

Workgroup: Network Working Group

Internet-Draft:

draft-chen-pim-srv6-p2mp-path-04

Published: 22 October 2021

Intended Status: Experimental

Expires: 25 April 2022

Authors: H. Chen M. McBride Y. Fan Z. Li
Futurewei Futurewei Casa Systems Huawei
X. Geng M. Toy A. Wang L. Liu
Huawei Verizon China Telecom Fujitsu
X. Liu
Volta Networks

Stateless SRV6 Point-to-Multipoint Path

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 25 April 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. Overview of P2MP Multicast Tree](#)
- [3. Encoding P2MP Multicast Tree](#)
- [4. Procedures/Behaviors](#)
 - [4.1. Procedure/Behavior on Ingress Node](#)
 - [4.2. Procedure/Behavior on Transit Node](#)
 - [4.3. Procedure/Behavior on Egress Node](#)
- [5. Stateless SRv6 P2MP Path for Ingress](#)
- [6. Protection](#)
 - [6.1. Global Protection](#)
 - [6.2. Local Protection](#)
- [7. IANA Considerations](#)
- [8. Security Considerations](#)
- [9. Acknowledgements](#)
- [10. References](#)
 - [10.1. Normative References](#)
 - [10.2. Informative References](#)
- [Appendix A. Example IPv6 Header using G-SRv6](#)
- [Authors' Addresses](#)

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; each adjacency of a node has a multicast adjacency SID, which is locally significant. A multicast SID of a node on a SR P2MP path is associated with the SIDs of the 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 the 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, it duplicates and sends the packet to each of the B next hop nodes along the P2MP path.

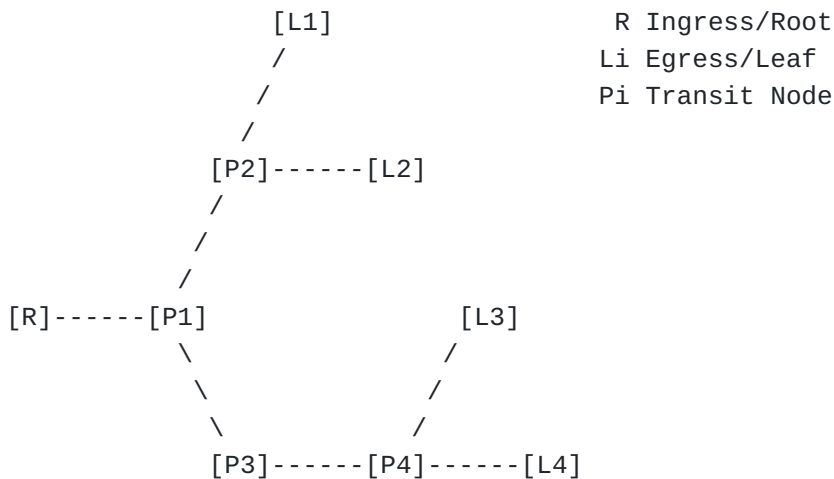


Figure 1: SR P2MP Path from R to L1, L2, L3 and L4

<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 transported by the P2MP path, it duplicates and sends the packet to the 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 of the nodes under node P1 is a value of another argument in P1-m, called N-SIDs. The value of N-SIDs in P1-m is 7, indicating that there are 7 SIDs following P1-m in the segment list.

There are 2 branches or next hops (i.e., L1 and L2) from node P2 and 2 SIDs (i.e., L1-m and L2-m) of the nodes under node P2. The values of N-Branched and N-SIDs in P2-m are 2 and 2. with this information, before sending the packet to node P2, node P1 pushes the SIDs under node P2 into the packet (i.e., the packet has a new segment list with the SIDs under node P2. The new segment list replaces the old one in the packet).

There are 1 branch or next hop (i.e., P4) from node P3 and 3 SIDs (i.e., P4-m, L3-m and L4-m) of the nodes under node P3. The values of N-Branched and N-SIDs in P3-m are 1 and 3 respectively. with this information, before sending the packet to node P3, node P1 pushes the SIDs under node P3 into the packet.

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 each sub-tree ST_i of a SR P2MP path from the ingress node of the P2MP path, suppose that

*the multicast SID of the next hop node NH_i is $mSID_i$;

*there are B_i branches (i.e., outgoing interfaces) to the next hop node BNH_j ($j = 1, \dots, B_i$) from node NH_i along the sub-tree, the multicast SID of BNH_j is $mSID_{ij}$;

*the number of branches (i.e., outgoing interfaces) under the node BNH_j ($j = 1, \dots, B_i$) is BB_j ; and the number of SIDs of the nodes under each of the B_i branches from node BNH_j is NS_j ($j = 1, \dots, B_i$).

Sub-tree ST_i is encoded as segment list

	< $mSID_i$,	$mSID_{i1}, \dots, mSID_{iB_i}$,	$SegSeq1, \dots, SegSeq_{B_i}$	> ,
	$\underbrace{\hspace{1cm}}$	$\underbrace{\hspace{2cm}}$	$\underbrace{\hspace{1cm}}$	$\underbrace{\hspace{1cm}}$
SIDs of	NH_i	B_i branches/next-hops	sub-tree	sub-tree
		BNH_j of node NH_i	from BNH_1	from BNH_{B_i}

where mSID_i contains the number of branches, B_i, in its N-Branches field, and the number of SIDs under mSID_i in its N-SIDs field; mSID_{ij} (j = 1, ..., B_i) contains the number of branches, BB_j, in its N-Branches field and the number of SIDs, NS_j, in its N-SIDs field; SegSeq_j (j = 1, ..., B_i) is the SID sequence in the segment list encoding the sub-trees from node BNH_j.

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.

For this sub-tree,

*the next hop node is P1 and the multicast SID of P1 is P1-m;

*there are 2 branches to the next hop nodes P2 and P3 from node P1 along the sub-tree; the number of SIDs of the nodes under P1 is 7; the multicast SIDs of P2 and P3 are P2-m and P3-m respectively;

*the numbers of SIDs of the nodes under these two branches are 2 and 3 respectively. The SIDs of the nodes under P2 are L1-m and L2-m. The SIDs of the nodes under P3 are P4-m, L3-m and L4-m.

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-tree	sub-tree
	P2 and P3 of node P1	from P2	from P3

where

P1-m's N-Branches field is set to 2 and its N-SIDs field to 7;
P2-m's N-Branches field is set to 2 and its N-SIDs field to 2;
P3-m's N-Branches field is set to 1 and its N-SIDs field to 3;

L1-m and L2-m are the SID sequence in the segment list encoding the sub-trees from P2;

P4-m, L3-m and L4-m are the SID sequence in the segment list encoding the sub-trees from P3; and

P4-m's N-Branches 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 P2MP multicast tree for the SR P2MP path from R to L1, L2, L3 and L4.

	N-Branches	N-SIDs		
P1's Multicast SID Locator	2	7	Arguments	P1
P2's Multicast SID Locator	2	2	Arguments	P2
P3's Multicast SID Locator	1	3	Arguments	P3
L1's Multicast SID Locator	0	0	Arguments	L1
L2's Multicast SID Locator	0	0	Arguments	L2
P4's Multicast SID Locator	2	2	Arguments	P4
L3's Multicast SID Locator	0	0	Arguments	L3
L4's Multicast SID Locator	0	0	Arguments	L4

Figure 2: Encoding of P2MP Multicast Tree from R to L1 - L4

SID P1-m indicates that there are 2 branches and 7 SIDs under P1.
 SID P2-m indicates that there are 2 branches and 2 SIDs under P2.
 SID P3-m indicates that there are 1 branch and 3 SIDs under P3. SIDs
 L1-m and L2-m indicate that there is no branch under them. SID P4-m
 indicates that there are 2 branches and 2 SIDs under P4. L3-m and
 L4-m indicate that there is no branch under them.

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 [[I-D.ietf-spring-srv6-network-programming](#)] 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 in 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:

$$\text{Pkt}' = (\text{SA}=\text{R}, \text{DA}=\text{P1-m}) \left(\underbrace{\text{L4-m, L3-m, ..., P3-m, P2-m}}_{\text{corresponds to: } \langle \text{P2-m, P3-m, ..., L3-m, L4-m} \rangle}; \text{SL}=7 \right) \text{Pkt0}$$

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 sub-trees under 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 n) of the DA indicates that there are n branches (or say sub-trees) under the transit node. The N-SIDs field of the DA indicates the number of SIDs for these n sub-trees under the transit node. The multicast SIDs of the next hop nodes of these n sub-trees are the first n 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 say sub-trees) under node P1. The N-SIDs field (which has value of 7) of the DA = P1-m indicates that there are 7 SIDs for these 2 sub-trees under node P1.

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

After the multicast SIDs of the next hop nodes, there are n blocks of SIDs for those n sub-trees. The N-SIDs field (which has value of B1) of the first multicast SID of the next hop nodes indicates that there are B1 SIDs in the first block for the first sub-tree; the N-SIDs field (which has value of B2) of the second multicast SID of the next hop nodes indicates that there are B2 SIDs in the second block for the second sub-tree after the first block; and so on.

For example, there are 2 blocks of SIDs for the 2 sub-trees under 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 2, indicating that there are 2 SIDs in the first block for the first sub-tree, which are L1-m and L2-m.

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 in the second block for the second sub-tree after the first block, which are P4-m, L3-m and L4-m.

The transit node duplicates the packet without top header for each sub-tree under it and adds a new header with a new segment list built from the SID block for the sub-tree to the duplicated packet by executing H.Encaps. It sets the DA of the packet to the multicast SID of the next hop node along the sub-tree and sends the packet to the DA.

For example, node P1 duplicates the packet for the first sub-tree towards L1 and L2 and adds a new header with a new segment list <L1-m, L2-m>. It sets DA = P2-m (multicast SID of next hop P2), and sends the packet to the DA (i.e., P2).

Suppose that the duplicated packet is Pkt0 for the sub-tree. The execution of H.Encaps pushes a new 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:

Pkt' = (SA=P1, DA=P2-m)(L2-m, L1-m; SL=2)Pkt0.

_____/

corresponds to: <L1-m, L2-m>

where DA=P2-m means that the destination address (DA) is set to P2-m; SA=P1 means that the source address (SA) is set to P1; SL=2 means that the number of Segments Left (SL) is 2.

Node P1 duplicates the packet for the second sub-tree via P3 towards L3 and L4 and adds a new header with a new segment list <P4-m, L3-m, L4-m>. It sets DA = P3-m (multicast SID of next hop P3), and sends the packet to the DA (i.e., P3).

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 a SID of the egress node. The egress node sends the packet to its destination accordingly. If the SID is a multicast SID of the egress, the N-Branched field and N-SIDs field are all zeros.

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 3](#). 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.

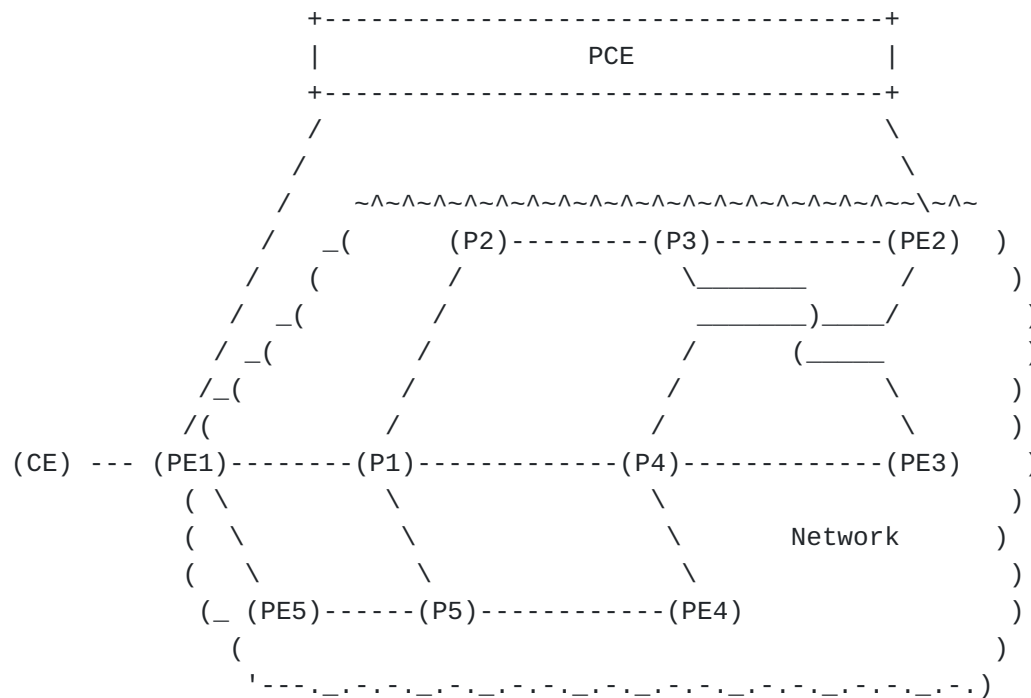


Figure 3: Architecture using PCE

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 ($i = 1, \dots, n$), a backup SR P2MP path from an ingress node $R1'$ to multiple egress nodes Li' ($i = 1, \dots, n$) 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 ($i = 1, \dots, n$), the failure of the egress nodes Li ($i = 1, \dots, n$) 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' ($i = 1, \dots, n$).

6.2. Local Protection

Local protection or say Fast Reroute (FRR) of a node and adjacency segment on 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 node and adjacency segment on 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 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)", Work in Progress, Internet-Draft, draft-ietf-6man-segment-routing-header-26, 22 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-6man-segment-routing-header-26.txt>>.

[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", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>>.

[I-D.ietf-rtgwg-srv6-egress-protection]

Hu, Z., Chen, H., Chen, H., Wu, P., Toy, M., Cao, C., He, T., Liu, L., and X. Liu, "SRv6 Path Egress Protection", Work in Progress, Internet-Draft, draft-ietf-rtgwg-srv6-egress-protection-04, 17 October 2021, <<https://>

www.ietf.org/archive/id/draft-ietf-rtgwg-srv6-egress-protection-04.txt>.

[I-D.ietf-spring-srv6-network-programming]

Filsfils, C., Garvia, P. C., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-network-programming-28, 29 December 2020, <<https://www.ietf.org/archive/id/draft-ietf-spring-srv6-network-programming-28.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>>.

[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] Filsfils, 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>>.

10.2. Informative References

[I-D.ietf-pim-sr-p2mp-policy] (editor), D. V., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", Work in Progress, Internet-Draft, draft-ietf-pim-sr-p2mp-policy-03, 23 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-pim-sr-p2mp-policy-03.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-05, 20 August

2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-replication-segment-05.txt>>.

[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", Work in Progress, Internet-Draft, draft-shen-spring-p2mp-transport-chain-04, 14 June 2021, <<https://www.ietf.org/archive/id/draft-shen-spring-p2mp-transport-chain-04.txt>>.

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 4](#). 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.

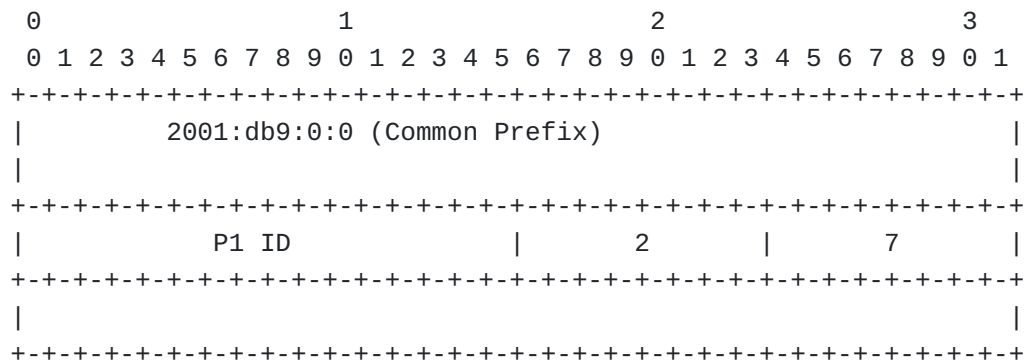


Figure 4: Destination Address (DA)

The IPv6 header is shown in [Figure 5](#). Ingress node R sends a packet with the IPv6 header to the DA.

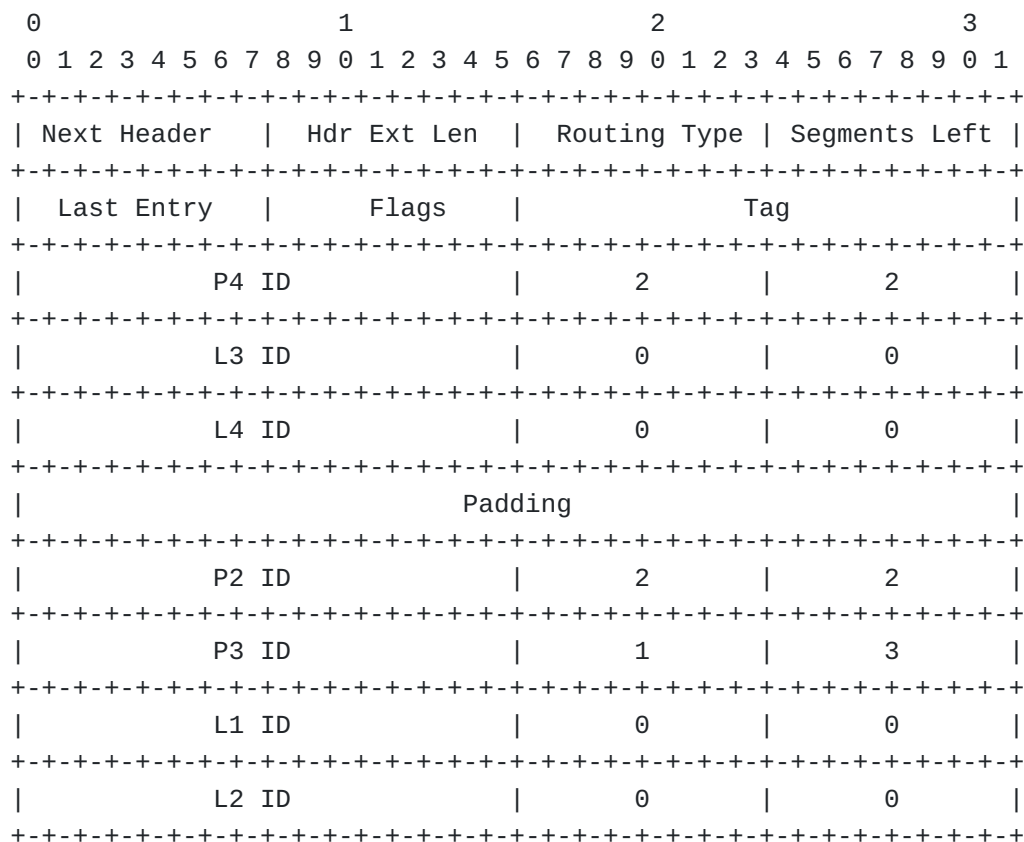


Figure 5: IPv6 Header

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Yanhe Fan
Casa Systems
United States of America

Email: yfan@casa-systems.com

Zhenbin Li
Huawei

Email: lizhenbin@huawei.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Mehmet Toy
Verizon
United States of America

Email: mehmet.toy@verizon.com

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

Email: wangaj3@chinatelecom.cn

Lei Liu
Fujitsu
United States of America

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks
McLean, VA
United States of America

Email: xufeng.liu.ietf@gmail.com