SPRING
Internet-Draft
Intended status: Informational
Expires: December 29, 2020
D. Dukes, Ed.
Cisco Systems
June 27, 2020

SRv6 Network Programming Overhead Analysis
draft-dukes-spring-srv6-overhead-analysis-00

Abstract

SRv6 network programming provides the framework for the best compression of an IPv6 header within an SR domain. This document provides the analysis to illustrate this fact.

Table of Contents

1. Introduction

SRv6 Network Programming [I-D.ietf-spring-srv6-network-programming] defines a framework to build a network program with topological and service segments carried in a Segment Routing header (SRH) [RFC8754]. The SRv6 Network Programming framework natively allows for long segment lists to be encoded in the SRH in a compressed fashion, as described in [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc].

This document provides a detailed analysis of the efficiency of this compression in a realistic SRv6 deployment scenario. In particular, it evaluates the encapsulation size of a compressed segment lists against uncompressed segment lists.

This document also shows that a mapping solution does not provide better compression than what can be achieved with the SRv6 mechanism. As such, analysis of the SRm6 proposal documented in [I-D.bonica-spring-sr-mapped-six] is provided for comparison.

2. Reference Topology

The following topology is used throughout the remainder of this document.

```
          + * * * * * * * * * * * * * * * * * * * * * * * * * * +
          *                                                       *
          * - - - - - - - - - + - - - - - - - - + - - - - - - - - *
          *                   |                 |                 *
          *   [M1_0]       [5]  [C_0]        [7]  [M2_0]          *
       [1]---[3]              |                 |               [4]---[2]
          *        [M1_i]  [6]        [C_j]  [8]       [M2_k]    *
          *                   |                 |                 *
          *     Metro 1       |      Core       |     Metro 2       *
          *- - - - - - - - - - - - - - - - - - - - - - - - - - - -*
          *                                                       *
          *                       SR domain                       *
          + * * * * * * * * * * * * * * * * * * * * * * * * * * +
```

Figure 1: Reference topology

- o 1 and 2 are hosts outside the SR domain
- o 3 and 4 are SR domain edge routers
- o Metro 1, Core and Metro 2 are independent IGP instances in the SR domain
- o 5 and 6 are border routers between the Metro 1 and Core instances
- o 7 and 8 are border routers between the Metro 2 and Core instances
- o M1_1..M1_i, C_1..C_j, and M2_1..M2_k are all SR domain routers

3. Reference Scenario

Consider a service provider offering a VPN service with underlay optimization.

Hosts 1 and 2 are located in two different sites of a VPN customer. The service provider associates VRF 5 with this VPN customer.

When host 1 sends a packet to host 2, the SR domain ingress router 3 steers it to the egress edge router 4 via an SR Policy that enforces a path through a number of underlay waypoints in Metro 1 (M1_1..M1_i), Core (C_1..C_j), and Metro 2 (M2_1..M2_k). The SR Policy ends with a SID that instructs the egress edge router 4 to decapsulate the packet and forward the inner packet within the appropriate VPN context towards host 2.

4. Compression Analysis

Compression is analyzed by first defining the encapsulation size of the compressed SRv6 segment list encoding schemes described in [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] called NEXT-C-SID and REPLACE-C-SID.

Subsequently, the encapsulation size of the SRm6 proposal [I-D.bonica-spring-sr-mapped-six] is defined.

The encapsulation savings is defined against the uncompressed SRv6 segment list for multiple reference scenarios.

4.1. SRv6

The formulae defining the length of a compressed SRv6 segment list, and an uncompressed one, along with the corresponding encapsulation size are defined in the subsequent sections.

4.1.1. NEXT-C-SID

4.1.1.1. Capacity of C-SID Container

The capacity of a C-SID container is calculated for the NEXT-C-SID flavor as

C_NEXT = floor((128 - B) / NF)

B and NF are as defined in Section 4 of [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc]

4.1.1.2. Length of a C-SID Sequence

Assuming that all SIDs in an uncompressed SID sequence S support a C-SID encoding, the length of the corresponding C-SID sequence is calculated for the NEXT-C-SID flavor as

L_NEXT(S) = ceil(|S| / C_NEXT)

4.1.2. REPLACE-C-SID

4.1.2.1. Capacity of C-SID Container

The capacity of a C-SID container is calculated for the REPLACE-C-SID flavor as

C_REPLACE = floor(128 / NF)

4.1.2.2. Length of a C-SID Sequence

Assuming that all SIDs in an uncompressed SID sequence S support a C-SID encoding, the length of the corresponding C-SID sequence is calculated for the REPLACE-C-SID flavor as

L_REPLACE(S) = 1 + ceil((|S| - 1) / C_REPLACE)

4.1.3. Encapsulation Size

The encapsulation size metric is defined in Section 2.4 of [I-D.filsfils-spring-analysis-fmwk-ext-srv6-encap] as follows.

> E(segment list): the number of bytes required to encapsulate a packet traversing the SR domain with segment list applied at an SR source node. Specifically, the number of bytes from the beginning of the encapsulating IPv6 header to the beginning of the packet traversing the SR domain, including any and all headers in between.

The value of this metric is calculated for reduced SRv6 encapsulation as

```
E(SL) = 40                         if |SL| = 1
        40 + 8 + (|SL| - 1) * 16   otherwise
```

where

- 40 octets is the length of the IPv6 header
- 8 octets is the length of the fixed SRH fields
- 16 octets is the length of each element in the SRH Segment List
- the conditional represents the option described in Section 4.1 of [RFC8754] to omit the SRH when the Segment List contains a single segment

The segment list SL may contain any combination of compressed and uncompressed SID sequences.

4.2. SRm6

The encapsulation size for SRm6 [I-D.bonica-spring-sr-mapped-six] includes a 40-octet IPv6 header, a CRH-16 or CRH-32 [I-D.bonica-6man-comp-rtg-hdr] routing header (when the segment list contains more than one element), and an 8-octet Destination Option header [I-D.bonica-6man-vpn-dest-opt].

4.2.1. SID list

The SID list for SRm6 does not include the VPN instruction. The VPN instruction is carried separately in a Destination Option header [I-D.bonica-6man-vpn-dest-opt].

Since SRm6 SID mappings are not redistributed across domains, an inter-domain SID list for SRm6 must include an intermediate SID on each border router between two domains. In the scenario described in Section 3, 2 additional SIDs are required.

4.2.2. Encapsulation size with CRH-16

```
E_CRH16(SL) = 40 + 8                                  if |SL| = 1
              40 + ceil((4 + |SL| * 2) / 8) * 8 + 8   otherwise
```

4.2.3. Encapsulation size with CRH-32

```
E_CRH32(SL) = 40 + 8                                  if |SL| = 1
              40 + ceil((4 + |SL| * 4) / 8) * 8 + 8   otherwise
```

4.3. Encapsulation size saving compared to uncompressed SRv6

The encapsulation size saving metric ES is defined as follows.

```
ES(SL) = 1 - E_compressed(SL) / E_uncompressed(SL)
```

where:

- "E_compressed(SL)" is the encapsulation size metric with a particular compression scheme (e.g., NEXT-C-SID, REPLACE-C-SID, SRm6)

- "E_uncompressed(SL)" is the encapsulation size metric for SRv6 without any compressed encoding. The uncompressed SRv6 encapsulation size is obtained by applying the formula in Section 4.1.3 to the uncompressed segment list.

4.4. Numerical Application

Applying the formulae presented in the previous sections, the encapsulation size savings is computed for the scenario described in Section 3.

4.4.1. Evaluation parameters

4.4.1.1. SID format

The NEXT-C-SID flavor is considered with a 32-bit SRv6 SID block and a 16-bit C-SID length.

The REPLACE-C-SID flavor is considered with both 16-bit and 32-bit C-SID lenghts. These two variants are referred to as REPLACE-16 and REPLACE-32, respectively.

The SRm6 proposal is considered with both the CRH-16 and CRH-32 routing headers.

4.4.1.2. Segment lists

The following segment lists are considered with equal number of segments in each domain:

- o "3D(3T).V": A segment list across 3 domains (metro 1, core, metro 2), with 3 underlay waypoints in each domain and a VPN identifier. 10 instructions in total.

- o "3D(4T).V": A segment list across 3 domains, with 4 underlay waypoints in each domain and a VPN identifier. 13 instructions in total.

- o "3D(6T).V": A segment list across 3 domains, with 6 underlay waypoints in each domain and a VPN identifier. 19 instructions in total.

4.4.2. Evaluation results

Metric	NEXT-C-SID	REPLACE-16	REPLACE-32	CRH-16	CRH-32
ES(3D(3T).V)	58.33%	33.33%	33.33%	58.33%	45.83%
ES(3D(4T).V)	66.67%	46.67%	46.67%	63.33%	53.33%
ES(3D(6T).V)	71.43%	61.90%	47.62%	71.43%	59.52%

Table 1: Encapsulation size saving for a multi-domain SR Policy

5. Conclusion

The NEXT-C-SID, REPLACE-C-SID and SRm6 methods all provide significant encapsulation size saving compared to the base SRv6 SIDs.

The mapping proposal, [I-D.bonica-spring-sr-mapped-six], does not bring any compression benefit compared to SRv6-native compression methods [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc].

The SRm6 proposal does have several deficiencies however, including:

- o A new data plane
- o A new control plane
- o A new ecosystem
- o Stateful NFV integration compared to SRv6 stateless NFV
- o Additional lookups at egress PE
  - * CRH requires 3 lookups at an egress PE vs 2 for SRv6.
  - * Additional lookups result in additional processing time in forwarding ASICs. Many modern ASIC architectures support 2 lookups per packet, additional lookups may require recirculation, effectively decreasing forwarding performance.

These and other metrics proposed for analysis are described in [I-D.filsfils-spring-analysis-fmwk-ext-srv6-encap].

6. References

6.1. Normative References

[I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc]
Cheng, W., Filsfils, C., Li, Z., Cai, D., Voyer, D., Clad, F., Shay, S., Guichard, J., and L. Aihua, "Compressed SRv6 Segment List Encoding in SRH", draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-01 (work in progress), May 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-15 (work in progress), March 2020.

[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>.

6.2. Informative References

[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-12 (work in progress), March 2020.

[I-D.bonica-spring-sr-mapped-six]
Bonica, R., Hegde, S., Kamite, Y., Alston, A., Henriques, D., Jalil, L., Halpern, J., Linkova, J., and G. Chen, "Segment Routing Mapped To IPv6 (SRm6)", draft-bonica-spring-sr-mapped-six-01 (work in progress), April 2020.

[I-D.filsfils-spring-analysis-fmwk-ext-srv6-encap]
Filsfils, C., Dukes, D., and K. Patel, "Analysis Framework For Extensions of SRv6 Encapsulation", draft-filsfils-spring-analysis-fmwk-ext-srv6-encap-00 (work in progress), January 2020.

Author's Address

Darren Dukes (editor)
Cisco Systems
Canada

Email: ddukes@cisco.com