PCEP extensions for p2mp sr policy
draft-hsd-pce-sr-p2mp-policy-00

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress." The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html
# 1. Introduction

This Internet-Draft will expire on October 8, 2017.

## Table of Contents

1. Introduction ........................................... 3
2. Conventions used in this document .................... 4
3. Overview of PCEP Operation in SR P2MP Network .......... 4
   3.1. High level view of a P2MP Policy Objects ........ 5
      3.1.1. Existing drafts used in defining the P2MP Policy .. 6
      3.1.2. P2MP Identification .......................... 7
3.2 High-Level Procedures for P2MP SR LSP Instantiation .. 7
   3.2.1. MVPN procedures ............................... 8
   3.2.2. Global Optimization for P2MP LSPs ............. 10
   3.2.3. Fast Reroute ................................ 10
      3.2.3. Connecting Replication Policy via Segment List ... 11
3.3. SR P2MP New TLVs and Objects ....................... 12
4. Object Format ......................................... 12
   4.1. Open Message and Capability Exchange ............. 12
   4.2. Symbolic Name in PCInitiate message from PCC ...... 13
   4.3. Replication policy ............................... 14
      4.3.1. P2MP Policy Association Group .................. 14
         4.3.1.1 P2MP SR Policy Association Group Policy Identifiers TLV ............... 14
         4.3.1.2 P2MP SR Policy Association Group Candidate Path Identifiers TLV ......... 15
         4.3.1.3 P2MP SR Policy Association Group Candidate Path Attributes TLV .......... 16
   4.4. PCEP Message Exchanges ............................ 17
      4.4.1 Extension of the LSP Object, SR-P2MP-LSPID-TLV .... 17
      4.5. SR-P2MP-CCI Object ............................. 18
      4.5.1 Optional IP-Address TLVs ....................... 19
   4.6. Root PCE Report message ........................... 21
1. Introduction

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

A Point-to-Multipoint (P2MP) segment connects a Root node to a set of Leaf nodes in a Segment Routing Domain. We also define a Replication segment, which corresponds to the state of a P2MP segment on a particular node.

A P2MP segment consists of replication segments for the root, leaves and optionally intermediate replication nodes.

A replication segment defines the forwarding behavior on a particular node on a particular P2MP segment.

For a P2MP segment, 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 segment: Spray and Replication.

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

Notice that Spray is actually a special form of TreeSID. Also notice that, the explicit path from the root or a replication node to a leaf or a downstream replication node can optionally be partially or completely specified by the controller PCE or determined locally via static configuration.

A PCE could computes a tree from a Root node to a set of Leaf nodes via a set of Replication nodes. A packet is replicated at the Root node and on Replication nodes towards each Leaf node. It should be noted that two replication nodes can be connected directly, or they can be connected via unicast SR segment or a segment list.

The leaves and the root can be explicitly configured on the PCE or PCC can updates the PCE with the information of the discovered root and leaves. As an example Multicast protocols like mvpn procedures 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 Segment 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 Operation in SR P2MP Network

For a P2MP SR policy, a PCE calculates a P2MP tree and programs the Root, Replication and Leaf nodes with information needed to forward a multicast stream from the root to a set of leaves. The PCE discovers the Root and the set of the Leaves via manual configuration on the PCE. On other hand the PCC (Root of the P2MP Tree) can providing the PCE with the relevant information by discovering the leaves via mvpn procedures or pim.

After discovering the Root and Leaves and computing the MPLS P2MP Tree and identifying the Replication routers, the PCE programs the PCCs with relevant information needed to create a P2MP Tree.
As per draft `draft-voyer-spring-sr-p2mp-policy` a P2MP Policy is defined by a root and set of leaves. A P2MP Policy contains replication policies. A replication policy 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. In addition since a P2MP policy is a variant of SR policy it uses the same concept as draft `draft-darth-pce-sement-routing-policy-cp`. In short a replication policy uses a collection of SR P2MP Candidate paths. The candidate paths are computed by the PCE and can be used for P2MP LSP redundancy. In short each candidate path in the replication policy is a unique P2MP lsp.

PCE could also calculate and download additional information such as next-hops for link/node protection or initiate a make-before-brake procedure for global Path optimization.

3.1. High level view of a P2MP Policy Objects

-SR P2MP Policy:

- Is a policy on PCE which contains information about:
  - root node of the P2MP Segment
  - leaf nodes of the P2MP Segment
  - optionally constrains used to build the tree from leaf nodes to the root node.
  - Tree-ID, which is a unique identifier of the P2MP tree on the root

-Replication Policy:

- Is the forwarding information needed on each node and is contained by P2MP policy. It is identified via:
  - Its node-ID, the node that replication policy belongs to.
  - The root of the P2MP Tree
  - Tree-ID, which is a unique identifier of the P2MP Tree on the root. As an example the could be MP-BGP opaque value as per [RFC6513]
  - It also contains a set of Candidate paths for P2MP tree
redundancy

-Candidate Path:

- Is used for P2MP Tree redundancy where the P2MP LSP with the highest preference is the active LSP

- It option can contain up to two P2MP LSP global optimization procedures, each identified with their own LSP-ID (i.e. make before break)

- It also contains the forwarding information for the P2MP LSPs, these forwarding information can be a replication SID or a segment list.

3.1.1. Existing drafts used in defining the P2MP Policy

P2MP Policy reuses current drafts and PCEP objects to identify the root and the leaves on the P2MP Segment and update the PCE with these information, and also to have PCE initiate and update P2MP Replication Policies on a the PCC.

In addition this draft will introduced new TLVs and Objects specific to a P2MP Policy.

This draft reuses the following pcep drafts:

- [RFC8231] The bases for a stateful PCE, and reuses the following objects or a variant of them
  <SRP Object>
  <Lsp Object>

- [RFC8236] P2MP capabilities advertisement
  Also a variation of the P2MP LSP Identifier specified in above RFC

- [draft-barth-pce-segment-routing-policy-cp-02] 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 precedence of the candidate path.

- [RFC 3209] defines the LSP-ID, LSP-ID is used for global optimization of a candidate path with in a P2MP policy. Each Candidate path can have 2 sub-lsps (LSP-IDs) for MBB and global optimization procedures.
- [draft-ietf-spring-segment-routing-policy] segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.

- [draft-ietf-pce-pcep-extension-for-pce-controller] a variant of CCI Object, used for downloading the replication policy forwarding instructions. These instructions can be incoming label and set of outgoing labels or fast reroute procedures or even downloading of a segment-list connecting two non-adjacent replication policy.

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

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

3.1.2. P2MP Identification

The key to identify a P2MP LSP is in LSP object and is as follow:

PLSP-ID: RFC 8231, is assigned by PCC and is unique per candidate path. It is constant for the lifetime of a PCEP session. Since PLSP-ID is unique per LSP, Stand-by P2MP LSPs will be downloaded with a new PLSP-ID. It should be noted a stand-by LSP is a LSP to protect the primary LSP and can be setup in parallel to the primary LSP. These stand-by LSPs are identified by the candidate path.

LSP-ID: LSP ID Identifier as defined in rfc 3209, and is used for global optimization of a P2MP LSP (Candidate path)

Tree-ID: is equivalent to Tunnel Identifier color which identifies a unique P2MP segment at a ROOT and is advertised via the PTA in the BGP AD route.

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

Note that the Tree-ID, Root and LSP-ID are part of a new SR-P2MP-LSP_Identifier TLV which will be identified in this draft.

3.2 High-Level Procedures for P2MP SR LSP Instantiation

A P2MP policy is consistent of Root and a set of Leaves and as such a set of replication policies for each node within the path of the P2MP segment. As mentioned previously a replication policy is a set of forwarding rules used on root, transit nodes and leaves.

The Root and Leaves can be discovered via many methods.
- They can be configured and identified on a controller
- They can be configured on the root node PCC and the root updates the PCE with this information
- They can be discovered via Multicast mechanisms like MVPN procedures or protocols like PIM.

### 3.2.1 MVPN procedures

In case of MVPN there can be pcc-initiated or pce-initiated p2mp policy. In either case MVPN procedures [RFC6513, RFC6514] are used to discover the leaves on PCC and report them up to the PCE.

1. **PCE-initiated 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 LSP for the request, by sending a PC Initiate message to the Root. The above PC Initiate message to the Root will contain the following information. PLSP-ID = 0, LSP-ID (SR-P2MP-LSP-ID-TLV defined in this document), symbolic path name, association object (association-id defined by PCE, association-type SR-P2MP-PAG), policy identifier TLV(root and tree-id(0)), candidate path identifier tlv(protocol, origin, discriminator (32 bit value, which needs to be generated by PCE, and uniquely identifies a candidate path. This will increment for every candidate path and starts from 0)). The CC-ID list will be empty since PCE has not discovered any leaves yet. Root in response to the PC Initiate message will generate PLSP-ID and tree-id for the LSP Identifier and the candidate path that was downloaded by the PCE for this replication policy. PCC reports back the PLSP-ID, LSP_ID(SR-P2MP-LSP-ID-TLV defined in this document) and tree-id, and any leaves that were discovered until then to PCE. PCE on discovering leaves from the root, will compute the path to the leaves and downloads the label-information by sending PC Initiate message on the transit and leaf nodes connected to PCE and sends a PC Update message to the Root. The PC Initiate message to the transit and leaf is like the PC Initiate message that was sent to the Root, but the Tree-ID will not be 0 and will be the Tree-ID that the Root communicated through the Pc Report.

2. **PCC Initiated Procedure**:

   Root sends a PC Report to PCE including the root, tree-id, PLSP-ID, LSP-ID(SR-P2MP-LSP-ID-TLV defined in this document), symbolic-path-name, and any leaves learnt until then. PCE on receiving this report, will compute the path and download label information on the leaf and
transit using PC Initiate message as PLSP-ID = 0, symbolic path name, association object (association-id defined by PCE, association-type SR-P2MP-PAG), policy identifier TLV(root and tree-id), candidate path identifier tlv(protocol, origin, discriminator (32 bit value, which needs to be generated by PCE, and uniquely identifies a candidate path. This will increment for every candidate path and starts from 0), CC-ID list of label downloads. On the root it will be an update message containing the PLSP-ID and other information that was earlier communicated by the Root. The association-ID used in the Root, transit and leaves will be the same for all candidate paths. Transit and Leaf on receiving the Initiate message will generate a PLSP-ID and report the status of the label downloads.

Beyond this, procedures for (1) and (2) are same.

[draft-ietf-pce-pcep-extension-for-pce-controller] indicates the PLSP-ID used in PC Initiate messages is the PLSP-ID defined at the ingress node, to allow correlation between transit instructions and the ingress LSP entity. For Replication Policy, as defined in the above procedures, other identifiers allow correlation of transit to root/ingress instructions to the downstream so the same PLSP-ID is not required. PLSP-ID’s are individually generated by every PCC in the P2MP path.

Any new leaves discovered from here on, are reported to PCE using the PLSP-ID of the active candidate path. If these leaves are discovered on routers that are part of the P2MP LSP path, then PC Update is sent from PCE to necessary PCCs (LEAVES, TRANSIT or ROOT) with the LSPs PLSP-ID. If the new leaves are discovered on routers that are not part of the P2MP Tree yet, then a PC Initiate message is sent down with PLSP-ID=0.

Any new candidate path is downloaded by PCE to its connected Root, transit and leaves by sending a PC Initiate message to them. Every candidate path is a different P2MP LSP which gets a unique PLSP-ID. Multiple candidate path are associated to the same Replication policy and each used as a redundant P2MP LSP.

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 SR-P2MP-POLICY to all the PCE connected nodes along the P2MP-LSP path. The R bit in the LSP Object as defined in rfc8231, refers to the removal of the LSP as identified by the SR-P2MP-LSP-ID-TLV (defined in this document). 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 gets paths one from the PCE and one from the CLI, and 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.

### 3.2.2. Global Optimization for P2MP LSPs

When a P2MP LSP needs to be optimized for any reason (i.e. it is taking a FRR Path or new routers are added to the network) a global optimization is possible. Note that optimization works per candidate path. Each candidate path is capable of global optimization. To do so each candidate path contains two P2MP LSP, each P2MP LSP is identified via the LSP-ID [RFC3209]. After calculating an optimized P2MP LSP path the PCE will program the candidate path with a 2nd LSP-ID and its set of CCI instructions. After the optimized LSP is downloaded a MBB procedure is performed and the previous instance of the P2MP LSP is deleted and removed from the corresponding PCCs. The globally optimized LSP is instantiated via the PCInitiate message. The PLSP-ID of this optimized LSP is same as the Current LSP which is being optimized, this is because both LSPs belong to the same candidate path. That said the LSP-ID of the optimize LSP is uniquely assigned by PCE and is different from that of the current LSP which is being optimized. In short, the LSP-ID uniquely identifies sub-instances of an LSP for optimization with in that candidate path. After the optimized LSP has been downloaded and verified via PCC PCRpt message, the MBB procedure can be performed to switch between the two instances of the LSP. The previous instance will be removed from PCCs.

### 3.2.3. 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.
As an example, the bypass path (unicast bypass) between the PLR and MP can be constructed via SR. The PCE needs to only update the PLR PCC with bypass path outgoing label and nexthop information, also PCE needs to update the MP PCC with bypass path ILM information. This information is presented via a P bit in the optional IPv4/IPv6-Address object as per upcoming section.

If one to one FRR is needed, then a second flag O should be defined in the IPv4/IPv6-Address object in future.

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 using treeSID procedures. 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. The PCE can download the prefix SID for F2 as a bypass path for R-T to R. Downloading the prefix SID for F2 will ensure an LFA detour for R-T. In addition, PHP procedure and implicit null label on the bypass path can be implemented to reduce the PCE programming on the MP PCC.

3.2.3. Connecting Replication Policy via Segment List

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

### 3.3. SR P2MP New TLVs and Objects

A new object `<SR-P2MP-CCI>` is defined for the controller to specify the forwarding instructions (label instructions) of the replication policy.

A new Association Type (P2MP SR Policy) is defined. Also a SR-P2MP policy identifiers TLV is defined to communicate the SR-P2MP policy and candidate path information.

A new P2MP-LSP identifier TLV is included to indicate the P2MP identifiers (Root, Tree-ID, LSP-ID).

The above new objects and TLV’s defined in this document can be included in PcReport, PcInitiate and PcUpdate messages.

It should be noted that every PcReport, PcInitiate and PCUPdate messages will contain full list of the Leaves and label and forwarding information that is needed to build the P2MP LSP. In short the PCC or PCE should never send the delta information related to the new leaves 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 PcReport, PcInitiate and PCUPdate messages is send via PCEP it maintains the previous instruction CC-IDs and create new CC-ID for the new instruction. This means the CC-IDs are maintained for each specific forwarding and label instructions until these instructions are deleted. For example, When the first leaf is added PCC gets instructions, CC-ID A on a particular transit node. On a second leaf add, according to the path calculated, PCE might just append the existing instruction A with B. This is done by sending a PC Update with CC-ID A and B.

### 4. Object Format

#### 4.1. Open Message and Capability Exchange

Format of the open object
All the nodes need to establish a PCEP connection with the PCE.

During PCEP Initialization Phase, PCEP Speakers advertise their support of PCECC extensions to include the new Path Setup Type [draft-ietf-pce-pcep-extension-for-pce-controller]. Also need to set flags N, M, P in the STATEFUL-PCE-CAPABILITY TLV as defined in [draft-ietf-pce-stateful-pce-p2mp] section capability advertisement.

We extend the PCEP OPEN object by defining an optional TLV to indicate the PCE’s capability to perform SR-P2MP path computations, 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 P2MP Policy.

In addition a Assoc-Type-List TLV as per [draft-ietf-pce-association-group-07] section 3.4 should be send via PCEP open object with following association type

<table>
<thead>
<tr>
<th>Association Type</th>
<th>Association Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>P2MP SR Policy Association</td>
<td>This document</td>
</tr>
</tbody>
</table>

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

4.2. Symbolic Name in PCInitiate 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 Root + Tree-ID of the LSP. The Tree-ID is a unique ID that identifies the P2MP LSP on the Root (Source) as such the combination of Root + Tree-ID will provide the P2MP LSP with a unique identification throughout the network. Depending on the Source IP, IPv4 vs. IPv6, the length of the TLV will vary.

4.3. Replication policy As per [draft-voyer-spring-sr-p2mp-policy] a replication policy is build of candidate paths. Each candidate path contains p2mp-cci object which may contain a single outgoing label, in case it is directly connected to another replication segment, or a segment list to connect two non adjacent replication segments.

The candidate path and segment list has been described in [draft-ietf-spring-segment-routing-policy].

Candidate paths can be used for P2MP LSP redundancy where the active candidate path in a replication policy has a higher precedence over other candidate paths.

As such and as per [draft-barth-pce-segment-routing-policy-cp] section-3.1 each candidate path of a Replication policy appears as a different SR P2MP LSP (identified via a PLSP-ID) in PCEP, it is useful to group together all the candidate paths that belong to the same Replication policy. Furthermore, it is useful for the PCE to have knowledge of the P2MP SR candidate path parameters such as Root, Tree-ID, protocol origin, discriminator, and preference.

In this draft we define new association group objects to make above possible.

4.3.1 P2MP Policy Association Group

Two ASSOCIATION object types for IPv4 and IPv6 are defined in [I-D.ietf-pce-association-group]. 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.

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.
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 [I-D.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 [I-D.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.

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

Length: 4.

Preference: Numerical preference of the candidate path, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.7.
If the TLV is missing, a default preference of 100 as specified in [I-D.ietf-spring-segment-routing-policy] is used.

4.4. PCEP Message Exchanges

PCE is informed of the P2MP request through manual configuration of root and leaves on the controller or through a Report from Root. On Reception of the P2MP Request, PCE initiates the P2MP LSP on the nodes connected along the P2MP Policy path that are connected to PCE. The object ordering of PC-Init, PC-Update and PC-Report messages are as per [draft-ietf-pce-association-group] section-6.3 (object Encoding in PCEP messages)

Format of PC InitiateMessage:
<Common Header>
<SRP>
<LSP>
<association-list>
<SR-P2MP-CCI>

Format of PC Update Message:
<Common Header>
<SRP>
<LSP>
<association-list>
<SR-P2MP-CCI>

Every SR-P2MP-LSP’s LSP Object MUST include the SR-P2MP-LSP-ID-TLV (IPV4/IpV6) which is defined by this document as below. This is a variation to the P2MP object defined in [draft-ietf-pce-stateful-pce-p2mp]

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 (P2MP-LSP) uniquely with in the Replication policy.

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

SR-IPV4-P2MP-LSP-IDENTIFIER TLV:
SR-IPV6-P2MP-LSP-IDENTIFIER TLV:

```
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                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LSP-ID              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

LSP-ID: Contains 16 Bit LSP ID defined in rfc 3209.

Root: Source Router IP Address

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

4.5. SR-P2MP-CCI Object

This is a variation of the CC-ID object defined in [draft-ietf-pce-pcep-extension-for-pce-controller]

The SR-P2MP-CCI is used to download label instruction to the nodes.
It can contain optional IPv4/IPv6 IP-Address TLVs that include forwarding instructions. These instructions include incoming label (incoming replication SID), or out-going label for adjacent replication policies or Fast Reroute labels. Even segment list labels connecting two non adjacent replication policy can be downloaded via this 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            CC-ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved            |              Flags           |0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

//                        Optional TLV                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

CC-ID: A PCEP-specific identifier for the CCI information. A PCE creates an CC-ID for each instruction, the value is unique within the scope of the PCE and is constant for the lifetime of a PCEP session. The values 0 and 0xFFFFFFFF are reserved and MUST NOT be used. Flags:
0 - Down - means label download was not successful 1 - Up - means label download was successful

4.5.1 Optional IP-Address TLVs

These optional IPv4/IPv6 TLVs can be including in P2MP CCI Object for forwarding information download.

Optional TLV: IPV4-ADDRESS 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 = 12                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             SID-List Size     |Rsvd           | Flag|I|B|S|E|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Interface ID                           |
~                          labels                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

SID-List Size:
- is the number of SIDs in the SID List

Flags:
I - Incoming Label:
If set means In Label, If not set means Out Label.

B - Bud Node Label:
If set this label is a bud node, the payload needs to be processed locally and also replicated if the S bit is set. In short if B is set then S needs to set also.

S - SWAP label:
0: If I bit is set and S bit is 0 it means pop the label and if the label’s S bit is set do a recursive lookup.
1 - If I bit is set and S bit is 1 it means swap this label with out label.

P - Protection NextHop
0 - Label information is not w.r.t protection next-hop.
1 - Label information is w.r.t protection next-hop.

Note: P flag is used at the PLR and MP to identify the facility tunnel.

E - Protected LTN, This bit is usually set with the an outgoing label, when the outgoing label is protected via a protection nextHop
  0 - Label information does not have a protection next-hop.
  1 - Label information has a protection next-hop.

IPV4 address and Interface Id:
correspond to the next-hop information in case of an OUT Label, and it corresponds to incoming interface information if it is an IN Label.

Labels:
can be a single label or a list of labels, with the first label in the list being the label on top of the stack and the last label in the list being the label at the bottom of the stack.
IPV6-ADDRESS TLV:

```
+---------------+------------------+
|             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 = 24      |
|               |                  |
+---------------+------------------+
| SID-List Size | Flag             |
|               |                  |
+---------------+------------------+
|                // IPv6 address (16 bytes)     |
|                //  |
+---------------+------------------+
| Interface ID  |                  |
|               |                  |
+---------------+------------------+
|                ~ labels                          |
+---------------+------------------+
```

UNNUMBERED-IPV4-ID-ADDRESS TLV:

```
+---------------+------------------+
|             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 = 16      |
|               |                  |
+---------------+------------------+
| SID-List Size | Flag             |
|               |                  |
+---------------+------------------+
| Node-ID       |                  |
|               |                  |
+---------------+------------------+
| Interface ID  |                  |
|               |                  |
+---------------+------------------+
|                ~ labels                          |
+---------------+------------------+
```

4.6. Root PCE Report message

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> | <SR-P2MP-CCI>]
```

4.6.1 END-POINTS Objects
IPV4-P2MP END-POINTS:

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
</table>
| +------------------------------------------+
| | Leaf type                                 |
| +------------------------------------------+
| | Source IPv4 address                       |
| +------------------------------------------+
| | Destination IPv4 address                  |
| +------------------------------------------+
| ~------------------------------------------+
| | Destination IPv4 address                  |
| +------------------------------------------+
| | +------------------------------------------+
| | | Source IPv4 address                       |
| +------------------------------------------+
| | Destination IPv4 address                  |
| +------------------------------------------+
| ~------------------------------------------+
| +------------------------------------------+
| | Destination IPv4 address                  |
| +------------------------------------------+
| +------------------------------------------+

IPV6-P2MP END-POINTS:

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
</table>
| +------------------------------------------+
| | Leaf type                                 |
| +------------------------------------------+
| | Source IPv6 address (16 bytes)             |
| +------------------------------------------+
| | Destination IPv6 address (16 bytes)        |
| +------------------------------------------+
| ~------------------------------------------+
| | Destination IPv6 address (16 bytes)        |
| +------------------------------------------+
| | +------------------------------------------+
| | | Source IPv6 address (16 bytes)             |
| +------------------------------------------+
| | Destination IPv6 address (16 bytes)        |
| +------------------------------------------+
| ~------------------------------------------+
| +------------------------------------------+
| | Destination IPv6 address (16 bytes)        |
| +------------------------------------------+
| +------------------------------------------+

Leaf Types (derived from RFC 8306 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.

5. Examples of PCEP messages between PCE and PCEP

```
+-------+  |LEAF D |                   +-------+
|       |  |       |                   | PCE  |
|Rep    |  |       |                   |  PCE  |
|Transit|  +-------+                   +-------+

|       |  +-------+  |       |
|      C |  |       |  |       |
| Non   |  |       |  |       |
|       |  +-------+  |       |
|Rep    |  |       |  |       |
|Transit|  |       |  |       |
|       |  +-------+  |       |
|       |  |       |  |       |
|       |  |       |  |       |
|       |  |       |  |       |
|       |  +-------+  |       |
|       |  |       |  |       |
|       |  |       |  |       |
```

5.1. PCE Initiate and PCC Leaves Update

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:
<SRP OBJECT>
| Flags = 0 |
| SRP-ID-number = 1 |
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
| PST = TBD |

<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 | Flags [0] | Reserved |

On Response to the above initiate message, PCC generates a Tree-ID, PLSP-ID for the candidate path identified by the candidate path identifier TLV and sends a report back to PCE. If leaves are discovered by the PCC at that point of time, that is communicated to the PCE in the same report message using the `<p2mp-end-point>` object in the Report message.

For PCC initiated P2MP Policy, if the Root wants to send a P2MP request to the PCE, the same is achieved through Root sending a PC Report to PCE indicating a P2MP Request.

Sample Report generated by the Root to the PCE for P2MP Request received from the Root Node:
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags = 0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SRP-ID-number = 1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TLV Type = 28 (PathSetupType), TLV Len = 4 |
| PST = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

<ASSOCIATION OBJECT>

```

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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 Flags | 0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Originator ASN |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
5.2. PCE P2MP LSP Calculation and Replication Policy download

Once the PC Report of leaves is sent to the PCE, PCE computes path to the leaf and would send a PC Initiate/PC Update to the connected PCC’s across the path to the leaf along with association object (defining association parameters, SR-P2MP policy identifier TLV, SR-P2MP-candidate path identifier TLV, candidate path attributes TLV) and label download information (SR-P2MP-CCI).

For example, say PCE computed 2 candidate paths <cp1 and cp2> that needs to be downloaded on the transit and root node, sample messages are explained below.

For cp1 -> on the root it will be a PC Update message sent from PCE, updating the empty candidate path it had sent earlier when it had intimated the root about the <root, tree-ID> it had known from NMS.
For cp2 -> on the root it will be a PC Initiate messages sent from PCE, initiating the new candidate path and associating it to the same SR-P2MP policy.

On the transit - for cp1, and cp2 since PCE is initiating both newly on those nodes, PCE will send one PC Initiate message with two LSP objects, defining each candidate path. Or PCE can send separate PC Initiate message for every candidate path. As defined in [draft-bartl-pce-segment-routing-policy-cp] section multiple candidate paths

A sample PC Update message sent to the Root for cp1 is as follows:
Note Root is connected to the next replication Segment C via non replication segment B. Hence a segment List is used.
<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 |
| LSP-ID = L1 |

<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 | Flags | 0 | Reserved |
| Originator ASN |
| Originator Address = <pce-address> |
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.
<thead><tr><th>Field</th><th>Value</th></tr></thead><tbody><tr><td>Reserved</td><td>0</td></tr><tr><td>Association type</td><td>SX-P2MP-PAG</td></tr><tr><td>Association ID</td><td>z</td></tr><tr><td>IPv4 Association Source</td><td>&lt;pce-address&gt;</td></tr><tr><td>Root</td><td>A</td></tr><tr><td>TREE-ID</td><td>Y</td></tr><tr><td>ProtOrigin</td><td>10</td></tr><tr><td>Originator ASN</td><td>&lt;pce-address&gt;</td></tr><tr><td>Discriminator</td><td>1</td></tr><tr><td>Preference</td><td>100</td></tr><tr><td>CC-ID</td><td>z10</td></tr><tr><td>IPv4 address</td><td>NH2</td></tr>
A sample PC Initiate message to the transit replication policy C for cp1

Let's assume C is connected to D and C via 2 fiber hence Fast Reroute is possible:

```
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                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LSP-ID =L1          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

<ASSOCIATION OBJECT>

|         Reserved              |            Flags            |0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Association type= SR-P2MP-PAG |      Association ID = z     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             IPv4 Association Source = <pce-address>          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
Internet-Draft      PCEP extensions for p2mp policy         July 6, 2019

+---------------------------------------------------------------+
| Root = A                                                        |
+---------------------------------------------------------------+
| TREE-ID                                                        |
+---------------------------------------------------------------+
| Type | Length |
| ProtOrigin 10 | Flags | 0 | Reserved |
+---------------------------------------------------------------+
| Originator ASN |
+---------------------------------------------------------------+
| Originator Address = <pce-address> |
+---------------------------------------------------------------+

+---------------------------------------------------------------+
| Discriminator = 0 |
+---------------------------------------------------------------+
| Type | Length |
| Preference = 100 <default> |
+---------------------------------------------------------------+

<CC-ID OBJECT>

| CC-ID = z20 |
+---------------------------------------------------------------+
| Reserved | Flags | 0 |
+---------------------------------------------------------------+
| Type=TBD | Length = 12 |
+---------------------------------------------------------------+
| sid-list-size = 1 | Rsvd | Flag|E|0|0|0|0|
+---------------------------------------------------------------+
| IPv4 address |
+---------------------------------------------------------------+
| Interface ID |
+---------------------------------------------------------------+
| Label= c1 |
+---------------------------------------------------------------+

<incoming label c1 swap with D1>  
CC-ID = z21

| Reserved | Flags | 0 |
+---------------------------------------------------------------+
| Type=TBD | Length = 12 |
+---------------------------------------------------------------+
| sid-list-size = 1 | Rsvd | Flag|0|0|S|E|0|
+---------------------------------------------------------------+
| IPv4 address =NHD1 |
+---------------------------------------------------------------+
5.3. 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.
<table>
<thead>
<tr>
<th>Flags = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRP-ID-number = 2</td>
</tr>
<tr>
<td>TLV Type = 28 (PathSetupType)</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<ASSOCIATION OBJECT>
| Reserved | Flags | 0 |
| Association type = SR-P2MP-PAG | Association ID = z |
| IPv4 Association Source = <pce-address> |
| Type | Length |
| Tunnel Sender Address Ipv4 or IPv6 =A |
| TREE-ID = Y |
| Type | Length |
| ProtOrigin 10 | Flags | 1 | Reserved |
| Originator ASN |
| Originator Address = <pce-address> |
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 workflow

As per slides submitted in IETF 105.

6. IANA Considerations

This document contains no actions for IANA.

7. Security Considerations

TBD

8. References

8.1. Normative References
8.2. Informative References


7. Acknowledgments

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

Authors’ Addresses

Hooman Bidgoli
Nokia
600 March Rd.
Ottawa, Ontario K2K 2E6
Canada

Email: hooman.bidgoli@nokia.com

Daniel Voyer
Bell Canada
Montreal
CA

Email: daniel.voyer@bell.ca

Siva Sivabalan
Cisco Systems
Ottawa
Canada

Email: msiva@cisco.com

Jayant Kotalwar
Nokia
380 N Bernardo Ave,
Mountain View, CA 94043
US

Email: jayant.kotalwar@nokia.com

Saranya Rajarathinam
Nokia
380 N Bernardo Ave,
Mountain View, CA 94043
US