PCEP extensions for p2mp sr policy
draft-hsd-pce-sr-p2mp-policy-01

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

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. shared tree vs non-shared tree .......................... 6
   3.1.1. Existing drafts used in defining the P2MP Policy ...... 6
   3.1.2. P2MP Policy Identification ............................. 8
   3.1.3. Replication Segment Identification ...................... 8
   3.2 High-Level Procedures for P2MP SR LSP Instantiation .......... 8
   3.2.1. MVPN procedures ...................................... 9
   3.2.2. Global Optimization for P2MP LSPs ...................... 11
   3.2.3. Fast Reroute ....................................... 11
   3.2.3. Connecting Replication Segment via Segment List .......... 12
   3.3. SR P2MP Policy and Replication Segment TLVs and Objects ... 12
   3.3.1. SR P2MP Policy Objects ................................. 13
   3.3.2. Replication Segment Objects ............................ 13
   3.3.3. P2MP Policy vs Replication Segment ..................... 13
   3.3.4. P2MP Policy and Replication Segment general considerations ........................................ 13
4. Object Format ................................................ 14
   4.1. Open Message and Capability Exchange ...................... 14
   4.2. Symbolic Name in PCInitiate message from PCC ............. 15
   4.3. P2MP Policy Specific Objects and TLVs .................... 15
   4.3.1. P2MP Policy Association Group for P2MP Policy .......... 15
   4.3.1.2. P2MP SR Policy Association Group Policy Identifiers TLV ............................................. 15
1. Introduction


A Point-to-Multipoint (P2MP) Policy connects a Root node to a set of Leaf nodes. We also define a 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 replication segments that 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 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 unicasts 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 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 updates the PCE with the information of the discovered root and leaves. As an example Multicast protocols like mvnp procedures [RFC 6514, RFC 6513] 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 Operation in SR P2MP Network

For a P2MP SR policy, a PCE calculates a P2MP tree and programs the Root, Transit 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 provide the PCE with the relevant information by discovering the leaves via mvnp 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-pim-sr-p2mp-policy a P2MP Policy is defined by a root and set of leaves. A P2MP policy is a variant of SR policy as such it uses the same concept as draft draft-darth-pce-sement-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. 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 is needed to be globally optimized two path-instances can be programmed on the node and via make before procedures the candidate path can be switched from path-instance 1 to the 2nd. The forwarding states of these path-instances are build via replication segments.

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 such as next-hops for link/node protection.

3.1. High level view of a P2MP Policy Objects

- SR P2MP Policy:
  - Is only relevant on the root of the P2MP tree 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
    - Path-instances which are P2MP LSPs under each candidate path for global optimization of the candidate path.
    - 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 P2MP LSP with the highest preference is the active LSP

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

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

- Replication Segment:

  - Is the forwarding information needed on each node for building the datapath 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 trees
      - Tree-ID and Root-ID and path-instance for non-shared trees.
      - Its node-ID, Replication-id, for shared trees

    - Contains forwarding instructions for the path-instances (P2MP LSPs), these instructions can be a replication SID or a segment list.

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

3.1.1. shared tree vs non-shared tree

A non-shared tree 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 tree is used when there is no upstream assigned label in the PTA and aggregate of MVPNs into a single P-Tunnel is not desired.

On other hand shared tree 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 P-Tunnel and the upstream assigned label distinguishes the MVPNs.

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 the PCC.

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

This draft reuses the following pcep drafts and concepts:
- [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
- [RFC8236] P2MP capabilities advertisement
- [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 preference of the candidate path.
- [RFC 3209] 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 sub-lsps (instance-IDs) for MBB and global optimization procedures. instance-ID is equivalent to LSP ID as per [RFC 3209]
- [draft-ietf-spring-segment-routing-policy] segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.
- [RFC8306] P2MP End Point objects, used for the PCC to update the PCE with discovered Leaves.
- [draft-sivabalan-pce-binding-label-sid-04#section-3] Path binding TLV is used to indicate the incoming replication SID
- [draft-koldychev-pce-multipath-01] 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.
It should be noted that the [draft-dhs-spring-sr-p2mp-policy-yang] can provide further details of the high level P2MP Policy Model.

3.1.2. P2MP Policy Identification

A P2MP Policy and its candidate path on the root can be identified on the Root via the P2MP LSP Object, this Object is a variation of the LSP ID Object defined in RFC 8231 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. Stand-by P2MP LSPs will be downloaded with a new PLSP-ID, since every P2MP LSP is a candidate path on its own. 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.

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

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.

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

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.

3.1.3. Replication Segment Identification

The key to identify a replication segment is also the P2MP LSP Object.

That said there are different rules for coding the SR-P2MP-LSP-IDENTIFIER TLV which will be explained in later sections.

3.2 High-Level Procedures for P2MP SR LSP Instantiation

A P2MP policy is defined by Root, Tree-Id, set of Leaves. It also contains a set of candidate-path and path-instances which are used for global optimization of the candidate paths. The P2MP policy is used to calculate and download a set of replication segment to connect the root to a set of leaves, optionally via a set of transit nodes. As mentioned previously a replication segments are set of forwarding rules used on root, transit nodes and leaves. Each path-instance of a candidate path will have a set of replication segments to connect the root the leaves via a set of transit nodes. A path-
instance can be viewed as a P2MP LSP.

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 [RFC 6513] and [RFC 6514] 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 Tunnel for the request, by sending a PC
  Initiate message to the Root and start programing the P2MP Policy on
  the root.

- Root in response to the PC Initiate message. It will generate PLSP-
  ID for the candidate paths, Path-Instance-ID for the P2MP LSP
  contained with in a candidate path and tree-id for the candidate
  paths and P2MP Policy. It will reports back the PLSP-ID, Instance ID
  and tree-id, and any leaves that were discovered until then to PCE.

- PCE on discovering leaves from the root, will compute the P2MP
  Policy from the Root to the leaves and downloads the replication
  segment by sending PC Initiate message on the transit and leaf nodes.

- PCE will also sends a PC Initiate message to the Root for the
  Replication Segment.

2. PCC Initiated Procedure:

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

- Root sends a PC Report for P2MP policy to PCE including the root,
tree-id, PLSP-ID, Path-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.
  - On Root it will send and update for P2MP policy with the PLSP-ID and the corresponding replication segment. It should be noted the replication segment is downloaded via an update message also.
  - It will download replication segments to leaves and transit nodes using PC Initiate message with PLSP-ID = 0, symbolic path name, Root-address, Tree-id(sent by the root), Instance-id(sent by root). The replication segment information(containing TE-Path binding TLV, ERO object, ERO attributes object, multipath backup TLV,SR-ERO sub-objects).

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

- Any new candidate path is downloaded by PCE to its connected Root by sending a PC Initiate message to them.
  - it should be noted, PLSP-ID, Path-Instance-ID and the Tree-ID are generated by the PCC

- The replication segments are downloaded via a PC Initiate message to the Root, Transit and Leaf nodes for these new candidate paths and their Path-Instance-ID.
  - it should be noted, a replication segment message never has the association object.

- Every candidate path is a different P2MP Tree which gets a unique PLSP-ID. All candidate path is associated to the same SR-P2MP policy.

- Any new leaves discovered from here on, are reported to PCE.

- 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.

- The candidate-path that was informed by PCE to PCC, is active or not is indicated by the PCC through the operational bits(Up/Active) of the LSP object in the PC Report. 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.

o 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-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.

o 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.

o Discovery of local replication policies is outside the scope of this document, however it may be achieved using methods such as learning via NMS, NetConf or extending dynamic discovery protocols such as extending draft-ietf-idr-te-lsp-distribution for sr replication policies

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 tunnel or new routers are added to the network) a global optimization is possible. At any instance, a PCE can download more than one candidate path to the existing SR-P2MP policy. Each candidate path has a unique PLSP-ID generated by PCC. Only one candidate path can be active at any instance. A candidate path is made active based on the preference of the path. On MBB, PCE will update the candidate path with a new path, and two paths will briefly co-exist. Two paths are uniquely identified by the 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.

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.

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.

3.2.3. 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.

3.3. SR P2MP Policy and Replication Segment TLVs and Objects
3.3.1. 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>]

3.3.2. Replication Segment Objects

Replication segment can be constructed via the following objects

<Common Header>
[SRP>
<LSP>
[replication-sid>]

as described in

[draft-sivabal-pce-binding-label-sid-06#section-3]

[ERO-Attributes Object>]

as per [draft-koldychev-pce-multipath]

3.3.3. P2MP Policy vs Replication Segment

Note P2MP Policy and Replication segments objects have a very close definition. They can be told apart via the following abstracts:

o The P2MP Policy will always have an association list object at least in its PC-Init message. While the replication segment does not have the association list object.

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

3.3.4. P2MP Policy and Replication Segment general considerations

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. They will 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 ERO path Id and generates new path Id for new instructions. This means the pathId are maintained for each specific forwarding and label instructions until these instructions are deleted. For example: When the first leaf is added I get instructions, path-id A, path-id B on a node. On a second leaf add, according to the path calculated, PCE might just append the existing instruction A,B with C,D. This is done by sending a PC Update with downstream-index A, B, C,D.

ERO path-Id requirements are mentioned in draft-koldychev-pce-multipath-01 and MUST be followed.

4. Object Format

4.1. Open Message and Capability 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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 https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-p2mp-08#section-5.2

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 https://tools.ietf.org/html/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 [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.

Finally the open message needs to include the MULTIPATH CAPABILITY TLV as defined in [draft-koldychev-pce-multipath-01]

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 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 [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

4.3.1.2. 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.

<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>
<tbody>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
</tbody>
</table>

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

4.3.1.3. 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.

<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>
<tbody>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+------------------------------------------</td>
</tr>
</tbody>
</table>

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.

4.3.1.4. 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

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
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Leaf type |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Source IPv4 address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination IPv4 address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        ...                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination IPv4 address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**IPV6-P2MP END-POINTS:**

```
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.

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 Path-Instance-ID to identify a P2MP LSP under 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
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 within 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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-voyer-spring-sr-replication-segment-00 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. shared vs non-shared replication segment

A non-shared tree 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 this case is when there is no upstream assigned label in the PTA and aggregation of MVPNs into one P-Tunnel is not desired.

A shared tree, on other hand, has a non Zero label field in the PTA and is used when there is an upstream assigned label in the PTA and aggregate of MVPNs into one P-Tunnel is desired.

4.5.2. 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 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.

4.5.3. Label action rules in replicating segment

If a node is ingress, transit, leaf or Bud, is implicitly identified
by the following information in the replication segment:

- If there is a next-hop with local loop back it is a Leaf

- If there is a next-hop with local loop back and a next-hop with swap downstream it is a Bud

- If incoming label and next-hop is downstream node loop back ip and there is p2mp policy then Ingress

- If incoming label and next-hop is downstream node loop back ip and no p2mp policy then it is a Transit

5. Examples of PCEP messages between PCE and PCEP

```
+--------+
| Rep    |
| Transit|
| Non    |
| Rep    |
| Transit|
| ROOT   |
| A      |
|--------+
```

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

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 |

<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 |
| I Pv4 Association Source = <pce-address> |
| Type | Length |
| Root = A |
| TREE-ID = 0 |
| Type | Length |
| ProtOrigin 10 | Reserved |
| Originator ASN |
On Response to the above initiate message, PCC generates a Tree-ID, PLSP-ID, Instance-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 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                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                             | PST = TBD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags = 0                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SRP-ID-number  = 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
|                                                                 |
|                  PST = TBD                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                             | PST = TBD       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          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       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags = 0                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SRP-ID-number  = 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
|                                                                 |
|                  PST = TBD                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags = 0                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SRP-ID-number  = 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
|                                                                 |
|                  PST = TBD                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Flags = 0                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SRP-ID-number  = 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TLV Type = 28 (PathSetupType)| TLV Len = 4                   |
|                                                                 |
|                  PST = TBD                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

<END POINT OBJECT>

|                          Leaf type =1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Leaf type =1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 address = A                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address = D                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address = E                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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 the node to download the replication segment (forwarding instructions).

5.2. PCE P2MP LSP Calculation and Replication Segment 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 TE-PATH-BINDING TLV and label download information using 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 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-barth-pce-segment-routing-policy-cp-02

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.
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.
<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 = 100 <default> |
+-----------------+----------+---+
| <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 = b1 |
+-----------------+----------+---+
| ipv4-address = NH2 |
A sample PC Initiate message to the transit replication policy C for cp1
Lets assume C is connected to D 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 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 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

<LSF 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>

<ERO-ATTRIBUTES OBJECT>

Flags  | Oper|
-------|-----|
|      |     |

Type   | Length|
-------|-------|
|       |       |

| Reserved |
-------|

ERO-path Id = 3

| Back-up ero path id = 4 |
-------|

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

<ERO-ATTRIBUTES OBJECT>

Flags  | Oper|
-------|-----|
|      |     |

Type   | Length|
-------|-------|
|       |       |

| 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|
-------|-------|
|       |       |

| Reserved |
-------|

ERO-path Id = 5
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.
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

Email: saranya.rajarathinam@nokia.com

Takre Saad
Juniper
Ottawa
Canada

tsaad@juniper.net