Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2021

H. Bidgoli, Ed.
Nokia
V. Voyer
Bell Canada
S. Rajarathinam
Nokia
E. Hemmati
Cisco System
T. Saad
Juniper Networks
S. Sivabalan
Ciena

October 30, 2020

PCEP extensions for p2mp sr policy
draft-hsd-pce-sr-p2mp-policy-02

Abstract

SR P2MP policies are 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 Leaves.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The draft [draft-ietf-pim-sr-p2mp-policy] defines a variant of the SR Policy [draft-ietf-spring-segment-routing-policy] for constructing a P2MP segment to support multicast service delivery.

A Point-to-Multipoint (P2MP) Policy connects a Root node to a set of Leaf nodes, optionally through a set of intermediate replication nodes. We also define a Replication segment [draft-ietf-spring-sr-replication-segment], which corresponds to the state of a P2MP segment on a particular node, as an example the forwarding instructions for the replication SID.

A P2MP Policy is relevant on the root of the P2MP Tree and it contains candidate paths. The candidate paths are made of path-instances and each path-instance is constructed via replication segments. These replication segments are programmed on the root, leaves and optionally intermediate replication nodes.

It should be noted that two replication segments can be connected directly, or they can be connected or steered via unicast SR segment or a segment list.

For a P2MP Tree, a controller may be used to compute paths from a Root node to a set of Leaf nodes, optionally via a set of replication

nodes. A packet is replicated at the root node and optionally on Replication nodes towards each Leaf node.

We define two types of a P2MP Tree: Spray and Replication.

A Point-to-Multipoint service delivery could be via Ingress Replication (aka Spray in some SR context), i.e., the root unicast individual copies of traffic to each leaf. The corresponding P2MP Policy consists of replication segments only for the root and the leaves and they are connected via a unicast SR Segment.

A Point-to-Multipoint service delivery could also be via Downstream Replication (aka TreeSID in some SR context), i.e., the root and some downstream replication nodes replicate the traffic along the way as it traverses closer to the leaves.

The leaves and the root can be explicitly configured on the PCE or PCC can update the PCE with the information of the discovered root and leaves. As an example Multicast protocols like mvpn procedures [RFC6513] or pim can be used to discovery the leaves and roots on the PCC and update the PCE with these relevant information. The controller can calculate the P2MP Policy based on these info.

In all of above cases a set of new PCEP object and TLVs are needed to update and instantiate the P2MP tree. This draft explains the procedure needed to instantiate a P2MP TreeSID.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Overview of PCEP Opration in SR P2MP Network

After discovering the root and the leaves on the PCE (via different mechanism mentioned in previous sections) computes the P2MP Tree and identifying the relevant Replication routers, the PCE programs the PCCs with relevant information needed to create a P2MP Tree.

As per draft [draft-ietf-pim-sr-p2mp-policy] a P2MP Policy is defined by Root-ID, Tree-ID and a set of leaves. A P2MP policy is a variant of SR policy as such it uses the same concept as draft [draft-ietf-pce-segment-routing-policy-cp]. In short a P2MP policy uses a collection of SR P2MP Candidate paths. Candidate paths are computed by the PCE and can be used for P2MP Tree redundancy, only a single candidate path is active at each time. Each candidate paths can be globally optimized and is consists of multiple path-instances.

A path-instance can be thought of as a P2MP LSP. If a candidate path needs to be globally optimized two path-instances can be programmed on the root node and via make before procedures the candidate path can be switched from path-instance 1 to the 2nd path-instance. The forwarding states of these path-instances are build via replication segments, in short each path-instance 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. As an example the push information on the Root node or swap and outgoing interface information on the transit nodes or pop information on the bud and leaves nodes.

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

3.1. High level view of P2MP Policy Objects

- SR P2MP Policy

  - Is only relevant on the Root of the P2MP path and is a policy on PCE which contains information about:

    - Root node of the P2MP Segment

    - Leaf nodes of the P2MP Segment

    - Tree-ID, which is a unique identifier of the P2MP tree on the Root

    - It also contains a set of Candidate paths and their path-instances for P2MP tree redundancy and global optimization

    - Optional Constrains used to build these candidate paths

    - P2MP policy information is downloaded only on the Root node and is identified via <Root-ID, Tree-ID>

- Candidate Path:

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

  - It can contain up to two path-instance for global optimization procedures (i.e. make before break), each identified with their own path-instance ID

- Contains information about protocol-id, originator, discriminator, preference, path-instances

o Replication Segment:

- Is the forwarding information needed on each node for building the forwarding path for each path-instance of the P2MP Candidate path.

- As it will be explained in upcoming sections there are 2 ways to identify the replication segment, shared and non-shared

  + It is identified via Tree-ID and Root-ID and path-instance for non-shared replication segment.

  + It is identified via Node-ID, Replication-ID, for shared replication segment

  + Contains forwarding instructions for the path-instances

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

  + Replication segment information is downloaded on Root, Transit and Leaf nodes respectively.

3.1.1. Shared Tree vs Non-Shared Replication Segment

A non-shared Replication Segment is used when the label field of the PMSI Tunnel Attribute (PTA) is set to zero as per [draft-parekh-bess-mvpn-sr-p2mp]. In short this is used when there is no upstream assigned label in the PTA (provider tunnel attribute) and aggregate of MVPNs into a single P-Tunnel is not desired.

On other hand shared Replication Segment is used when the label field of the PTA is not set to Zero and there is an upstream assigned label in the PTA. In this case multiple MVPNs (VRFs) can be aggregate into a single Provider Tunnel and the upstream assigned label distinguishes the MVPNs context.

It should be noted that the shared Replication Segment can also be used to build a bypass tunnel for the purpose of FRR. This might be desirable if the bypass tunnel is build via the PCE and downloaded to the PCC for link protection. In this case multiple non-shared Replication Segments can use the shared replication segment as their bypass tunnel for link protection. This replication segments used in this bypass tunnel should only create a unicast bypass tunnel to

protect the link between two replication segments on the primary path.

3.2. existing drafts used for defining a P2MP Policy

P2MP Policy reuses current drafts and PCEP objects to update the PCE with Root and Leaves information when PCC Initiated method is used. Also current drafts are used as much as possible to update the PCC with relevant information to build the P2MP Policy and its Replication Segments.

In addition this draft will introduced new TLVs and Objects specific to a programing P2MP policy and its replication segment.

3.2.1. Existing drafts/rfcs used by this draft

o [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

o [RFC8236] P2MP capabilities advertisement

o [draft-ietf-pce-segment-routing-policy-cp] Candidate paths for P2MP Policy is used for Tree Redundancy. As an example a P2MP Policy can have multiple candidate paths. Each protecting the primary candidate path. The active path is chosen via the preference of the candidate path.

o [RFC3209] Defines the instance-ID, instance-ID is used for global optimization of a candidate path with in a P2MP policy. Each Candidate path can have 2 path-instances. These path-instances are equivalent to sub-lsps (instance-IDs). There are used for MBB and global optimization procedures. instance-ID is equivalent to LSP ID

o [draft-ietf-spring-segment-routing-policy] Segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.

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

- [draft-sivabalan-pce-binding-label-sid] Section 3; Path binding TLV is used to indicate the incoming replication SID

- [draft-koldychev-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.

It should be noted that the [draft-dhs-spring-sr-p2mp-policy-yang] can provide further details of the high level P2MP Policy Model.

3.2.2. P2MP Policy Identification

A P2MP Policy and its candidate path 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. Stand-by candidate paths will be assigned a new PLSP-ID by PCC. Stand-by candidate paths can co-exist with the active candidate path.

  * Note: Every candidate path in the SR-P2MP Policy is unique with its own unique PLSP-ID and Instance-ID. But the same Tree-ID is used for all candidate paths as they are part of the same P2MP Tree.

- Root-ID: is equivalent to the first node on the P2MP path, as per [RFC3209], Section 4.6.2.1

- Tree-ID: is equivalent to Tunnel Identifier color which identifies a unique P2MP Policy at a ROOT and is advertised via the PTA in the BGP AD route or can be assigned manually on the root. Tree-ID needs to be unique on the root.

- Instance-ID: LSP ID Identifier as defined in RFC 3209, is the path-instance identifier and is assigned by the PCC. As it was mentioned the candidate path can have up to two path-instance for global optimization. Note that the Root-ID, Tree-ID and Instance-ID are part of a new SR- P2MP-LSP-IDENTIFIER TLV which will be identified in this draft.

  * Note: each Path-instance on the Root node is assigned a unique Instance-ID

3.2.3. Replication Segment Identificaton

The key to identify a replication segment is also a P2MP LSP Object. That said there are different rules for coding the SR-P2MP-LSP-IDENTIFIER TLV which will be explained in later sections. In short for replication segment the P2MP LSP Object does not have the association object.

3.3. High Level Procedures for P2MP SR LSP Instantiation

A P2MP policy can be instantiated via the PCC or the PCE depending on how the root and the leaves are discovered. In theory there is two way to discover the root and the leaves:

- They can be configured and identified on the controller, PCE initiated.

- They can be discovered on the PCC via MVPN procedures [RFC6513] or legacy multicast protocols like PIM or IGMP etc... PCC initiated.

3.3.1. PCE-Init Procedure

- PCE is informed of the P2MP request through it's API or configuration mechanism to instantiate a P2MP tunnel.

- PCE will initiate the P2MP Policy for the request, by sending a PCInitiate message to the Root. This PCInitiation message will have the association object to identify the Candidate Path

  * Optionally the EROs can be added to the PCInitiate message to construct the replication segment on the root.

- Root in response to the PCInitiate message. It will generate PLSP-ID for the candidate paths and an Instance-ID for the Path-Instance (LSP-ID) contained with in a candidate path. The tree-id for the P2MP Policy is also filled. PCC will reports back the PLSP-ID, Instance-ID and tree-id via PCRpt message

  * Optionally, the Root can add any additional leaves that were discovered by multicast procedures in this PCRpt message.

- PCE based on any update to the configured leaves or if any new discovered leaves, can re-compute the P2MP Policy and its replication segments from the Root to the leaves.

  * Any new EROs are send via PCInitiate message, without the association object

  - Any update to the existing EROs are send via PCUpd message, without the association object

- PCE will also sends a PCInitiate message to the Transit and the leaves for the Replication Segment.

3.3.2. PCC-Init Procedure

After Root (PCC) discovers the leaves (as an example via MVPN Procedures or other mechanism), the following communication happens between the PCE and PCCs

- Root sends a PCRpt message for P2MP policy to PCE including the Root-ID, Tree-ID, PLSP-ID, Instance-ID, symbolic-path-name, and any leaves discovered until then.

- PCE on receiving of this report, will compute the P2MP Policy and its replication segments.

  - The PCE will send a PCUpd message to Root for P2MP policy with the Tree-ID, PLSP-ID and the Instance-ID assigned by the PCC. It should be noted the replication segment for root is also downloaded via this update message. In short a single update message that includes the association object will create the P2MP Policy and its replication segment on the Root

    - Note: in this scenario no PCInitiate message was send from the PCE to the PCC to instantiate the P2MP Policy and its Replication segment. This is because for an PCInitiate message a brand new PLSP-ID and Instance-ID is assigned by PCC which is undesirable, since they are already assigned on the first step of this procedure.

  - PCE will also sends a PCInitiate message to the Transit and the leaves for the Replication Segment.

3.3.3. Comon Procedure

Beyond this, the following procedures are the same for PCE or PCC Init.

- PCE will download the replication segments for the Candidate-path's path-instances to all the leaves and transit nodes using PCInitiate message with PLSP-ID = 0, Instance-ID =0, symbolic path name, Root-address, Tree-id(assigned by the root). This PCInitiate message includes the EROs needed for the replication segments.

- Any new candidate path for the P2MP Policy is downloaded by PCE to its connected Root by sending a PCInitiate message

  * it should be noted, PLSP-ID, Path-Instance ID and the Tree-ID are generated by the PCC for these new candidate paths and their Path-instances

  * The ERO objects can be included in this Initiate message

  * The PCC will reply with a PCRpt message

  * Any update to the Candidate Paths or Replication Segments is done via the PCUpd message. Association object need to be present for Candidate Path updates.

- The PCE will also download the necessary replication segment for the candidate path and its path-instances to the leaves and the transit nodes via a PCInit message

- New leaves can be discovered via Multicast procedures, and new replication segments can be instantiated or existing one updated to reach these leaves

  * If these leaves reside on routers that are part of the P2MP LSP path, then PCUpd is sent from PCE to necessary PCCs (LEAVES, TRANSIT or ROOT) with the correct PLSP-ID, Instance-ID and Tree-ID

  * If the new leaves are residing on routers that are not part of the P2MP Tree yet, then a PCInitiate message is sent down with PLSP- ID=0 and Instance-ID=0 on the corresponding routers.

- The active candidate-path is indicated by the PCC through the operational bits(Up/Active) of the LSP object in the PCRpt message. If a candidate path needs to be removed, PCE sends PC Initiate message, setting the R-flag in the LSP object and R bit in the SRP-object.

- To remove the entire P2MP-LSP, PCE needs to send PC Initiate remove messages for every candidate path of the P2MP POLICY to all the PCCs on the P2MP Tree. The R bit in the LSP Object as defined in [RFC8231], refers to the removal of the LSP as identified by the SR-P2MP-POLICY-ID-TLV (defined in this document). An all zero (SR-P2MP-LSP-ID-TLV defines to remove all the state of the corresponding PLSP-ID.

- A candidate path is made active based on the preference of the path. If the Root is programed with multiple candidate paths 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 path will be active

3.3.4. Global Optimiation 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 candidate path is possible.

Each Candidate Path can contain two Path-Instances. The current unoptimized Path-Instance is the active instance and its replication segments are forwarding the multicast PDUs from the root to the leaves. How ever the second optimized Path-Instance will be setup with its own unique replication segments throughout the network, from the Root to the leaves. These two Path-Instances can co-exist. The two Path-Instances are uniquely identified by their Instance-ID in the SR-P2MP-POLICY-ID-TLV (defined in this document). After the optimized LSP has been downloaded successfully PCC MUST send two reports, reporting UP of the new path indicating the new LSP-ID, and a second reporting the tear down of the old path with the R bit of the LSP Object SET with the old Instance-ID in the SR-P2MP-POLICY-ID-TLV. This MBB procedure will move the multicast PDUs to the optimized Path-Instance.

The leaf should be able to accept traffic from both Path-Instances to minimize the traffic outage by the Make Before Break process.

3.3.5. Fast Reroute

Currently this draft identifies the Facility FRR procedures. In addition, only LINK Protection procedures are defined. How the Facility Path is built and instantiated is beyond the scope of this document.

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

Figure 1

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

Figure 2

As an example, the bypass path (unicast bypass) between the PLR and MP can be constructed via SR or even via a shared tree (replication segment).

As an example, in figure 1 the detour path between R and T is the 2nd fiber between these nodes. As such the bypass path could be setup on the 2nd fiber. That said in figure 2 the bypass path is traversing multiple nodes and this example a unicast SR path might be ideal for setting up the detour path.

In addition, PHP procedure and implicit null label on the bypass path can be implemented to reduce the PCE programming on the MP PCC.

Optional shared replication segments can be used in networks that do not have unicast SR turned on. These shared replication segments can be programmed on the bypass nodes without a P2MP Policy. The replication segments on primary path can use these shared replication segments as a protection tunnel to protect links.

3.3.6. Connecting Replication Segment via Segment List

There could be nodes between two replication segment that do not understand P2MP Policy or Replication segment. It is possible to connect two non-adjacent Replication segment via a unicast binding SID or segment-list.

Replication segment does support the concept of a segment-list. A list of unicast SIDs (Binding SID, Adjacency SIDs or Node SIDs) can be programmed on a Replication segment via the SR-ERO sub-objects and ERO-attributes object.

How ever it should be noted that there needs to be a Replication SID as the bottom of the stack in all cases.

3.4. SR P2MP Policy and Replication Segment TLVs and Objects

3.4.1. SR P2MP Policy Objects

SR P2MP Policy can be constructed via the following objects

<Common Header>

[<SRP>]

<LSP>

[<association-list>]

optionally if the root is updating the PCE with end point list the end-point-list object can be added.

[<end-points-list>]

3.4.2. Replication Segment Objects

Replication segment can be constructed via the following objects

<Common Header>

[<SRP>]

<LSP>

[<replication-sid>]

as described in [draft-sivabalan-pce-binding-label-sid]

[<ERO-Attributes Object>]

as per [draft-koldychev-pce-multipath]

3.4.3. P2MP Policy vs Replication Segment

Note on the root the P2MP Policy and Replication Segment can be downloaded via the same message that includes the association object. That said on the transit or leaf nodes the replication segment needs to be downloaded individually as P2MP Policy is only relevant to the Root node. P2MP Policy and Replication segments objects have a very close definition, they can be told apart via the following abstracts:

- The P2MP Policy will always have an association list object for the Candidate Paths in its PCInitiate message. While the replication segment does not have the association list object. That said they can be downloaded simultaneously by inserting the association list object and the ERO object in the same PCInitiate or PCUdp message.

- Both P2MP Policy and Replication segment have the PLSP-ID and it is set to 0 in the PCInitiate message. For both Objects the PLSP-ID is set via the PCC.

3.4.4. P2MP Policy and Replication Segment general considerations

The above new objects and TLV's defined in this document can be included in PCRpt, PcInitiate and PcUpd messages.

It should be noted that every PcRpt, PcInitiate and PCUpd messages will contain full list of the Leaves and labels and forwarding information that is needed to build the Candidate path and its Replication segments. They will never send the delta information related to the new leaves or forwarding information that need to be added or updated. This is necessary to ensure that PCE or any new PCE is in sync with the PCC.

As such when a PcRpt, PcInitiate and PCUpd messages is send via PCEP it maintains the previous ERO Path IDs and generates new Path IDs for new instructions, as per [draft-koldychev-pce-multipath]. This means the PATH IDs are maintained for each specific forwarding instructions until these instructions are deleted. For example: When the first leaf is added the PCE will be update with PathI ID 1 to the PCC. When the second leaf is add, 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. Object Format

4.1. Open Message and Capablity Exchange

Format of the open Object:

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Ver |   Flags |   Keepalive   |  DeadTimer    |      SID      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    //                        Optional TLVs                        //
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

All the nodes need to establish a PCEP connection with the PCE.

During PCEP Initialization Phase, PCEP Speakers need to set flags N, M, P in the STATEFUL-PCE-CAPABILITY TLV as defined in [draft-ietf-pce-stateful-pce-p2mp] section-5.2

This draft extends the PCEP OPEN object by defining an optional TLV to indicate the PCE's capability to perform SR-P2MP path computations with a new IANA capability type.

The inclusion of this TLV in an OPEN object indicates that the sender can perform SR-P2MP path computations. This will be similar to the P2MP-CAPABILITY defined in [RFC8306] section-3.1.2 and a new value needs to be defined for SR-P2MP.

In addition 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.

Finally the open message needs to include the MULTIPATH CAPABILITY TLV as defined in [draft-koldychev-pce-multipath]

4.2. Symbolic Name in PCInit Message from PCC

As per [RFC8231] section 7.3.2. a Symbolic Path Name TLV should uniquely identify the P2MP path on the PCC. This symbolic path name is a human-readable string that identifies an P2MP LSP in the network. It needs to be constant through the life time of the P2MP path.

As an example in the case of P2MP LSP the symbolic name can be p2mp policy name + candidate path name of the LSP.

4.3. P2MP Policy Specific Objects and TLVs

4.3.1. P2MP Policy Association Group for P2MP Policy

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 "P2MP SR Policy Association Type" for SR Policy Association Group (P2MP SRPAG). As per [draft-barth-pce-segment-routing-policy-cp] section 5, three new TLVs are identified to carry association information: P2MP-SRPAG-POL-ID-TLV, P2MP-SRPAG-CPATH-ID-TLV, P2MP-SRPAG-CPATH-ATTR-TLV

4.3.1.1. P2MP SR Policy Association Group Policy Identifiers TLV

The P2MP-SRPOLICY-POL-ID TLV is a mandatory TLV for the P2MP-SRPAG Association. Only one P2MP-SRPOLICY-POL-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.

```
    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              |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Root                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           TREE-ID                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD2 for "P2MP-SRPOLICY-POL-ID" TLV.

Length: 8 or 20, depending on length of End-point (IPv4 or IPv6)

Tunnel Sender Address : Can be either IPv4 or IPv6, this value is the value of the root loopback IP.

Tree-ID: Tree ID that the replication segment is part of as per draft-ietf-spring-sr-p2mp-policy

4.3.1.2. P2MP SR Policy Association Group Candidate Path Identifiers TLV

The P2MP-SRPOLICY-CPATH-ID TLV is a mandatory TLV for the P2MPSRPAG Association. Only one P2MP-SRPOLICY-CPATH-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.

```
     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              |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Proto. Origin |Flags        |A|    Reserved                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Originator ASN                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       Originator Address                      |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Discriminator                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD3 for "P2MP-SRPOLICY-CPATH-ID" TLV.

Length: 28.

Protocol Origin: 8-bit value that encodes the protocol origin, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.3.

Flags : A: This candidate path is active. At any instance only one candidate path can be active. PCC indicates the active candidate path to PCE through this bit. Reserved: MUST be set to zero on transmission and ignored on receipt.

Originator ASN: Represented as 4 byte number, part of the originator identifier, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.4.

Originator Address: Represented as 128 bit value where IPv4 address are encoded in lowest 32 bits, part of the originator identifier, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.4.

Discriminator: 32-bit value that encodes the Discriminator of the candidate path.

4.3.1.3. P2MP SR Policy Association Group Candidate Path Attributes TLV

The P2MP-SRPOLICY-CPATH-ATTR TLV is an optional TLV for the SRPAG Association. Only one P2MP-SRPOLICY-CPATH-ATTR TLV can be carried and only the first occurrence is processed and any others MUST be ignored.

```
     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              |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Preference                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: TBD4 for "P2MP-SRPOLICY-CPATH-ATTR" TLV.

Length: 4. Preference: Numerical preference of the candidate path, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.7.

If the TLV is missing, a default preference of 100 as specified in [draft-ietf-spring-segment-routing-policy] is used.

4.3.2. P2MP-END-POINTS Object

In order for the Root to indicate operations of its leaves(Add/Remove/Modify/DoNotModify), the PC Report message is extended to include P2MP End Point <P2MP End-points> Object which is defined in [RFC8306]

The format of the PC Report message is as follow:

<Common Header>

[<SRP>]

<LSP>

[<association-list>]

[<end-points-list>]

```
      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. Old leaves whose path can be modified/reoptimized (leaf type = 3), Future reserved not used for tree SID as of now.

4. Old leaves whose path must be left unchanged (leaf type = 4)

A given P2MP END-POINTS object gathers the leaves of a given type. Note that a P2MP report can mix the different types of leaves 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.

4.4. P2MP Policy and Replication Segment Identifier Object and TLV

As it was mentioned previously both P2MP Policy and Replication Segment are identified via the LSP object and more precisely via the SR-P2MP-LSPID-TLV

The P2MP Policy uses the PLSP-ID to identify the Candidate Paths and the Instance-ID to identify a Path-Instance with in the Candidate path.

On other hand the Replication Segment uses the SR-P2MP-LSPID-TLV to identify and correlate a Replication Segment to a P2MP Policy

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

4.4.1. 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 tunnel, the PLSP-ID identify a Candidate Path uniquely with in the P2MP policy.

The LSP Object MUST include the new SR-P2MP-POLICY-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]

SR-IPV4-P2MP-POLICY-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                                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |            Path-Instance-ID   |      Reserved                 |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SR-IPV6-P2MP-POLICY-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                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Path-Instance-ID  |      Reserved                 |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 : Contains 16 Bit instance ID.

4.5. Replication Segment

As per [draft-ietf-spring-sr-replication-segment] 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 at the bottom of the stack. This

means two replication segments can be directly connected or connected via a SR domain.

4.5.1. The format of the replication segment message

As it was mentioned in previous chapters the format of the replication segment message is close to P2MP Policy. That said the P2MP Policy contains the association object and the replication segment message does not contain the association object. The replication segment may be downloaded individually on transit and leaf nodes without the P2MP Policy. The P2MP Policy is a Root Concept. The replication segment uses SR-P2MP-LSPID-TLV as its identifier. That said this TLV is coded differently for shared and on shared case.

- In the case of a replication segment being shared, the Tree-ID in the SR-P2MP-POLICY Identifier TLV is the replication-id of the replication segment and Root = 0, Instance-Id = 0. When downloading a shared replication segment from PCE through a PcInitiate message, the SR-P2MP-POLICY Identifier TLV is all 0, and on the report back from PCC, PCC generates PLSP-ID, Replication-id (Tree-id field will be populated with replication-id). Instance-id will be 0.

4.5.2. Label action rules in replicating segment

The node action, ingress, transit, leaf or Bud, is indicated via a new Node Role TLV. This document introduces a new SR-P2MP-NODE-ROLE TLV (Type To be assigned by IANA) that will be present in the PATH-ATTRIB object.

```
     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            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Role Type   |                  Reserved                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- ingress, role type = 1

- transit, role type = 2

- leaf, role type = 3

- bud, role type = 4

4.5.3. SR-ERO Rules

Forwarding information of a replication segment can be configured and steered via many different mechanisms.

As an example a replication SID can be steered via:

1. Replication SID steered with an IPv4/IPv6 directly connected nexthop

   * 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.

   * 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.

   * In addition a new flag D is added to the SR-ERO to signal that the Loopback nexthop is connected to the directly attached router.

3. Replication SID steered with unnumbered IPv4/IPv6 directly connected Interface

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 SID

   * The Replication SID SR-ERO is at the bottom while all other SR-EROs are on the top in order.

4.5.3.1. SR-ERO subobject changes

SR-ERO from RFC 8664 is used to construct the forwarding information needed for Replication Segment.

A new D flag was added to indicate a loopback nexthop that is residing on the directly attached router. It should be noted that this flag should be set only for the loopback case and not for a local interface as a nexthop.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |L|   Type=36   |     Length    |  NT   |     Flags   |D|F|S|C|M|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         SID (optional)                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       //                   NAI (variable, optional)                  //
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flags : F, S, C, M are already defined in rfc8664.

This document defines a new flag D: If the next-hop in NAI field is system IP or loopback, this bit indicates whether the system IP / loopback is directly connected router or not. If set indicates directly connected address. When this bit is set, F bit should be 0 (meaning NAI should be present)

5. Examples of PCEP messages between PCE and PCEP

```
                                   +--------+
                                   |        |
                        +--------+ |LEAF D |                  +--------+
                        |Rep    |  |        |                  |  PCE   |
                        |Transit|  +--------+                  +--------+
                 +------|C      |
                 | Non  |       |  +--------+
                 | Rep  +--------+ |        |
                 | Transit|        |LEAF E |
          +------| B      |        |        |
          |Rep   +---------+       +--------+
          |ROOT    |
          |A       |
          +---------+
```

5.1. PCE Initiate

For a PCE Initiate P2MP Policy a sample PC Initiate message from the PCE to the root is provided below. This is on reception of a P2MP Policy creation on 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
                                 <SRP OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                          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:1   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Type=17             |           Length=<var>        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            symbolic path name                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type=TBD          |  Length                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Root   = A                                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Tree-ID = 0                                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |         Instance-id = 0       |         Reserved              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                            <ASSOCIATION OBJECT>
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Reserved              |            Flags           |0|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | Association type= SR-P2MP-PAG |      Association ID = z       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |              IPv4 Association Source = <pce-address>          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Type              |             Length           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Root  = A                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                           TREE-ID = 0                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Type              |             Length           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |ProtOrigin 10  |    Reserved                                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Originator ASN                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                              |
       |                       Originator Address = <pce-address>     |
       |                                                              |
       |                                                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Discriminator = 1                    |
```

```
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             Type              |             Length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                          Preference = 100 <default>           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Presence of an association object with Tree-ID = 0 in the Initiate message, is an indication to the node to create a P2MP policy and associated candidate path. An initiate message without an association object, is an indication to PCC that a Replication Segment (forwarding instructions) is being instantiated.

5.2. PCC Initiate or PCE Initiate Respond

For PCC initiated P2MP Policy, the Root will send a P2MP request to the PCE, this is achieved through Root sending a PCRpt to PCE with the Tree-ID, PLSP-ID and Instance-ID Set. Below is a sample Report generated by the Root (PCC) to the PCE

In addition for the PCE Initiated case the same PCRpt message can be send from Root (PCC) to the PCE. The Root will generate the Tree-ID, PLSP-ID, Instance-ID for the candidate path identified by the candidate path identifier TLV and sends a report back to PCE. Note, in this case the End point object is optional. The end point object (optionally) is added if the root has discovered any new leaves on the PCC.

Sample Report generated by the Root to the PCE for Leaf Add

```
       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       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Type=17             |           Length=<var>        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            symbolic path name                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type=TBD          |  Length                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Root = A                                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Tree-ID = Y                                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Instance-ID =L1            |      Reserved                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                              <END POINT OBJECT>
      |                          Leaf type =1                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                     Source IPv4 address = A                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Destination IPv4 address = D                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Destination IPv4 address = E                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.3. PCE P2MP Path-Instance Calculation and Replication Segment download

Once the PCRpt message including the endpoints (optional in case of PCE Initiated) is sent to the PCE, PCE computes the path from root to the leaves and would send a PCInitiate to the transit and leaf nodes for instantiating the replication segment for the Path-Instance.

In addition a PCUpd is send to the Root node to construct the replication segment.

The forwarding information is downloaded via the ERO object, ERO-attribute object and SR-ERO sub-objects. For example, say PCE computed 2 candidate paths <cp1 and cp2> that needs to be downloaded on the root and their corresponding Replication Segment download to the root, transit and leaf nodes. The sample messages are explained below.

For cp1:

- For PCC initiate case, the PCE will send a PCUpd message to download the Candidate Paths and the replication segment. Note on the root a single message with association object will achieve this.

- For PCE initiate case, the PCE optionally sends a PCUpd message to instantiate the replication segment that were newly discovered by the PCC and send to the PCE via the PCRpt message.. Note for this case the association object might not be needed is there is no update to the P2MP Policy.

For cp2:

- For both PCC/PCE initiate, a PCInitiate messages sent from PCE, initiating the new Candidate Path and its associated Replication Segments.

For both CP1 and CP2 on the transit and leaves, since PCE is initiating newly Replication Segments, PCE will send one PCInitiate message with two LSP objects and no association object, defining the Replication Semgnets on each candidate path. On other hand, PCE can send separate PCInitiate message for every Replication Segment. As defined in [draft-barth-pce-segment-routing-policy-cp]

A sample PCUpd message sent to the Root for cp1 is as follows, NOTE in the below example the Node B is not Replication Segment Capable as such there is a sid-list programmed on A with node SID B as steering followed by node SID of C and finally the Replication SID C at the bottom :

Note:

1. Root is connected to the next replication Segment C via non replication segment B. Hence a segment List is used.

2. The following PCUpd message send to the root is for PCC Initiated case as such it has the association object to instantiate the Candidate Path and the Replication Segment via a single message on the root.

3. For PCE Initiate message the association object can be omitted sense it is only used for instantiating or updating the Replication Segment only.

```
       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  = 2                     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  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        |         Reserved              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      BT= 0    |                 Reserved                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Binding value = incoming replication SID           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ASSOCIATION OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |         Reserved              |            Flags            |0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Association type= SR-P2MP-PAG |      Association ID = z       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              IPv4 Association Source = <pce-address>          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Root  = A                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           TREE-ID = 0                         |
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |            Length             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |ProtOrigin 10  |    Reserved                                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Originator ASN                        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      |                       Originator Address = <pce-address>      |
      |                                                               |
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Discriminator = 1                     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |            Length             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           Preference = 100 <default>          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                       <ERO-ATTRIBUTES OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |            Length             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|0|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 1                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 0            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = node sid b                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = node sid c                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = RSID C                          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A sample PC Initiate message to the Root for cp2 is as follows: Note cp2 can be either on the same path as cp1 or on a separate path, assuming that there is a 2nd path connecting A to B to C. In this example a 2nd interface is used on A and B, hence the adjacency SIDs are programmed

```
       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  = 3                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                             | PST = TBD       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                             <LSP OBJECT>
      |                PLSP-ID = 0            |     A:1,D:1,N:1,C:1   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Type=17             |           Length=<var>        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            symbolic path name                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type=TBD          |  Length                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Root = A                                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Tree-ID = Y                                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Instance-ID = 0     |   reserved                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      BT       |                 Reserved                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Binding Value= incoming replication sid            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                           <ASSOCIATION OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |         Reserved              |            Flags            |0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Association type= SR-P2MP-PAG |      Association ID = z       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              IPv4 Association Source = <pce-address>          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                       Root = A                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           TREE-ID = Y                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |ProtOrigin 10  |    Reserved                   |
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Originator ASN                        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      |                       Originator Address = <pce-address>      |
      |                                                               |
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Discriminator = 2                     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           Preference = 50 <Lower Pref>        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                             <ERO-ATTRIBUTES OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|0|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 2                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 0            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |               SID = adjacency sid a-b-int2                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                SID = adjacency sid b-c-int2                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = RSID C                          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A sample PC Initiate message to the transit Replication Segment C for cp1 Lets assume C is connected to D and E via 2 fiber hence Fast Reroute is possible. Below example sets up the forwarding plane fro C to Leaves D and E

```
       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  = 4                     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                             | PST = TBD       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                             <LSP OBJECT>
      |                PLSP-ID = 0            |     A:1,D:1,N:1,C:1   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Type=17             |           Length=<var>        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            symbolic path name                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type=TBD          |  Length                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Root=A                                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Tree-ID = Y                                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Instance-ID =L1     |        Reserved               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      BT       |                 Reserved                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Binding Value= incoming replication sid = c1       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                             <With FRR over NHD2>
                            <ERO-ATTRIBUTES OBJECT>
                            <incoming label c1 swap with D1>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|1|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 3                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 4            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = d1                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                ipv4-address  = NHD1           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ERO-ATTRIBUTES OBJECT>
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |0|1|0|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 4                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 0            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = d protect                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                ipv4-address  = NHD2           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                     <incoming label c1 swap with E1>
                        <ERO-ATTRIBUTES OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|1|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 5                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 6            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |L|   Type=36   |     Length    |  NT= 1|     Flags     |0|0|0|0|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         SID = e1                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                ipv4-address  = NHE1           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            <ERO-ATTRIBUTES OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           | Oper|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |0|1|0|                 Reserved                                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 6                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 0            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

5.4. PCC Rpt for PCE Update and Init Messages

In response to the PC Initiate message / PC Update message , PCC will send PC Reports to PCE indicating the state of the label download for that particular candidate path. PCC's will generate PLSP-ID for newly initiated candidate path. Here is an PC Report Message send for the root PCE Init message with cp2 on the root.

```
       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  = 2                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  TLV Type = 28 (PathSetupType)| TLV Len = 4                  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                             | PST = TBD      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                               <LSP OBJECT>
      |                PLSP-ID = 1            |    O:1,A:1,D:1,N:1,C:1|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Type=17             |           Length=<var>       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            symbolic path name                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type=TBD          |  Length                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Root                                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                  Tree-ID                                     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Instance-ID = L1    |          Reserved            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                               <ERO-ATTRIBUTE OBJECT>
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Flags                           |O =Up|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             Type              |             Length           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|1|                 Reserved                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              ERO-path Id = 5                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                            Back-up ero path id = 6           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6. Tree Deletion

To delete the entire tree (P2MP LSP) , Root send a PCRpt message with the R bit of the LSP object set and all the fields of the SR-P2MP-LSP-ID TLV set to 0(indicating to remove all state associated with this P2MP tunnel). The controller in response sends a PCInitiate message with R bit in the SRP object SET to all nodes along the path to indicate deletion of a label entry.

7. Fragmentation

The Fragmentation bit in the LSP object (F bit) can be used to indicate a fragmented PCEP message

8. Example Workflows

As per slides submitted in IETF 105.

9. IANA Consideration

1. This draft extends the PCEP OPEN object by defining an optional TLV to indicate the PCE's capability to perform SR-P2MP path computations with a new IANA capability type (TBD).

2. PCEP open object with a new association type " P2MP SR Policy Association " value (TBD).

3. A new Association type. Association type = TBD1 "P2MP SR Policy Association Type" for SR Policy Association Group (P2MP SRPAG)

    1. three new TLVs are identified to carry association information: P2MP-SRPAG- POL-ID-TLV, P2MP-SRPAG-CPATH-ID-TLV, P2MP-SRPAG-CPATH-ATTR-TLV

4. Two new TLVs for Identifying the P2MP Policy and the Replication segment SR-IPV4-P2MP-POLICY-ID TLV and SR-IPV6-P2MP-POLICY-ID TLV

5. A new SR-P2MP-NODE-ROLE TLV (Type To be assigned by IANA) that will be present in the PATH-ATTRIB object

10. Security Considerations

TBD

11. Acknowledgments

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

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

12.2. Informative References

[draft-barth-pce-segment-routing-policy-cp]
.

[draft-dhs-spring-sr-p2mp-policy-yang]
.

[draft-ietf-pce-segment-routing-policy-cp]
.

[draft-ietf-pce-stateful-pce-p2mp]
.

[draft-ietf-pim-sr-p2mp-policy]
"D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang, "draft-voyer-pim-sr-p2mp-policy"", October 2019.

[draft-ietf-spring-segment-routing-policy]
.

[draft-ietf-spring-sr-replication-segment]
"D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang, "draft-voyer-pim-sr-p2mp-policy "draft-voyer-spring-sr-replication-segment"", July 2020.

[draft-koldychev-pce-multipath]
.

[draft-parekh-bess-mvpn-sr-p2mp]
.

[draft-sivabalan-pce-binding-label-sid]
.

[RFC3209] .

[RFC6513] .

[RFC8231] .

[RFC8236] .

[RFC8306] .

[RFC8664] .

[RFC8697] .

Authors' Addresses

Hooman Bidgoli (editor)
Nokia
Ottawa
Canada

Email: hooman.bidgoli@nokia.com

Daniel Voyer
Bell Canada
Montreal
Canada

Email: daniel.yover@bell.ca

Saranya Rajarathinam
Nokia
Mountain View
US

Email: saranya.Rajarathinam@nokia.com

Ehsan Hemmati
Cisco System
San Jose
USA

Email: ehemmati@cisco.com

Tarek Saad
Juniper Networks
Ottawa
Canada

Email: tsaad@juniper.com

Siva Sivabalan
Ciena
Ottawa
Canada

Email: ssivabal@ciena.com