PCEP extensions for p2mp sr policy
draft-dhs-spring-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
This Internet-Draft will expire on October 8, 2017.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................. 3
2. Conventions used in this document .......... 3
3. Overview of PCEP Operation in SR P2MP Network .... 3
   3.1. P2MP Identification ..................... 4
   3.2 High-Level Procedures for P2MP SR LSP Instantiation .... 4
      3.2.1. New Objects for SR P2MP LSP instantiation ... 5
   3.3. High-Level Procedures for P2MP Global Optimization ... 5
   3.4. High-Level Procedures for Fast Reroute ...... 6
4. Object Format .................................. 7
   4.1. Open Message and Capability Exchange ...... 7
   4.2. Symbolic Name in PCInitiate message from PCC .... 7
   4.3. PCE Update, Root Report message .......... 8
      4.3.1. Extension of the LSP Object, SR-P2MP-LSPID-TLV ... 8
      4.3.2. END-POINTS Objects ................. 9
   4.4. PCInitiate Message and P2MP Tree Instantiation ... 12
      4.4.1. New SR-P2MP-CCI Object ............... 13
      4.4.2. New Optional IPV4-ADDRESS TLV ........ 13
      4.4.3. UNNUMBERED IPV4-ID-ADDRESS TLV: ........ 15
      4.4.4. New Optional IPV6-ADDRESS TLV .......... 15
   4.5. Global Optimization and Make Before Break .... 15
   4.7. Fragmentation ............................. 16
5. Example workflow ............................. 16
6. IANA Considerations .......................... 18
7. Security Considerations ..................... 18
8. References .................................. 19
   8.1. Normative References .................... 19
   8.2. Informative References .................. 19
7. Acknowledgments .............................. 19
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. We call it an SR Replication Policy.

A Point-to-Multipoint (P2MP) segment connects a Root node to a set of Leaf nodes in a Segment Routing Domain. We define two types of a P2MP segment: Spray and Tree.

Spray P2MP segment enables a Root node to directly replicate a packet using a SR path to each Leaf node.

For a TreeSID P2MP segment, a controller 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.

For a TreeSID the tree can be built manually via the PCE or PCC by indicating the root and the leaves or dynamically via MVPN procedures, where the root of a tree discovers the leaves via MVPN procedures and updates the PCE of the root and the set of leaves. In either case the PCE computes and programs the P2MP Tree on the PCCs.

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 Tree P2MP SR policy, TreeSID network, 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 or PCC providing the PCE with the relevant information. The PCC can discover the Root and the leaves via MVPN procedures or via manual configuration.
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 from the Root to the set of the Leaves. This information specifies label operation on the relevant nodes. As an example for a label Push operation, outgoing interface and their corresponding labels are programmed on the PCC via the PCE. In the case of swap operation, an incoming label and a set of outgoing interfaces and their corresponding labels are programmed on the PCCs. In the Case of pop operation, incoming label that needs to be removed is programmed. PCE could also calculate and download additional information such as next-hops for link/node protection or initiate a make-before-brake for P2MP Path optimization.

3.1. 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 LSP that is constant for the lifetime of a PCEP session. When an LSP is being globally optimized it will maintain its PLSP-ID over the 2 instances of the LSP, the current LSP and the new optimized LSP. It is possible to establish standby P2MP LSPs to protect primary P2MP SR policies, and standby P2MP LSPs can co-exist along with primary P2MP LSP. Since PLSP-ID is unique per LSP, each standby P2MP LSP has a unique PLSP-ID.

LSP-ID: is assigned by PCE for each PLSP-ID. It is used for global optimization of an existing LSP via Make-Before-Break (MBB) procedures. An optimized P2MP LSP is instantiated with the same PLSP-ID but a unique LSP-ID. MBB procedures first switches traffic from the previous to the new P2MP LSP, and then destroys the previous P2MP LSP.

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

IPv4/IPv6 Sender Address: is equivalent to the first MPLS node on the path, as per [RFC3209], Section 4.6.2.1

3.2 High-Level Procedures for P2MP SR LSP Instantiation

A P2MP LSP’s ROOT and Leaves can be discovered via MVPN procedures or static configuration of the ROOT and Leaves on the PCE or PCC.

In case of MVPN Procedures, PCC will use Report messages to update
the PCE with discovered leaf or a set of leaves as per P2MP [draft-sr-mvpn-procedure or alike]. PCC will assign a PLSP-ID and use the P2MP-ID it had specified in the BGP MVPN PTA. PCC will update the PCE with the root and discovered leaves via a PCRpt message. Note the LSP-ID will be set to 0 or NA in this first report message as the PCE has not instantiated the P2MP LSP yet.

PCE will instantiate the P2MP LSP and update PCC (root, transit and leaves) via a PCInit message. The PCInit message will contain a valid LSP-ID assigned by PCE for this P2MP path. It will also download the corresponding label and local protection (Fast Re-Route) information.

In short, each P2MP LSP will have a unique PLSP-ID assigned by PCC and LSP-ID assigned by PCE. This is true for LSP redundancy where the Primary LSP is protected by one or multiple backup LSPs. The backup LSPs will have a unique PLSP-ID and LSP-ID. PCE will use the same PLSP-ID and LSP-ID for any new leaves that are discovered from here on. If these leaves are discovered on routers that are part of the P2MP LSP path, then PCUpd is sent from PCE to necessary PCCs (root, transit and leaves) with the PLSP-ID and LSP-ID. If the new leaves are discovered that are not part of the P2MP Tree yet, then an PCInit message is sent down to the relevant transit and/or leaf nodes with the current PLSP-ID and LSP-ID.

3.2.1. New Objects for SR P2MP LSP instantiation

A new object <SR-P2MP-CCI> is defined for the controller to specify the forwarding instructions (label instructions). This can be included in PcRpt, PcInitiate and PcUpd messages.

It should be noted that every PcRpt, PcInitiate and PCUpd messages will contain full list of the Leaves and labels and forwarding information that is needed to build the 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 discovered 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.

3.3. High-Level Procedures for P2MP Global Optimization

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. After the optimized LSP is downloaded a MBB
procedure is performed and the previous instance of the 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. 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. After the optimized LSP has been downloaded and verified via PCC PCRpt message, the MBB procedure can be performed to switch between the 2 instances of the LSP. The previous instance will be removed from PCCs.

3.4. High-Level Procedures for 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.

![Figure 1](image1)

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 section 4.4.2, 4.4.3, 4.4.4.

![Figure 2](image2)

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.

4. Object Format

4.1. Open Message and Capability Exchange

Format of the open object

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

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

During PCEP session establishment phase, PCEP Speakers advertise their capabilities to support PCECC extensions to include the new Path Setup Type as per draft-ietf-pce-pcep-extension-for-pce-controller-00. In addition they need to set flags N, M, P in the STATEFUL-PCE-CAPABILITY TLV as defined in 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 rfc8306 section-3.1.2 and a new value needs to be defined for SR-P2MP.

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 the source IP + P2MP-ID of the LSP. The P2MP-ID is a unique ID that identifies the P2MP on the Root (Source) as such the combination of Source IP + P2MP-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. PCE Update, Root Report message

The Root node on reception of a P2MP path request, via MVPN procedures or manual Configuration on the PCCs, will initiates an Report to PCE through a PCRpt message.

The format of the PC Report message is as follows:

```
<Common Header>
[<SRP>]
<LSP>
[<end-points-list> | <SR-P2MP-CCI>]
```

4.3.1 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 P2MP TE LSP uniquely.

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-08

SR-IPv4-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 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Type=TBD       |     Length=12     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       LSP ID       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      IPv4 Tunnel Sender Address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      P2MP ID(color) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
SR-IPV6-P2MP-LSP-IDENTIFIER TLV:

```
+-------------+-----------------------------+
|           Type=TBD            |           Length=20           |
+-------------+-----------------------------+
|            LSP ID             |                               |
+-------------+-----------------------------+
|                          +                               |
|                           | IPv6 tunnel sender address   |
|                           | (16 octets)                 |
|                           | +                            |
|                           | +                            |
|                           | +                            |
+-------------+-----------------------------+
|            P2MP ID(color)      |                               |
+-------------+-----------------------------+
```

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

LDS ID: is a unique ID assigned by PCE to indicate sub-instances of an P2MP LSP for global or local optimization

IPv4/IPv6 Sender address: contains the sender node’s IP address (Root Node)

P2MP ID (Color): contains the 32-bit 'P2MP ID' identifier used in the PTA of BGP AD Route.

4.3.2 END-POINTS Objects

In order for the Root to indicate the leaves and its corresponding operation (Add/Remove/Modify/DoNotModify), the PC Report message is extended to include P2MP End Point <P2MP End-points> Object which is defined in rfc8306

IPV4-P2MP END-POINTS:
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.

A sample Report message of a P2MP tunnel request, and leaf Add report is noted below:

Note as it was mentioned before:

P2MP-ID (color) = Tunnel Identifier color which identifies a unique P2MP segment at the Root and is advertised via the PTA (BGP)

LSP ID is used for Global optimization of the P2MP Tree. contains the 16-bit ‘LSP ID’ identifier

P2MP Tunnel Request, Report generated by the Root to the PCE. The Root should populate the Tunnel-sender Addr, P2MP-ID(color), PLSP-ID. While the LSP ID is set to 0 for the PCE to assign.

    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Flags = 0                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        SRP-ID-number  = 0                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  TLV Type = 28 (PathSetupType) | TLV Len = 4               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                             | PST = TBD       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  PLSP-ID = X            |     A:1,D:1,N:1       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                Type=TBD            |           Length=8            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                   LSP ID = 0         |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       IPv4 Tunnel Sender Address = x.y.z.w        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                   P2MP_ID(color) = Y                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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  = 0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
| PST = TBD |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PLSP-ID = X | A:1,D:1,N:1 |
| Type=TBD | Length=8 |
| LSP ID = 0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv4 Tunnel Sender Address = x.y.z.w |
| P2MP_ID(color) = Y |
| Leaf type =1 |
| Source IPv4 address = x.y.z.w |
| Destination IPv4 address = a.b.c.d |
| Destination IPv4 address = a.b.c.e |

4.4. PCInitiate Message and P2MP Tree Instantiation

In response to a new leaf add PCRpt message, the PCE calculates the path and assigns labels along the path and sets up the path by sending a PCInitiate Message(label download) to each node along the path (Root, transits nodes and leaves). A new object <SR-P2MP-CCI> is defined for the controller to specify the forwarding instructions (label instructions). This can be included in PCRpt, PcInitiate and PcUpdate messages.

For optimization it is recommended for the PCE to send the PCInitiate message starting from the LEAVES to the Transit nodes and finally the Root.

The transit and leaf nodes generate a Path Computation State Report (PCRpt) with SR-P2MP-LSP-ID TLV and SR-P2MP-CCI object to indicate the state of the label download. Once the controller gets a report from all the leaves and transit nodes that the label download was
successful, it will send a PCInit Message with the SR-P2MP-CCI object (label instruction) to the Root node.

A successful report (including SR-P2MP-CCI object and P2MP LSP ID TLV) back from the root for the label download, means the P2MP policy is successfully UP. The PLSP-ID used in the PCInit message is the original identifier used by the ingress PCC, so the transit LSR could have multiple central controller instructions that have the same PLSP-ID. The PLSP-ID in combination with the SR-P2MP-LSP-ID-TLV MUST be unique.

Format of PC Initiate Message:
<Common Header>  
<SRP>  
<LSP>  
<SR-P2MP-CCI>

Format of Pc Update Message: is like PC Initiate Message, but the Create Bit in the LSP Object will be Set to 0.

4.4.1. New SR-P2MP-CCI Object

This is a variation of the CC-ID object defined in draft-ietf-pce-pcep-extension-for-pce-controller-00 SR-P2MP-CCI Object-Class is TBD.

CCI Object-Type is 1 for the MPLS Label.

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 - 0 - Down - means label download was not successful 1 - Up - means label download was successful

4.4.2. New Optional IPv4-ADDRESS TLV
The IP-Address TLVs download the label and its instructions to the
PCC. These instructions could include information about local
Protection information or if the label is an incoming label or an
outgoing label. Additionally they could identify a node as a BUD node
or just a transit node.

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
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.

4.4.3. UNNUMBERED IPV4-ID-ADDRESS TLV:

```
+---------------+---------------+-----------------+
| Type=TBD      | Length = 16   |                |
| Label         | Rsvd          | Flag|I|B|S|E|P|
+---------------+---------------+-----------------+
                     | Node-ID       |
+---------------+---------------+-----------------+
                     | Interface ID  |
+---------------+---------------+-----------------+
```

4.4.4. New Optional IPv6-ADDRESS TLV

```
+---------------+---------------+-----------------+
| Type=TBD      | Length = 24   |                |
| Label         | Rsvd          | Flag|I|B|S|E|P|
+---------------+---------------+-----------------+
                     |                |
+---------------+---------------+-----------------+
                     // IPv6 address (16 bytes)
+---------------+---------------+-----------------+
                     |                |
+---------------+---------------+-----------------+
```

4.5. Global Optimization and Make Before Break

If controller optimizes a tree for any reason, it will download the tree with same PLSP-ID and P2MP-ID (Color) but a new LSP-ID.
The new optimize LSP is downloaded via the same mechanisms explained above and a new LSP-ID for this instance of the LSP.

After the new tree is established and the report messages arrive from the source, transit and leaf nodes confirming successful download, the previous tree is deleted starting from the root. This delete of the previous LSP-ID will force the root to do a switch (MBB) to the new LSP-ID P2MP tree.

4.6. 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.

4.7. Fragmentation

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

5. Example workflow

In the workflow below after discovering the root and leaf nodes via MVPN procedures the root generates a PCRpt message to the PCE with the discovered root and leaves via the endpoint objects, which in order forces the PCE to compute the P2MP Tree and generates a PCInitiate to the Root, downloading the label information to the Root, Bud/Transit and Leaves.

A PC Initiate message on Root,transit and leaf nodes - indicates a new label instruction and the existing label instruction should remain unmodified(if any present). If the old instructions are no longer needed, a PC Update message indicating the old label instructions and the R bit in the LSP object SET is sent to the nodes(to remove the labels).

A PCUpd message on any node (root, transit or leaf) - indicates an overwrite of the existing label instruction if any present.

A sample message flow is depicted below:

When leaves(C,D) are discovered, at R a PCInitiate with 2 label instruction is instantiated by the PCE. The two labels correspond to an egress label and a protection nextHop for FRR. The PCE will download a PCInitiate message with CC-ID X, egress label(A), flag
E (as the Protection NextHop CCID will Follow) and a IPv4 nextHop of N1, in addition CC-ID Y = egress label (B), flag P (as it is a protection NextHop) and IPv4 nextHop of N2, are instantiated by the PCE.

The Root will use a PCRpt message to acknowledge this initiate.

The PC will also download the necessary information to the BUD/Transit router. In case of the BUD/Transit router the flags <I, B, S> are set. This means that for the incoming label the label should be swapped with an outgoing label, but also the payload should be replicated and a copy forwarded locally as there is a BUD present on this router.

The BUD/Transit node will use a PCRpt message to acknowledge this update as well. The Leafs (not shown) will also get the PCInit message with Flag of only <I> which indicates a pop and forward operation.
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 at Nokia for his 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