP TLV Type Indicators

This draft extends the PCEP OPEN object by defining a new optional TLV to indicate the PCE’s capability to perform SR-P2MP path computation.

Further, this draft defines two new TLVs for Identifying the SR P2MP Policy and the Replication Segment with IPv4 or IPv6 root address.

IANA is requested to allocate a new value from the IANA Registry "PCEP TLV Type Indicators"

TLV Type Value	Description	Reference
73	SR-P2MP-POLICY-CAPABILITY	This document
74	IPV4-SR-P2MP-INSTANCE-ID TLV	This document
75	IPV6-SR-P2MP-INSTANCE-ID TLV	This document

6.4. New CCI Object Type

This draft defines a new CCI Object type SR P2MP Policy.

IANA is requested to allocate new codepoints in the "PCEP Objects" sub-registry as follows:

Object Class Value	Name	Reference
44	CCI Object Object-Type 3 : SR P2MP Policy	This document

7. Security Considerations

The security considerations described in [RFC5440], [RFC8231], [RFC8281], [RFC8664], [RFC8697], [RFC9256] and [RFC9524] are applicable to this specification.

As per [RFC8231], it is RECOMMENDED that these PCEP extensions can only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using Transport Layer Security (TLS) [RFC8253].

No additional security measures are required for SR P2MP Policy.

8. Implementation Status

Note to the RFC Editor: please remove this section before publication. This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC7942 . The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist. According to RFC7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

8.1. Cisco Implementation

Cisco has implemented this document on both IOS XR PCE and PCC. Both PCE-Initiated and PCC-Initiated SR P2MP Policy types have been implemented. Only one Candidate Path can be created with the SR P2MP Policy and only one PTI is supported within a CP.

9. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC8231], [RFC8664], and [RFC9256] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

9.1. Operatoin Consideration

With P2MP policy any router that is acting as a Root, Transit (replication point) and Leaf needs to have a PCEP connectivity to the controller. This might not be the case for unicast where only the node that is instantiating the SR Policy and the ordered list of the segments needs PCEP connectivity. An operator might need to enable PCEP on more nodes for enabling P2MP Policy on a network that is already supporting SR Policy, as such the operator needs to ensure that the controller can handle the extra PCEP connection scale. Otherwise the PCE control needs to be able to The operation consideration of P2MP policy is in par with [RFC5440], [RFC8231], [RFC8664], [RFC9256] and [RFC9862].

9.2. Liveness Detection and Monitoring

Since P2MP Policy does not use any underlay signaling to detect failure it is recommended to use (Operations, Administration, and Maintenance) OAM features to detect failure on a tree. [draft-ietf-pim-p2mp-policy-ping] defines extensions to Ping and Traceroute mechanism for SR P2MP Policy with MPLS encapsulation to provide OAM capabilities. The extensions enable operators to verify connectivity, diagnose failures and troubleshoot forwarding issues within SR P2MP Policy multicast trees. The implementation can trigger P2MP Policy Ping or Traceroute periodically to test the end-to-end connectivity and trigger a global optimization of the tree from the Root to the Leaves. Any local failure can be detected via monitoring the state of the outgoing links and triger a local optimization. In addition, as the controller gets updated with the network topology it also triggers a global optimization of the tree based on new discovered optimal paths from the Root to the Leaves.

9.3. Verify Correct Operations

Operation verification requirements already listed in [RFC5440], [RFC8231], [RFC8664], [RFC9256] are applicable to mechanisms defined in this document.

An implementation MUST allow the operator to view SR P2MP Policy Identifier and each SR Replication segment Candidate Path Identifier advertised in PCEP.

An implementation SHOULD allow the operator to view the capabilities defined in this document.

An implementation SHOULD allow the operator to view each Replication segment Candidate Path associated with specific SR P2MP Policy.

9.4. Requirements On Other Protocols

The PCEP extensions defined in this document do not imply any new requirements on other protocols.

9.5. Impact On Network Operations

The mechanisms defined in [RFC5440], [RFC8231], [RFC9256] also apply to the PCEP extensions defined in this document.

10. Acknowledgments

The authors would like to thank Tanmoy Kundu and Stone Andrew at Nokia and Tarek Saad at Cisco for their feedback and major contribution to this draft.

11. Contributors

Tarek Saad
Cisco Systems
Email: tsaad.net@gmail.com

Saranya Rajarathinam
Saranya Rajarathinam
Nokia
Email: saranya.Rajarathinam@nokia.com

12. References

12.1. Normative References

[draft-ietf-pce-multipath]
"M. Koldychev, S. Sivabalan, T. Saad, H. Bidgoli, B. Yadav, S. Peng, G. Mishra "PCEP Extensions for Signaling Multipath Information"", November 2022.

[draft-ietf-pim-p2mp-policy-ping]
"h.bidgoli, z.ali, z.zhang, A.Budhirajac, D.Voyer "Segment Routing MPLS P2MP Policy Ping"".

[draft-ietf-pim-sr-p2mp-policy]
"D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang, "Segment Routing Point-to-Multipoint Policy"", October 2019.

[RFC2119] ""Key words for use in RFCs to Indicate Requirement Levels"", March 1997.

[RFC3209] "D. Awduche, L. Berger, T. Li, V. Srinivasan, G. Swallow "RSVP-TE: Extensions to RSVP for LSP Tunnels"", December 2001.

[RFC5440] "JP. Vasseur, JL. Le Roux "Path Computation Element (PCE) Communication Protocol (PCEP)"", March 2009.

[RFC6513] "E. Rosen, R. Aggerwal "Multicast in MPLS/BGP IP VPNs"", February 2012.

[RFC7988] "E. Rosen, K. Subramanian, Z. Zhang "Ingress Replication Tunnels in Multicast VPN"", October 2016.

[RFC8126] "M. Cotton, B. Leiba, T. Narten "Guidelines for Writing an IANA Considerations Section in RFCs"", June 2017.

[RFC8174] ""Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words"", May 2017.

[RFC8231] "E. Crabbe, I. Minei, J. Medved, R. Varga "PCEP Extensions for Stateful PCE"", September 2017.

[RFC8253] "D. Lopez, Q. Wu. D.Dhody "Usage of TLS to Provide a Secure Transport for the PCEP"", October 2017.

[RFC8281] "E. Crabbe, I. Minei, S. Sivabalan, R. Varga "PCEP Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model"", December 2017.

[RFC8306] "Q. Zhao, D. Dhody, R. Palleti, D.King "PCEP for Point-to-Multipoint Traffic Engineering Label Switched Paths"", November 2017.

[RFC8664] "S. Sivabalan, C. Filsfils, J. Tantsura, W. Henderickx, J. Hardwick "PCEP Extensions for Segment Routing"", December 2019.

[RFC8697] "I. Minei, E. Crabbe, S. Sivabalan, H. Ananthakrishnan, D. Dhody, Y. Tanaka "PCEP Extensions for Establishing Relationships between Sets of LSPs"", January 2020.

[RFC8779] "C. Margaria, O. Gonzalez, F. Zhang "PCEP extensions for GMPLS"", July 2020.

[RFC9050] "Z. Li, S. Peng, M. Negi, Q. Zhao, C. Zhou "PCEP Procedures and Extensions for Using the PCECC"", July 2021.

[RFC9256] "C. Filsfils, K. Talaulikar, D. Voyer, A. Bogdanov, P. Mattes "Segment Routing Policy Architecture"", July 2022.

[RFC9524] "D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang, "SR Replication Segment for Multi-point Service Delivery"", July 2020.

[RFC9862] "M. Koldychev, S. Sivabalan, C. Barth, S. Peng, H. Bidgoli "PCEP Extensions for SR Policy Candidatte Paths"", October 2025.

12.2. Informative References

[draft-ietf-pce-pcep-extension-pce-controller-sr]
"Z. Li, S. Peng, M. negi, Q. Zhao, C. Zhau "PCEP Extensions for Using PCE as a PCECC for SR MPLS SID"".

[draft-ietf-pce-stateful-pce-p2mp]
"U. Palle, D. Dhody, Y.Tanaka, V. Beeram "Protocol Extensions for Stateful PCE usage for Point-to-Multipoint Traffic Engineering Label"", April 2019.

Appendix A. Packet Examples

A.1. Report for Leaf Add

This is an example of PCC initiated P2MP Policy. The PCC will send a Report message to the PCE to initiat a P2MP Policy with a set of leaves that are discovered via Next Generation MVPN procedures as per [RFC6513].

In addition, since the PCC is initiating the P2MP Policy, it does populate the PLSP-ID for the Candidate Path. PCC will leave the Instance-ID for the PTI to 0 and the Instance-ID is assigned by the PCE.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Flags = 0                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SRP-ID-number  = 0                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                             | PST = TBD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                           <LSP OBJECT>
|                PLSP-ID = 1            |     A:1,D:1,N:1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=17             |           Length=<var>        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             symbolic path name                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type=TBD          |  Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Root = A                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Tree-ID = Y                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Instance-ID = 0                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                       <P2MP END POINT OBJECT>
|                          Leaf type = 5                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source IPv4 address = A                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IPv4 address = D               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IPv4 address = E               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A.2. SR P2MP Policy Candidate Path Init

The following packet is the PCE creating the CP for the SR P2MP Policy and downloading the Replication Segment with the same message on the root.

It should be noted combining the SR P2MP Policy CP creation and Replication Segment only is possible on the root.

As such this message contains both association object and the CCI object. The CCI Object contains the incoming Binding SID and wraps all the Path Attribute messages and their corresponding EROs.

The PLSP-ID are populated with the same ID as the previous PCC report message and the Instance-ID is assigned by the PCE.

```
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          Flags = 0                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        SRP-ID-number  = 1                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                             | PST = TBD       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                          <LSP OBJECT>
    |                PLSP-ID = 1            |     A:1,D:1,N:1,C:0   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Type=17             |           Length=<var>        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            symbolic path name                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type=TBD          |  Length                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Root =A                                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Tree-ID = Y                                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Instance-ID = L1                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                         <ASSOCIATION OBJECT>
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Reserved              |             Flags           |0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Association type= SR-P2MP-PAG |      Association ID = z       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             IPv4 Association Source = <pce-address>           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type=P2MP SR Policy ID    |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Root  = A                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          TREE-ID = 0                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Type=P2MP SR Policy Name   |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                     P2MP SR Policy Name                       ~
    |                                                               |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=P2MP SR Pol Cand-path ID  |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|ProtOrigin 10  |    Reserved                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Originator ASN                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Originator Address = <pce-address>      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Discriminator = 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=P2MP SR Pol Cand-path Name|             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~               P2MP SR Policy Candidate Path Name              ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=P2MP SR Pol Cand-Path Pre |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Preference = 100 <default>          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                       <P2MP END POINT OBJECT>
|                          Leaf type = 5                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 address = A                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address = D                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address = E                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                     <CCI OBJECT>
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             CC-ID = X                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved1            |             Flags        |0|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Label = 0             |     Reserved2         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                <PATH-ATTRIBUTES OBJECT>
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Flags                         | Oper|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
        |                              ERO-path Id = 1                  |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |             Type              |             Length            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |       Backup Path Count = 1   |             Flags           |0|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         Backup Path ID 2                      |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |           Type=Node Role      |           Length=4            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |Role = ingress |                Reserved                       |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ERO-OBJECT>
                          <SR-ERO-SUB OBJECT>

        |L|   Type=36   |     Length    |  NT= 1|     Flags   |0|0|1|0|0|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                ipv4-address  = NHD1           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |L|   Type=36   |     Length    |  NT= 0|     Flags   |0|1|0|0|0|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         SID = d1                              |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        <PATH-ATTRIBUTES OBJECT>
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         Flags                           | Oper|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                              ERO-path Id = 2                  |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |             Type              |             Length            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |       Backup Path Count = 0   |             Flags           |1|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |           Type=TBD            |           Length=4            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |Role = ingress |                Reserved                       |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ERO-OBJECT>
                          <SR-ERO-SUB OBJECT>

        |L|   Type=36   |     Length    |  NT= 1|     Flags   |0|0|1|0|0|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                ipv4-address  = NHD2           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |L|   Type=36   |     Length    |  NT= 0|     Flags   |0|1|0|0|0|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         SID = d2                              |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A.3. Replication segment PCE Initiated on Transit and LEaves

The following packet examples shows the Replication Segment initiation via a PCE on transit nodes and/or leaves node.

Note:

1. This packet is generated from PCE to instantiate the Replication segment, as such the PLSP-ID is set to zero. PCC will assign these value and report them back to PCE.

2. The Instance-ID was assigned by the PCE for the entire PTI path on the root, transit and leaves (P2MP tree)

3. This is a Replication Segment instantiation as such there is no association object.

4. The LSP Object Root A and Tree-ID Y associates this Replication Segment to the corresponding CP and PTI on the root node.

there is no association object present in the packet.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Flags = 0                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SRP-ID-number  = 1                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                             | PST = TBD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                       <LSP OBJECT>
|                 PLSP-ID = 0           |     A:1,D:1,N:1,C:0   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=17             |           Length=<var>        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             symbolic path name                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type=TBD          |  Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Root =A                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Tree-ID = Y                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Instance-ID = L1                       |
```

```
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                             <CCI OBJECT>
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                            CC-ID = X                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |          Reserved1            |             Flags         |0|0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                 Label = d1            |     Reserved2         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        <PATH-ATTRIBUTES OBJECT>
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Flags                           | Oper|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ERO-path Id = 1                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Type              |             Length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |       Backup Path Count = 1   |             Flags           |0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Backup Path ID 2                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           Type=TBD            |           Length=4            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Role           |                 Reserved                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ERO-OBJECT>
                          <SR-ERO-SUB OBJECT>

       |L|   Type=36   |      Length    |  NT= 1|     Flags   |0|0|1|0|0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                ipv4-address  = NHE1           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |L|   Type=36   |      Length    |  NT= 0|     Flags   |0|1|0|0|0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         SID = e1                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        <PATH-ATTRIBUTES OBJECT>
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Flags                           | Oper|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ERO-path Id = 2                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Type              |             Length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |       Backup Path Count = 0   |             Flags           |1|
```

```
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           Type=TBD            |           Length=4            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Role           |                 Reserved                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                <ERO-OBJECT>
                              <SR-ERO-SUB OBJECT>

       |L|   Type=36   |     Length    |  NT= 1|     Flags   |0|0|1|0|0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                ipv4-address  = NHE2           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |L|   Type=36   |     Length    |  NT= 0|     Flags   |0|1|0|0|0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         SID = e2                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix B. Example Workflows

```
                +-------+                            +-------+
                  |PCC    |                            |  PCE  |
                  |Root   |                            +-------+
           +------|       |                                |
           | PCC  +-------+                                |
           | Transit| |                                    |
    +------|        | |---PCRpt,PLSP-ID=1,SRP-ID=1--------->| PCECC LSP
    |PCC   +--------+ |   N=1,root-addr,Tree-ID=a,          | SR-Policy
    |        |  |     |   Instance-ID =0,                   | Report to
    |Leaf    |  |     |   P2MP-end-points(LeafType=5)(optnl)| PCE
    +--------+  |     |   association-obj                   |
        |       |     |                                     |
        |       |     |<--PCUpdate,PLSP-ID=1, SRP-ID =1,    | Update
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| CP
        |       |     |   P2MP-end-points, association-obj  |
        |       |     |                                     |
        |       |     |-------PCRpt,PLSP-ID=1, SRP-ID = 1,->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |      P2MP-end-points(LeafType=5)    |
        |       |     |      association-object             |
        |       |     |                                     |
        |<--------------PCInitiate,PLSP-ID=0, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Leaf
        |       |     |  CC-ID=Z,C=0,                       | Replication
        |       |     |  O=0,L=z,path-attribute,ERO,SR-ERO  | Segment(RS)
        |       |     |                                     |
        |-------------------PCRpt,PLSP-ID=1-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Z,Label=z,O=0,          |
```

```
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |<-------PCInitiate,PLSP-ID=0, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Transit
        |       |     |  CC-ID=Y,C=0,                       | RS
        |       |     |  O=0,L=y,path-attribute,ERO,SR-ERO  |
        |       |     |                                     |
        |       |-------------PCRpt,PLSP-ID=2-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Y,Label=y,O=0,          |
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |     |<--PCInitiate,PLSP-ID=1,             | Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Root
        |       |     |   CC-ID=X,C=0,                      | RS
        |       |     | O=0,L=x,P2MP-end-                   |
        |       |     |   points(LeafType=5),path-attribute,|
        |       |     |   ERO,SR-ERO                        |
        |       |     |                                     |
        |       |     |-------PCRpt,PLSP-ID=1-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |    CC-ID=X,Label=x,O=0,             |
        |       |     |      P2MP-end-points(LeafType=5)    |
        |       |     |      path-attriute,ERO,SR-ERO       |
        |       |     |                                     |
        |       |     |<--PCUpdate,PLSP-ID=1, SRP-ID =2,    |
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Activate
        |       |     |   P2MP-end-points                   | CP to last
        |       |     |                                     | RS
        |       |     |-------PCRpt,PLSP-ID=1, SRP-ID =2, ->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |      P2MP-end-points(LeafType=5)    |
        |       |     |                                     |
```

Note that on transit / leaf Initiate is with PLSP-ID = 0. Therefore PLSP-ID is locally unique to a node. It should be noted that the CC-ID does not need to be constant across all nodes that make up the path.

PCE-Initiated workflow

```
                 +-------+                             +-------+
                  |PCC    |                             |  PCE  |
                  |Root   |                             +-------+
           +------|       |                                 |
           | PCC  +-------+                                 |
           | Transit| |                                     |
    +------|        | |                                     | PCECC LSP
```

```
    |PCC   +--------+ |                                     |
    |        |  |     |                                     |
    |Leaf    |  |     |                                     |
    +--------+  |     |                                     |
        |       |     |<--PCInitiate,PLSP-ID=0,             | Initiate
        |       |     |   root-addr,Tree-ID=0,Instance-ID=b,| CP
        |       |     |   P2MP-end-points, association-obj  |
        |       |     |                                     |
        |       |     |-------PCRpt,PLSP-ID=1,------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |      P2MP-end-points(LeafType=5)    |
        |       |     |      association-object,            |
        |       |     |                                     |
        |       |     |<--PCUpdt,PLSP-ID=1,                 | Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Root RS
        |       |     |   CC-ID=X,C=0,                      |
        |       |     | O=0,L=x,P2MP-end-                   |
        |       |     |   points(LeafType=5),path-attribute,|
        |       |     |   ERO,SR-ERO                        |
        |       |     |                                     |
        |       |     |-------PCRpt,PLSP-ID=1-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |    CC-ID=X,Label=x,O=0,             |
        |       |     |      P2MP-end-points(LeafType=5)    |
        |       |     |      path-attriute,ERO,SR-ERO       |
        |       |     |                                     |
        |<--------------PCInitiate,PLSP-ID=0, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Leaf RS
        |       |     |  CC-ID=Z,C=0,                       |
        |       |     |  O=0,L=z,path-attribute,ERO,SR-ERO  |
        |       |     |                                     |
        |--------------------PCRpt,PLSP-ID=1-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Z,Label=z,O=0,          |
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |<-------PCInitiate,PLSP-ID=0, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Transit RS
        |       |     |  CC-ID=Y,C=0,                       |
        |       |     |  O=0,L=y,path-attribute,ERO,SR-ERO  |
        |       |     |                                     |
        |       |-------------PCRpt,PLSP-ID=2-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=c,|
        |       |     |       CC-ID=Y,Label=y,O=0,          |
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |<-------PCInitiate,PLSP-ID=0, -------------|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
```

```
        |       |     |  CC-ID=Y,C=0,                       |
        |       |     |  O=0,L=y,path-attribute,ERO,SR-ERO  |
        |       |     |                                     |
        |       |-------------PCRpt,PLSP-ID=2-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Y,Label=y,O=0,          |
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |     |<--PCUpdate,PLSP-ID=1, SRP-ID =1,    | Bind and
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Activate
        |       |     |   P2MP-end-points,                  | CP to last
        |       |     |                                     | RS
        |       |     |-------PCRpt,PLSP-ID=1, SRP-ID = 1,->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |      P2MP-end-points(LeafType=5)    |

   MBB Workflow:

Common (PCE-INIT, PCC-INIT) MBB

                 +-------+                             +-------+
                  |PCC    |                             |  PCE  |
                  |Root   |                             +-------+
           +------|       |                                 |
           | PCC  +-------+                                 |
           | Transit| |                                     |
    +------|        | |                                     | PCECC LSP
    |PCC   +--------+ |                                     |
    |        |  |     |                                     |
    |Leaf    |  |     |                                     |
    +--------+  |     |                                     |
        |<---------------PCInitiate,PLSP-ID=1, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| new RS on
        |       |     |  CC-ID=Z1,C=0,                      | Leaf
        |       |     |  O=0,L=z1,path-attribute,ERO,SR-ERO |
        |       |     |                                     |
        |---------------------PCRpt,PLSP-ID=1-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Z1,Label=z1,O=0,        |
        |       |     |      path-attribute,ERO,SR-ERO      |
        |       |     |                                     |
        |       |<-------PCInitiate,PLSP-ID=2, -------------| Download
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| new RS on
        |       |     |  CC-ID=Y1,C=0,                      | Transit
        |       |     |  O=0,L=y1,path-attribute,ERO,SR-ERO |
        |       |     |                                     |
        |       |-------------PCRpt,PLSP-ID=2-------------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
```

```
         |       |     |       CC-ID=Y1,Label=y1,O=0,        |
         |       |     |      path-attribute,ERO,SR-ERO      |
         |       |     |                                     |
         |       |     |<--PCInitiate,PLSP-ID=1,             | Download
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| new RS on
         |       |     |   CC-ID=X1,C=0,                     | Root
         |       |     | O=0,L=x1,P2MP-end-                  |
         |       |     |   points(LeafType=5),path-attribute,|
         |       |     |   ERO,SR-ERO                        |
         |       |     |                                     |
         |       |     |-------PCRpt,PLSP-ID=1-------------->|
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
         |       |     |    CC-ID=X1,Label=x1,O=0,           |
         |       |     |      P2MP-end-points(LeafType=5)    |
         |       |     |      path-attriute,ERO,SR-ERO       |
         |       |     |                                     |
         |       |     |<--PCUpdate,PLSP-ID=1, SRP-ID =1,    | Bind and
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| Activate
,        |       |     |   P2MP-end-points,                  | CP to last
         |       |     |                                     | RS
         |       |     |-------PCRpt,PLSP-ID=1, SRP-ID = 1,->|
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
         |       |     |      P2MP-end-points(LeafType=5)    |
         |       |     |                                     |
         |       |     |<--PCInitiate,PLSP-ID=1,R=1          | Remove
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| the old RS
         |       |     |   CC-ID=X1,C=0                      | from Leaf
         |       |     | O=0,L=x1,P2MP-end-                  |
         |       |     |   points(LeafType=5),path-attribute,|
         |       |     |   ERO,SR-ERO                        |
         |       |     |                                     |
         |       |     |-------PCRpt,PLSP-ID=1, R=1--------->|
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
         |       |     |    CC-ID=X1,Label=x1,O=0,           |
         |       |     |      P2MP-end-points(LeafType=5)    |
         |       |     |      path-attriute,ERO,SR-ERO       |
         |       |     |                                     |
         |       |<-------PCInitiate,PLSP-ID=2, R=1----------| Remove the
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| old RS from
         |       |     |  CC-ID=Y1,C=0,                      | Transit
         |       |     |  O=0,L=y1,path-attribute,ERO,SR-ERO |
         |       |     |                                     |
         |       |-------------PCRpt,PLSP-ID=2, R=1--------->|
         |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
         |       |     |       CC-ID=Y1,Label=y1,O=0,        |
         |       |     |      path-attribute,ERO,SR-ERO      |
         |       |     |                                     |
         |<---------------PCInitiate,PLSP-ID=1,R=1-----------| Remove the
```

```
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,| old RS from
        |       |     |  CC-ID=Z1,C=0,                      | Root
        |       |     |  O=0,L=z1,path-attribute,ERO,SR-ERO |
        |       |     |                                     |
        |--------------------PCRpt,PLSP-ID=1,R=1---------->|
        |       |     |   root-addr,Tree-ID=a,Instance-ID=b,|
        |       |     |       CC-ID=Z1,Label=z1,O=0,        |
        |       |     |      path-attribute,ERO,SR-ERO      |
```

Authors’ Addresses

Hooman Bidgoli (editor)
Nokia
Ottawa
Canada
Email: hooman.bidgoli@nokia.com

Daniel Voyer
Cisco Systems
Montreal
Canada
Email: davoyer@cisco.com

Anuj Budhiraja
Cisco System
San Jose,
United States of America
Email: abudhira@cisco.com

Rishabh
Arrcus
San Jose,
United States of America
Email: rishabh@arrcus.com

Siva Sivabalan
Ciena
Ottawa
Canada
Email: ssivabal@ciena.com