SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 7, 2021

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 4, 2020

Segment Routing Mapped To IPv6 (SRm6)
draft-bonica-spring-sr-mapped-six-02

Abstract

This document describes Segment Routing Mapped to IPv6 (SRm6). SRm6 is a Segment Routing (SR) solution that supports a wide variety of use-cases while complying with IPv6 specifications. SRm6 is optimized for ASIC-based routers 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 they can forward packets through SR paths. An SR path provides 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 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 sends packets to the segment egress node.

SR nodes can also execute service instructions. Segment egress nodes execute Per Segment Service Instructions (PSSI). Likewise, path egress nodes execute Per Path Service Instructions (PPSI). (Section 2) of this document describes 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 shorter than the topological instruction that it identifies. While a SID is 16 or 32 bits long, the topological instruction that it identifies is at least 128 bits long.

To forward a packet through an SR path, the SR ingress node encodes a list of SIDs in the packet header. It can also encode service instructions in the packet header.

Because the SR ingress node is also the first segment ingress node, it executes the first segment's topological instruction and sends the packet to the first segment egress node. When the first segment egress node receives the packet, it executes the first segment's PSSI, if one is present.

If the SR path contains only one segment, the first segment egress node is also the path egress node. In this case, that node executes the PPSI, if one is present.

If the SR path contains multiple segments, the first segment's egress node is also the second segment's ingress node. In this case, that node executes the second segment's topological instruction. This

procedure continues until the packet arrives at the path egress node. If the packet includes a PPSI, the path egress node executes it.

In SR, only the path ingress node maintains path information. The segment ingress node maintains a topological instruction, but it does not maintain path information unless it is also a path ingress node.

SR can use either an MPLS [RFC3031] data plane or an IPv6 [RFC8200] data plane. SR-MPLS [RFC8660] uses MPLS. SRv6 [I-D.ietf-spring-srv6-network-programming] uses IPv6.

This document describes Segment Routing Mapped to IPv6 (SRm6). SRm6 is an SR solution that also uses IPv6. It supports a wide variety of use-cases while adhering to IPv6 specifications.

Section 8 of this document describes the differences between SRv6 and SRm6.

2. Paths, Segments And Instructions

```
     ----      ----      ----      ----      ----      ----
    |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 connectivity from its ingress node (i.e., Node A) to its egress node (i.e., Node F). It contains Segments A-C, C-D and 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. So, Node A executes the segment's topological instruction. If the packet includes a PSSI for the segment, Node C executes it.

In Segment C-D, Node C is the ingress node and Node D is the egress node. So, Node C executes the segment's topological instruction. If the packet includes a PSSI for the segment, Node D executes it.

In Segment D-F, Node D is the ingress node, Node E is a transit node, and Node F is the egress node. So, Node D executes the segment's topological instruction. If the packet includes a PSSI for the segment, Node F executes it.

Node F is also the path egress node. So, if the packet includes a PPSI, Node F executes it.

Other paths that are not included in the figure also include Segments A-C, C-D, and D-F.

3. Topological Instructions

SRm6 supports the following topological instruction types:

- Adjacency.

- Node.

- Binding.

3.1. Adjacency Instructions

Adjacency instructions send packets through a single link that connects the segment ingress node to the segment egress node. An adjacency instruction includes the following information:

- SE-ADDR: The IPv6 address of an interface on the segment egress node.

- IFACE: An interface identifier.

The instruction behaves as follows:

- If the interface identified by IFACE is not operational, discard the packet and send an ICMPv6 [RFC4443] Destination Unreachable message to the packet's source node.

- Overwrite the packet's Destination Address with SE-ADDR.

- Send the packet through the interface identified by IFACE.

3.2. Node Instructions

Node instructions send packets through the least-cost path from the segment ingress node to the segment egress node. A node instruction includes an SE-ADDR. The SE-ADDR is the IPv6 address of an interface on the segment egress node.

The instruction behaves as follows:

- o If the segment ingress node does not have a viable route to SE-ADDR, discard the packet and send an ICMPv6 Destination Unreachable message to the packet's source node.

- o Overwrite the packet's Destination Address with SE-ADDR.

- o Send the packet to the next-hop along the least-cost path to SE-ADDR.

3.3. Binding Instructions

Binding instructions send packets through tunnels that connect the segment ingress node to the segment egress node. Because the tunnel is also an SRm6 path, it is called an SRm6 tunnel.

A binding instruction includes the following information:

- o SE-ADDR: The IPv6 address of an interface on the segment egress node.

- o Tunnel-SID-List: A SID list that determines the path of the SRm6 tunnel.

The instruction behaves as follows:

- o Overwrite the packet's Destination Address with SE-ADDR.

- o Prepend an SRm6 tunnel header to the packet. The SRm6 tunnel header includes the Tunnel-SID-List.

- o If the SRm6 tunnel is not operational, discard the packet and send an ICMPv6 Destination Unreachable message to the packet's source node.

- o Send the packet through the SRm6 tunnel.

4. Service Instructions

SRm6 supports the following service instruction types:

- o Per-Segment Service Instructions (PSSI).

- o Per-Path Service Instructions (PPSI).

4.1. PSSI

The PSSI, if present, is executed on the segment egress node. Because the path egress node is also a segment egress node, it can execute a PSSI.

The following are examples of PSSIs:

- Expose a packet to a firewall policy.

- Expose a packet to a sampling policy.

4.2. PPSI

A PPSI, if present, is executed on the path egress node.

The following are examples of PPSIs:

- 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.

5. Segment Identifiers (SID)

A Segment Identifier (SID) is an unsigned integer that identifies a segment. It can be either 16 or 32 bits long. 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 shorter than the topological instruction that it identifies. While a SID is 16 or 32 bits long, the topological instruction that it identifies is at least 128 bits long.

SIDs have node-local significance. This means that a segment ingress node identifies each segment that it originates with a unique SID. However, a single SID value can be used to identify:

- 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.

A single SID value can identify both segments because they originate on different nodes.

SID values 0 through 15 are reserved for future use.

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

5.1. 16-Bit SIDs Versus 32-Bit SIDs

The maximum 16-bit SID value is 65,535. Because SIDs 0 through 15 are reserved for future use, a 16-bit SID offers 65,520 usable values.

The maximum 32-bit SID value is 4,294,967,295. Because SIDs 0 through 15 are reserved for future use, a 32-bit SID offers 4,294,967,280 usable values.

To minimize packet length, network operators should use 16-bit SIDs whenever possible. However, when more than 65,520 SIDs are required, network operators must use 32-bit SIDs.

Consider a network that contains 60,000 nodes. Each node connects to 200 or fewer neighbors through point-to-point links. In this network, we will determine how many SIDs are required under the following conditions:

- If the network contains adjacency segments only.

- If the network contains node segments only.

- If the network contains both adjacency and node segments.

If the network contains adjacency segments only, and each node originates an adjacency segment to each of its neighbors, each node originates 200 segments or fewer. Because SIDs have node-local significance (i.e., they can be reused across nodes), the network requires only 200 SIDs. The network operator can use 16-bit SIDs, so long as each node supports 65,520 point-to-point links or fewer.

If the network contains node segments only, and every node originates a node segment to every other node, every node originates 59,999 segments. Because SIDs have node-local significance, the network requires only 59,999 SIDs. The network operator can use 16-bit SIDs so long as the number of nodes is less than 65,520.

If the network contains both adjacency and node segments, each node will originate 60,199 segments or fewer. This value is the sum of:

- The number of node segments that each node originates (i.e., 59,999).

- The number of adjacency segments that each node originates (i.e., 200 or fewer).

Because SIDs have node-local significance, only 60,199 SIDs are required. The network operator can use 16-bit SIDs so long as the number of nodes plus the maximum number of links per node is less than 65,520.

5.2. Assigning SIDs

Network operators can establish conventions for assigning SIDs to segments. These conventions can simplify network operations.

For example, a network operator who uses 16-bit SIDs can:

- Assign SIDs 16 - 1000 to adjacency segments

- Assign SIDs 1001 - 62,000 to node segments

- Assign SIDs 62,001 to 65,535 to binding segments

The network operator can also assign node SIDs so that all node segments ending on a node have the same SID (i.e., all node instructions that include the same information are identified by the same SID).

Segment Ingress Node	Segment Egress Node	SID
2	1	1001
3	1	1001
1	2	1002
3	2	1002
1	3	1003
2	3	1003

Table 1: Node SID Assignments

Table 1 illustrates this convention for Nodes 1, 2 and 3.

6. Forwarding Plane

```
                    SRm6 Path from node B to node C
                     <------------------------>
                   SR Path                    SR Path
                   Ingress                    Egress
                   Node                       Node
    +-+            +-+                        +-+            +-+
    |A|-->--//-->--|B|=====>=====//=====>=====|C|-->--//-->--|D|
    +-+            +-+                        +-+            +-+
    Original                                                 Original
    Packet                                                   Packet
    Source                                                   Destination
    Node                                                     Node
```

Figure 2: SRm6 Path As Tunnel

SRm6 is an application of IPv6 tunneling [RFC2473]. Figure 2 illustrates how an SRm6 ingress node receives an original packet, encapsulates it in an SRm6 header, and forwards the resulting SRm6 packet through an SRm6 path to an SRm6 egress node. The SRm6 egress node removes the SRm6 header and forwards the original packet to its destination.

```
                           +----------------------------------//-----+
                             | Original |                              |
                             |          |   Original Packet Payload    |
                             | Header   |                              |
                             +----------------------------------//-----+
                              <            Original Packet            >
                                               |
                                               v
        <     SRm6 Header    > <       Original Packet                 >
       +---------+ - - - - - +-------------------------//--------------+
       | IPv6    | IPv6      |                                         |
       |         | Extension |        Original Packet                  |
       | Header  | Headers   |                                         |
       +---------+ - - - - - +-------------------------//--------------+
        <                              SRm6 Packet                     >
```

Figure 3: SRm6 Encapsulation

Figure 3 illustrates SRm6 encapsulation. In the figure, the SRm6 header contains:

- An IPv6 header.

- One or more IPv6 extension headers.

The following rules govern the use of extension headers in the SRm6 header:

- The SRm6 header can contain any valid combination of extension headers.

- Extension headers are arranged in the order recommended by Section 4.1 of [RFC8200].

- Extension headers are processed in the order that the appear in the packet, as described in Section 4.1 of [RFC8200].

- If the SR path contains multiple segments, the SID list is encoded in a Compressed Routing Header (CRH) [I-D.bonica-6man-comp-rtg-hdr].

- If the SRm6 header contains a PSSI, it is encoded in a Destination Option header that precedes the CRH. Destination options will be defined as needed.

- If the SRm6 header contains a PPSI, it is encoded in the IPv6 Tunnel Payload Forwarding (TPF) Option [I-D.bonica-6man-vpn-dest-opt]. The TPF option appears in a Destination Options header that immediately precedes the upper-layer header.

Therefore, PSSI's are processed at each segment egress node, while the PPSI is processed at the path egress node only.

An SRm6 header contains only the extension headers that it needs. For example, an SRm6 header can contain:

- A CRH and a TPF Option - This packet traverses an SRm6 path that contains multiple segments and executes a PPSI at the path egress node.

- A CRH only - This packet traverses an SRm6 path that contains multiple segments and does not execute a PPSI at the path egress node.

- A TPF Option only - This packet traverses an SRm6 path that contains a single segment and executes a PPSI at the path egress node.

SRm6 inherits Hop Limit, traffic class and Flow Label processing procedures from I [RFC2473].

7. Control Plane

The following documents describe control plane extensions that support the CRH and the TPF Option:

- IS-IS Support for CRH [I-D.bonica-lsr-crh-isis-extensions]

- BGP Support for the IPv6 TPF Option [I-D.ssangli-idr-bgp-vpn-srv6-plus], [I-D.alston-spring-crh-bgp-signalling]

8. Differences Between SRm6 and SRv6

8.1. Instruction Encoding

SRm6 encodes topological instructions in 16 or 32-bit SIDs that appear in the CRH. It also encodes service instructions in IPv6 Destination Options.

SRv6 encodes all instructions in the low-order bits of the IPv6 Destination Address.

8.2. IPv6 Address Semantics

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

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

- A network interface

- An SRv6 SID, whose high-order bits are used for routing and low-order bits represent an instruction.

8.3. OAM

SRm6 does not require any new OAM mechanisms. Because SRv6 modifies IPv6 address semantics, it requires the OAM mechanisms described in [I-D.ietf-6man-spring-srv6-oam].

8.4. Routing Extension Header Size

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	48
11	184	32	48
12	200	32	52
13	216	32	52
14	232	32	56
15	248	40	56
16	264	40	60
17	280	40	60
18	296	40	64

Table 2: Routing Header Size (in Bytes) As A Function Of Routing Header Type and Number Of SIDs

SRv6 and SRm6 both encode path information in a Routing extension header. SRv6 uses the Segment Routing Header (SRH) [RFC8754], while SRm6 uses either the 16 or 32-bit version of the CRH. (Table 2) reports 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 all headers from buffer memory to on-chip memory. As header sizes increase, 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 is excessive.

8.5. Authentication

An SRm6 packet can include any valid combination of IPv6 extension headers. However, the IPv6 Authentication Header (AH) [RFC4302] is not defined in SRv6.

9. IANA Considerations

This document does not request any actions by IANA.

10. Security Considerations

SRm6 inherits the security consideration of IPv6 tunneling [RFC2473], the Compressed Routing Header (CRH) [I-D.bonica-6man-comp-rtg-hdr], and the IPv6 Tunnel Payload Forwarding (TPF) Option [I-D.bonica-6man-vpn-dest-opt].

11. Acknowledgements

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

12. References

12.1. 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., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-22 (work in progress), May 2020.

[I-D.bonica-6man-vpn-dest-opt]
Bonica, R., Kamite, Y., Jalil, L., Zhou, Y., and G. Chen, "The IPv6 Tunnel Payload Forwarding (TPF) Option", draft-bonica-6man-vpn-dest-opt-13 (work in progress), August 2020.

[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-03 (work in progress), August 2020.

[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.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[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>.

[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>.

12.2. Informative References

[I-D.filsfils-spring-net-pgm-extension-srv6-usid]
Filsfils, C., Camarillo, P., Cai, D., Voyer, D., Meilik, I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman, D., Liu, Y., and J. Guichard, "Network Programming extension: SRv6 uSID instruction", draft-filsfils-spring-net-pgm-extension-srv6-usid-07 (work in progress), May 2020.

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-07 (work in progress), July 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-20 (work in progress), September 2020.

[I-D.li-spring-compressed-srv6-np]
Li, Z., Li, C., Xie, C., LEE, K., Tian, H., Zhao, F., Guichard, J., Cong, L., and S. Peng, "Compressed SRv6 Network Programming", draft-li-spring-compressed-srv6-np-02 (work in progress), February 2020.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., and G. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-07 (work in progress), July 2020.

[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>.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

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