SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2020

R. Bonica
S. Hegde
Juniper Networks
Y. Kamite
NTT Communications Corporation
A. Alston
D. Henriques
Liquid Telecom
L. Jalil
Verizon
J. Halpern
Ericsson
J. Linkova
Google
G. Chen
Baidu

October 15, 2019

Segment Routing Mapped To IPv6 (SRm6)
draft-bonica-spring-srv6-plus-06

Abstract

This document describes Segment Routing mapped to IPv6 (SRm6). SRm6 is a Segment Routing (SR) solution that leverages IPv6. It supports a wide variety of use-cases while remaining in strict compliance with IPv6 specifications. SRm6 is optimized for ASIC-based forwarding devices that operate at high data rates.

Status of This Memo

Copyright Notice

Table of Contents

1. Overview

Network operators deploy Segment Routing (SR) [RFC8402] so that they can forward packets through SR paths. An SR path provides unidirectional connectivity from its ingress node to its egress node. While an SR path can follow the least cost path from ingress to egress, it can also follow any other path.

An SR path contains one or more segments. A segment provides unidirectional connectivity from its ingress node to its egress node. It includes a topological instruction that controls its behavior.

The topological instruction is executed on the segment ingress node. It determines the segment egress node and the method by which the segment ingress node forwards packets to the segment egress node.

Per-segment service instructions can augment a segment. Per-segment service instructions, if present, are executed on the segment egress node.

Likewise, a per-path service instruction can augment a path. The per-path service instruction, if present, is executed on the path egress node. Section 3 of this document illustrates the relationship between SR paths, segments and instructions.

A Segment Identifier (SID) identifies each segment. Because there is a one-to-one relationship between segments and the topological instructions that control them, the SID that identifies a segment also identifies the topological instruction that controls it.

A SID is different from the topological instruction that it identifies. While a SID identifies a topological instruction, it does not contain the topological instruction that it identifies. Therefore, a SID can be encoded in relatively few bits, while the topological instruction that it identifies may require many more bits for encoding.

An SR path can be represented by its ingress node as an ordered sequence of SIDs. In order to forward a packet through an SR path, the SR ingress node encodes the SR path into the packet as an ordered sequence of SIDs. It can also augment the packet with service instructions.

Because the SR ingress node is also the first segment ingress node, it executes the topological instruction associated with the first segment. This causes the packet to be forwarded to the first segment egress node. When the first segment egress node receives the packet, it executes any per-segment service instructions that augment the first segment.

If the SR path contains exactly one segment, the first segment egress node is also the path egress node. In this case, that node executes any per-path service instruction that augments the path, and SR forwarding is complete.

If the SR path contains multiple segments, the first segment egress node is also the second segment ingress node. In this case, that node executes the topological instruction associated with the second segment. The above-described procedure continues until the packet arrives at the SR egress node.

In the above-described procedure, only the SR ingress node maintains path information. Segment ingress and egress nodes maintain information regarding the segments in which they participate, but they do not maintain path information.

The SR architecture, described above, can leverage either an MPLS [RFC3031] data plane or an IPv6 [RFC8200] data plane. SR-MPLS [I-D.ietf-spring-segment-routing-mpls] leverages MPLS. SRv6 [I-D.ietf-spring-srv6-network-programming] [I-D.ietf-6man-segment-routing-header] leverages IPv6.

This document describes Segment Routing mapped to IPv6 (SRm6). SRm6 is an SR variant that leverages IPv6. It supports a wide variety of use-cases while remaining in strict compliance with IPv6 specifications. SRm6 is optimized for ASIC-based forwarding devices that operate at high data rates. Section 9 of this document highlights differences between SRv6 and SRm6.

2. 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.

3. Paths, Segments And Instructions

An SRm6 path is determined by the segments that it contains. It can be represented by its ingress node as an ordered sequence of SIDs.

A segment is determined by its ingress node and by the topological instruction that controls its behavior. The topological instruction determines the segment egress node and the method by which the segment ingress node forwards packets to the segment egress node.

Per-segment service instructions augment, but do not determine, segments. A segment ingress node can:

- Send one packet through a segment with one per-segment service instruction.

- Send another packet through the same segment with a different per-segment service instruction.

- Send another packet through the same segment without any per-segment service instructions.

Likewise, per-path service instructions augment, but do not determine, paths.

```
     ----      ----      ----      ----      ----      ----
    |Node|----|Node|----|Node|----|Node|----|Node|----|Node|
    | A  |    | B  |    | C  |    | D  |    | E  |    | F  |
     ----      ----      ----      ----      ----      ----
       |                   |         |                   |
        -------------------|         |-------------------|
           Segment A-C     |---------|    Segment D-F
                           Segment C-D
       |                                                 |
        -------------------------------------------------
                             SRm6 Path
```

Figure 1: Paths, Segments And Instructions

Figure 1 depicts an SRm6 path. The path provides unidirectional connectivity from its ingress node (i.e., Node A) to its egress node (i.e., Node F). It contains Segment A-C, Segment C-D and Segment D-F.

In Segment A-C, Node A is the ingress node, Node B is a transit node, and Node C is the egress node. Therefore, the topological instruction that controls the segment is executed on Node A, while per-segment service instructions that augment the segment (if any exist) are executed on Node C.

In Segment C-D, Node C is the ingress node and Node D is the egress node. Therefore, the topological instruction that controls the segment is executed on Node C, while per-segment service instructions that augment the segment (if any exist) are executed on Node D.

In Segment D-F, Node D is the ingress node, Node E is a transit node, and Node F is the egress node. Therefore, the topological instruction that controls the segment is executed on Node D, while per-segment service instructions that augment the segment (if any exist) are executed on Node F.

Node F is also the path egress node. Therefore, if a per-path service instruction augments the path, it is executed on Node F.

Segments A-C, C-D and D-F are also contained by other paths that are not included in the figure.

4. Segment Types

SRm6 supports the following segment types:

- Adjacency.

- Node.

- Binding.

Adjacency segments forward packets through a specified link that connects the segment ingress node to the segment egress node. Node segments forward packets through the least cost path from the segment ingress node to the segment egress node. Binding segments facilitate recursive application of SRm6. They cause SRm6 paths to be nested in a hierarchy.

Each segment type is described below.

4.1. Adjacency Segments

When a packet is submitted to an adjacency segment, the topological instruction associated with that segment operates upon the packet. The topological instruction executes on the segment ingress node and receives the following parameters:

- An IPv6 address that identifies an interface on the segment egress node.

- An interface identifier.

The topological instruction behaves as follows:

- If the interface that was received as a parameter is not operational, discard the packet and send an ICMPv6 [RFC4443] Destination Unreachable message (Code: 5, Source Route Failed) to the packet's source node.

- Overwrite the packet's Destination Address with the IPv6 address that was received as a parameter.

- Forward the packet through the above-mentioned interface.

For further processing details, see [I-D.bonica-6man-comp-rtg-hdr].

4.2. Node Segments

When a packet is submitted to a node segment, the topological instruction associated with that segment operates upon the packet. The topological instruction executes on the segment ingress node and receives an IPv6 address as a parameter. The IPv6 address identifies an interface on the segment egress node.

The topological instruction behaves as follows:

- If the segment ingress node does not have a viable route to the IPv6 address received as a parameter, discard the packet and send an ICMPv6 Destination Unreachable message (Code:1 Net Unreachable) to the packet's source node.

- Overwrite the packet's Destination Address with the destination address that was received as a parameter.

- Forward the packet to the next hop along the least cost path to the segment egress node. If there are multiple least cost paths to the segment egress node (i.e., Equal Cost Multipath), execute procedures so that all packets belonging to a flow are forwarded through the same next hop.

For further processing details, see [I-D.bonica-6man-comp-rtg-hdr].

4.3. Binding Segments

When a packet is submitted to a binding segment, the topological instruction associated with that segment operates upon the packet. The topological instruction executes on the segment ingress node and receives the following parameters:

- An IPv6 address.

- A SID list length.

- A SID list.

The topological instruction behaves as follows:

- If the segment ingress node does not have a viable route to the IPv6 address received as a parameter, discard the packet and send an ICMPv6 Destination Unreachable message (Code:1 Net Unreachable) to the packet’s source node.

- Prepend a Compressed Routing Header (CRH) [I-D.bonica-6man-comp-rtg-hdr] to the packet. Copy the SID list length, received as a parameter, to the CRH Segments Left field. Also copy the SID list, received as a parameter, to the CRH SID list.

- Prepend an IPv6 header to the packet. Copy the IPv6 address, received as a parameter, to the IPv6 Destination Address.

- Forward the packet to the next hop along the least cost path to the IPv6 address received as a parameter. If there are multiple least cost paths to the IPv6 address received as a parameter (i.e., Equal Cost Multipath), execute procedures so that all packets belonging to a flow are forwarded through the same next hop.

For further processing details, see [I-D.bonica-6man-comp-rtg-hdr].

5. Segment Identifiers (SID)

A Segment Identifier (SID) is an unsigned integer that identifies a segment. Because there is a one-to-one relationship between segments and the topological instructions that control them, the SID that identifies a segment also identifies the topological instruction that controls it.

A SID is different from the topological instruction that it identifies. While a SID identifies a topological instruction, it does not contain the topological instruction that it identifies. Therefore, a SID can be encoded in relatively few bits, while the topological instruction that it identifies may require many more bits for encoding.

SIDs have node-local significance. This means that a segment ingress node MUST identify each segment that it originates with a unique SID.

However, a SID that is used by one segment ingress node to identify a segment that it originates can be used by another segment ingress node to identify another segment. For example, SID S can identify both of the following:

- A segment whose ingress is Node A and whose egress is Node Z.

- Another segment whose ingress is Node B and whose egress is also node Z.

Although SIDs have node-local significance, an SRm6 path can be uniquely identified by its ingress node and an ordered sequence of SIDs. This is because the topological instruction associated with each segment determines the ingress node of the next segment (i.e., the node upon which the next SID has significance.)

SIDs can be assigned in a manner that simplifies network operations. See Section 5.2 and Section 5.3 for details.

5.1. Range

SID values range from 0 to a configurable Maximum SID Value (MSV). The values 0 through 15 are reserved for future use. The following are valid MSVs:

- 65,535 (i.e., 2**16 minus 1).

- 4,294,967,295 (i.e., 2**32 minus 1).

In order to optimize packet encoding (Section 7.1), network operators can configure all nodes within an SRm6 domain to have the smallest feasible MSV. The following paragraphs explain how an operator determines the smallest feasible MSV.

Consider an SRm6 domain that contains 5,000 nodes connected to one another by point-to-point infrastructure links. The network topology is not a full-mesh. In fact, each node supports 200 point-to-point infrastructure links or fewer. Given this SRm6 domain, we will determine the smallest feasible MSV under the following conditions:

- The SRm6 domain contains adjacency segments only.

- The SRm6 domain contains node segments only.

- The SRm6 domain contains both adjacency and node segments.

If an SRm6 domain contains adjacency segments only, and each node creates a adjacency segment to each of its neighbors, each node will create 200 segments or fewer and consume 200 SIDs or fewer. This is because each node has 200 neighbors or fewer. Because SIDs have node-local significance (i.e., they can be reused across nodes), the smallest feasible MSV is 65,535.

Adding nodes to this SRm6 domain will not increase the smallest feasible MSV, so long as each node continues to support 65,519 point-to-point infrastructure links or fewer. If a single node is added to the domain and that node supports 65,520 infrastructure links, the smallest feasible MSV will increase to 4,294,967,295.

If an SRm6 domain contains node segments only, and every node creates a node segment to every other node, every node will create 4,999 segments and consume 4,999 SIDs. This is because the domain contains 5,000 nodes. Because SIDs have node-local significance (i.e., they can be reused across nodes), the smallest feasible MSV is 65,535.

Adding nodes to this SRm6 domain will not increase the smallest feasible MSV until the number of nodes exceeds 65,519. When the smallest feasible MSV increases, it becomes 4,294,967,295.

If an SRm6 domain contains both adjacency and node segments, each node will create 5,199 segments or fewer and consume 5,199 SIDs or fewer. This value is the sum of the following:

- The number of node segments that each node will create, given that every node creates a node segment to every other node (i.e., 4,999).

- The number of adjacency segments that each node will create, given that each node creates a adjacency segment to each of its neighbors (i.e., 200 or fewer).

Because SIDs have node-local significance (i.e., they can be reused across nodes), the smallest feasible MSV is 65,535.

Adding nodes to this SRm6 domain will not increase the smallest feasible MSV until the number of nodes plus the maximum number of infrastructure links per node exceeds 65,519. When the smallest feasible MSV increases, it becomes 4,294,967,295.

5.2. Assigning SIDs to Adjacency Segments

Network operators can establish conventions by which they assign SIDs to adjacency segments. These conventions can simplify network operations.

For example, a network operator can reserved a range of SIDs for adjacency segments. It can further divide that range into subranges, so that all segments sharing a common egress node are identified by SIDs from the same subrange.

5.3. Assigning SIDs to Node Segments

In order to simplify network operations, all node segments that share a common egress node are identified by the same SID. In order to maintain this discipline, network wide co-ordination is required.

For example, assume that an SRm6 domain contains N nodes. Network administrators reserve a block of N SIDs and configure one of those SIDs on each node. Each node advertises its SID into the control plane. When another node receives that advertisement, it creates a node segment between itself and the advertising node. It also associates the SID that it received in the advertisement with the newly created segment. See [I-D.bonica-lsr-crh-isis-extensions] for details.

5.4. Assigning SIDs to Binding Segments

Network operators can establish conventions by which they assign SIDs to binding segments. These conventions can simplify network operations.

For example, a network operator can reserve a range of SIDs for binding segments. It can further divide that range into subranges, so that all segments sharing a common egress node are identified by SIDs from the same subrange.

6. Service Instructions

SRm6 supports the following service instruction types:

- Per-segment.

- Per-path.

Each is described below.

6.1. Per-Segment

Per-segment service instructions can augment a segment. Per-segment service instructions, if present, are executed on the segment egress node. Because the path egress node is also a segment egress node, it can execute per-segment service instructions.

The following are examples of per-segment service instructions:

- Expose a packet to a firewall policy.

- Expose a packet to a sampling policy.

Per-segment Service Instruction Identifiers identify a set of service instructions. Per-segment Service Instruction Identifiers are allocated and distributed by a controller. They have domain-wide significance.

6.2. Per-Path

A per-path service instruction can augment a path. The per-path service instruction, if present, is executed on the path egress node.

The following are examples of per-path service instructions:

- De-encapsulate a packet and forward its newly exposed payload through a specified interface.

- De-encapsulate a packet and forward its newly exposed payload using a specified routing table.

Per-path Service Instruction Identifiers identify per-path service instructions. Per-path Service Instruction Identifiers are allocated and distributed by the processing node (i.e., the path egress node). They have node-local significance. This means that the path egress node MUST allocate a unique Per-path Service Instruction Identifier for each per-path service instruction that it instantiates.

7. The IPv6 Data Plane

SRm6 ingress nodes generate IPv6 header chains that represent SRm6 paths. An IPv6 header chain contains an IPv6 header. It can also contain one or more extension headers.

An extension header chain that represents an SRm6 path can contain any valid combination of IPv6 extension headers. The following bullet points describe how SRm6 leverages IPv6 extension headers:

- If an SRm6 path contains multiple segments, the IPv6 header chain that represents it MUST contain a Routing header. The SRm6 path MUST be encoded in the Routing header as an ordered sequence of SIDs.

- If an SRm6 path is augmented by a per-path service instruction, the IPv6 header chain that represents it MUST contain a

Destination Options header. The Destination Options header MUST immediately precede an upper-layer header and it MUST include a Per-Path Service Instruction Identifier.

- If an SRm6 path contains a segment that is augmented by a per-segment service instruction, the IPv6 chain that represents it MUST contain a Routing header and a Destination Options header. The Destination Options header MUST immediately precede a Routing header and it MUST include the Per-Segment Service Instruction Identifier.

The following subsections describe how SRm6 uses the Routing header and the Destination Options header.

7.1. The Routing Header

SRm6 defines two new Routing header types. Generically, they are called the Compressed Routing Header (CRH) [I-D.bonica-6man-comp-rtg-hdr]. More specifically, the 16-bit version of the CRH is called the CRH-16, while the 32-bit version of the CRH is called the CRH-32.

Both CRH versions contain the following fields:

- Next Header - Identifies the header immediately following the CRH.

- Hdr Ext Len - Length of the CRH.

- Routing Type - Identifies the Routing header variant (i.e., CRH-16 or CRH-32).

- Segments Left - The number of segments still to be traversed before reaching the path egress node.

- SID List - Represents the SRm6 path as an ordered list of SIDs. SIDs are listed in reverse order, with SID[0] representing the final segment, SID[1] representing the penultimate segment, and so forth. SIDs are listed in reverse order so that Segments Left can be used as an index to the SID List. The SID indexed by Segments Left is called the current SID.

In the CRH-16, each SID list entry is encoded in 16-bits. In the CRH-32, each SID list entry is encoded in 32-bits. In networks where the smallest feasible MSV (Section 5.1) is greater than 65,635, CRH-32 is required. Otherwise, CRH-16 is preferred.

As per [RFC8200], when an IPv6 node receives a packet, it examines the packet's destination address. If the destination address represents an interface belonging to the node, the node processes the next header. If the node encounters and recognizes the CRH, it processes the CRH as follows:

- If Segments Left equal 0, skip over the CRH and process the next header in the packet.

- Decrement Segments Left.

- Search for the current SID in a local table that maps SID's to topological instructions. If the current SID cannot be found in that table, send an ICMPv6 Parameter Problem message to the packet's Source Address and discard the packet.

- Execute the topological instruction found in the table as described in Section 4. This causes the packet to be forwarded to the segment egress node.

When the packet arrives at the segment egress node, the above-described procedure is repeated. For further processing details, see [I-D.bonica-6man-comp-rtg-hdr].

7.2. The Destination Options Header

According to [RFC8200], the Destination Options header contains one or more IPv6 options. It can occur twice within a packet, once before a Routing header and once before an upper-layer header. The Destination Options header that occurs before a Routing header is processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations that are listed in the Routing header. The Destination Options header that occurs before an upper-layer header is processed by the packet's final destination only.

Therefore, SRm6 defines the following new IPv6 options:

- The SRm6 Per-Segment Service Instruction Option [I-D.bonica-6man-seg-end-opt]

- The SRm6 Per-Path Service Instruction Option [I-D.bonica-6man-vpn-dest-opt]

The SRm6 Per-Segment Service Instruction Option is encoded in a Destination Options header that precedes the CRH. Therefore, it is processed by every segment egress node. It includes a Per-Segment Service Instruction Identifier and causes segment egress nodes to execute per-segment service instructions.

The SRm6 Per-Path Service Instruction Option is encoded in a Destination Options header that precedes the upper-layer header. Therefore, it is processed by the path egress node only. It includes a Per-Path Service Instruction Identifier and causes the path egress node to execute a per-path service instruction.

8. Control Plane

IS-IS extensions [I-D.bonica-lsr-crh-isis-extensions] have been defined for the following purposes:

- So that SRm6 segment ingress nodes can flood information regarding adjacency segments that they originate.

- So that SRm6 segment egress nodes can flood information regarding node segments that they terminate.

BGP extensions [I-D.ssangli-idr-bgp-vpn-srv6-plus] are defined so that SRm6 path egress nodes can associate path-terminating service instructions with Network Layer Reachability Information (NLRI). Additional BGP extensions [I-D.alston-spring-crh-bgp-signalling] are defined so that SIDs can be mapped to the IPv6 addresses that they represent.

9. Differences Between SRv6 and SRv6+

9.1. Routing Header Size

SRv6 defines a Routing header type, called the Segment Routing Header (SRH). The SRH contains a field that represents the SRv6 path as an ordered sequence of SIDs. Each SID contained by that field is 128 bits long.

Likewise, SRm6 defines two Routing Header Types, called CRH-16 and CRH-32. Both contain a field that represents the SRv6 path as an ordered sequence of SIDs. In the CRH-16, each SID is 16 bits long. In the CRH-32, each SID is 32 bits long.

SIDs	SRv6 SRH (128-bit SID)	SRm6 CRH-16	SRm6 CRH-32
1	24	8	8
2	40	8	16
3	56	16	16
4	72	16	24
5	88	16	24
6	104	16	32
7	120	24	32
8	136	24	40
9	152	24	40
10	168	24	N/A
11	184	32	N/A
12	200	32	N/A
13	216	32	N/A
14	232	32	N/A
15	248	40	N/A
16	264	40	N/A
17	280	40	N/A
18	296	40	N/A

Table 1: Routing Header Size (in Bytes) As A Function Of Routing Header Type and Number Of SIDs

Table 1 reflects Routing header size as a function of Routing header type and number of SIDs contained by the Routing header. Due to their relative immaturity, [I-D.filsfils-spring-net-pgm-extension-srv6-usid], [I-D.li-spring-compressed-srv6-np] and [I-D.mirsky-6man-unified-id-sr] are omitted from this analysis.

Large Routing headers are undesirable for the following reasons:

- Many ASIC-based forwarders copy the entire IPv6 extension header chain from buffer memory to on-chip memory. As the size of the IPv6 extension header chain increases, so does the cost of this copy.

- Because Path MTU Discovery (PMTUD) [RFC8201] is not entirely reliable, many IPv6 hosts refrain from sending packets larger than the IPv6 minimum link MTU (i.e., 1280 bytes). When packets are small, the overhead imposed by large Routing headers becomes pronounced.

9.2. Decoupling of Topological and Service Instructions

SRm6 decouples topological instructions from service instructions. Topological instructions are invoked at the segment ingress node, as a result of CRH processing, while service instructions are invoked at the segment egress node, as a result of Destination Option processing. Therefore, network operators can use SRm6 mechanisms to support topological instructions, service instructions, or both.

```
                  ----------           ----------          ----------
                 | Ethernet |         | Ethernet |        | Ethernet |
                  ----------           ----------          ----------
    Service      |  VXLAN   |         |  Dest    |        |   Dest   |
    Instruction   ----------          |          |        |          |
                 |   UDP    |         |  Option  |        |  Option  |
                  ----------           ----------          ----------
    Topological  |                    |          |        |          |
    Instructions |   CRH    |         |          |        |    CRH   |
                  ----------          |          |         ----------
                 |   IPv6   |         |   IPv6   |        |   IPv6   |
                  ----------           ----------          ----------

                  Option 1             Option 2             Option 3
```

Figure 2: EVPN Design Alternatives

Figure 2 illustrates this point by depicting design options available to network operators offering Ethernet Virtual Private Network [RFC7432] services over Virtual eXtensible Local Area Network (VXLAN) [RFC7348]. In Option 1, the network operator encodes topological instructions in the CRH, while encoding service instructions in a VXLAN header. In Option 2, the network operator encodes service instructions in a Destination Options header, while allowing traffic to traverse the least cost path between the ingress and egress Provider Edge (PE) routers. In Option 3, the network operator encodes topological instructions in the CRH, and encodes service instructions in a Destination Options header.

9.3. Authentication

The IPv6 Authentication Header (AH) [RFC4302] can be used to authenticate SRm6 packets. However, AH processing is not defined in SRv6.

9.4. Traffic Engineering Capability

SRm6 supports traffic engineering solutions that rely exclusively upon adjacency segments. For example, consider an SRm6 network whose diameter is 12 hops and whose minimum feasible MSV is 65,525. In that network, in the worst case, SRm6 overhead is 72 bytes (i.e., a 40-byte IPv6 header and a 32-byte CRH-16).

SRv6 also supports traffic engineering solutions that rely exclusively upon adjacency segments (i.e., END.X SIDs). However, SRv6 overhead may be prohibitive. For example, consider an SRv6 network whose diameter is 12 hops. In the worst case, SRv6 overhead is 240 bytes (i.e., a 40 byte IPv6 header and a 200-byte SRH).

9.5. IP Addressing Architecture

In SRv6, an IPv6 address can represent either of the following:

- A network interface

- An instruction instantiated on a node (i.e., an SRv6 SID)

In SRm6 an IPv6 address always represents a network interface, as per [RFC4291].

10. Compliance

In order to be compliant with this specification, an SRm6 implementation MUST:

- Be able to process IPv6 options as described in Section 4.2 of [RFC8200].

- Be able to process the Routing header as described in Section 4.4 of [RFC8200].

- Be able to process the Destination Options header as described in Section 4.6 of [RFC8200].

- Support the CRH-16 and the CRH-32

Additionally, an SRm6 implementation MAY:

- Recognize the Per-Segment Service Instruction Option.

- Recognize the Per-Path Service Instruction Option.

11. Operational Considerations

11.1. Ping and Traceroute

Ping and Traceroute [RFC2151] both operate correctly in SRm6 (i.e., in the presence of the CRH).

11.2. ICMPv6 Rate Limiting

As per [RFC4443], SRm6 nodes rate limit the ICMPv6 messages that they emit.

12. IANA Considerations

SID values 0-15 are reserved for future use. They may be assigned by IANA, based on IETF Consensus.

IANA is requested to establish a "Registry of SRm6 Reserved SIDs". Values 0-15 are reserved for future use.

13. Security Considerations

SRm6 domains MUST NOT span security domains. In order to enforce this requirement, security domain edge routers MUST do one of the following:

- Discard all inbound SRm6 packets whose IPv6 destination address represents domain infrastructure.

- Authenticate [RFC4302] [RFC4303] all inbound SRm6 packets whose IPv6 destination address represents domain infrastructure.

14. Acknowledgements

The authors wish to acknowledge Dr. Vanessa Ameen, Reji Thomas, Parag Kaneriya, Rejesh Shetty, Nancy Shaw, and John Scudder.

15. References

15.1. Normative References

[I-D.alston-spring-crh-bgp-signalling]
Alston, A., Henriques, D., and R. Bonica, "BGP Extensions for IPv6 Compressed Routing Header (CRH)", draft-alston-spring-crh-bgp-signalling-01 (work in progress), July 2019.

[I-D.bonica-6man-comp-rtg-hdr]
Bonica, R., Kamite, Y., Niwa, T., Alston, A., Henriques, D., So, N., Xu, F., Chen, G., Zhu, Y., Yang, G., and Y. Zhou, "The IPv6 Compressed Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-07 (work in progress), September 2019.

[I-D.bonica-6man-seg-end-opt]
Bonica, R., Halpern, J., Kamite, Y., Niwa, T., So, N., Xu, F., Chen, G., Zhu, Y., Yang, G., and Y. Zhou, "The Per-Segment Service Instruction (PSSI) Option", draft-bonica-6man-seg-end-opt-04 (work in progress), July 2019.

[I-D.bonica-6man-vpn-dest-opt]
Bonica, R., Kamite, Y., Lenart, C., So, N., Xu, F., Presbury, G., Chen, G., Zhu, Y., Yang, G., and Y. Zhou, "The Per-Path Service Instruction (PPSI) Option", draft-bonica-6man-vpn-dest-opt-06 (work in progress), July 2019.

[I-D.bonica-lsr-crh-isis-extensions]
Kaneriya, P., Shetty, R., Hegde, S., and R. Bonica, "IS-IS Extensions To Support The IPv6 Compressed Routing Header (CRH)", draft-bonica-lsr-crh-isis-extensions-00 (work in progress), May 2019.

[I-D.ssangli-idr-bgp-vpn-srv6-plus]
Ramachandra, S. and R. Bonica, "BGP based Virtual Private Network (VPN) Services over SRv6+ enabled IPv6 networks", draft-ssangli-idr-bgp-vpn-srv6-plus-02 (work in progress), July 2019.

[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>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[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>.

15.2. Informative References

[I-D.filsfils-spring-net-pgm-extension-srv6-usid] Filsfils, C., Camarillo, P., Cai, D., Jiang, Z., daniel.voyer@bell.ca, d., Shawky, A., Leymann, N., Steinberg, D., Zandi, S., Dawra, G., Meilik, I., Uttaro, J., Jalil, L., So, N., Fiumano, M., and M. Khaddam, "Network Programming extension: SRv6 uSID instruction", draft-filsfils-spring-net-pgm-extension-srv6-usid-02 (work in progress), August 2019.

[I-D.ietf-6man-segment-routing-header] Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-24 (work in progress), October 2019.

[I-D.ietf-spring-segment-routing-mpls] Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-04 (work in progress), October 2019.

[I-D.li-spring-compressed-srv6-np] Li, Z., Li, C., Peng, S., Wang, Z., and B. Liu, "Compressed SRv6 Network Programming", draft-li-spring-compressed-srv6-np-00 (work in progress), July 2019.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., Wan, X., and C. Wei, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-03 (work in progress), July 2019.

[RFC2151] Kessler, G. and S. Shepard, "A Primer On Internet and TCP/ IP Tools and Utilities", FYI 30, RFC 2151, DOI 10.17487/RFC2151, June 1997, <https://www.rfc-editor.org/info/rfc2151>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

Authors' Addresses

Ron Bonica
Juniper Networks
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Shraddha Hegde
Juniper Networks
Embassy Business Park
Bangalore, KA 560093
India

Email: shraddha@juniper.net

Yuji Kamite
NTT Communications Corporation
3-4-1 Shibaura, Minato-ku
Tokyo 108-8118
Japan

Email: y.kamite@ntt.com

Andrew Alston
Liquid Telecom
Nairobi
Kenya

Email: Andrew.Alston@liquidtelecom.com

Daniam Henriques
Liquid Telecom
Johannesburg
South Africa

Email: daniam.henriques@liquidtelecom.com

Luay Jalil
Verizon
Richardson, Texas
USA

Email: luay.jalil@one.verizon.com

Joel Halpern
Ericsson
P. O. Box 6049
Leesburg, Virginia 20178
USA

Email: joel.halpern@ericsson.com

Jen Linkova
Google
Mountain View, California 94043
USA

Email: furry@google.com

Gang Chen
Baidu
No.10 Xibeiwang East Road Haidian District
Beijing 100193
P.R. China

Email: phdgang@gmail.com