

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: November 1, 2022

H. Chen  
M. McBride  
Futurewei  
Y. Fan  
Casa Systems  
Z. Li  
X. Geng  
Huawei  
M. Toy  
G. Mishra  
Verizon  
A. Wang  
China Telecom  
L. Liu  
Fujitsu  
X. Liu  
Volta Networks  
April 30, 2022

Stateless SRv6 Point-to-Multipoint Path  
draft-chen-pim-srv6-p2mp-path-06

Abstract

This document describes a solution for a SRv6 Point-to-Multipoint (P2MP) Path/Tree to deliver the traffic from the ingress of the path to the multiple egresses/leaves of the path in a SR domain. There is no state stored in the core of the network for a SR P2MP path like a SR Point-to-Point (P2P) path in this solution.

Requirements Language

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on November 1, 2022.

#### Copyright Notice

Copyright (c) 2022 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 (<https://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. Overview of P2MP Multicast Tree . . . . .	3
3. Encoding P2MP Multicast Tree . . . . .	5
4. Procedures/Behaviors . . . . .	8
4.1. Procedure/Behavior on Ingress Node . . . . .	8
4.2. Procedure/Behavior on Transit Node . . . . .	9
4.3. Procedure/Behavior on Egress Node . . . . .	11
5. Stateless SRv6 P2MP Path for Ingress . . . . .	11
6. Protection . . . . .	13
6.1. Global Protection . . . . .	13
6.2. Local Protection . . . . .	13
7. IANA Considerations . . . . .	13
8. Security Considerations . . . . .	13
9. Acknowledgements . . . . .	14
10. References . . . . .	14
10.1. Normative References . . . . .	14
10.2. Informative References . . . . .	15
Appendix A. Example IPv6 Header using G-SRv6 . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

The Segment Routing (SR) for unicast or Point-to-Point (P2P) path is described in [RFC8402]. For SR multicast or Point-to-Multipoint (P2MP) path/tree, it may be implemented through using multiple SR P2P paths. The function of a SR P2MP path/tree from an ingress node to multiple (say n) egress/leaf nodes is implemented by n SR P2P paths. These n P2P paths are from the ingress to those n egress/leaf nodes of the P2MP path/tree. This solution may waste some network resources such as link bandwidth.

An alternative solution proposed in [I-D.shen-spring-p2mp-transport-chain] uses a number of P2MP chain tunnels to implement a P2MP path/tree from an ingress to n egress/leaf nodes. Each P2MP chain tunnel is a tunnel from the ingress to a leaf node as its tail end and may have some leaf nodes as its bud nodes along the tunnel. This alternative solution improves the usage of network resources over the solution above using pure P2P paths. However, these two solutions are based on SR P2P paths.

A solution for a SR P2MP path/tree using a P2MP multicast tree is proposed in [I-D.ietf-pim-sr-p2mp-policy]. For a SR P2MP path/tree from an ingress/root to multiple egress/leaf nodes, a multicast P2MP tree is created to deliver the traffic from the ingress/root to the egress/leaf nodes. The state of the tree is instantiated in the forwarding plane by a controller such as PCE at Root node, intermediate Replication nodes and Leaf nodes of the tree. This is not consistent with the SR principles in which no state is stored at the core of the network.

This document describes a new solution for a SRv6 Point-to-Multipoint (P2MP) Path/Tree to deliver the traffic from the ingress of the path to the multiple egresses/leaves of the path in a SR domain. This solution uses a P2MP multicast tree without storing its state in the core of the network for a SR P2MP path/tree like a SR P2P path. For distinguishing a SRv6 P2MP path/tree used in the other solutions with storing some states in the core, a new name, called stateless SRv6 P2MP path/tree, is used in the solution in this document. Even though SRv6 P2MP path/tree and stateless SRv6 P2MP path/tree are used interchangeably in the document, they both mean stateless SRv6 P2MP path/tree.

## 2. Overview of P2MP Multicast Tree

For a SR P2P path from its ingress to its egress, a segment list for the path is provided to the ingress. The ingress pushes the list into a packet, and the packet is delivered to the egress according to the segment list without any state in the core of the network.

For a SR P2MP path from its ingress to multiple egress/leaf nodes, a segment list for the P2MP path is provided to the ingress. The ingress pushes the list into a packet, and the packet is delivered to the multiple egress/leaf nodes according to the segment list without any state in the core of the network.

Figure 1 shows a SR P2MP path from ingress/root R to four egress/leaf nodes L1, L2, L3 and L4. Nodes P1, P2, P3 and P4 are the transit nodes of the P2MP path.

Suppose that X-m is the segment identifier (SID) of node X. X-m is an adjacent SID or node SID. For simplicity, we assume X-m is a node SID in the illustrations below. R-m, P1-m, P2-m, P3-m, P4-m, L1-m, L2-m, L3-m and L4-m are the SIDs of the nodes on the SR P2MP path. They are multicast SIDs or replication SIDs in general.

A multicast SID is a SID from a multicast SID block. In a SR domain supporting SR multicast, each node has a multicast node SID, which is globally significant. A multicast SID of a node on a SR P2MP path is associated with the SIDs of its next hop (or say downstream) nodes. When the node receives a packet with its multicast SID, it duplicates and sends the packet to each of its next hop nodes according to their SIDs.

If node P on a SR P2MP path has B ( $B > 1$ ) next hop nodes along the path, the SID of node P, P-m, MUST be a multicast SID when it is in the segment list for the P2MP path. The SIDs of the B next hop nodes just follow P-m in the segment list. When node P receives the packet with P-m as destination address (DA), it duplicates and sends the packet to each of the B next hop nodes along the P2MP path.

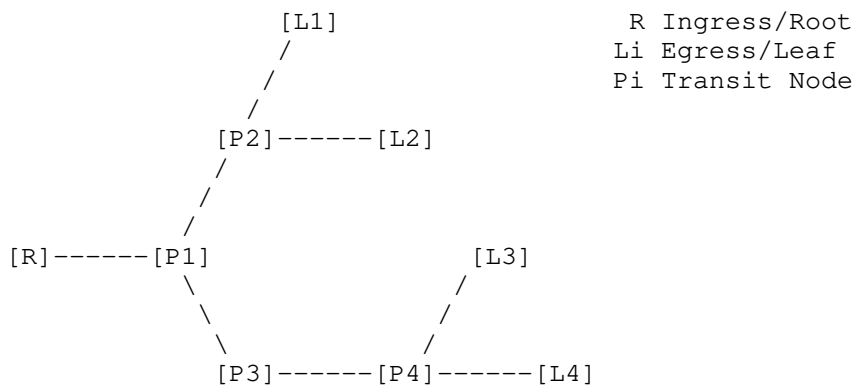


Figure 1: SR P2MP Path from R to L1, L2, L3 and L4

<P1-m, P2-m, P3-m, L1-m, L2-m, P4-m, L3-m, L4-m> is a segment list for the SR P2MP path in Figure 1 to be pushed into a packet at ingress/root R. Node P1 has 2 next hop nodes P2 and P3 along the P2MP path. The next hop nodes' SIDs P2-m and P3-m follow P1-m, which is P1's multicast SID. When P1 receives a packet with DA = P1-m transported by the P2MP path, it duplicates and sends the packet to its next hop nodes P2 and P3 according to P1-m, P2-m and P3-m.

The number of branches or next hops from node P1 is a value of one argument in P1-m, called N-Branches. The value of N-Branches in P1-m is 2. With this information, node P1 duplicates and sends the packet to 2 next hop nodes P2 and P3, which are indicated by the 2 SIDs P2-m and P3-m following P1-m.

The number of SIDs under node P1 is a value of another argument in P1-m, called N-SIDs. It is the number of the SIDs encoding the sub-trees from P1 and the SIDs following. The sub-trees are encoded by 7 SIDs following P1-m in the segment list. The value of N-SIDs in P1-m is 7.

Since there are 2 branches or next hops (i.e., L1 and L2) from node P2, the value of N-Branches in P2-m is 2. The two sub-trees from P2 are encoded by 2 SIDs (i.e., L1-m and L2-m) and there are 3 SIDs (i.e., P4-m, L3-m, L4-m) following them. The value of N-SIDs in P2-m is 5 (2 + 3). With this information, before sending the packet to node P2, node P1 sets DA to P2-m, SL in SRH to 5 (the N-SIDs in DA = P2-m), and sends the packet to DA (i.e., P2).

Since there are 1 branch or next hop (i.e., P4) from node P3, the value of N-Branches in P3-m is 1. The sub-tree from P3 is encoded by 3 SIDs (i.e., P4-m, L3-m and L4-m) and no SIDs following them. The value of N-SIDs in P3-m is 3. With this information, before sending the packet to node P3, node P1 sets DA to P3-m, SL in SRH to 3 (the N-SIDs in DA = P3-m), and sends the packet to DA (i.e., P3).

Each node on the SR P2MP path sends the packet to its next hop nodes according to the segment list and no state is stored in any transit node (i.e., the core of the network). The packet is delivered to the egress/leaf nodes from the ingress.

### 3. Encoding P2MP Multicast Tree

For a sub-tree ST of a SR P2MP path from the ingress node of the P2MP path, suppose that

- o the multicast SID of the next hop node NH is mSID;

- o there are B branches (i.e., outgoing interfaces) to the next hop node BNH-j (j = 1, ..., B) from node NH along the sub-tree, the multicast SID of BNH-j is mSID-j;
- o SidSeq-j (j = 1, ..., B) is the SID sequence in the segment list encoding the sub-trees from node BNH-j.

Sub-tree ST is encoded as segment list

	$\langle$	$\text{mSID},$	$\text{mSID-1}, \dots, \text{mSID-B},$	$\text{SidSeq-1}, \dots,$	$\text{SidSeq-B} \rangle$
		$\underbrace{\hspace{1.5cm}}$	$\underbrace{\hspace{1.5cm}}$	$\underbrace{\hspace{1.5cm}}$	$\underbrace{\hspace{1.5cm}}$
SIDs of	NH	B branches/next-hops	sub-trees	sub-trees	
		BNH-1 of node NH	from BNH-1	from BNH-B	

where mSID contains the number of branches in its N-Branched field, which is B, and the number of SIDs in its N-SIDs field, which is the number of the SIDs encoding the sub-trees from NH and the SIDs following (No SID following in this case). The SIDs following mSID encode the sub-trees. The value of N-SIDs field in mSID is B plus the number of the SIDs in SidSeq-1, ..., SidSeq-B. mSID-j (j = 1, ..., B) contains the number of branches in its N-Branched field, which is the number of branches from node BNH-j, and the number of SIDs in its N-SIDs field, which is the number of the SIDs in SidSeq-j to SidSeq-B.

For the P2MP path in Figure 1 from ingress node R to egress nodes L1, L2, L3 and L4, there is one sub-tree from R. Suppose that the multicast SIDs of P1, P2, P3, P4, L1, L2, L3 and L4 are P1-m, P2-m, P3-m, P4-m, L1-m, L2-m, L3-m and L4-m respectively.

The sub-tree is encoded as segment list

	< P1-m,	P2-m, P3-m,	L1-m, L2-m,	P4-m, L3-m, L4-m >
	\	\	\	\
SIDs of	P1	2 branches/next-hops	sub-trees	sub-tree
		P2 and P3 of node P1	from P2	from P3

where

- o L1-m, L2-m is the SID sequence (SidSeq-1) in the segment list encoding the sub-trees from P2.
- o P4-m, L3-m, L4-m is the SID sequence (SidSeq-2) in the segment list encoding the sub-tree from P3.
- o P1-m's N-Branched field is set to 2 since there are 2 branches from P1 and its N-SIDs field to 7 since there are 7 SIDs following P1-m, which "points" to the sub-tree from P1.

- o P2-m's N-Branched field is set to 2 since there are 2 branches from P2 and its N-SIDs field to 5 since there are 5 SIDs in SidSeq-1 and SidSeq-2. The N-SIDs = 5 acts as a pointer to the sub-tree from P2.
- o P3-m's N-Branched field is set to 1 since there is 1 branch from P3 and its N-SIDs field to 3 since there are 3 SIDs in SidSeq-2. The SIDs = 3 acts as a pointer to the sub-tree from P3.
- o P4-m's N-Branched field is set to 2 and its N-SIDs field to 2.

Figure 2 shows in details the segment list, which is an encoding of the sub-tree of the SR P2MP path from R via P1 to L1, L2, L3 and L4.

		N-Branched	N-SIDs		
1	L4's Multicast SID Locator	0	0	Arguments	L4-m
2	L3's Multicast SID Locator	0	0	Arguments	L3-m
3	P4's Multicast SID Locator	2	2	Arguments	P4-m
4	L2's Multicast SID Locator	0	0	Arguments	L2-m
5	L1's Multicast SID Locator	0	0	Arguments	L1-m
6	P3's Multicast SID Locator	1	3	Arguments	P3-m
7	P2's Multicast SID Locator	2	5	Arguments	P2-m
8	P1's Multicast SID Locator	2	7	Arguments	P1-m

Figure 2: Encoding of sub-tree of path from R via P1 to L1 - L4

A bud node is considered as a loopback leaf of itself. The bud node will have one more branch for this loopback leaf. For example, suppose that L4 is a bud node and connected to a leaf L5 (not shown in Figure 1). The N-Branched in L4-m as multicast SID of bud L4 is 2 since there are 2 branches from L4: one to L5 and the other to L4 itself as a leaf.

Figure 3 shows in details the segment list, which is an encoding of the sub-tree of the SR P2MP path from R via P1 to L1, L2, L3, L4 and L5.

For L4-m as multicast SID of bud L4, its N-Branched = 2, N-SIDs = 2. The N-SIDs = 2 acts as a pointer to the sub-tree from L4. This sub-

tree has 2 branches: one from L4 to L5, and the other from L4 (loopback) to L4 itself.

The others in Figure 3 are the same as or similar to those in Figure 2.

		N-Branches	N-SIDs		
1	L4's Multicast SID Locator	0	0	Arguments	L5-m
2	L4's Multicast SID Locator	0	0	Arguments	L4-m
3	L4's Multicast SID Locator	2	2	Arguments	L4-m
4	L3's Multicast SID Locator	0	0	Arguments	L3-m
5	P4's Multicast SID Locator	2	4	Arguments	P4-m
6	L2's Multicast SID Locator	0	0	Arguments	L2-m
7	L1's Multicast SID Locator	0	0	Arguments	L1-m
8	P3's Multicast SID Locator	1	5	Arguments	P3-m
9	P2's Multicast SID Locator	2	7	Arguments	P2-m
	P1's Multicast SID Locator	2	9	Arguments	P1-m

Figure 3: Encoding of sub-tree of path from R via P1 to L1 - L5

#### 4. Procedures/Behaviors

This section describes the procedures or behaviors on the ingress, transit and egress/leaf node of a SR P2MP path to deliver a packet received from the path to its destinations.

##### 4.1. Procedure/Behavior on Ingress Node

For a packet to be transported by a SR P2MP Path, the ingress of the P2MP path duplicates the packet for each sub-tree of the SR P2MP path branching from the ingress, pushes the segment list encoding the sub-tree into the packet by executing H.Encaps [RFC8986] and sends the packet to the next hop node along the sub-tree.

Regarding to the finite size of the segment list, a sub-tree can be "split" into multiple sub-trees such that each of the sub-trees can be encoded in the segment list of the finite size.



For example, there is one sub-tree from the ingress R of the SR P2MP path in Figure 1 via next hop node P1 towards egress/leaf nodes L1, L2, L3 and L4.

For this sub-tree, the ingress R duplicates the packet, set the destination address (DA) to P1-m (i.e., multicast SID of node P1), pushes the segment list without P1-m (i.e., <P2-m, P3-m, L1-m, L2-m, P4-m, L3-m, L4-m>) encoding the sub-tree into a Segment Routing Header (SRH) of the packet by executing H.Encaps and sends the packet to DA (i.e., node P1). The contents of the multicast SIDs P1-m, P2-m, P3-m, L1-m, L2-m, P4-m, L3-m, L4-m are shown in Figure 2.

Suppose that the duplicated packet is Pkt0 for the sub-tree. The execution of H.Encaps pushes an IPv6 header (i.e., SRH) to Pkt0 and sets some fields in the header to produce an encapsulated packet Pkt'. Pkt' is represented in the following:

$$\text{Pkt}' = (\text{SA}=\text{R}, \text{DA}=\text{P1-m}) \left( \text{L4-m}, \text{L3-m}, \dots, \text{P3-m}, \text{P2-m}; \text{SL}=7 \right) \text{Pkt0}$$

corresponds to: <P2-m, P3-m, ..., L3-m, L4-m>

where DA=P1-m means that the destination address (DA) is set to P1-m; SA=R means that the source address (SA) is set to R; SL=7 means that the number of Segments Left (SL) is 7.

#### 4.2. Procedure/Behavior on Transit Node

When a transit node of a SR P2MP path receives a packet transported by the P2MP path, the DA of the packet is a multicast SID of the node and the packet contains a segment list for the next hops and the sub-trees of the transit node. The DA and the segment list comprise the information for encoding the sub-trees.

For example, when node P1 receives a packet transported by the SR P2MP path in Figure 1, the packet's DA is P1-m (which is a multicast SID of node P1) and the segment list in the packet is <P2-m, P3-m, L1-m, L2-m, P4-m, L3-m, L4-m>.

The N-Branches field (which has value of B) of the DA indicates that there are B branches or next hops from the transit node. The N-SIDs field of the DA indicates the number of SIDs for the B sub-trees from the transit node. The multicast SIDs of the B next hop nodes are the first B multicast SIDs of the segment list in the packet.

For example, the N-Branches field (which has value of 2) of DA = P1-m indicates that there are 2 branches or next hops from node P1. The N-SIDs field (which has value of 7) of the DA = P1-m indicates that there are 7 SIDs for the 2 sub-trees from node P1.

The first multicast SID (P2-m) of the segment list is the SID of the first next hop node (P2); The second multicast SID (P3-m) of the segment list is the SID of the second next hop node (P3).

After the multicast SIDs of the next hop nodes, there are B SidSeqs (SIDs sequences) for the B sub-trees. The N-SIDs field (which has value of S1) of the first multicast SID of the next hop nodes indicates that there are S1 SIDs from SidSeq-1 to SidSeq-B; the N-SIDs field (which has value of S2) of the second multicast SID of the next hop nodes indicates that there are S2 SIDs from SidSeq-2 to SidSeq-B; and so on.

For example, there are 2 SidSeqs for the 2 sub-trees from node P1 after the multicast SIDs P2-m and P3-m of the next hop nodes P2 and P3. The N-SIDs field of P2-m (the first multicast SID of the next hop nodes) has value of 5, indicating that there are 5 SIDs from SidSeq-1 to SidSeq-2.

The N-SIDs field of P3-m (the second multicast SID of the next hop nodes) has value of 3, indicating that there are 3 SIDs from SidSeq-2.

The transit node duplicates the packet for each next hop under it, sets the DA of the duplicated packet to the multicast SID of the next hop, SL in SRH to the N-SIDs in the DA, and sends the packet to the DA (i.e., the next hop).

For example, node P1 duplicates the packet for the first next hop P2, sets DA to P2-m (multicast SID of P2), SL in SRH to 5 (N-SIDs in P2-m), and sends the packet Pkt' to DA (i.e., P2).

$$\text{Pkt}' = (\text{SA}=\text{R}, \text{DA}=\text{P2-m}) (\text{L4-m}, \text{L3-m}, \text{P4-m}, \text{L2-m}, \text{L1-m}; \text{SL}=5) \text{Pkt0}$$

corresponds to:  $\langle \text{L1-m}, \text{L2-m}, \text{P4-m}, \text{L3-m}, \text{L4-m} \rangle$

Node P1 duplicates the packet for the second next hop P3, sets DA to P3-m (multicast SID of P3), SL in SRH to 3 (N-SIDs in P3-m), and sends the packet Pkt' to DA (i.e., P3).

$$\text{Pkt}' = (\text{SA}=\text{R}, \text{DA}=\text{P3-m}) (\text{L4-m}, \text{L3-m}, \text{P4-m}; \text{SL}=3) \text{Pkt0}$$

corresponds to:  $\langle \text{P4-m}, \text{L3-m}, \text{L4-m} \rangle$

The behavior of Multicast SID is executed by node N when the DA of the packet received by N is N's Multicast SID. It is a variant of the Endpoint behavior in Section 4.1 of [RFC8986] with the change from S13 - S15 to S13a - S15b below.

```
S13a. Duplicate the packet B times (where B = N-Branches in DA)
S13b. FOR (i = 1 to B) {
S13c.   Set SL of the i-th duplicated packet to N-SIDs in the i-th SID
S14a.   Set IPv6 DA of the i-th duplicated packet to the i-th SID
S15a.   Submit the i-th duplicated packet to the egress IPv6 FIB
        lookup for transmission to the new destination
s15b. }
```

This change duplicates the packet for each of B branches or sub-trees from N, sends the duplicated packet to the next hop node along the branch through setting the DA of the duplicated packet to the multicast SID of the next hop node, SL in SRH to the N-SIDs in DA to pop SIDs and have the SIDs sequence encoding the sub-trees from the next hop at the top of the segment list in SRH, and submitting the duplicated packet to the egress IPv6 FIB lookup for transmission to the new destination DA (i.e., the next hop).

#### 4.3. Procedure/Behavior on Egress Node

When an egress node of a SR P2MP path receives a packet transported by the P2MP path, the DA of the packet is the Multicast SID of the egress node and SL = 0. The egress node proceeds to process the next header in the packet (refer to S03 in Section 4.1 of [RFC8986]).

#### 5. Stateless SRv6 P2MP Path for Ingress

A controller such as PCE can compute a stateless SRv6 P2MP path and send it to its ingress. For a packet to be transported by the path, the ingress encapsulates the packet with the path and the packet will be delivered to the egresses of the path without any states in the network core.

An example architecture using PCE as a controller is illustrated in Figure 4. There is a connection (i.e., PCE session) between the PCE and (the PCC running on) each of the PEs, which are possible ingress nodes in the network domain. Note that some of connections between the PCE and PEs are not shown in the figure.

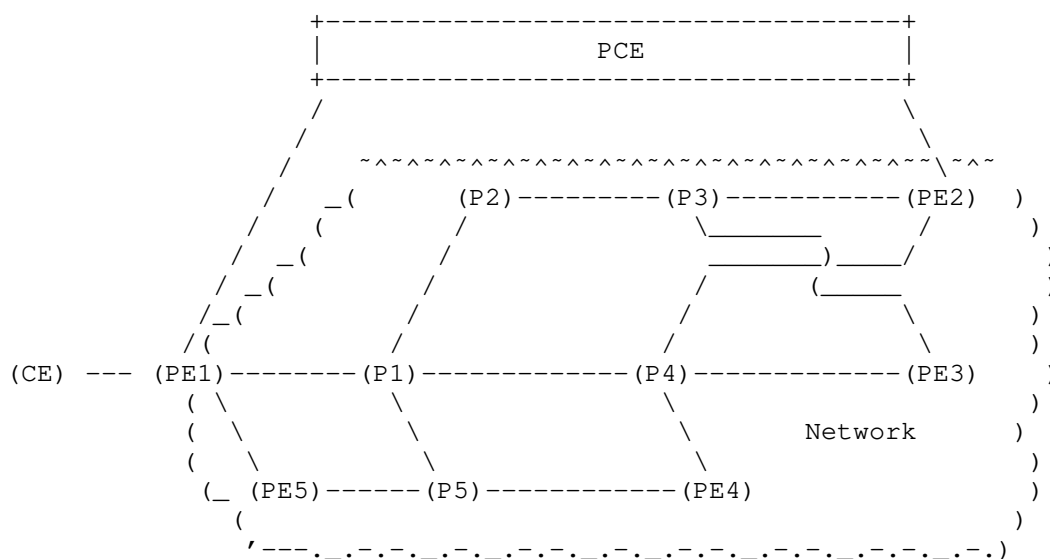


Figure 4: Architecture using PCE

The PCE has the information about the network domain from the IGP or BGP (BGP-LS). The information includes link bandwidth, link colors, node SIDs, and so on. A separate multicast SID could be provisioned on every replication node and the PCE gets the SID on the node from IGP or BGP.

The PCE maintains the current status of the network resource usage in its local TED (Traffic Engineering Database), and the status of every stateless SRv6 P2MP path in its local LSP-DB (Label Switch Path Database).

Upon receiving a request for a stateless SRv6 P2MP path from a user or application, the PCE computes a path based on the network resource availability stored in the TED. After a path satisfying the given constraints is found, the PCE constructs a stateless SRv6 P2MP path using the multicast SIDs of the nodes on the path and encodes the structure of the P2MP path/tree into the parameters of the SIDs. In fact, the stateless SRv6 P2MP path is a segment list consisting of multicast SIDs with parameter values.

And then the PCE sends the segment list representing the path to the ingress node of the path in a PCEP message such as PCInitiate. After receiving the path from the PCE, the ingress node establishes the path by creating a forwarding entry in its FIB. For every multicast packet to be transported by the path, the forwarding entry encapsulates the packet with the segment list and the packet will be

delivered to the egress nodes of the path along the path without any state in the core of the network.

## 6. Protection

Protections for a SR P2MP path can be classified into two types: global protection and local protection.

### 6.1. Global Protection

For a primary SR P2MP path from an ingress node R1 to multiple egress nodes Li (i = 1, ..., n), a backup SR P2MP path from an ingress node R1' to multiple egress nodes Li' (i = 1, ..., n) is set up to provide global protection for the primary SR P2MP path. If R1' is the same as R1, the failure of the ingress node R1 of the primary SR P2MP path is not protected; otherwise (i.e., R1' and R1 are different and connected to the same traffic source), the failure of the ingress node R1 is protected. If Li' is the same as Li (i = 1, ..., n), the failure of the egress nodes Li (i = 1, ..., n) of the primary SR P2MP path is not protected; otherwise (i.e., Li' and Li are different and connected to the same destination), the failure of the egress nodes Li is protected.

When a failure happens on the primary SR P2MP path and is detected by the source of the traffic or other entity, the traffic to be transported by the primary SR P2MP path is switched to the backup SR P2MP path, which sends the traffic from its ingress node R1' to its egress nodes Li' (i = 1, ..., n).

### 6.2. Local Protection

Local protection or say Fast Reroute (FRR) of a SR P2P path is proposed in [I-D.ietf-rtgwg-segment-routing-ti-lfa] and [I-D.ietf-rtgwg-srv6-egress-protection]. It can be applied to FRR of a SR P2MP path in a similar way. But FRR for SR P2MP path is more complicated.

More details will be added later.

## 7. IANA Considerations

TBD

## 8. Security Considerations

TBD

## 9. Acknowledgements

The authors would like to thank Acee Lindem, Jeffrey Zhang, Rishabh Parekh, Arvind Venkateswaran and Daniel Voyer for their valuable comments and suggestions on this draft.

## 10. References

### 10.1. Normative References

- [I-D.ietf-6man-segment-routing-header]  
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa]  
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-08 (work in progress), January 2022.
- [I-D.ietf-rtgwg-srv6-egress-protection]  
Hu, Z., Chen, H., Chen, H., Wu, P., Toy, M., Cao, C., Liu, L., and X. Liu, "SRv6 Path Egress Protection", draft-ietf-rtgwg-srv6-egress-protection-05 (work in progress), April 2022.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

- [RFC8754] Filtsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8986] Filtsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

## 10.2. Informative References

- [I-D.ietf-pim-sr-p2mp-policy]  
(editor), D. V., Filtsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-04 (work in progress), March 2022.
- [I-D.ietf-spring-sr-replication-segment]  
(editor), D. V., Filtsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-07 (work in progress), March 2022.
- [I-D.shen-spring-p2mp-transport-chain]  
Shen, Y., Zhang, Z., Parekh, R., Bidgoli, H., and Y. Kamite, "Point-to-Multipoint Transport Using Chain Replication in Segment Routing", draft-shen-spring-p2mp-transport-chain-04 (work in progress), June 2021.

## Appendix A. Example IPv6 Header using G-SRv6

For simplicity, 64 bits for Common Prefix, 16 bits for Node ID, 8 bits for the number of branches (N-Branches) and 8 bits for the number of SIDs (N-SIDs) are used when G-SRv6 compression method is applied for <P1-m, P2-m, P3-m, L1-m, L2-m, P4-m, L3-m, L4-m> at ingress node R in Figure 1. The Destination Address (DA) is illustrated below in Figure 5. It contains the Common Prefix of 64 bits, node P1's ID of 16 bits, the value 2 for the number of branches (N-Branches) of 8 bits, and the value 7 for the number of SIDs (N-SIDs) of 8 bits.

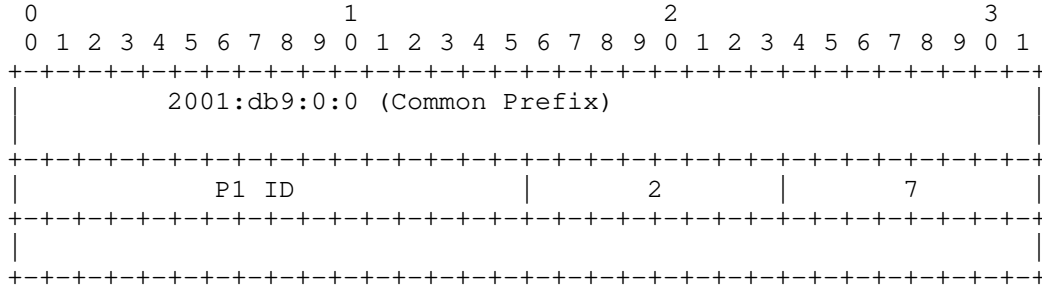


Figure 5: Destination Address (DA)

The IPv6 header is shown in Figure 6. Ingress node R sends a packet with the IPv6 header to the DA.

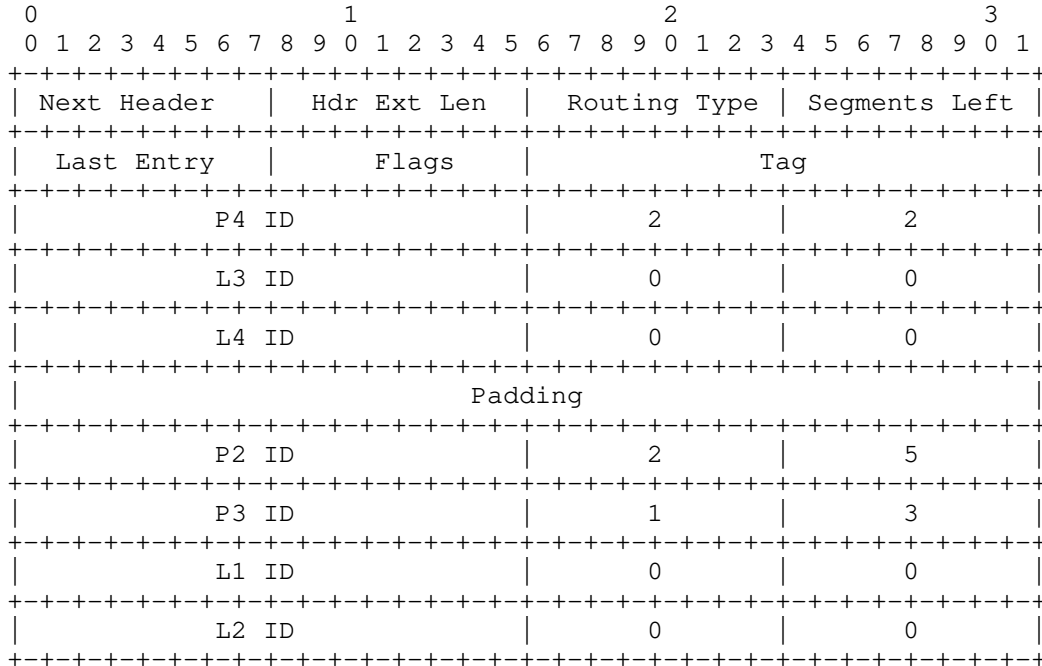


Figure 6: IPv6 Header

Authors' Addresses



Huaimo Chen  
Futurewei  
Boston, MA  
USA

Email: [Huaimo.chen@futurewei.com](mailto:Huaimo.chen@futurewei.com)

Mike McBride  
Futurewei

Email: [michael.mcbride@futurewei.com](mailto:michael.mcbride@futurewei.com)

Yanhe Fan  
Casa Systems  
USA

Email: [yfan@casa-systems.com](mailto:yfan@casa-systems.com)

Zhenbin Li  
Huawei

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Xuesong Geng  
Huawei

Email: [gengxuesong@huawei.com](mailto:gengxuesong@huawei.com)

Mehmet Toy  
Verizon  
USA

Email: [mehmet.toy@verizon.com](mailto:mehmet.toy@verizon.com)

Gyan S. Mishra  
Verizon  
13101 Columbia Pike  
Silver Spring MD 20904  
USA

Phone: 301 502-1347  
Email: [gyan.s.mishra@verizon.com](mailto:gyan.s.mishra@verizon.com)

Aijun Wang  
China Telecom  
Beiqijia Town, Changping District  
Beijing 102209  
China

Email: wangaj3@chinatelecom.cn

Lei Liu  
Fujitsu  
USA

Email: liulei.kddi@gmail.com

Xufeng Liu  
Volta Networks  
McLean, VA  
USA

Email: xufeng.liu.ietf@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 10 November 2022

M. Sivakumar  
Juniper Networks  
S. Venaas  
Cisco Systems, Inc.  
Z. Zhang  
ZTE Corporation  
H. Asaeda  
NICT  
9 May 2022

Internet Group Management Protocol version 3 (IGMPv3) and Multicast  
Listener Discovery version 2 (MLDv2) Message Extension  
draft-ietf-pim-igmp-mld-extension-07

## Abstract

This document specifies a generic mechanism to extend IGMPv3 and MLDv2 by using a list of TLVs (Type, Length and Value).

## 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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 10 November 2022.

## Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions used in this document . . . . .	3
3. Extension Format . . . . .	3
3.1. Multicast Listener Query Extension . . . . .	4
3.2. Version 2 Multicast Listener Report Extension . . . . .	5
3.3. IGMP Membership Query Extension . . . . .	6
3.4. IGMP Version 3 Membership Report Extension . . . . .	7
4. No-op TLV . . . . .	8
5. Processing the extension . . . . .	9
6. Applicability and backwards compatibility . . . . .	10
7. Security Considerations . . . . .	10
8. IANA Considerations . . . . .	11
9. Acknowledgements . . . . .	11
10. References . . . . .	11
10.1. Normative References . . . . .	11
10.2. Informative References . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

This document defines a generic method to extend IGMPv3 [RFC3376] and MLDv2 [RFC3810] messages to accommodate information other than what is contained in the current message formats. This is done by allowing a list of TLVs (Type, Length and Value) to be used in the Additional Data section of IGMPv3 and MLDv2 messages. This document defines a registry for such TLVs, while other documents will define the specific types and their values, and their semantics. The extension would only be used when at least one TLV is to be added to the message. This extension also applies to the lightweight versions of IGMPv3 and MLDv2 as defined in [RFC5790].

When this extension mechanism is used, it replaces the Additional Data section defined in IGMPv3/MLDv2 for TLVs.

Additional Data is defined for Query messages in IGMPv3 [RFC3376] Section 4.1.10 and MLDv2 [RFC3810] Section 5.1.12, and for Report messages in IGMPv3 [RFC3376] Section 4.2.11 and MLDv2 [RFC3810] Section 5.2.11.

## 2. Conventions used in this document

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

## 3. Extension Format

For each of the IGMPv3 and MLDv2 headers, a previously reserved bit is used to indicate the presence of this extension. When this extension is used, the Additional Data of IGMPv3 and MLDv2 messages is formatted as follows. Note that this format contains a variable number of TLVs. It MUST contain at least one TLV.

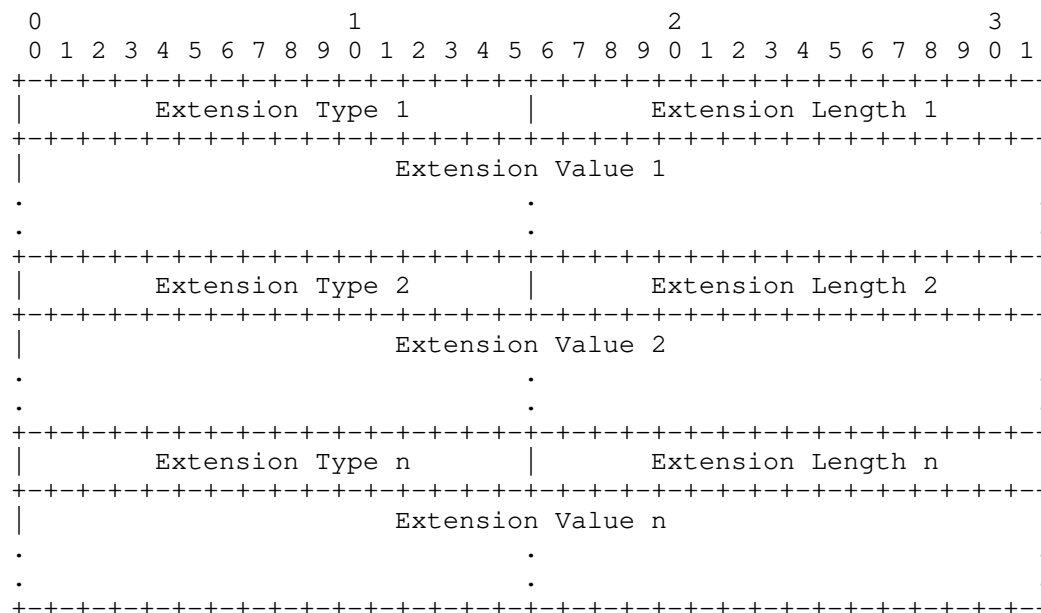


Figure 1: Figure 1: Extension Format

Extension Type: 2 octets. This identifies a particular Extension Type as defined in the IGMP/MLD Extension Type Registry. If this is not the first TLV, it will follow immediately after the end of the previous one. There is no alignment or padding.

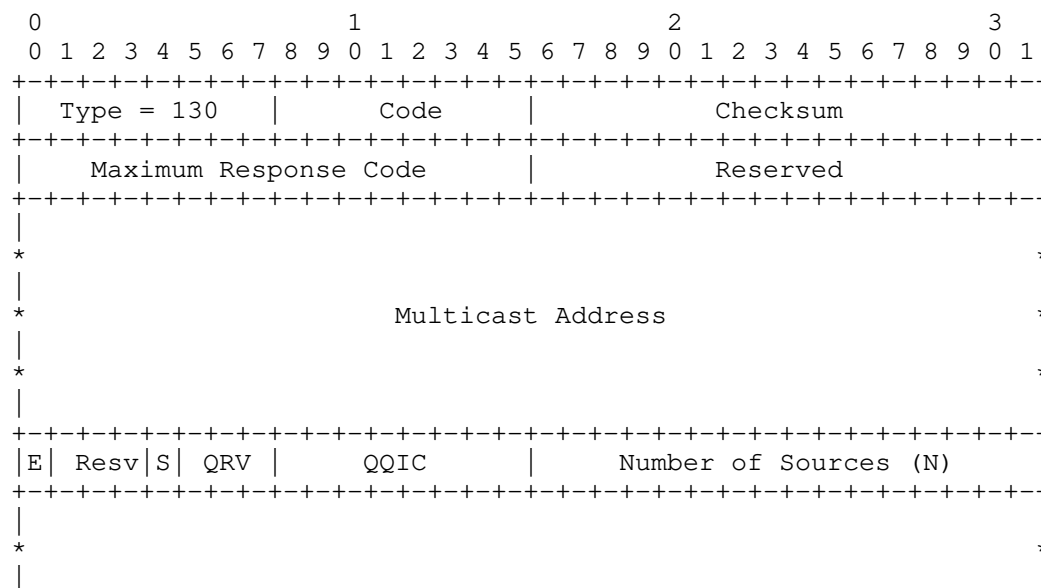
Extension Length: 2 octets. This specifies the length in octets of the following Extension Value field. The length may be zero if no value is needed.

Extension Value: This field contains the value. The length and the contents of this field is according to the specification of the Extension Type.

IGMPv3 and MLDv2 messages are defined so that they can fit within the network MTU, in order to avoid fragmentation. An IGMPv3/MLDv2 report message contains a number of records. The records are called Group Records for IGMPv3, and Address Records for MLDv2. When this extension mechanism is used, the number of records in each Report message SHOULD be kept small enough that the entire message, including any extension TLVs can fit within the network MTU.

### 3.1. Multicast Listener Query Extension

The MLDv2 Query Message format [RFC3810] with extension is shown below. The E-bit MUST be set to 1 to indicate that the extension is present. Otherwise, it MUST be 0.



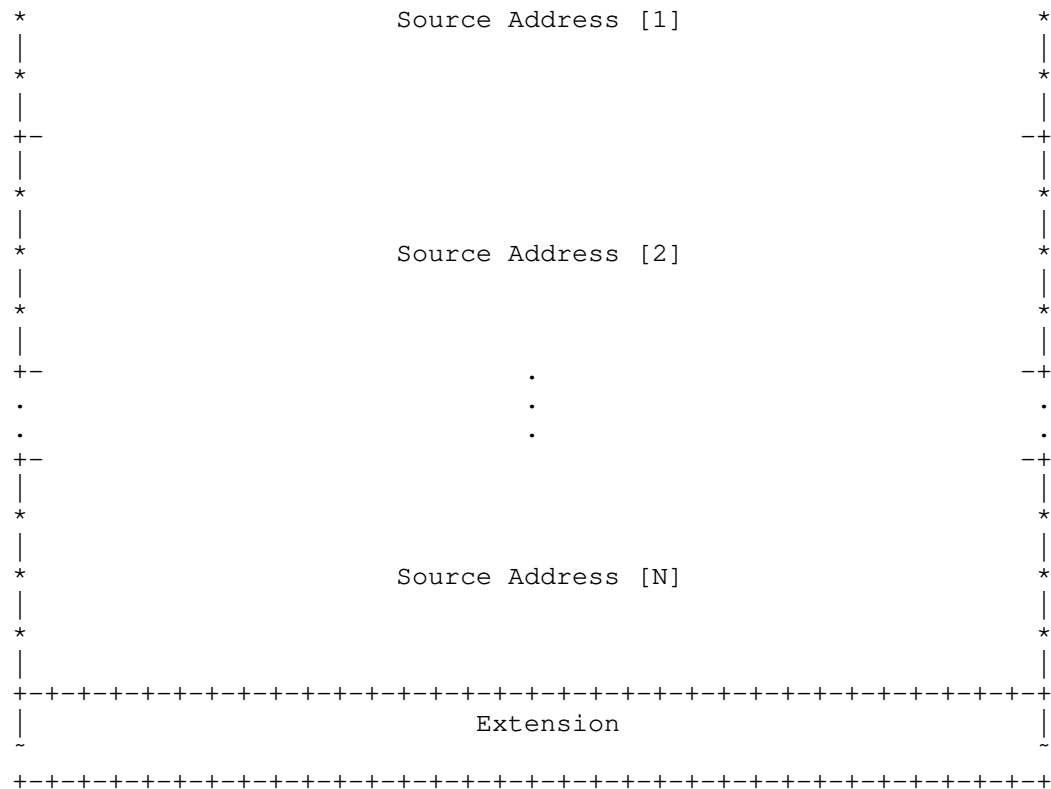


Figure 2: Figure 2: MLD Query Extension

### 3.2. Version 2 Multicast Listener Report Extension

The MLDv2 Report Message format [RFC3810] with extension is shown below. The E-bit MUST be set to 1 to indicate that the extension is present. Otherwise, it MUST be 0.

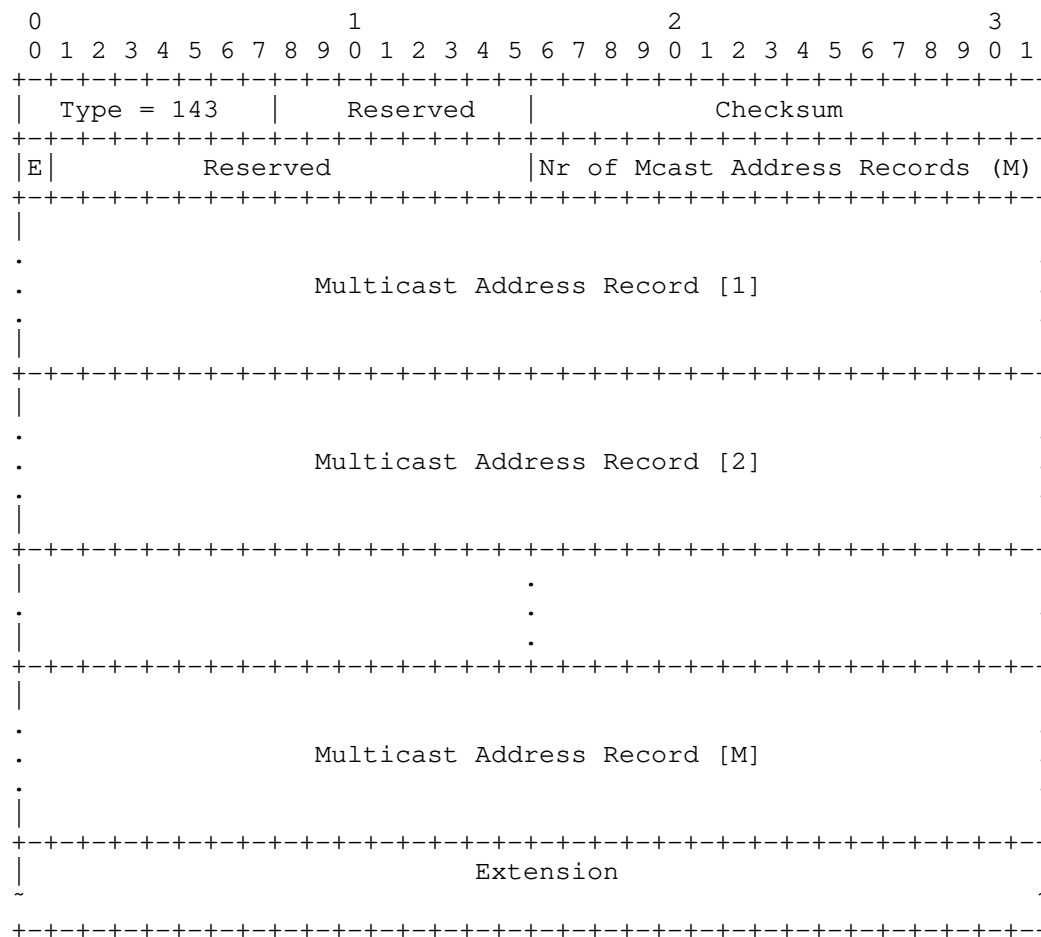


Figure 3: Figure 3: MLD Report Extension

### 3.3. IGMP Membership Query Extension

The IGMPv3 Query Message format [RFC3376] with the extension is shown below. The E-bit MUST be set to 1 to indicate that the extension is present. Otherwise, it MUST be 0.



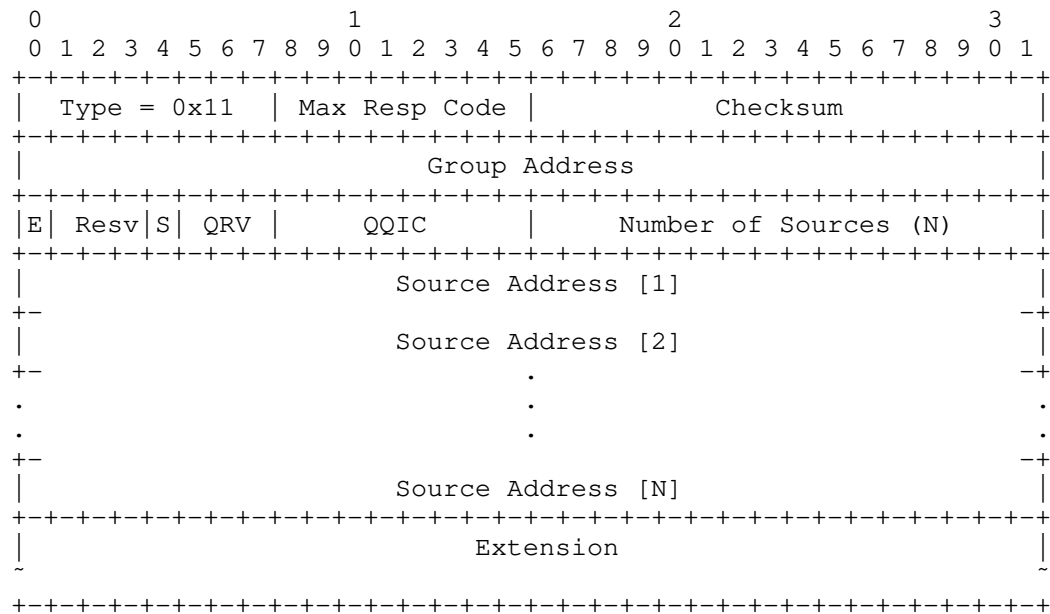


Figure 4: Figure 4: IGMP Query Extension

### 3.4. IGMP Version 3 Membership Report Extension

The IGMPv3 Report Message format [RFC3376] with the extension is shown below. The E-bit MUST be set to 1 to indicate that the extension is present. Otherwise, it MUST be 0.

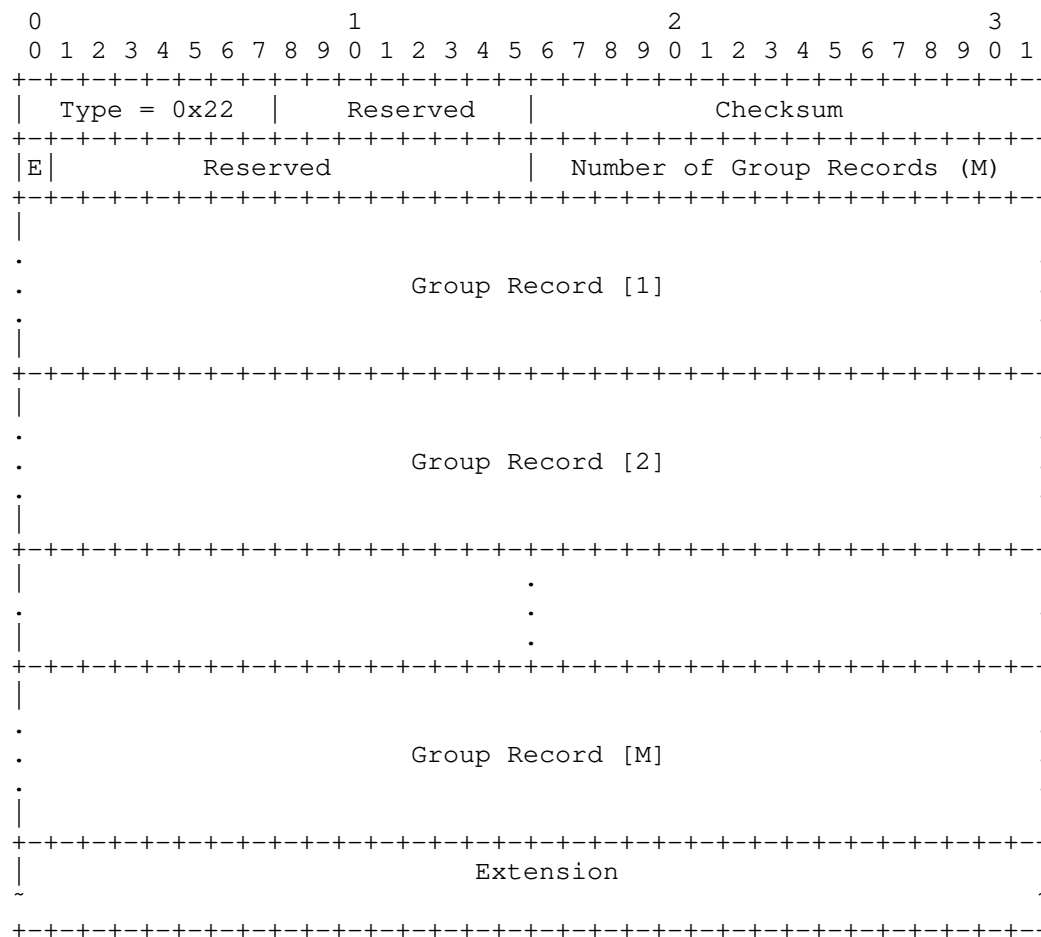


Figure 5: Figure 5: IGMP Report Extension

#### 4. No-op TLV

The no-op TLV is a No-Operation TLV that **MUST** be ignored during processing. This TLV may be useful for verifying that implementations correctly implement this extension mechanism. Note that there is no alignment requirement, so there is no need to use this type to provide alignment.

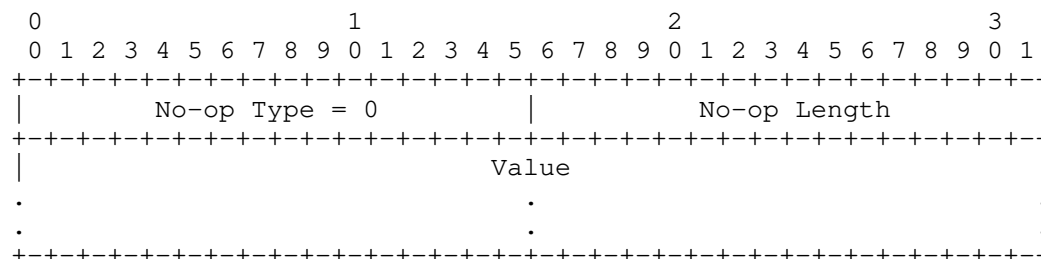


Figure 6: Figure 6: No-op TLV Format

**No-op Type:** 2 octets. The type of the No-op TLV extension is the value 0.

**Extension Length:** 2 octets. This specifies the length in octets of the following Value field. The length may be zero if no value is needed.

**Value:** This field contains the value. As this type is always ignored, the value can be arbitrary data. The number of octets used **MUST** match the specified length. contents of this field is according to the specification of the Extension Type.

## 5. Processing the extension

The procedure specified in this document applies only when the E-bit is set.

If the validation of the TLVs fails, the entire Additional Data field **MUST** be ignored as specified in IGMPv3 [RFC3376] and MLDv2 [RFC3810]. The following checks must pass for the validation of the TLVs not to fail:

At least one TLV **MUST** be present.

There **MUST NOT** be any data in the IP payload after the last TLV. To check this, the parser needs to walk through each of the TLVs until there are less than four octets left in the IP payload. If there are any octets left, validation fails.

The total length of the Extension **MUST NOT** exceed the remainder of the IP payload length. For this validation, one only examines the content of the Extension Length fields.

Future documents defining a new type **MUST** specify any additional processing and validation. These rules, if any, will be examined only after the general validation (above) succeeds.

TLVs with unsupported types MUST be ignored.

## 6. Applicability and backwards compatibility

IGMP and MLD implementations, particularly implementations on hosts, rarely change, and the adoption process of this extension mechanism is expected to be slow. Also, as new extension TLVs are defined, it may take a long time before they are supported. Due to this, defining new extension TLVs should not be taken lightly, and it is crucial to consider backwards compatibility.

Implementations that do not support this extension mechanism will ignore it, as specified in [RFC3376] and [RFC3810]. Also, as mentioned in the previous section, unsupported extension TLVs are ignored.

It is possible that a new extension TLV only applies to queries, or only to reports, or there may be other specific conditions for when it is to be used. A document defining a new type MUST specify under what conditions the new type should be used, including for which message types. It MUST also be specified what the behavior should be if a message is not used in the defined manner, e.g., if it is present in a query message, when it was only expected to be used in reports.

When defining new types, care should be taken to consider the effect of partial support for the new TLV, by either the hosts or routers, on the same link. Further, it must be considered whether there are any dependencies or restrictions on combinations between the new types and any pre-existing types.

This document defines an extension mechanism only for IGMPv3 and MLDv2. Hence, this mechanism does not apply if hosts or routers send older version messages.

## 7. Security Considerations

The Security Considerations of [RFC3376] and [RFC3810] also apply here.

This document extends the IGMP and MLD message formats, allowing for a variable number of TLVs. Implementations must take care when parsing the TLVs to not exceed the packet boundary, an attacker could intentionally specify a TLV with a length exceeding the boundary.

An implementation could add a large number of minimal TLVs in a message to increase the cost of processing the message to magnify a Denial of Service attack.

## 8. IANA Considerations

IANA is asked to create a new registry called "IGMP/MLD Extension Types" in the "Internet Group Management Protocol (IGMP) Type Numbers" section, with registration procedure "IETF Review" [RFC8126], and with this document as a reference. The registry is common for IGMP and MLD. Two extension types are provided for "Experimental Use" [RFC8126]. These types can be used by experiments without any need for an assignment. Any experiments should be confined to closed environments where it is unlikely that it may conflict with other experiments. An experimental type might be used in public deployments if the experimental use is not enabled by default. The default should be to ignore all experimental types. The initial content of the registry should be as below.

Type	Length	Name	Reference
0	variable	No-op	[this document]
1-65533		Unassigned	
65534	variable	Experimental use	
65535	variable	Experimental use	

## 9. Acknowledgements

The authors thank Ron Bonica, Ian Duncan, Wesley Eddy, Leonard Giuliano, Jake Holland, Tommy Pauly, Pete Resnick, Alvaro Retana and Zhaohui Zhang for reviewing the document and providing valuable feedback.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 10.2. Informative References

- [RFC5790] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, DOI 10.17487/RFC5790, February 2010, <<https://www.rfc-editor.org/info/rfc5790>>.

## Authors' Addresses

Mahesh Sivakumar  
Juniper Networks  
64 Butler St  
Milpitas, CA 95035  
United States of America  
Email: [sivakumar.mahesh@gmail.com](mailto:sivakumar.mahesh@gmail.com)

Stig Venaas  
Cisco Systems, Inc.  
Tasman Drive  
San Jose, CA 95134  
United States of America  
Email: [stig@cisco.com](mailto:stig@cisco.com)

Zheng(Sandy) Zhang  
ZTE Corporation  
No. 50 Software Ave, Yuhuatai District  
Nanjing  
210000  
China  
Email: [zhang.zheng@zte.com.cn](mailto:zhang.zheng@zte.com.cn)

Hitoshi Asaeda  
National Institute of Information and Communications Technology  
4-2-1 Nukui-Kitamachi,  
184-8795  
Japan  
Email: asaeda@nict.go.jp

PIM Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: February 28, 2022

H. Zhao  
Ericsson  
X. Liu  
Volta  
Y. Liu  
China Mobile  
M. Panchanathan  
Cisco  
M. Sivakumar  
Juniper

August 30, 2021

A Yang Data Model for IGMP/MLD Proxy  
draft-ietf-pim-igmp-mld-proxy-yang-06.txt

## Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

## 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/lid-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 February 28, 2022.

## Copyright Notice

Copyright (c) 2021 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
1.1. Terminology.....	3
1.2. Conventions Used in This Document.....	3
1.3. Tree Diagrams.....	4
1.4. Prefixes in Data Node Names.....	4
2. Design of Data Model.....	4
2.1. Overview.....	5
2.2. Optional Capabilities.....	5
2.3. Position of Address Family in Hierarchy.....	5
3. Module Structure.....	6
3.1. IGMP Proxy Configuration and Operational State.....	6
3.2. MLD Proxy Configuration and Operational State.....	7
4. IGMP/MLD Proxy YANG Module.....	8
5. Security Considerations.....	15
6. IANA Considerations.....	16
6.1. XML Registry.....	16
6.2. YANG Module Names Registry.....	16
7. References.....	17
7.1. Normative References.....	17
7.2. Informative References.....	18
Appendix. Data Tree Example.....	19
Authors' Addresses.....	22

## 1. Introduction

This document defines a YANG [RFC7950] data model for the management of Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) Proxy [RFC4605] devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

### 1.1. Terminology

The terminology for describing YANG data models is found in [RFC6020] and [RFC7950], including:

- \* augment
- \* data model
- \* data node
- \* identity
- \* module

The following abbreviations are used in this document and defined model:

IGMP: Internet Group Management Protocol [RFC3376].

MLD: Multicast Listener Discovery [RFC3810].

PIM: Protocol Independent Multicast [RFC7761].

### 1.2. Conventions Used in This Document

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

### 1.3. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

### 1.4. Prefixes in Data Node Names

In this document, names of data nodes, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
rt-types	ietf-routing-types	[RFC8294]
pim-base	ietf-pim-base	[draft-ietf-pim-yang]

Table 1: Prefixes and Corresponding YANG Modules

## 2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying") [RFC4605].

The goal of this document is to define a data model that provides a common user interface to IGMP/MLD Proxy. This document provides freedom for vendors to adapt the data model to their product implementations.

## 2.1. Overview

The model defined in this document has all the common building blocks for the IGMP/MLD Proxy devices. It can be used to configure IGMP/MLD Proxy. The operational state data and statistics can also be retrieved by it.

The model contains all the basic configuration parameters. The occasionally implemented parameters are modeled as optional features in this model, while the rarely implemented parameters are not included in this model and left for augmentation.

## 2.2. Optional Capabilities

This model is designed to represent the basic capability subsets of IGMP / MLD Proxy. The main design goals of this document are that the basic capabilities described in the model are supported by any major now-existing implementation, and that the configuration of all implementations meeting the specifications is easy to express through some combination of the optional features in the model and simple vendor augmentations.

There is also value in widely supported features being standardized, to provide a standardized way to access these features, to save work for individual vendors, and so that mapping between different vendors' configuration is not needlessly complicated. Therefore, this model declares a number of features representing capabilities that not all deployed devices support.

The extensive use of feature declarations should also substantially simplify the capability negotiation process for a vendor's IGMP / MLD Proxy implementations.

## 2.3. Position of Address Family in Hierarchy

IGMP Proxy only supports IPv4, while MLD Proxy only supports IPv6. The data model defined in this document can be used for both IPv4 and IPv6 address families.

This document defines IGMP Proxy and MLD Proxy as separate schema branches in the structure. The benefits are:

- \* The model can support IGMP Proxy (IPv4), MLD Proxy (IPv6), or both optionally and independently. Such flexibility cannot be achieved cleanly with a combined branch.

\* The structure is consistent with other YANG data models such as [RFC8652], which uses separate branches for IPv4 and IPv6.

\* Having separate branches for IGMP Proxy and MLD Proxy allows minor differences in their behavior to be modelled more simply and cleanly. The two branches can better support different features and node types.

### 3. Module Structure

This model augments the core routing data model specified in [RFC8349].

```
+--rw routing
  +--rw router-id?
  +--rw control-plane-protocols
    |   +--rw control-plane-protocol* [type name]
    |   |   +--rw type
    |   |   +--rw name
    |   |   +--rw igmp-proxy <= Augmented by this Model
    |   |   ...
    |   |   +--rw mld-proxy <= Augmented by this Model
```

The "igmp-proxy" container instantiates IGMP Proxy. The "mld-proxy" container instantiates MLD Proxy.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407].

#### 3.1. IGMP Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the igmp-proxy container.

All the IGMP Proxy related attributes are defined in the igmp-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The igmp-version represents version of IGMP protocol, and default value is 2. If the value of enable is true, it means IGMP Proxy is enabled.

The interface list under igmp-proxy contains upstream interfaces for IGMP proxy. There is also a constraint to make sure the upstream interface for IGMP proxy should not be configured PIM.

To configure a downstream interface for IGMP proxy, it is needed to enable IGMP on that interface. This is defined in the YANG Data Model

for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) [RFC8652].

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw igmp-proxy {igmp-proxy}?
      +--rw interfaces
        +--rw interface* [interface-name]
          +--rw interface-name          if:interface-ref
          +--rw igmp-version?            uint8
          +--rw enable?                  boolean
          +--rw sender-source-address?   inet:ipv4-address
          +--ro group* [group-address]
            +--ro group-address
              | rt-types:ipv4-multicast-group-address
            +--ro up-time?                uint32
            +--ro filter-mode             enumeration
            +--ro source* [source-address]
              +--ro source-address        inet:ipv4-address
              +--ro up-time?              uint32
              +--ro downstream-interface* [interface-name]
                +--ro interface-name     if:interface-ref

```

### 3.2. MLD Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the mld-proxy container.

All the MLD Proxy related attributes are defined in the mld-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The mld-version represents version of MLD protocol, and default value is 2. If the value of enable is true, it means MLD Proxy is enabled.

The interface list under mld-proxy contains upstream interfaces for MLD proxy. There is also a constraint to make sure the upstream interface for MLD proxy should not be configured PIM.

To configure a downstream interface for MLD proxy, enable MLD on that interface. This is defined in the YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) [RFC8652].

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw mld-proxy {mld-proxy}?
      +--rw interfaces
        +--rw interface* [interface-name]

```

```

+--rw interface-name          if:interface-ref
+--rw mld-version?            uint8
+--rw enable?                 boolean
+--rw sender-source-address?  inet:ipv6-address
+--ro group* [group-address]
  +--ro group-address
    |   rt-types:ipv6-multicast-group-address
  +--ro up-time?              uint32
  +--ro filter-mode           enumeration
  +--ro source* [source-address]
    +--ro source-address      inet:ipv6-address
    +--ro up-time?            uint32
    +--ro downstream-interface* [interface-name]
      +--ro interface-name    if:interface-ref

```

#### 4. IGMP/MLD Proxy YANG Module

This module references [RFC4605], [RFC6991], [RFC8294], [RFC8343], [RFC8349] and [draft-ietf-pim-yang].

```

<CODE BEGINS> file ietf-igmp-mld-proxy@2021-04-21.yang
module ietf-igmp-mld-proxy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy";
  // replace with IANA namespace when assigned
  prefix igmp-mld-proxy;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ietf-routing {
    prefix rt;
  }
  import ietf-routing-types {
    prefix rt-types;
  }
  import ietf-pim-base {
    prefix pim-base;
  }

  organization
    "IETF PIM Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/pim/>
    WG List:  <mailto:pim@ietf.org>

```

Editors: Hongji Zhao  
<mailto:hongji.zhao@ericsson.com>  
  
Xufeng Liu  
<mailto:xufeng.liu.ietf@gmail.com>  
  
Yisong Liu  
<mailto:liuyisong@chinamobile.com>  
  
Mani Panchanathan  
<mailto:mapancha@cisco.com>  
  
Mahesh Sivakumar  
<mailto:sivakumar.mahesh@gmail.com>

";

description

"The module defines a collection of YANG definitions common for all Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxy devices.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2021-04-21 {  
  description  
    "Initial revision.";  
  reference  
    "RFC XXXX: A YANG Data Model for IGMP and MLD Proxy";  
}
```

```
/*  
 * Features  
 */
```

```
feature igmp-proxy {  
  description  
    "Support IGMP Proxy protocol.";  
  reference  
    "RFC 4605";
```



```
}

feature mld-proxy {
  description
    "Support MLD Proxy protocol.";
  reference
    "RFC 4605";
}

/*
 * Identities
 */

identity igmp-proxy {
  base rt:control-plane-protocol;
  description
    "IGMP Proxy protocol";
}

identity mld-proxy {
  base rt:control-plane-protocol;
  description
    "MLD Proxy protocol";
}

/*
 * Groupings
 */

grouping per-interface-config-attributes {
  description "Config attributes under interface view";
  leaf enable {
    type boolean;
    default false;
    description
      "Set the value to true to enable IGMP/MLD proxy";
  }
} // per-interface-config-attributes

grouping state-group-attributes {
  description
    "State group attributes";
  leaf up-time {
    type uint32;
    units seconds;
    description
      "The elapsed time for (S,G) or (*,G).";
  }
  leaf filter-mode {
    type enumeration {
      enum "include" {
```

```
        description
            "In include mode, reception of packets sent
            to the specified multicast address is requested
            only from those IP source addresses listed in the
            source-list parameter";
    }
    enum "exclude" {
        description
            "In exclude mode, reception of packets sent
            to the given multicast address is requested
            from all IP source addresses except those
            listed in the source-list parameter.";
    }
}
mandatory true;
description
    "Filter mode for a multicast group,
    may be either include or exclude.";
}
} // state-group-attributes

/* augments */

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when
        "derived-from-or-self(rt:type, 'igmp-mld-proxy:igmp-proxy')" {
            description
                "This augmentation is only valid for IGMP Proxy.";
        }
    description
        "IGMP Proxy augmentation to routing control plane protocol
        configuration and state.";
    container igmp-proxy {
        if-feature "igmp-proxy";
        description "IGMP proxy";
        container interfaces {
            description
                "Containing a list of upstream interfaces.";
            list interface {
                key "interface-name";
                description
                    "List of upstream interfaces.";
                leaf interface-name {
                    type if:interface-ref;
                    must "not( current() = /rt:routing"+
                        "/rt:control-plane-protocols/pim-base:pim"+
                        "/pim-base:interfaces/pim-base:interface"+
                        "/pim-base:name )" {
                        description
```

```
        "The upstream interface for IGMP proxy
        should not be configured PIM.";
    }
    description "The upstream interface name.";
}
leaf igmp-version {
    type uint8 {
        range "1..3";
    }
    default 2;
    description "IGMP version.";
}
uses per-interface-config-attributes;
leaf sender-source-address {
    type inet:ipv4-address;
    description
        "The sender source address of
        IGMP membership report or leave.";
}
list group {
    key "group-address";
    config false;
    description
        "Multicast group membership information
        that joined on the interface.";
    leaf group-address {
        type rt-types:ipv4-multicast-group-address;
        description
            "Multicast group address.";
    }
}
uses state-group-attributes;
list source {
    key "source-address";
    description
        "List of multicast source information
        of the multicast group.";
    leaf source-address {
        type inet:ipv4-address;
        description
            "Multicast source address";
    }
}
leaf up-time {
    type uint32;
    units seconds;
    description
        "The elapsed time for (S,G) or (*,G).";
}
list downstream-interface {
    key "interface-name";
    description "The downstream interfaces list.";
    leaf interface-name {
```

```

        type if:interface-ref;
        description
            "Downstream interfaces
             for each upstream-interface";
    }
    }
    } // list source
    } // list group
    } // interface
    } // interfaces
}
}

augment "/rt:routing/rt:control-plane-protocols"+
    "/rt:control-plane-protocol" {
    when
        "derived-from-or-self(rt:type, 'igmp-mld-proxy:mld-proxy')" {
        description
            "This augmentation is only valid for MLD Proxy.";
        }
    description
        "MLD Proxy augmentation to routing control plane protocol
         configuration and state.";
    container mld-proxy {
        if-feature "mld-proxy";
        description "MLD proxy";
        container interfaces {
            description
                "Containing a list of upstream interfaces.";
            list interface {
                key "interface-name";
                description
                    "List of upstream interfaces.";
                leaf interface-name {
                    type if:interface-ref;
                    must "not( current() = /rt:routing"+
                        "/rt:control-plane-protocols/pim-base:pim"+
                        "/pim-base:interfaces/pim-base:interface"+
                        "/pim-base:name )" {
                        description
                            "The upstream interface for MLD proxy
                             should not be configured PIM.";
                    }
                }
                description "The upstream interface name.";
            }
            leaf mld-version {
                type uint8 {
                    range "1..2";
                }
                default 2;
                description "MLD version.";
            }
        }
    }
}

```

```
    }
    uses per-interface-config-attributes;
    leaf sender-source-address {
        type inet:ipv6-address;
        description
            "The sender source address of
             MLD membership report or leave.";
    }
    list group {
        key "group-address";
        config false;
        description
            "Multicast group membership information
             that joined on the interface.";
        leaf group-address {
            type rt-types:ipv6-multicast-group-address;
            description
                "Multicast group address.";
        }
    }
    uses state-group-attributes;
    list source {
        key "source-address";
        description
            "List of multicast source information
             of the multicast group.";
        leaf source-address {
            type inet:ipv6-address;
            description
                "Multicast source address";
        }
        leaf up-time {
            type uint32;
            units seconds;
            description
                "The elapsed time for (S,G) or (*,G).";
        }
        list downstream-interface {
            key "interface-name";
            description "The downstream interfaces list.";
            leaf interface-name {
                type if:interface-ref;
                description
                    "Downstream interfaces
                     for each upstream-interface";
            }
        }
    } // list source
} // list group
} // interface
} // interfaces
}
```

```
    }  
  }  
<CODE ENDS>
```

## 5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

igmp-mld-proxy:igmp-proxy

igmp-mld-proxy:mld-proxy

Unauthorized access to any data node of these subtrees can adversely affect the IGMP / MLD Proxy subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

igmp-mld-proxy:igmp-proxy

igmp-mld-proxy:mld-proxy

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP / MLD Proxy on this device. The group/source information may expose multicast group memberships.

## 6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

### 6.1. XML Registry

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

---

URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy  
Registrant Contact: The IETF.  
XML: N/A, the requested URI is an XML namespace.

---

### 6.2. YANG Module Names Registry

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

---

name:	ietf-igmp-mld-proxy
namespace:	urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy
prefix:	igmp-mld-proxy
reference:	RFC XXXX

---

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4605] B. Fenner, H. He, B. Haberman and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] R. Enns, Ed., M. Bjorklund, Ed., J. Schoenwaelder, Ed., A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.
- [RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8174] B. Leiba, "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.
- [RFC8294] X. Liu, Y. Qu, A. Lindem, C. Hopps, L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, December 2017.
- [RFC8340] M. Bjorklund, and L. Berger, Ed., "YANG Tree Diagrams", RFC 8340, March 2018.



- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.
- [RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.
- [RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.
- [RFC8349] L. Lhotka, A. Lindem, Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, March 2018.
- [RFC8407] A. Bierman, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", RFC 8407, October 2018.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.
- [RFC8652] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", RFC 8652, November 2019.
- [draft-ietf-pim-yang] X. Liu, P. McAllister, A. Peter, M. Sivakumar, Y. Liu, F. Hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (RFC Editor state is MISSREF), May 2018.

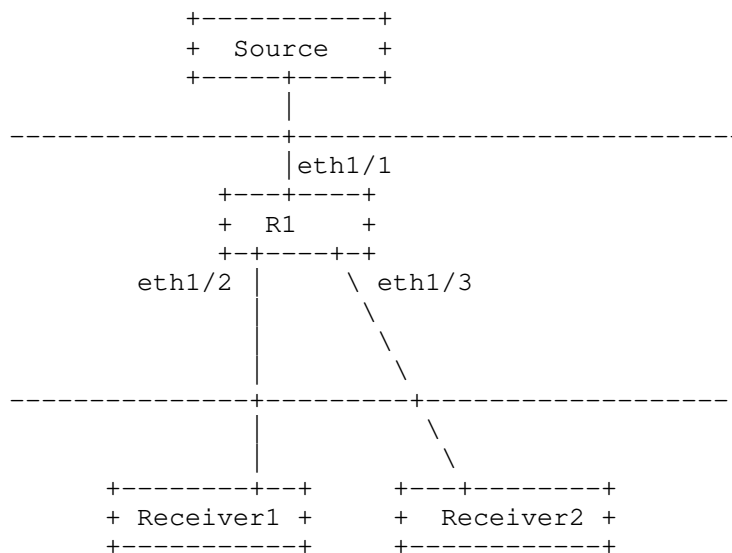
## 7.2. Informative References

- [RFC7761] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 7761, March 2016.
- [RFC7951] L. Lhotka, "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.

## Appendix. Data Tree Example

This section contains an example for IGMP Proxy in the JSON encoding [RFC7951], containing both configuration and state data. In the example IGMP Proxy is enabled on interface eth1/1.

It is also needed to enable IGMP on eth1/2 and eth1/3. The configuration details are omitted here because this document is focused on IGMP/MLD Proxy.



The configuration data for R1 in the above figure could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1/1",
        "type": "iana-if-type:ipForward",
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "11.0.0.1",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  }
}

```

```

    ]
  },
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "ietf-igmp-mld-proxy:igmp-proxy",
          "name": "proxy1",
          "ietf-igmp-mld-proxy:igmp-proxy": {
            "interfaces": {
              "interface": [
                {
                  "interface-name": "eth1/1",
                  "igmp-version": 3,
                  "enable": true
                }
              ]
            }
          }
        }
      ]
    }
  }
}

```

The corresponding operational state data for R1 could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1/1",
        "type": "iana-if-type:ipForward",
        "admin-status": "up",
        "oper-status": "up",
        "if-index": 25678136,
        "statistics": {
          "discontinuity-time": "2021-05-23T10:34:56-06:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "11.0.0.1",
              "prefix-length": 24
            }
          ]
        }
      }
    ]
  }
}

```

```

"ietf-routing:routing": {
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-igmp-mld-proxy:igmp-proxy",
        "name": "proxyl",
        "ietf-igmp-mld-proxy:igmp-proxy": {
          "interfaces": {
            "interface": [
              {
                "interface-name": "eth1/1",
                "igmp-version": 3,
                "enable": true,
                "group": [
                  {
                    "group-address": "225.0.0.1",
                    "filter-mode": "include",
                    "source": [
                      {
                        "source-address": "1.1.1.1",
                        "downstream-interface": [
                          {
                            "interface-name": "eth1/2"
                          },
                          {
                            "interface-name": "eth1/3"
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}

```

Authors' Addresses

Hongji Zhao  
Ericsson (China) Communications Company Ltd.  
Ericsson Tower, No. 5 Lize East Street,  
Chaoyang District Beijing 100102, China  
Email: hongji.zhao@ericsson.com

Xufeng Liu  
Volta Networks  
USA  
EMail: Xufeng.liu.ietf@gmail.com

Yisong Liu  
China Mobile  
China  
Email: liuyisong@chinamobile.com

Mani Panchanathan  
Cisco  
India  
Email: mapancha@cisco.com

Mahesh Sivakumar  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, California  
USA  
EMail: sivakumar.mahesh@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 11 May 2022

V. Kamath  
VMware  
R. Chokkanathapuram Sundaram  
Cisco Systems, Inc.  
R. Banthia  
Apstra  
A. Gopal  
Cisco Systems, Inc.  
7 November 2021

PIM Null-Register packing  
draft-ietf-pim-null-register-packing-11

## Abstract

In PIM-SM networks PIM Null-Register messages are sent by the Designated Router (DR) to the Rendezvous Point (RP) to signal the presence of Multicast sources in the network. There are periodic PIM Null-Registers sent from the DR to the RP to keep the state alive at the RP as long as the source is active. The PIM Null-Register message carries information about a single Multicast source and group.

This document defines a standard to send multiple Multicast source and group information in a single PIM Packed Null-Register message. We will refer to the new packed formats as the PIM Packed Null-Register format and PIM Packed Register-Stop format throughout the document. This document also discusses interoperability between the PIM routers which do not understand the PIM Packed Null-Register format and routers which do understand it.

## 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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 11 May 2022.

## Copyright Notice

Copyright (c) 2021 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 (<https://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 . . . . .	2
1.1. Conventions used in this document . . . . .	3
1.2. Terminology . . . . .	3
2. Packed Null-Register Capability . . . . .	3
3. PIM Packed Null-Register message format . . . . .	4
4. PIM Packed Register-Stop message format . . . . .	5
5. Protocol operation . . . . .	6
6. Operational Considerations . . . . .	7
7. PIM Anycast RP Considerations . . . . .	7
8. PIM RP router version downgrade . . . . .	7
9. Fragmentation Considerations . . . . .	7
10. Security Considerations . . . . .	8
11. IANA Considerations . . . . .	8
12. Acknowledgments . . . . .	8
13. References . . . . .	8
13.1. Normative References . . . . .	8
13.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

PIM Null-Registers are sent by the DR periodically for Multicast streams to keep the states active on the RP, as long as the multicast source is alive. As the number of multicast sources increases, the number of PIM Null-Register messages that are sent also increases. This results in more PIM packet processing at the RP and the DR.

The control plane policing (COPP), monitors the packets that are processed by the control plane. The high rate at which Null-Registers are received at the RP can lead to COPP drops of Multicast PIM Null-Register messages. This draft proposes a method to efficiently pack multiple PIM Null-Registers [RFC7761] (Section 4.4) and Register-Stops [RFC7761] (Section 3.2) into a single message as these packets anyway do not contain encapsulated data.

The draft also discusses interoperability with PIM routers that do not understand the new packet format.

### 1.1. Conventions used in this document

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

### 1.2. Terminology

RP: Rendezvous Point

DR: Designated Router

## 2. Packed Null-Register Capability

A router (DR) can decide to pack multiple Null-Register messages based on the capability received from the RP as part of the PIM Register-Stop. This ensures compatibility with routers that do not support processing of the new format. The capability information can be indicated by the RP via the PIM Register-Stop message sent to the DR. Thus a DR will switch to the new format only when it learns that the RP is capable of handling the PIM Packed Null-Register messages.

Conversely, a DR that does not support the packed format can continue generating the PIM Null-Register as defined in [RFC7761] (Section 4.4). To exchange the capability information in the Register-Stop message, the "Reserved" field can be used to indicate this capability in those Register-Stop messages. One bit of the Reserved field is used to indicate the "packing" capability (P bit). The rest of the bits in the "Reserved" field will be retained for future use.



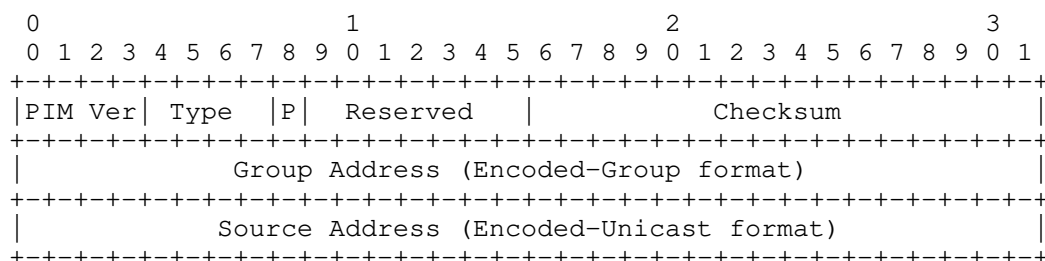


Figure 1: PIM Register-Stop message with capability option

PIM Version, Type, Checksum, Group Address, Source Address:

Same as [RFC7761] (Section 4.9.4)

P:

Capability bit (flag bit 7) used to indicate support for the  
Packed Null-Register Capability

### 3. PIM Packed Null-Register message format

PIM Packed Null-Register message format includes a count to indicate the number of Null-Register records in the message.

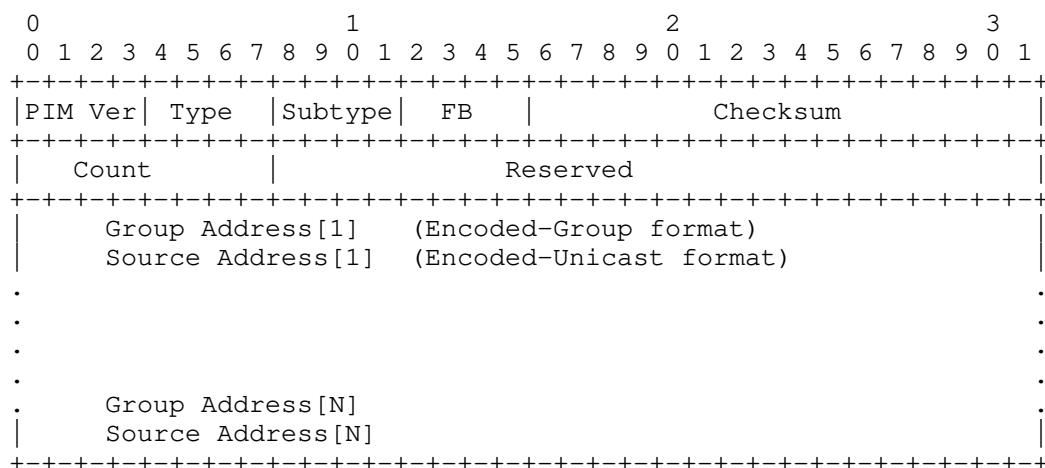


Figure 2: PIM Packed Null-Register message format

PIM Version, Reserved, Checksum:

Same as [RFC7761] (Section 4.9.3)

Type, SubType:

The new packed Null-Register Type and SubType values TBD.  
[RFC8736]

Count:

The number of packed Null-Register records. A record consists of a Group Address and Source Address pair.

Group Address, Source Address:

Same as [RFC7761] (Section 4.9.4)

#### 4. PIM Packed Register-Stop message format

The PIM Packed Register-Stop message includes a count to indicate the number of records that are present in the message.

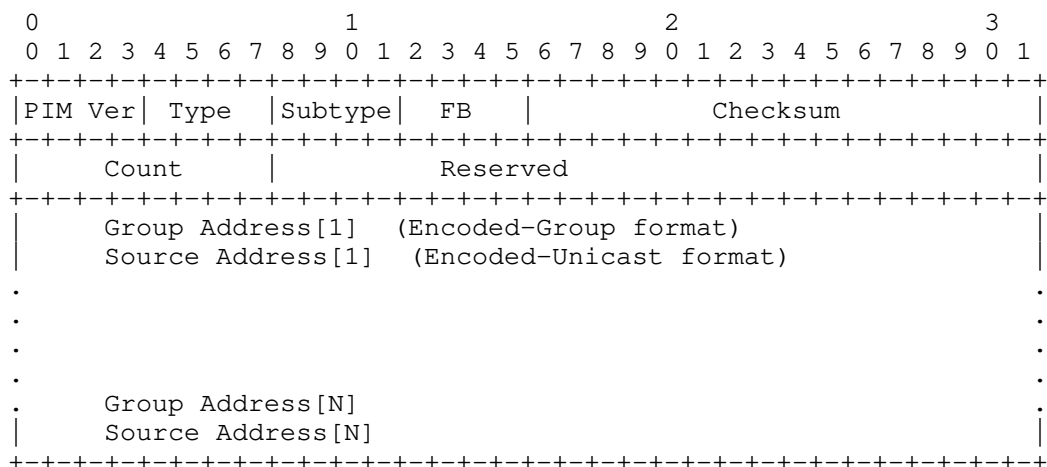


Figure 3: PIM Packed Register-Stop message format

PIM Version, Reserved, Checksum:

Same as [RFC7761] (Section 4.9.4)

Type:

The new Register Stop Type and SubType values TBD

Count:

The number of PIM packed Register-Stop records. A record consists of a Group Address and Source Address pair.

Group Address, Source Address:

Same as [RFC7761] (Section 4.9.4)

## 5. Protocol operation

The following combinations exist -

1. DR and RP both support the PIM Packed Null-Register and PIM Packed Register-Stop formats:
  - \* As specified in [RFC7761], the DR sends PIM Register messages towards the RP when a new source is detected.
  - \* An RP supporting this specification MUST set the P-bit in the corresponding Register-Stop messages.
  - \* When a Register-Stop message with the P-bit set is received, the DR SHOULD send PIM Packed Null-Register messages (Section 3) to the RP instead of multiple Register messages with the N-bit set [RFC7761].
  - \* The RP, after receiving a PIM Packed Null-Register message SHOULD start sending PIM Packed Register-Stop messages (Section 4) to the corresponding DR instead of individual Register-Stop messages.
2. DR supports but RP does not support the PIM Packed Null-Register and PIM Packed Register-Stop formats:
  - \* As specified in [RFC7761], DR sends PIM Null-Registers towards the RP.
  - \* After receiving DR's PIM Null-Register message, RP sends a normal Register-Stop without any capability information.
  - \* DR then sends PIM Null-Registers in the unpacked format [RFC7761].
3. RP supports but DR does not support the PIM Packed Null-Register and PIM Packed Register-Stop formats:

- \* As specified in [RFC7761], DR sends the PIM Null-Register towards the RP.
- \* After receiving DR's PIM Null-Register message, RP sends a PIM Packed Register-Stop towards the DR that includes capability information.
- \* Since DR does not support the new format, it sends PIM Null-Registers in the unpacked format [RFC7761].

## 6. Operational Considerations

In case the network manager disables the packed capability at the RP, the router should not advertise the capability. However, an implementation MAY choose to still parse any packed registers if they are received. This may be particularly useful in the transitional period after the network manager disables it.

## 7. PIM Anycast RP Considerations

The PIM Packed Null-Register format should be enabled only if it is supported by all PIM Anycast RP [RFC4610] members in the RP set for the RP address. This consideration applies to PIM Anycast RP with MSDP [RFC3446] as well.

## 8. PIM RP router version downgrade

Consider a PIM RP router that supports PIM Packed Null-Registers and PIM Packed Register-Stops. When this router downgrades to a software version which does not support PIM Packed Null-Registers and PIM Packed Register-Stops, the DR that sends the PIM Packed Null-Register message will not get a PIM Register-Stop message back from the RP. In such scenarios the DR can send an unpacked PIM Null-Register and check the PIM Register-Stop to see if the capability bit (P-bit) for PIM Packed Null-Register is set or not. If it is not set then the DR will continue sending unpacked PIM Null-Register messages.

## 9. Fragmentation Considerations

When building a PIM Packed Null-Register message or PIM Packed Register-Stop message, a router should include as many records as possible based on the path MTU towards RP, if path MTU discovery is done. Otherwise, the number of records should be limited by the MTU of the outgoing interface.

## 10. Security Considerations

General Register messages security considerations from [RFC7761] apply. As mentioned in [RFC7761], PIM Null-Register messages and Register-Stop messages are forwarded by intermediate routers to their destination using normal IP forwarding. Without data origin authentication, an attacker who is located anywhere in the network may be able to forge a Null-Register or Register-Stop message. We next consider the effect of a forgery of each of these messages. By forging a Register message, an attacker can cause the RP to inject forged traffic onto the shared multicast tree.

By forging a Register-Stop message, an attacker can prevent a legitimate DR from registering packets to the RP. This can prevent local hosts on that LAN from sending multicast packets. The above two PIM messages are not changed by intermediate routers and need only be examined by the intended receiver. Thus, these messages can be authenticated end-to-end. Attacks on Register and Register-Stop messages do not apply to a PIM-SSM-only implementation, as these messages are not used in PIM-SSM.

There is another case where a spoofed Register-Stop can be sent to make it appear that is from the RP, and that the RP supports this new packed capability when it does not. This can cause Null-Registers to be sent to an RP that doesn't support this packed format. But standard methods to prevent spoofing should take care of this case. For example, uRPF can be used to filter out packets coming from the outside from addresses that belong to routers inside.

## 11. IANA Considerations

This document requires the assignment of Capability bit (P-bit), flag bit 7 in the PIM Register-Stop message.

This document requires the assignment of 2 new PIM message types for the PIM Packed Null-Register and PIM Packed Register-Stop.

## 12. Acknowledgments

The authors would like to thank Stig Venaas, Anish Peter, Zheng Zhang and Umesh Dudani for their helpful comments on the draft.

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <<https://www.rfc-editor.org/info/rfc4610>>.
- [RFC8736] Venaas, S. and A. Retana, "PIM Message Type Space Extension and Reserved Bits", RFC 8736, DOI 10.17487/RFC8736, February 2020, <<https://www.rfc-editor.org/info/rfc8736>>.

### 13.2. Informative References

- [RFC3446] Kim, D., Meyer, D., Kilmer, H., and D. Farinacci, "Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)", RFC 3446, DOI 10.17487/RFC3446, January 2003, <<https://www.rfc-editor.org/info/rfc3446>>.

### Authors' Addresses

Vikas Ramesh Kamath  
VMware  
3401 Hillview Ave  
Palo Alto, CA 94304  
United States of America  
  
Email: [vkamath@vmware.com](mailto:vkamath@vmware.com)

Ramakrishnan Chokkanathapuram Sundaram  
Cisco Systems, Inc.  
Tasman Drive  
San Jose, CA 95134  
United States of America

Email: ramaksun@cisco.com

Raunak Banthia  
Apstra  
333 Middlefield Rd STE 200  
Menlo Park, CA 94025  
United States of America

Email: rbanthia@apstra.com

Ananya Gopal  
Cisco Systems, Inc.  
Tasman Drive  
San Jose, CA 95134  
United States of America

Email: ananygop@cisco.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 September 2022

D. Voyer, Ed.  
Bell Canada  
C. Filsfils  
R. Parekh  
Cisco Systems, Inc.  
H. Bidgoli  
Nokia  
Z. Zhang  
Juniper Networks  
7 March 2022

Segment Routing Point-to-Multipoint Policy  
draft-ietf-pim-sr-p2mp-policy-04

Abstract

This document describes an architecture to construct a Point-to-Multipoint (P2MP) tree to deliver Multi-point services in a Segment Routing domain. A SR P2MP tree is constructed by stitching a set of Replication segments together. A SR Point-to-Multipoint (SR P2MP) Policy is used to define and instantiate a P2MP tree which is computed by a PCE.

Requirements Language

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].

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 8 September 2022.



## Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. P2MP Tree . . . . .	3
2.1. Sharing Replication segments across P2MP trees . . . . .	4
3. SR P2MP Policy . . . . .	5
4. Using Controller to build a P2MP Tree . . . . .	6
4.1. Provisioning SR P2MP Policy Creation . . . . .	6
4.1.1. API . . . . .	7
4.1.2. Invoking API . . . . .	7
4.2. P2MP Tree Computation . . . . .	7
4.2.1. Topology Discovery . . . . .	8
4.2.2. Capability and Attribute Discovery . . . . .	8
4.3. Instantiating P2MP tree on nodes . . . . .	8
4.3.1. PCEP . . . . .	9
4.3.2. BGP . . . . .	9
4.3.3. NetConf . . . . .	9
4.4. Protection . . . . .	9
4.4.1. Local Protection . . . . .	9
4.4.2. Path Protection . . . . .	9
5. IANA Considerations . . . . .	9
6. Security Considerations . . . . .	9
7. Acknowledgements . . . . .	10
8. Contributors . . . . .	10
9. References . . . . .	10
9.1. Normative References . . . . .	10
9.2. Informative References . . . . .	11
Appendix A. Illustration of SR P2MP Policy and P2MP Tree . . . . .	12
A.1. P2MP Tree with non-adjacent Replication Segments . . . . .	13
A.1.1. SR-MPLS . . . . .	14
A.1.2. SRv6 . . . . .	15
A.2. P2MP Tree with adjacent Replication Segments . . . . .	17
A.2.1. SR-MPLS . . . . .	17
A.2.2. SRv6 . . . . .	19

Authors' Addresses . . . . . 20

## 1. Introduction

A Multi-point service delivery could be realized via P2MP trees in a Segment Routing domain [RFC8402]. A P2MP tree spans from a Root node to a set of Leaf nodes via intermediate Replication Nodes. It consists of a Replication segment [I-D.ietf-spring-sr-replication-segment] at the root node, one or more Replication segments at Leaf nodes and intermediate Replication Nodes. The Replication segments are stitched together.

A Segment Routing P2MP policy, a variant of the SR Policy [I-D.ietf-spring-segment-routing-policy], is used to define a P2MP tree. A PCE is used to compute the tree from the Root node to the set of Leaf nodes via a set of Replication Nodes. The PCE then instantiates the P2MP tree in the SR domain by signaling Replication segments to Root, replication and Leaf nodes using various protocols (PCEP, BGP, NetConf etc.). Replication segments of a P2MP tree can be instantiated for SR-MPLS and SRv6 dataplanes.

## 2. P2MP Tree

A P2MP tree in a SR domain connects a Root to a set of Leaf nodes via a set of intermediate Replication Nodes. It consists of a Replication segment at the root stitched to Replication segments at intermediate Replication Nodes eventually reaching the Leaf nodes.

The Replication SID of the Replication segment at Root node is called Tree-SID. The Tree-SID SHOULD also be used as Replication SID of Replication segments at Replication and Leaf nodes. The Replication segments at Replication and Leaf nodes MAY use Replication SIDs that are not same as the Tree-SID.

The Replication segment at Root of a P2MP tree MUST be associated with that P2MP tree (i.e. <Root, Tree-ID> identifier in SR P2MP policy section below) to map a Multi-point service to the tree. A Replication segment that terminates a P2MP tree at a Leaf node MUST be associated with the P2MP tree to determine the context for a Multi-point service. The information that can be used to derive this association is specific to encoding of the protocol (PCEP, BGP, NetConf etc.) used to instantiate the Replication segment for a P2MP tree. Replication segments at intermediate Replication Nodes of a tree are also associated with that tree.

For SR-MPLS, a PCE MAY decide not to instantiate Replication segments at Leaf nodes of a P2MP tree if it is known a priori that Multi-point services mapped to the P2MP tree can be identified using a context

that is globally unique in SR domain. In this case, Replication Nodes connecting to Leaf nodes effectively does Penultimate-Hop Pop (PHP) behavior to pop Tree-SID from a packet. A Multi-point service context assigned from "Domain-wide Common Block" (DCB) [I-D.ietf-bess-mvpn-evpn-aggregation-label] is an example of globally unique context.

A packet steered into a P2MP tree is replicated by the Replication segment at Root node to each downstream node in the Replication segment, with the Replication SID of the Replication segment at the downstream node. A downstream node could be a Leaf node or an intermediate Replication Node. In the latter case, replication continues with the Replication segments until all Leaf nodes are reached. A packet is steered into a P2MP tree in two ways:

- \* Based on a local policy-based routing at the Root node.
- \* Based on steering via the Tree-SID at the Root node.

#### 2.1. Sharing Replication segments across P2MP trees

Two or more P2MP trees MAY share a Replication segment at Root or Replication Nodes if at minimum the first condition below is satisfied. A tree always has its own Replication segment at its root even if shares another Replication segment. A tree that shares another Replication segment may or may not have its own Replication segment on its Leaf nodes. If not, the second and third conditions apply to such situations.

1. The Leaf nodes reached via a shared Replication segment must be subset of Leaf or Replication Nodes of the P2MP trees that share this segment. Note if a Replication segment is shared, all its downstream Replication segments are also shared.
2. Some Multi-point services realized by the P2MP trees may need service context (e.g. packets are for certain VPNs, and/or from certain nodes). If the trees do not have their own Replication segments at their Leaf nodes then the packets transported on the P2MP trees MUST carry a service context that does not rely on the tree or root identification, e.g. a service label assigned from Domain-wide Common Block or common SRGB for SR-MPLS.

3. For some Multi-point services using P2MP trees that share Replication segments, packets transported on these trees MAY require a Tree context (e.g. MVPN Extranet [RFC7900] to avoid certain ambiguities - see Section 2.3.1 of RFC 7900). In this case, the trees MUST have their own Replication segments on the Leaf nodes. For SR-MPLS, this is similar to "tunnel stacking" concept.

Sharing of a Replication segment for P2MP trees is OPTIONAL. Exact procedures to ensure validity of above conditions across PM2P services on nodes of a Segment Routing domain are outside the scope of this document.

### 3. SR P2MP Policy

The SR P2MP policy is a variant of an SR policy[I-D.ietf-spring-segment-routing-policy] and is used to instantiate SR P2MP trees.

A SR P2MP Policy is identified by the tuple <Root, Tree-ID>, where:

- \* Root: The address of Root node of P2MP tree instantiated by the SR P2MP Policy
- \* Tree-ID: A identifier that is unique in context of the Root. This is an unsigned 32-bit number.

A SR P2MP Policy is defined by following elements:

- \* Leaf nodes: A set of nodes that terminate the P2MP trees.
- \* Candidate Paths: See below.

A SR P2MP policy is provisioned on a PCE to instantiate the P2MP tree. The Tree-SID SHOULD be used as Binding SID of the P2MP policy. A PCE computes the P2MP tree and instantiates Replication segments at Root, Replication and Leaf nodes. When Replication segments are not shared across P2MP trees, the Root and Tree-ID of the SR P2MP policy are mapped to Replication-ID element of the Replication segment identifier i.e the SR Replication segment identifier is <Root, Tree-ID, Node-ID>. A shared Replication segment MAY be identified with zero Root-ID address (0.0.0.0 for IPv4 and :: for IPv6) and a Replication-ID that is unique in context of Node address where the Replication segment is instantiated when it is not associated a particular tree.

A SR P2MP Policy has one or more Candidate paths. The active Candidate path is selected based on the tie breaking rules amongst the candidate-paths as specified in [I-D.ietf-spring-segment-routing-policy]. Each candidate path has a set of topological/resource constraints and/or optimization objectives which determine the P2MP tree for that Candidate path. Tree-SID is an identifier of the P2MP tree of the candidate path in the forwarding plane. It is instantiated in the forwarding plane at Root node, intermediate Replication Nodes and Leaf nodes. The Tree-SID MAY be different at Replication and Leaf nodes.

#### 4. Using Controller to build a P2MP Tree

A P2MP tree can be built using a Path Computation Element (PCE). This section outlines a high-level architecture for such an approach.

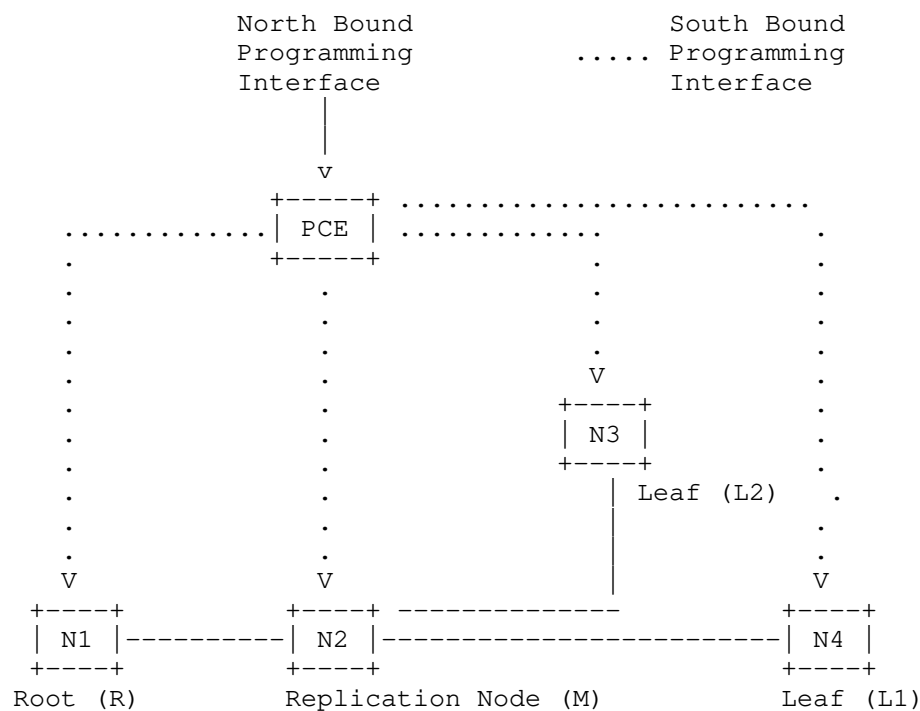


Figure 1: Centralized Control Plane Model

##### 4.1. Provisioning SR P2MP Policy Creation

A SR P2MP policy can be instantiated and maintained in a centralized fashion using a Path Computation Element (PCE).

#### 4.1.1. API

North-bound APIs on a PCE can be used to:

1. Create SR P2MP policy: `CreateSRP2MPPolicy<Root, Tree-ID>`
2. Delete SR P2MP policy: `DeleteSRP2MPPolicy<Root, Tree-ID>`
3. Modify SR P2MP policy Leaf Set: `SRP2MPPolicyLeafSetModify<Root, Tree-ID, {Leaf Set}>`
4. Create a Candidate Path for SR P2MP policy:  
`CreateSRP2MPCandidatePath<Root, Tree-ID, <CP-ID>>`
5. Delete a Candidate Path for SR P2MP policy:  
`DeleteSRP2MPCandidatePath<Root, Tree-ID, <CP-ID>>`
6. Update a Candidate Path for SR P2MP policy:  
`UpdateSRP2MPCandidatePath<Root, Tree-ID, <CP-ID>, Preference, Constraints, Optimization, ...>`

CP-ID is identifier of a Candidate Path within a SR P2MP policy. One possible identifier is the tuple `<Protocol-Origin, originator, discriminator>` as specified in [I-D.ietf-spring-segment-routing-policy].

Note these are conceptual APIs. Actual implementations may offer different APIs as long as they provide same functionality. For example, API might allow symbolic name to be assigned for a P2MP policy or APIs might allow individual Leaf nodes to be added or deleted from a policy instead of an update operation.

#### 4.1.2. Invoking API

Interaction with a PCE can be via PCEP, REST, Netconf, gRPC, CLI. Yang model shall be developed for this purpose as well.

#### 4.2. P2MP Tree Computation

An entity (an operator, a network node or a machine) provisions a SR P2MP policy by specifying the addresses of the root (R) and set of leaves {L} as well as Traffic Engineering (TE) attributes of Candidate paths via a suitable North-Bound API. The PCE computes the tree of Active candidate path. The PCE MAY compute P2MP trees for all Candidate paths., If tree computation is successful, PCE instantiates the P2MP tree(s) using Replication segments on Root, Replication, and Leaf nodes.

Candidate path constraints shall include link color affinity, bandwidth, disjointness (link, node, SRLG), delay bound, link loss, etc. Candidate path shall be optimized based on IGP or TE metric or link latency.

The Tree SID of Candidate path of a SR P2MP policy can be either dynamically allocated by the PCE or statically assigned by entity provisioning the SR P2MP policy. Ideally, same Tree-SID SHOULD be used for Replication segments at Root, Replication, and Leaf nodes. Different Tree-SIDs MAY be used at Replication Node(s) if it is not feasible to use same Tree SID.

A PCE can modify a P2MP tree following network element failure or in case a better path can be found based on the new network state. In this case, the PCE may want to setup the new instance of the tree and remove the old instance of the tree from the network in order to minimize traffic loss. In this case, the instances of trees for all the Candidate paths of a P2MP policy can be identified by an Instance-ID which is unique in context of the P2MP policy. As such, the identifier of non-shared Replication segments used to instantiate these trees becomes <Root-ID, Tree-ID, Node-ID, Instance-ID>.

A PCE shall be capable of computing paths across multiple IGP areas or levels as well as Autonomous Systems (ASs).

#### 4.2.1. Topology Discovery

A PCE shall learn network topology, TE attributes of link/node as well as SIDs via dynamic routing protocols (IGP and/or BGP-LS). It may be possible for entities to pass topology information to PCE via north-bound API.

#### 4.2.2. Capability and Attribute Discovery

It shall be possible for a node to advertise SR P2MP tree capability via IGP and/or BGP-LS. Similarly, a PCE can also advertise its P2MP tree computation capability via IGP and/or BGP-LS. Capability advertisement allows a network node to dynamically choose one or more PCE(s) to obtain services pertaining to SR P2MP policies, as well as a PCE to dynamically identify SR P2MP tree capable nodes.

#### 4.3. Instantiating P2MP tree on nodes

Once a PCE computes a P2MP tree for Candidate path of SR P2MP policy, it needs to instantiate the tree on the relevant network nodes via Replication segments. The PCE can use various protocols to program the Replication segments as described below.

#### 4.3.1. PCEP

PCE Protocol (PCEP) has been traditionally used:

1. For a head-end to obtain paths from a PCE.
2. A PCE to instantiate SR policies.

PCEP protocol can be stateful in that a PCE can have a stateful control of an SR policy on a head-end which has delegated the control of the SR policy to the PCE. PCEP shall be extended to provision and maintain SR P2MP trees in a stateful fashion.

#### 4.3.2. BGP

BGP has been extended to instantiate and report SR policies. It shall be extended to instantiate and maintain P2MP trees for SR P2MP policies.

#### 4.3.3. NetConf

TBD

#### 4.4. Protection

##### 4.4.1. Local Protection

A network link, node or path on the tree of a P2MP tree can be protected using SR policies computed by PCE. The backup SR policies shall be programmed in forwarding plane in order to minimize traffic loss when the protected link/node fails. It is also possible to use node local Fast Re-Route protection mechanisms (LFA) to protect link/nodes of P2MP tree.

##### 4.4.2. Path Protection

It is possible for PCE create a disjoint backup tree for providing end-to-end path protection.

#### 5. IANA Considerations

This document makes no request of IANA.

#### 6. Security Considerations

There are no additional security risks introduced by this design.



## 7. Acknowledgements

The authors would like to acknowledge Siva Sivabalan, Mike Koldychev and Vishnu Pavan Beeram for their valuable inputs..

## 8. Contributors

Clayton Hassen Bell Canada Vancouver Canada

Email: clayton.hassen@bell.ca

Kurtis Gillis Bell Canada Halifax Canada

Email: kurtis.gillis@bell.ca

Arvind Venkateswaran Cisco Systems, Inc. San Jose US

Email: arvvenka@cisco.com

Zafar Ali Cisco Systems, Inc. US

Email: zali@cisco.com

Swadesh Agrawal Cisco Systems, Inc. San Jose US

Email: swaagraw@cisco.com

Jayant Kotalwar Nokia Mountain View US

Email: jayant.kotalwar@nokia.com

Tanmoy Kundu Nokia Mountain View US

Email: tanmoy.kundu@nokia.com

Andrew Stone Nokia Ottawa Canada

Email: andrew.stone@nokia.com

Tarek Saad Juniper Networks Canada

Email:tsaad@juniper.net

## 9. References

### 9.1. Normative References

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-18, 17 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-18.txt>>.
- [I-D.ietf-spring-sr-replication-segment]  
(editor), D. V., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", Work in Progress, Internet-Draft, draft-ietf-spring-sr-replication-segment-06, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-replication-segment-06.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

## 9.2. Informative References

- [I-D.filsfils-spring-srv6-net-pgm-illustration]  
Filsfils, C., Garvia, P. C., Li, Z., Matsushima, S., Decraene, B., Steinberg, D., Lebrun, D., Raszuk, R., and J. Leddy, "Illustrations for SRv6 Network Programming", Work in Progress, Internet-Draft, draft-filsfils-spring-srv6-net-pgm-illustration-04, 30 March 2021, <<https://www.ietf.org/archive/id/draft-filsfils-spring-srv6-net-pgm-illustration-04.txt>>.
- [I-D.ietf-bess-mvpn-evpn-aggregation-label]  
Zhang, Z., Rosen, E., Lin, W., Li, Z., and I. Wijnands, "MVPN/EVPN Tunnel Aggregation with Common Labels", Work in Progress, Internet-Draft, draft-ietf-bess-mvpn-evpn-aggregation-label-08, 20 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-bess-mvpn-evpn-aggregation-label-08.txt>>.

- [RFC7900] Rekhter, Y., Ed., Rosen, E., Ed., Aggarwal, R., Cai, Y., and T. Morin, "Extranet Multicast in BGP/IP MPLS VPNs", RFC 7900, DOI 10.17487/RFC7900, June 2016, <<https://www.rfc-editor.org/info/rfc7900>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

## Appendix A. Illustration of SR P2MP Policy and P2MP Tree

Consider the following topology:

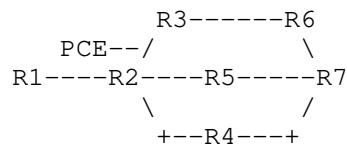


Figure 2: Figure 1

In these examples, the Node-SID of a node  $R_n$  is  $N\text{-}SID_n$  and Adjacency-SID from node  $R_m$  to node  $R_n$  is  $A\text{-}SID_{mn}$ . Interface between  $R_m$  and  $R_n$  is  $I_{mn}$ .

For SRv6, the reader is expected to be familiar with SRv6 Network Programming [RFC8986] to follow the examples. We use SID allocation scheme, reproduced below, from Illustrations for SRv6 Network Programming [I-D.filsfils-spring-srv6-net-pgm-illustration]

- \* 2001:db8::/32 is an IPv6 block allocated by a RIR to the operator
- \* 2001:db8:0::/48 is dedicated to the internal address space
- \* 2001:db8:cccc::/48 is dedicated to the internal SRv6 SID space
- \* We assume a location expressed in 64 bits and a function expressed in 16 bits
- \* Node  $k$  has a classic IPv6 loopback address 2001:db8:: $k$ /128 which is advertised in the IGP
- \* Node  $k$  has 2001:db8:cccc: $k$ ::/64 for its local SID space. Its SIDs will be explicitly assigned from that block
- \* Node  $k$  advertises 2001:db8:cccc: $k$ ::/64 in its IGP

- \* Function :1:: (function 1, for short) represents the End function with PSP support
- \* Function :Cn:: (function Cn, for short) represents the End.X function to Node n
- \* Function :Cln: (function Cln for short) represents the End.X function to Node n with USD

Each node k has:

- \* An explicit SID instantiation 2001:db8:cccc:k:1::/128 bound to an End function with additional support for PSP
- \* An explicit SID instantiation 2001:db8:cccc:k:Cj::/128 bound to an End.X function to neighbor J with additional support for PSP
- \* An explicit SID instantiation 2001:db8:cccc:k:Clj::/128 bound to an End.X function to neighbor J with additional support for USD

Assume PCE is provisioned following SR P2MP policy at Root R1 with Tree-ID T-ID:

```
SR P2MP Policy <R1,T-ID>:
  Leaf Nodes: {R2, R6, R7}
  Candidate-path 1:
    Optimize: IGP metric
    Tree-SID: T-SID1
```

The PCE is responsible for P2MP tree computation. Assume PCE instantiates P2MP trees by signalling non-shared Replication segments i.e. Replication-ID of these Replication segments is <Root, Tree-ID>. If a Candidate-path can have multiple instances of P2MP trees, the Replication-ID is <Root, Tree-ID, Instance-ID>. In this example, we assume one instance of P2MP tree for a candidate-path. All Replication segments use the Tree-SID T-SID1 as Replication-SID. For SRv6, assume the Replication SID at node k, bound to an End.Replcate function, is 2001:db8:cccc:k:FA::/128.

#### A.1. P2MP Tree with non-adjacent Replication Segments

Assume PCE computes a P2MP tree with Root node R1, Intermediate and Leaf node R2, and Leaf nodes R6 and R7. The PCE instantiates the P2MP tree by stitching Replication segments at R1, R2, R6 and R7. Replication segment at R1 replicates to R2. Replication segment at R2 replicates to R6 and R7. Note nodes R3, R4 and R5 do not have any Replication segment state for the tree.

## A.1.1.1. SR-MPLS

The Replication segment state at nodes R1, R2, R6 and R7 is shown below.

Replication segment at R1:

Replication segment <R1,T-ID,R1>:

Replication SID: T-SID1

Replication State:

R2: <T-SID1->L12>

Replication to R2 steers packet directly to the node on interface L12.

Replication segment at R2:

Replication segment <R1,T-ID,R2>:

Replication SID: T-SID1

Replication State:

R2: <Leaf>

R6: <N-SID6, T-SID1>

R7: <N-SID7, T-SID1>

R2 is a Bud-Node. It performs role of Leaf as well as a transit node replicating to R6 and R7. Replication to R6, using N-SID6, steers packet via IGP shortest path to that node. Replication to R7, using N-SID7, steers packet via IGP shortest path to R7 via either R5 or R4 based on ECMP hashing.

Replication segment at R6:

Replication segment <R1,T-ID,R6>:

Replication SID: T-SID1

Replication State:

R6: <Leaf>

Replication segment at R7:

Replication segment <R1,T-ID,R7>:

Replication SID: T-SID1

Replication State:

R7: <Leaf>

When a packet is steered into the SR P2MP Policy at R1:

- \* Since R1 is directly connected to R2, R1 performs PUSH operation with just <T-SID1> label for the replicated copy and sends it to R2 on interface L12.
- \* R2, as Leaf, performs NEXT operation, pops T-SID1 label and delivers the payload. For replication to R6, R2 performs a PUSH operation of N-SID6, to send <N-SID6,T-SID1> label stack to R3. R3 is the penultimate hop for N-SID6; it performs penultimate hop popping, which corresponds to the NEXT operation and the packet is then sent to R6 with <T-SID1> in the label stack. For replication to R7, R2 performs a PUSH operation of N-SID7, to send <N-SID7,T-SID1> label stack to R4, one of IGP ECMP nexthops towards R7. R4 is the penultimate hop for N-SID6; it performs penultimate hop popping, which corresponds to the NEXT operation and the packet is then sent to R7 with <T-SID1> in the label stack.
- \* R6, as Leaf, performs NEXT operation, pops T-SID1 label and delivers the payload.
- \* R7, as Leaf, performs NEXT operation, pops R-SID7 label and delivers the payload.

#### A.1.1.2. SRv6

For SRv6, the replicated packet from R2 to R7 has to traverse R4 using a SR-TE policy, Policy27. The policy has one SID in segment list: End.X function with USD of R4 to R7. The Replication segment state at nodes R1, R2, R6 and R7 is shown below.

Policy27: <2001:db8:cccc:4:C17::>

Replication segment at R1:

Replication segment <R1,T-ID,R1>:  
 Replication SID: 2001:db8:cccc:1:FA::  
 Replication State:  
 R2: <2001:db8:cccc:2:FA::->L12>

Replication to R2 steers packet directly to the node on interface L12.

Replication segment at R2:

Replication segment <R1,T-ID,R2>:  
Replication SID: 2001:db8:cccc:2:FA::  
Replication State:  
R2: <Leaf>  
R6: <2001:db8:cccc:6:FA::>  
R7: <2001:db8:cccc:7:FA:: -> Policy27>

R2 is a Bud-Node. It performs role of Leaf as well as a transit node replicating to R6 and R7. Replication to R6, steers packet via IGP shortest path to that node. Replication to R7, via SR-TE policy, first encapsulates the packet using H.Encaps and then steers the outer packet to R4. End.X USD on R4 decapsulates outer header and sends the original inner packet to R7.

Replication segment at R6:

Replication segment <R1,T-ID,R6>:  
Replication SID: 2001:db8:cccc:6:FA::  
Replication State:  
R6: <Leaf>

Replication segment at R7:

Replication segment <R1,T-ID,R7>:  
Replication SID: 2001:db8:cccc:7:FA::  
Replication State:  
R7: <Leaf>

When a packet (A,B2) is steered into the SR P2MP Policy at R1 using H.Encaps.Replicate behavior:

- \* Since R1 is directly connected to R2, R1 sends replicated copy (2001:db8::1, 2001:db8:cccc:2:FA::) (A,B2) to R2 on interface L12.
- \* R2, as Leaf removes outer IPv6 header and delivers the payload. R2, as a bud node, also replicates the packet.
  - For replication to R6, R2 sends (2001:db8::1, 2001:db8:cccc:6:FA::) (A,B2) to R3. R3 forwards the packet using 2001:db8:cccc:6::/64 packet to R6.
  - For replication to R7 using Policy27, R2 encapsulates and sends (2001:db8::2, 2001:db8:cccc:4:C17::) (2001:db8::1, 2001:db8:cccc:7:FA::) (A,B2) to R4. R4 performs End.X USD behavior, decapsulates outer IPv6 header and sends (2001:db8::1, 2001:db8:cccc:7:FA::) (A,B2) to R7.
- \* R6, as Leaf, removes outer IPv6 header and delivers the payload.

- \* R7, as Leaf, removes outer IPv6 header and delivers the payload.

#### A.2. P2MP Tree with adjacent Replication Segments

Assume PCE computes a P2MP tree with Root node R1, Intermediate and Leaf node R2, Intermediate nodes R3 and R5, and Leaf nodes R6 and R7. The PCE instantiates the P2MP tree by stitching Replication segments at R1, R2, R3, R5, R6 and R7. Replication segment at R1 replicates to R2. Replication segment at R2 replicates to R3 and R5. Replication segment at R3 replicates to R6. Replication segment at R5 replicates to R7. Note node R4 does not have any Replication segment state for the tree.

##### A.2.1. SR-MPLS

The Replication segment state at nodes R1, R2, R3, R5, R6 and R7 is shown below.

Replication segment at R1:

```
Replication segment <R1,T-ID,R1>:
  Replication SID: T-SID1
  Replication State:
    R2: <T-SID1->L12>
```

Replication to R2 steers packet directly to the node on interface L12.

Replication segment at R2:

```
Replication segment <R1,T-ID,R2>:
  Replication SID: T-SID1
  Replication State:
    R2: <Leaf>
    R3: <T-SID1->L23>
    R5: <T-SID1->L25>
```

R2 is a Bud-Node. It performs role of Leaf as well as a transit node replicating to R3 and R5. Replication to R3, steers packet directly to the node on L23. Replication to R5, steers packet directly to the node on L25.

Replication segment at R3:

```
Replication segment <R1,T-ID,R3>:
  Replication SID: T-SID1
  Replication State:
    R6: <T-SID1->L36>
```



Replication to R6, steers packet directly to the node on L36.

Replication segment at R5:

Replication segment <R1,T-ID,R5>:  
Replication SID: T-SID1  
Replication State:  
R7: <T-SID1->L57>

Replication to R7, steers packet directly to the node on L57.

Replication segment at R6:

Replication segment <R1,T-ID,R6>:  
Replication SID: T-SID1  
Replication State:  
R6: <Leaf>

Replication segment at R7:

Replication segment <R1,T-ID,R7>:  
Replication SID: T-SID1  
Replication State:  
R7: <Leaf>

When a packet is steered into the SR P2MP Policy at R1:

- \* Since R1 is directly connected to R2, R1 performs PUSH operation with just <T-SID1> label for the replicated copy and sends it to R2 on interface L12.
- \* R2, as Leaf, performs NEXT operation, pops T-SID1 label and delivers the payload. It also performs PUSH operation on T-SID1 for replication to R3 and R5. For replication to R6, R2 sends <T-SID1> label stack to R3 on interface L23. For replication to R5, R2 sends <T-SID1> label stack to R5 on interface L25.
- \* R3 performs NEXT operation on T-SID1 and performs a PUSH operation for replication to R6 and sends <T-SID1> label stack to R6 on interface L36.
- \* R5 performs NEXT operation on T-SID1 and performs a PUSH operation for replication to R7 and sends <T-SID1> label stack to R7 on interface L57.
- \* R6, as Leaf, performs NEXT operation, pops T-SID1 label and delivers the payload.

- \* R7, as Leaf, performs NEXT operation, pops R-SID7 label and delivers the payload.

#### A.2.2. SRv6

The Replication segment state at nodes R1, R2, R3, R5, R6 and R7 is shown below.

Replication segment at R1:

Replication segment <R1,T-ID,R1>:  
Replication SID: 2001:db8:cccc:1:FA::  
Replication State:  
R2: <2001:db8:cccc:2:FA::->L12>

Replication to R2 steers packet directly to the node on interface L12.

Replication segment at R2:

Replication segment <R1,T-ID,R2>:  
Replication SID: 2001:db8:cccc:2:FA::  
Replication State:  
R2: <Leaf>  
R3: <2001:db8:cccc:3:FA::->L23>  
R5: <2001:db8:cccc:5:FA::->L25>

R2 is a Bud-Node. It performs role of Leaf as well as a transit node replicating to R3 and R5. Replication to R3, steers packet directly to the node on L23. Replication to R5, steers packet directly to the node on L25.

Replication segment at R3:

Replication segment <R1,T-ID,R3>:  
Replication SID: 2001:db8:cccc:3:FA::  
Replication State:  
R6: <2001:db8:cccc:6:FA::->L36>

Replication to R6, steers packet directly to the node on L36.

Replication segment at R5:

Replication segment <R1,T-ID,R5>:  
Replication SID: 2001:db8:cccc:5:FA::  
Replication State:  
R7: <2001:db8:cccc:7:FA::->L57>

Replication to R7, steers packet directly to the node on L57.

Replication segment at R6:

Replication segment <R1,T-ID,R6>:  
Replication SID: 2001:db8:cccc:6:FA::  
Replication State:  
R6: <Leaf>

Replication segment at R7:

Replication segment <R1,T-ID,R7>:  
Replication SID: 2001:db8:cccc:7:FA::  
Replication State:  
R7: <Leaf>

When a packet (A,B2) is steered into the SR P2MP Policy at R1 using H.Encaps.Replicate behavior:

- \* Since R1 is directly connected to R2, R1 sends replicated copy (2001:db8::1, 2001:db8:cccc:2:FA::) (A,B2) to R2 on interface L12.
- \* R2, as Leaf, removes outer IPv6 header and delivers the payload. R2, as a bud node, also replicates the packet. For replication to R3, R2 sends (2001:db8::1, 2001:db8:cccc:3:FA::) (A,B2) to R3 on interface L23. For replication to R5, R2 sends (2001:db8::1, 2001:db8:cccc:5:FA::) (A,B2) to R5 on interface L25.
- \* R3 replicates and sends (2001:db8::1, 2001:db8:cccc:6:FA::) (A,B2) to R6 on interface L36.
- \* R5 replicates and sends (2001:db8::1, 2001:db8:cccc:7:FA::) (A,B2) to R7 on interface L57.
- \* R6, as Leaf, removes outer IPv6 header and delivers the payload.
- \* R7, as Leaf, removes outer IPv6 header and delivers the payload.

#### Authors' Addresses

Daniel Voyer (editor)  
Bell Canada  
Montreal  
Canada  
Email: daniel.voyer@bell.ca

Clarence Filsfils  
Cisco Systems, Inc.  
Brussels  
Belgium  
Email: cfilsfil@cisco.com

Rishabh Parekh  
Cisco Systems, Inc.  
San Jose,  
United States of America  
Email: riparekh@cisco.com

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

Zhaohui Zhang  
Juniper Networks  
Email: zzhang@juniper.net

PIM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 2, 2021

O. Komolafe  
Arista Networks  
September 29, 2020

IGMPv3 and MLDv2 Survey Report  
draft-komolafe-pim-igmp-mld-survey-report-00

## Abstract

The PIM WG intends to progress IGMPv3 and MLDv2 from Proposed Standards to Internet Standards. The WG decided to conduct a survey of operators, vendors and implementors of these and related protocols to gather information about their implementation and deployment. This document presents the results of the survey and briefly summarizes the key findings. The survey indicates that there is widespread deployment and usage of IGMPv3 and MLDv2, with numerous independent implementations interoperating successfully. No major issues with either protocol were identified and, similarly, no major unused features in the specifications were highlighted. These findings suggest that IGMPv3 and MLDv2 are indeed ready for progression to Internet Standards.

## 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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 2, 2021.

## Copyright Notice

Copyright (c) 2020 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

(<https://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 . . . . .	2
2. Approach . . . . .	3
2.1. Methodology . . . . .	3
3. Responses for Vendors or Host Implementors . . . . .	4
3.1. Protocols Implemented . . . . .	4
3.2. Features Supported . . . . .	4
3.3. Issues Identified . . . . .	4
3.4. Suggestions . . . . .	4
4. Responses for Network Operators . . . . .	5
4.1. Protocols Deployed . . . . .	5
4.2. Features Enabled . . . . .	5
4.3. Interoperability Issues . . . . .	5
4.4. Fallback Mechanism . . . . .	5
4.5. Strengths and Weaknesses of IGMPv3 and MLDv2 . . . . .	5
5. Conclusions . . . . .	6
6. Acknowledgements . . . . .	6
7. References . . . . .	6
7.1. Normative References . . . . .	6
7.2. Informative References . . . . .	7
Appendix A. Questionnaire . . . . .	7
A.1. Questionnaire for Vendors or Host Implementors . . . . .	7
A.1.1. Implementation Status . . . . .	7
A.1.2. Implementation Specifics . . . . .	7
A.1.3. Implementation Perspectives . . . . .	8
A.2. Questionnaire for Network Operators . . . . .	8
A.2.1. Deployment Status . . . . .	8
A.2.2. Deployment Specifics . . . . .	9
A.2.3. Deployment Perspectives . . . . .	10
Author's Address . . . . .	10

## 1. Introduction

Internet Group Management Protocol Version 3 (IGMPv3) [RFC3376] and Multicast Listener Discovery Version 2 (MLDv2) for IPv6 [RFC3810] are currently Proposed Standards. Given the fact that multiple independent implementations of these protocols exist and they have been successfully and widely used operationally, the PIM WG is keen to progress these protocols to Internet Standards. As such, it is

critical to establish if there are features specified in [RFC3376] and [RFC3810] that have not been widely used and also to determine any interoperability issues that have arisen from using the protocols.

Following approach taken for PIM-SM, documented in [RFC7063], the PIM WG has decided that conducting a comprehensive survey on implementations and deployment of IGMPv3 and MLDv2 will provide valuable information to facilitate their progression to Internet Standard.

This document summarizes the findings of the survey.

## 2. Approach

### 2.1. Methodology

The raw survey questions are shown in Appendix A. In order to make the submission and processing of responses as convenient as possible, Tim Chown kindly formatted and posted the survey online using the JISC online surveys tool. The PIM WG chairs subsequently announced the survey, publicizing the URL at which the survey could be completed. In addition to announcing the survey on the relevant IETF WG mailing lists, effort was made to distribute the survey to other forums such as NANOG.

The survey was targeted at:

- Network operators

- Router vendors

- Switch vendors

- Host implementors

Once the deadline for the survey elapsed, Tim Chown collated the responses, anonymizing the data so the responses from a specific operator, vendor or implementor could not be identified.

The questions targeted at vendors or host implementors were answered by 10 respondents. The network operators questions were answered by 14 respondents. (These numbers are comparable with the number of responses to the PIM-SM survey [RFC7063].)

### 3. Responses for Vendors or Host Implementors

#### 3.1. Protocols Implemented

80% or more of the respondents had implemented each of IGMPv1, IGMPv2, IGMPv3, MLDv1 and MLDv2, with IGMPv3 being the only protocol that had been implemented by all the respondents. In contrast, Lightweight IGMPv3 and Lightweight MLDv2 had been implemented by only 20% of the respondents.

#### 3.2. Features Supported

All the respondents supported source filtering with include list. Snooping querier was also a popular feature, with 80% of respondents supporting it. Source filtering with exclude list, snooping proxy, snooping filtering, L2 report flooding, host proxy were moderately popular, with 40%-70% of respondents supporting each of these features. Unicast queries/reports were supported by only 20% of the respondents.

#### 3.3. Issues Identified

No ambiguities or inconsistencies in [RFC3376] and [RFC3810] that made the implementation challenging were identified by any respondent.

#### 3.4. Suggestions

A number of respondents made suggestions to the PIM WG regarding progressing IGMPv3 and MLDv2 to full standards:

- o Add source discovery mechanism to SSM in addition to existing application-based source discovery
- o Improve scalability of query/response messages
- o Deprecate older versions and streamline IGMPv3
- o Allow reports to be sent without a querier
- o Remove source filtering with exclude list as it is not widely used and makes state machine unnecessarily complicated

Each of these points was raised by a different respondent, apart from the last point which was raised by two separate respondents.



## 4. Responses for Network Operators

### 4.1. Protocols Deployed

IGMPv2 was the most widely deployed protocol, with 86% of respondents indicating it is running in their network. Next was IGMPv3 with 79% of respondents indicating it is deployed. However, between only 20% and 36% of respondents indicated they had deployed IGMPv1, MLDv1 and MLDv2. Lightweight IGMPv3 and Lightweight MLDv2 were undeployed.

### 4.2. Features Enabled

Between 20% and 30% of respondents indicated that had enabled Source filtering with include list, source filtering with exclude list, snooping querier, snooping filtering or unicast queries/reports. Snooping proxy and L2 report flooding were enabled by 7% of respondents. No respondent was using host proxy.

### 4.3. Interoperability Issues

Half the respondents indicated they were using equipment with multi-vendor implementations in their network. No interoperability issues were identified.

### 4.4. Fallback Mechanism

36% of respondents indicated there are dependent on the fallback mechanisms between the different protocol versions. 7% of respondents have experienced issues related to this fallback mechanism.

### 4.5. Strengths and Weaknesses of IGMPv3 and MLDv2

A respondent indicated that a significant strength of IGMPv3 was the simplicity introduced by using SSM, avoiding the complexities associated with ASM. The weaknesses associated with IGMPv3 which were identified were:

- o No CPE implementations
- o Automatic fallback makes deployments challenging
- o ASM provides better source filtering (by potentially restricting the acceptance of register messages at the RP) whereas SSM allows only data plane filtering using multicast boundary

## 5. Conclusions

There were a total of 24 respondents to the survey which asked vendors/implementors and network operators questions about IGMPv1, IGMPv2, IGMPv3, Lightweight IGMPv3, MLDv1, MLDv2 and Lightweight MLDv2. A reasonable number of responses were gathered to the survey, allowing some interesting observations to be made. Firstly, and perhaps unsurprisingly, operators use a lower number of protocols and protocol features than have been implemented. Furthermore, there is a relatively lower deployment of the different MLD versions, suggesting that IPv6 multicast is less widely used than IPv4 multicast. No major flaws, inconsistencies or ambiguity in the IGMPv3 [RFC3376] and MLDv2 [RFC3810] specifications were identified. However, a number of issues were raised about the usage of these protocols, notably concerns about the automatic fallback from IGMPv3 to IGMPv2 being sometimes problematic and the loss of certain useful features offered by the ASM control plane with the transition to SSM.

These findings suggest that IGMPv3 and MLDv2 are indeed ready for progression to Internet Standards.

## 6. Acknowledgements

The authors are grateful to Tim Chown for posting the survey online, and for collating and anonymizing the responses.

## 7. References

### 7.1. Normative References

- [RFC1112] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.
- [RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC5790] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.

## 7.2. Informative References

[RFC7063] Zheng, L., Zhang, Z., and R. Parekh, "Survey Report on Protocol Independent Multicast - Sparse Mode (PIM-SM) Implementations and Deployments", RFC 7063, December 2013.

## Appendix A. Questionnaire

### A.1. Questionnaire for Vendors or Host Implementors

Name:

Affiliation/Organization:

Contact Email:

Do you wish to complete the survey anonymously?: Y/N

#### A.1.1. Implementation Status

Which of the following have you implemented?

1. IGMPv1 [RFC1112]?
2. IGMPv2 [RFC2236]?
3. IGMPv3 [RFC3376]?
4. Lightweight IGMPv3 [RFC5790]?
5. MLDv1 [RFC2710]?
6. MLDv2 [RFC3810]?
7. Lightweight MLDv2 [RFC5790]?

#### A.1.2. Implementation Specifics

1. Which IGMPv3 and MLDv2 features have you implemented?
  - A. Source filtering with include list?
  - B. Source filtering with exclude list?

- C. Snooping proxy?
  - D. Snooping querier?
  - E. Snooping filtering?
  - F. L2 Report flooding?
  - G. Host proxy?
  - H. Unicast queries/reports?
- 2. Have you carried out IGMPv3 or MLDv2 interoperability tests with other implementations?
    - A. What issues, if any, arose during these tests?
    - B. How could [RFC3376] and [RFC3810] have helped minimize these issues?

#### A.1.3. Implementation Perspectives

- 1. Which ambiguities or inconsistencies in RFC 3376 or RFC 3810 made the implementation challenging?
- 2. What suggestions would you make to the PIM WG as it seeks to progress IGMPv3 and MLDv2 to Internet Standard?

#### A.2. Questionnaire for Network Operators

Name:

Affiliation/Organization:

Contact Email:

Do you wish to complete the survey anonymously?: Y/N:

##### A.2.1. Deployment Status

Which of the following have you deployed in your network?

- 1. IGMPv1 [RFC1112]?
- 2. IGMPv2 [RFC2236]?
- 3. IGMPv3 [RFC3376]?

4. Lightweight IGMPv3 [RFC5790]?
5. MLDv1 [RFC2710]?
6. MLDv2 [RFC3810]?
7. Lightweight MLDv2 [RFC5790]?

#### A.2.2. Deployment Specifics

1. Which IGMPv3 and MLDv2 features do you use?
  - A. Source filtering with include list?
  - B. Source filtering with exclude list?
  - C. Snooping proxy?
  - D. Snooping querier?
  - E. Snooping filtering?
  - F. L2 Report flooding?
  - G. Host proxy?
  - H. Unicast queries/reports?
2. Are you using equipment with multi-vendor implementations in your IGMPv3/MLDv2 deployment?
  - A. What inter-operability issues, if any, have you experienced?
  - B. How could [RFC3376] and [RFC3810] have helped minimize these issues?
3. Are you using different IGMP versions or different MLD versions in your network?
  - A. Are you dependent on the fallback mechanism between the different versions?
  - B. Have you experienced any issues related to the fallback mechanism between the different versions?
  - C. How could [RFC3376] and [RFC3810] have helped minimize these issues?

A.2.3. Deployment Perspectives

1. Based on your operational experience, What have you found to be the strengths of IGMPv3 or MLDv2?
2. What have you found to be the weaknesses of IGMPv3 or MLDv2?
3. What suggestions would you make to the PIM WG as it seeks to progress IGMPv3 and MLDv2 to Internet Standard?

Author's Address

Olufemi Komolafe  
Arista Networks  
UK

Email: femi@arista.com

MBONED  
Internet-Draft  
Intended status: Standards Track  
Expires: July 23, 2021

H. Song  
M. McBride  
Futurewei Technologies  
G. Mirsky  
ZTE Corp.  
G. Mishra  
Verizon Inc.  
January 19, 2021

Multicast On-path Telemetry Solutions  
draft-song-multicast-telemetry-07

Abstract

This document discusses the requirement of on-path telemetry for multicast traffic. The existing solutions are examined and their issues are identified. Solution modifications are proposed to allow the original multicast tree to be correctly reconstructed without unnecessary replication of telemetry information.

Requirements Language

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

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on July 23, 2021.

## Copyright Notice

Copyright (c) 2021 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 (<https://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 . . . . .	2
2. Requirements for Multicast Traffic Telemetry . . . . .	3
3. Issues of Existing Techniques . . . . .	4
4. Proposed Modifications to Existing Techniques . . . . .	4
4.1. Per-hop postcard using IOAM DEX . . . . .	5
4.2. Per-section postcard . . . . .	7
5. Considerations for Different Multicast Protocols . . . . .	8
5.1. Application in PIM . . . . .	8
5.2. Application in P2MP . . . . .	9
5.3. Application in BIER . . . . .	9
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Contributors . . . . .	10
9. Acknowledgments . . . . .	10
10. References . . . . .	10
10.1. Normative References . . . . .	10
10.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

Multicast traffic is an important traffic type in today's Internet. Multicast provides services that are often real time (e.g., online meeting) or have strict QoS requirements (e.g., IPTV, Market Data). Multicast packet drop and delay can severely affect the application performance and user experience.

It is important to monitor the performance of the multicast traffic. Existing OAM techniques cannot gain direct and accurate information about the multicast traffic. New on-path telemetry techniques such as In-situ OAM [I-D.ietf-ippm-ioam-data], Postcard-based Telemetry



[I-D.song-ippm-postcard-based-telemetry], and Hybrid Two-Step (HTS) [I-D.mirsky-ippm-hybrid-two-step] provide promising means to directly monitor the network experience of multicast traffic. However, multicast traffic has some unique characteristics which pose some challenges on efficiently applying such techniques.

When a network contains multicast (p2mp) trees there will be redundant data as data is replicated at branch points. The IP Multicast S,G data is identical from one branch to another on it's way to multiple receivers. When adding iOAM trace data, to multicast packets, we enlarge data packets thus consuming more network bandwidth. Instead of adding iOAM trace data, it could be more efficient to collect the telemetry information using solutions, such as iOAM postcard or HTS, to cut down on the redundant iOAM data. The problem is that a postcard type solution doesn't have a branch identifier.

This draft proposes a set of solutions to this iOAM data redundancy problem. The requirements for multicast traffic telemetry are discussed along with the issues of the existing on-path telemetry techniques. We propose modifications to make these techniques adapt to multicast in order for the original multicast tree to be correctly reconstructed while eliminating redundant data.

## 2. Requirements for Multicast Traffic Telemetry

Multicast traffic is forwarded through a multicast tree. With PIM and P2MP (MLDP, RSVP-TE) the forwarding tree is established and maintained by the multicast routing protocol. With BIER, no state is created in the network to establish a forwarding tree, instead, a bier header provides the necessary information for each packet to know the egress points. Multicast packets are only replicated at each tree branch node for efficiency.

There are several requirements for multicast traffic telemetry, a few of which are:

- o Reconstruct and visualize the multicast tree through data plane monitoring.
- o Gather the multicast packet delay and jitter performance.
- o Find the multicast packet drop location and reason.
- o Gather the VPN state and tunnel information in case of P2MP multicast.

In order to meet these requirements, we need the ability to directly monitor the multicast traffic and derive data from the multicast packets. The conventional OAM mechanisms, such as multicast ping and trace, may not be sufficient to meet these requirements.

### 3. Issues of Existing Techniques

On-path Telemetry techniques that directly retrieve data from multicast traffic's live network experience are ideal to address the above mentioned requirements. The representative techniques include In-situ OAM (IOAM) Trace option [I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX) option [I-D.ioamteam-ippm-ioam-direct-export], and Postcard-based Telemetry with Packet Marking (PBT-M) [I-D.song-ippm-postcard-based-telemetry]. However, unlike unicast, multicast poses some unique challenges to applying these techniques.

Multicast packets are replicated at each branch node in the corresponding multicast tree. Therefore, there are multiple copies of packets in the network.

If the IOAM trace option is used for on-path data collection, the partial trace data will also be replicated into multiple copies. The end result is that each copy of the multicast packet has a complete trace. Most of the data, however, is redundant. Data redundancy introduces unnecessary header overhead, wastes network bandwidth, and complicates the data processing. In case the multicast tree is large, and the path is long, the redundancy problem becomes severe.

The PBT solutions, including the IOAM DEX option and PBT-M, can be used to eliminate such data redundancy, because each node on the tree only sends a postcard covering local data. However, they cannot track the tree branches properly so it can bring confusion about the multicast tree topology. For example, Node A has two branches, one to Node B and the other to node D, and Node B leads to Node C and Node D leads to Node E. From the received postcards, one cannot tell whether or not Node C(E) is the next hop of Node B(D).

The fundamental reason for this problem is that there is not an identifier (either implicit or explicit) to correlate the data on each branch.

### 4. Proposed Modifications to Existing Techniques

Two solutions are proposed to address the above issues. One is built on PBT and requires augmentation or modification to the instruction header of the IOAM Direct Export Option; the other combines the IOAM trace option and PBT for an optimized solution.

#### 4.1. Per-hop postcard using IOAM DEX

One way to mitigate PBT's multiple tree tracking weakness is to augment it with a branch identifier field. Note that this works for the IOAM DEX option but not for PBT-M because the IOAM DEX option uses an instruction header. To make the branch identifier globally unique, the branch node ID plus an index is used. For example, if Node A has two branches, one to Node B and one to Node C, Node A will use [A, 0] as the branch identifier for the branch to B, and [A, 1] for the branch to C. The identifier is unchanged for each multicast tree instance and carried with the multicast packet until the next branch node. Each postcard needs to include the branch identifier in the export data. The branch identifier, along with the other fields such as flow ID and sequence number, is sufficient for the data analyzer to reconstruct the topology of the multicast tree.

Figure 1 shows an example of this solution. "P" stands for the postcard packet. The square brackets contains the branch identifier. The curly brace contains the telemetry data about a specific node.

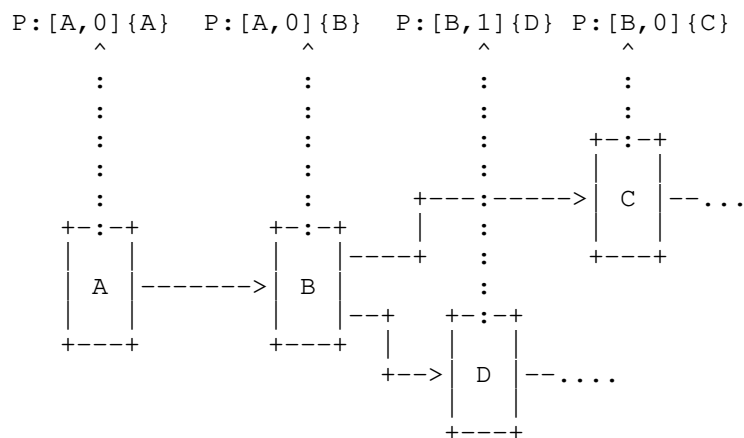


Figure 1: Per-hop Postcard

Each branch fork node need to generate the branch ID for each branch in its multicast tree instance and include it in the IOAM DEX option header so the downstream node can learn it. The branch ID contains two parts: the branch fork node ID and a unique branch index.

Figure 2 shows that the branch ID is carried as an optional field after the flow ID and sequence number optional fields in the IOAM DEX option header. A bit "M" in the Flags field is reserved to indicate

the presence of the branch index field. The "M" flag position will be determined later after the other flags are specified in [I-D.ioamteam-ippm-ioam-direct-export].

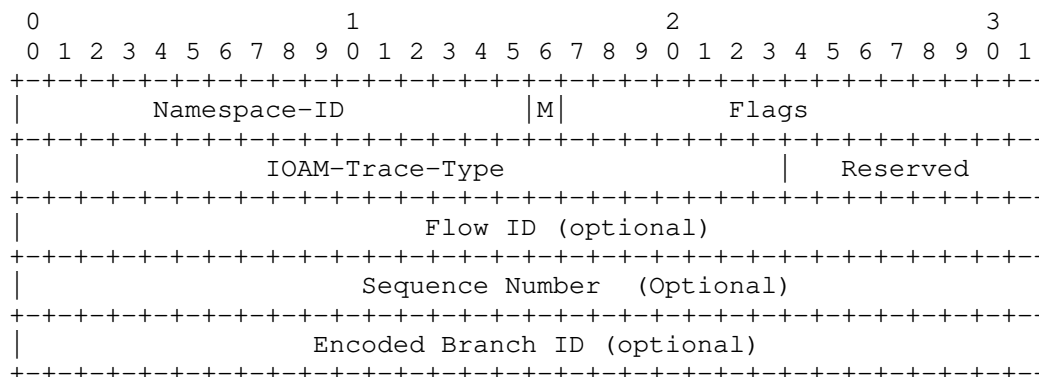


Figure 2: Carry Branch Index in IOAM DEX option header

To avoid introducing a new type of data field to the IOAM DEX option header, we can encode the branch identifier using the existing node ID data field as defined in [I-D.ietf-ippm-ioam-data]. Currently, the node ID field occupies three octets. A simple solution is to shorten the node ID field so a number of bits can be saved to encode the branch index, as shown in Figure 3.

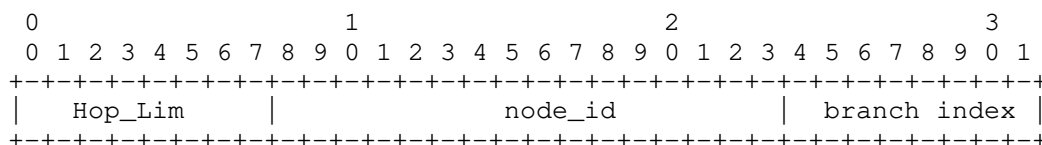


Figure 3: Encode Branch Index with Node ID Method 1

Another encoding method is to use the sum of the node ID and the branch index as the new node ID, as shown in Figure 4. As long as the node IDs are assigned with large enough gap, the telemetry data analyzer can still successfully recover the original node ID and branch index.

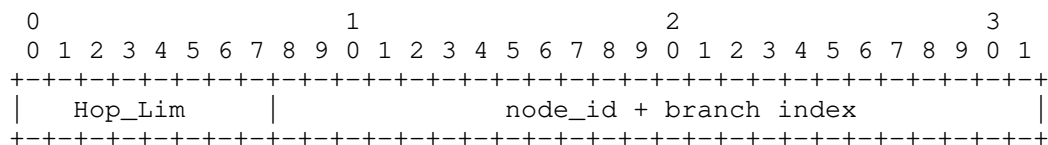


Figure 4: Encode Branch Index with Node ID Method 2

Once a node gets the branch ID information from the upstream, it **MUST** carry this information in its telemetry data export postcards, so the original multicast tree can be correctly reconstructed based on the postcards.

#### 4.2. Per-section postcard

The second solution is a combination of the IOAM trace mode and PBT. To avoid data redundancy at each branch node, the trace data accumulated, to that point, is exported by a postcard before the packet is replicated. In this case, each branch still needs to maintain some identifier to help correlate the postcards for each tree section. The natural way to accomplish this is to simply carry the branch node's data (including its ID) in the trace of each branch. This is also necessary because each replicated multicast packet can have different telemetry data pertaining to this particular copy (e.g., node delay, egress timestamp, and egress interface). As a consequence, the local data exported by each branch node can only contain partial data (e.g., ingress interface and ingress timestamp).

Figure 5 shows an example in a segment of a multicast tree. Node B and D are two branch nodes and they will export a postcard covering the trace data for the previous section. The end node of each path will also need to export the data of the last section as a postcard.

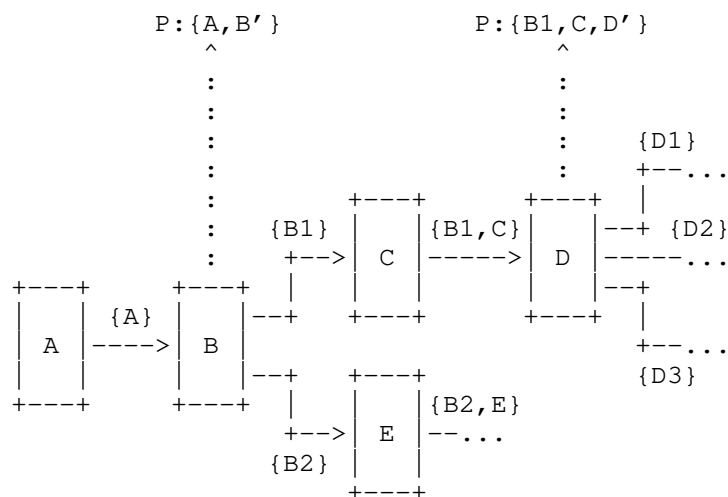


Figure 5: Per-section Postcard

There is no need to modify the IOAM trace mode header format. We just need to configure the branch node to export the postcard and refresh the IOAM header and data.

## 5. Considerations for Different Multicast Protocols

MTRACEv2 [RFC8487] provides an active probing approach for the tracing of an IP multicast routing path. Mtrace can also provide information such as the packet rates and losses, as well as other diagnostic information. New on-path telemetry techniques will enhance Mtrace, and other existing OAM solutions, with more granular and realtime network status data through direct measurements. There are various multicast protocols that are used to forward the multicast data. Each will require their own unique on-path telemetry solution.

### 5.1. Application in PIM

PIM-SM [RFC7761] is the most widely used multicast routing protocol deployed today. Of the various PIM modes (PIM-SM, PIM-DM, BIDIR-PIM, PIM-SSM), PIM-SSM is the preferred method due to its simplicity and removal of network source discovery complexity. With all PIM modes, control plane state is established in the network in order to forward multicast UDP data packets. But with PIM-SSM, the discovery of multicast sources is performed outside of the network via HTTP, SDN, etc. IP Multicast packets fall within the range of 224.0.0.0 through

239.255.255.255. The telemetry solution will need to work within this address range and provide telemetry data for this UDP traffic.

The proposed solutions for encapsulating the telemetry instruction header and metadata in IPv4/IPv6 UDP packets are described in [I-D.herbert-ipv4-udpencap-eh] and [I-D.ioametal-ippm-6man-ioam-ipv6-deployment].

## 5.2. Application in P2MP

Multicast Label Distribution Protocol (MLDP) and P2MP RSVP-TE are commonly used within a Multicast VPN (MVPN) environment. MLDP provides extensions to LDP to establish point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) in MPLS networks. P2MP RSVP-TE provides extensions to RSVP-TE for establish traffic-engineered P2MP LSPs in MPLS networks. The telemetry solution will need to be able to follow these P2MP paths. The telemetry instruction header and data should be encapsulated into MPLS packets on P2MP paths. A corresponding proposal is described in [I-D.song-mpls-extension-header].

## 5.3. Application in BIER

BIER [RFC8279] adds a new header to multicast packets and allows the multicast packets to be forwarded according to the header only. By eliminating the requirement of maintaining per multicast group state, BIER is more scalable than the traditional multicast solutions.

OAM Requirements for BIER [I-D.ietf-bier-oam-requirements] lists many of the requirements for OAM at the BIER layer which will help in the forming of on-path telemetry requirements as well.

There is also current work to provide solutions for BIER forwarding in ipv6 networks. For instance, a solution, BIER in Non-MPLS IPv6 Networks [I-D.xie-bier-ipv6-encapsulation], proposes a new bier Option Type codepoint from the "Destination Options and Hop-by-Hop Options" IPv6 sub-registry. This is similar to what IOAM proposes for IPv6 transport.

Depending on how the BIER header is encapsulated into packets with different transport protocols, the method to encapsulate the telemetry instruction header and metadata also varies. It is also possible to make the instruction header and metadata a part of the BIER header itself, such as in a TLV.

## 6. Security Considerations

No new security issues are identified other than those discovered by the IOAM and PBT drafts.

## 7. IANA Considerations

The document makes no request of IANA.

## 8. Contributors

TBD

## 9. Acknowledgments

The authors would like to thank Frank Brockners, Tianran Zhou for the comments and advice.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4687] Yasukawa, S., Farrel, A., King, D., and T. Nadeau, "Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks", RFC 4687, DOI 10.17487/RFC4687, September 2006, <<https://www.rfc-editor.org/info/rfc4687>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.



[RFC8487] Asaeda, H., Meyer, K., and W. Lee. Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", RFC 8487, DOI 10.17487/RFC8487, October 2018, <<https://www.rfc-editor.org/info/rfc8487>>.

## 10.2. Informative References

- [I-D.herbert-ipv4-udpencap-eh]  
Herbert, T., "IPv4 Extension Headers and UDP Encapsulated Extension Headers", draft-herbert-ipv4-udpencap-eh-01 (work in progress), March 2019.
- [I-D.ietf-bier-oam-requirements]  
Mirsky, G., Nainar, N., Chen, M., and S. Pallagatti, "Operations, Administration and Maintenance (OAM) Requirements for Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-oam-requirements-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-data]  
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ioametal-ippm-6man-ioam-ipv6-deployment]  
Bhandari, S., Brockners, F., Mizrahi, T., Kfir, A., Gafni, B., Spiegel, M., Krishnan, S., and M. Smith, "Deployment Considerations for In-situ OAM with IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-deployment-03 (work in progress), March 2020.
- [I-D.ioamteam-ippm-ioam-direct-export]  
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.
- [I-D.mirsky-ippm-hybrid-two-step]  
Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-07 (work in progress), December 2020.
- [I-D.song-ippm-postcard-based-telemetry]  
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., and L. Andersson, "MPLS Extension Header", draft-song-mpls-extension-header-02 (work in progress), February 2019.

[I-D.xie-bier-ipv6-encapsulation]

Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-09 (work in progress), January 2021.

#### Authors' Addresses

Haoyu Song  
Futurewei Technologies  
2330 Central Expressway  
Santa Clara  
USA

Email: [hsong@futurewei.com](mailto:hsong@futurewei.com)

Mike McBride  
Futurewei Technologies  
2330 Central Expressway  
Santa Clara  
USA

Email: [mmcbride@futurewei.com](mailto:mmcbride@futurewei.com)

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Gyan Mishra  
Verizon Inc.

Email: [gyan.s.mishra@verizon.com](mailto:gyan.s.mishra@verizon.com)