

6man
Internet-Draft
Intended status: Standards Track
Expires: 16 May 2022

R. Bonica
Juniper Networks
Y. Kamite
NTT Communications Corporation
A. Alston
D. Henriques
Liquid Telecom
L. Jalil
Verizon
12 November 2021

The IPv6 Compact Routing Header (CRH)
draft-bonica-6man-comp-rtg-hdr-27

Abstract

This document defines two new Routing header types. Collectively, they are called the Compact Routing Headers (CRH). Individually, they are called CRH-16 and CRH-32.

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 16 May 2022.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. The Compressed Routing Headers (CRH)	3
4. The CRH Forwarding Information Base (CRH-FIB)	4
5. Processing Rules	6
5.1. Computing Minimum CRH Length	7
5.2. CRH Removal Procedure	8
6. Mutability	8
7. Applications And SIDs	8
8. Management Considerations	9
9. Security Considerations	9
10. Implementation and Deployment Status	9
11. IANA Considerations	10
12. Acknowledgements	10
13. Contributors	10
14. References	11
14.1. Normative References	11
14.2. Informative References	11
Appendix A. CRH Processing Examples	12
A.1. The SID List Contains One Entry For Each Segment In The Path	13
A.2. The SID List Omits The First Entry In The Path	14
Appendix B. A Packet Recycling Use-Case	14
Authors' Addresses	15

1. Introduction

IPv6 [RFC8200] source nodes use Routing headers to specify the path that a packet takes to its destination. The IETF has defined several Routing header types [IANA-RH]. This document defines two new Routing header types. Collectively, they are called the Compact Routing Headers (CRH). Individually, they are called CRH-16 and CRH-32.

The CRH allows IPv6 source nodes to specify the path that a packet takes to its destination. The CRH:

- * Can be encoded in relatively few bytes.
- * Is designed to operate within a network domain. (See Section 9).

The following are reasons for encoding the CRH in as few bytes as possible:

- * Many ASIC-based forwarders copy 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.

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. The Compressed Routing Headers (CRH)

Both CRH versions (i.e., CRH-16 and CRH-32) contain the following fields:

- * Next Header - Defined in [RFC8200].
- * Hdr Ext Len - Defined in [RFC8200].
- * Routing Type - Defined in [RFC8200]. Value TBD by IANA. (For CRH-16, the suggested value is 5. For CRH-32, the suggested value is 6.)
- * Segments Left - Defined in [RFC8200].
- * Type-specific Data - Described in [RFC8200].

In the CRH, the Type-specific data field contains a list of Segment Identifiers (SIDs). Each SID represents both of the following:

- * A segment of the path that the packet takes to its destination.
- * An entry in the CRH Forwarding Information Base (CRH-FIB) (Section 4).

SIDs are listed in reverse order. So, the first SID in the list represents the final segment in the path. Because segments are listed in reverse order, the Segments Left field can be used as an index into the SID list. In this document, the "current SID" is the SID list entry referenced by the Segments Left field.

The first segment in the path can be omitted from the list. See Appendix A for examples.

In the CRH-16 (Figure 1), each SID is encoded in 16-bits. In the CRH-32 (Figure 2), each SID is encoded in 32-bits.

In all cases, the CRH MUST end on a 64-bit boundary. So, the Type-specific data field MUST be padded with zeros if the CRH would otherwise not end on a 64-bit boundary.

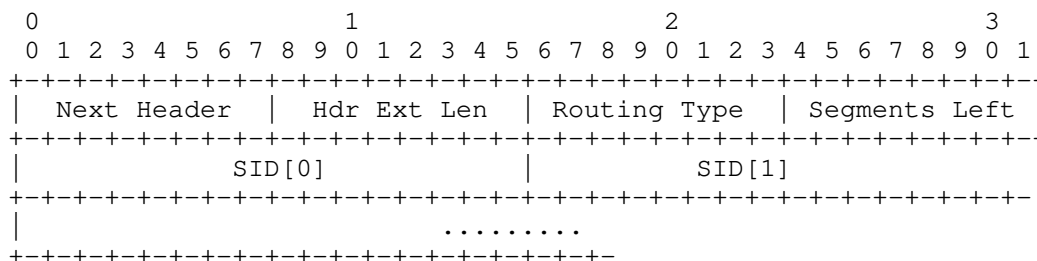


Figure 1: CRH-16

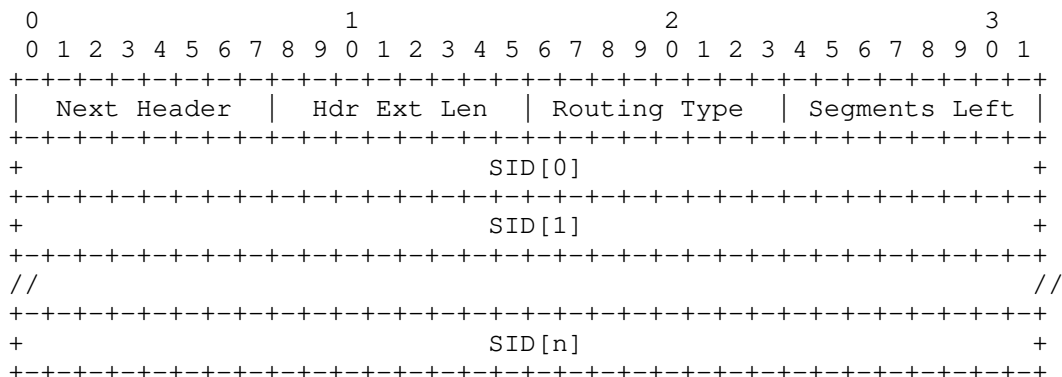


Figure 2: CRH-32

4. The CRH Forwarding Information Base (CRH-FIB)

Each SID identifies a CRH-FIB entry.

Each CRH-FIB entry contains:

- * An IPv6 address (optional).
- * A topological function.
- * Arguments for the topological function (optional).
- * Flags.
- * A service function (optional).
- * Arguments for the service function (optional).

The IPv6 address can represent either:

- * An interface on the next segment endpoint.
- * An SRv6 SID [RFC8986], instantiated on the next segment endpoint.

The first ten bits of the IPv6 address MUST NOT be fe00. That prefix is reserved for link-local [RFC6890] addresses.

The topological function specifies how the processing node forwards the packet to the next segment endpoint. The following are examples:

- * Forward the packet through the least-cost path to the next segment endpoint.
- * Forward the packet through a specified interface.
- * Encapsulate the packet in another IPv6 header of any type (e.g., MPLS, IPv6) and forward either through the least cost path or a specified interface.
- * Recycle the packet, as if the node had forwarded to one of its own interfaces. When recycling is complete, process the next SID. See Appendix B for a packet recycling use-case.

Some topological functions require parameters. For example, a topological function might require a parameter that identifies the interface through which the packet should be forwarded.

The following flags are defined:

- * The PSP flag indicates whether the penultimate segment endpoint (i.e., the node that sets Segments Left to 0) MAY remove the CRH.

- * The OAM flag indicates whether the processing node should invoke OAM procedures for which it is configured.

The service function is optional. If present, it invokes a node specific procedure. The following are examples of node specific procedures:

- * Emit telemetry.
- * Subject the packet's payload to a firewall rule.
- * Replicate the packet, forwarding one copy and retaining the other for sampling, analysis, or other purposes.

Node specific procedures are not subject to standardization. A node can support any number of node specific procedures and associate them with any SIDs.

Some service functions require parameters. For example, an instruction to emit telemetry might require an IP address to which telemetry should be sent.

The CRH-FIB can be populated:

- * By an operator, using a Command Line Interface (CLI).
- * By a controller, using the Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] or the Network Configuration Protocol (NETCONF) [RFC6241].
- * By a distributed routing protocol [ISO10589-Second-Edition], [RFC5340], [RFC4271].

5. Processing Rules

The following rules describe CRH processing:

- * If Segments Left equals 0, skip over the CRH and process the next header in the packet.
- * If Hdr Ext Len indicates that the CRH is larger than the implementation can process, discard the packet and send an ICMPv6 [RFC4443] Parameter Problem, Code 0, message to the Source Address, pointing to the Hdr Ext Len field.
- * Compute L, the minimum CRH length (Section 5.1).

- * If L is greater than Hdr Ext Len, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the Segments Left field.
- * Decrement Segments Left.
- * Search for the current SID in the CRH-FIB. In this document, the "current SID" is the SID list entry referenced by the Segments Left field.
- * If the search does not return a CRH-FIB entry, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the current SID.
- * If Segments Left is greater than 0 and the CRH-FIB entry contains a multicast address, discard the packet and send an ICMPv6 Parameter Problem, Code 0, message to the Source Address, pointing to the current SID.
- * If present, copy the IPv6 address from the CRH-FIB entry to the Destination Address field in the IPv6 header.
- * Decrement the IPv6 Hop Limit.
- * If the CRH-FIB entry contains a service function, execute it.
- * If Segments Left is equal to zero, and the PSP flag in the CRH-FIB entry is set, execute the CRH removal procedure (Section 5.2).
- * Submit the packet, its topological function and its parameters to the IPv6 module. See NOTE.

NOTE: By default, the IPv6 module determines the next-hop and forwards the packet. However, the topological function may elicit another behavior. For example, the IPv6 module may forward the packet through a specified interface.

5.1. Computing Minimum CRH Length

The algorithm described in this section accepts the following CRH fields as its input parameters:

- * Routing Type (i.e., CRH-16 or CRH-32).
- * Segments Left.

It yields L, the minimum CRH length. The minimum CRH length is measured in 8-octet units, not including the first 8 octets.

```
<CODE BEGINS>
switch(Routing Type) {
  case CRH-16:
    if (Segments Left <= 2)
      return(0)
    sidsBeyondFirstWord = Segments Left - 2;
    sidPerWord = 4;
  case CRH-32:
    if (Segments Left <= 1)
      return(0)
    sidsBeyondFirstWord = Segments Left - 1;
    sidsPerWord = 2;
  case default:
    return(0xFF);
}

words = sidsBeyondFirstWord div sidsPerWord;
if (sidsBeyondFirstWord mod sidsPerWord)
  words++;

return(words)
<CODE ENDS>
```

5.2. CRH Removal Procedure

The processing node SHOULD execute the following procedure, if it is capable of doing so:

- * Update the Next Header field in the header preceding the CRH using a value taken from the Next Header field in the CRH.
- * Decrease the Payload Length field in the IPv6 header by $8 \times (x+1)$, where value of x is equal to the value of the Hdr Ext Len field in the CRH.
- * Remove the CRH from the IPv6 header chain.

6. Mutability

In the CRH, the Segments Left field is mutable. All remaining fields are immutable.

7. Applications And SIDs

A CRH contains one or more SIDs. Each SID is processed by exactly one node.

Therefore, a SID is not required to have domain-wide significance. Applications can:

- * Allocate SIDs so that they have domain-wide significance.
- * Allocate SIDs so that they have node-local significance.

8. Management Considerations

PING and TRACEROUTE [RFC2151] both operate correctly in the presence of the CRH.

9. Security Considerations

Networks that process the CRH MUST NOT accept packets containing the CRH from untrusted sources. Their border routers SHOULD discard packets that satisfy the following criteria:

- * The packet contains a CRH
- * The Segments Left field in the CRH has a value greater than 0
- * The Destination Address field in the IPv6 header represents an interface that resides inside of the network.

Many border routers cannot filter packets based upon the Segments Left value. These border routers MAY discard packets that satisfy the following criteria:

- * The packet contains a CRH
- * The Destination Address field in the IPv6 header represents an interface that resides inside of the network.

10. Implementation and Deployment Status

Juniper Networks has produced experimental implementations of the CRH on:

- * A LINUX-based software platform
- * The MX-series (ASIC-based) router

Liquid Telecom has deployed the CRH, on a limited basis, in their network. Other experimental deployments are in progress.

11. IANA Considerations

This document makes the following registrations in the "Internet Protocol Version 6 (IPv6) Parameters" "Routing Types" subregistry maintained by IANA:

Value	Description	Reference
5	CRH-16	This document
6	CRH-32	This document

12. Acknowledgements

Thanks to Dr. Vanessa Ameen, Fernando Gont, Naveen Kottapalli, Joel Halpern, Tony Li, Gerald Schmidt, Nancy Shaw, Ketan Talaulikar, and Chandra Venkatraman for their contributions to this document.

13. Contributors

Gang Chen

Baidu

No.10 Xibeiwang East Road Haidian District

Beijing 100193 P.R. China

Email: phdgang@gmail.com

Yifeng Zhou

ByteDance

Building 1, AVIC Plaza, 43 N 3rd Ring W Rd Haidian District

Beijing 100000 P.R. China

Email: yifeng.zhou@bytedance.com

Gyan Mishra

Verizon

Silver Spring, Maryland, USA

Email: hayabusagsm@gmail.com

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [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>>.

14.2. Informative References

- [IANA-RH] IANA, "Routing Headers", <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-3>>.
- [ISO10589-Second-Edition] International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, November 2001.

- [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>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<https://www.rfc-editor.org/info/rfc6890>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

Appendix A. CRH Processing Examples

This appendix demonstrates CRH processing in the following scenarios:

- * The SID list contains one entry for each segment in the path (Appendix A.1).
- * The SID list omits the first entry in the path (Appendix A.2).

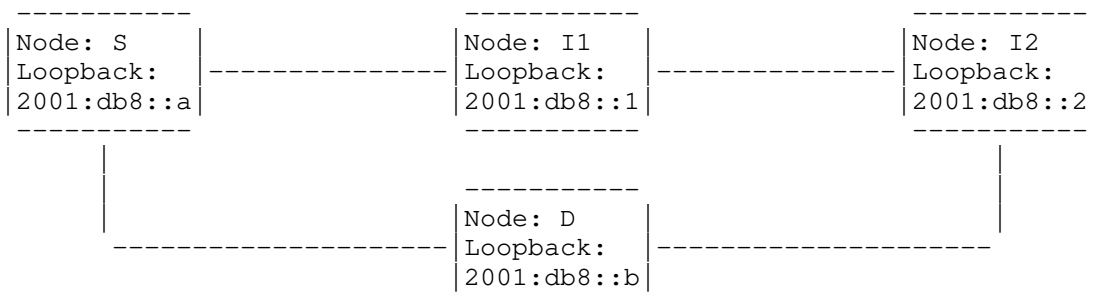


Figure 3: Reference Topology

Figure 3 provides a reference topology that is used in all examples.

SID	IPv6 Address	Forwarding Method
2	2001:db8::2	Least-cost path
11	2001:db8::b	Least-cost path

Table 1: Node SIDs

Table 1 describes two entries that appear in each node's CRH-FIB.

A.1. The SID List Contains One Entry For Each Segment In The Path

In this example, Node S sends a packet to Node D, via I2. In this example, I2 appears in the CRH segment list.

As the packet travels from S to I2:	
Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11
	SID[1] = 2

Table 2

As the packet travels from I2 to D:	
Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11
	SID[1] = 2

Table 3

A.2. The SID List Omits The First Entry In The Path

In this example, Node S sends a packet to Node D, via I2. In this example, I2 does not appear in the CRH segment list.

As the packet travels from S to I2:	
Source Address = 2001:db8::a	Segments Left = 1
Destination Address = 2001:db8::2	SID[0] = 11

Table 4

As the packet travels from I2 to D:	
Source Address = 2001:db8::a	Segments Left = 0
Destination Address = 2001:db8::b	SID[0] = 11

Table 5

Appendix B. A Packet Recycling Use-Case

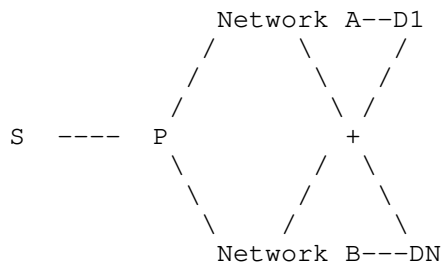


Figure 4: Packet Recycling Use-case

In Figure 4:

- * The SR domain contains Node S, Node P, and a set of destination nodes (D1 through DN)
- * S is connected to P
- * P is connected to Network A and to Network B. Neither of these networks are SR-capable.
- * The destination nodes connect to both Network A and Network B

S needs to reach each destination node through two SR paths. One SR path traverses Network A while the other traverses Network B.

Uncompressed SRv6 can encode this SR Path in two segments, with one segment instantiated on P and the other on the destination. To support this strategy, P instantiates two END.X SIDs (one per network).

CRH compressed SRv6 can encode this SR Path in two or three segments. When it encodes the path in two segments, one segment instantiated on P and the other on the destination. To support this strategy, P instantiates 2*N SIDs (one per network per destination). When CRH compressed SRv6 encodes the path in three segments, two segments are instantiated on P and the other on the destination. The first segment on P updates the IPv6 Destination address without forwarding the packet, while the other segment on P forwards the packet without updating the IPv6 destination address. To support this strategy, P instantiates 2+N SIDs (one per network and one per destination).

Authors' Addresses

Ron Bonica
 Juniper Networks
 2251 Corporate Park Drive

Herndon, Virginia 20171
United States of America

Email: rbonica@juniper.net

Yuji Kamite
NTT Communications Corporation
3-4-1 Shibaura, Minato-ku,
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
United States of America

Email: luay.jalil@one.verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 22, 2019

M. Chen
X. Geng
Huawei
Z. Li
China Mobile
October 19, 2018

Segment Routing (SR) Based Bounded Latency
draft-chen-detnet-sr-based-bounded-latency-00

Abstract

One of the goals of DetNet is to provide bounded end-to-end latency for critical flows. This document defines how to leverage Segment Routing (SR) to implement bounded latency. Specifically, the SR Identifier (SID) is used to specify transmission time (cycles) of a packet. When forwarding devices along the path follow the instructions carried in the packet, the bounded latency is achieved. This is called Cycle Specified Queuing and Forwarding (CSQF) in this document.

Since SR is a source routing technology, no per-flow state is maintained at intermediate and egress nodes, SR-based CSQF naturally supports flow aggregation that is deemed to be a key capability to allow DetNet to scale to large networks.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 22, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Cycle Specified Queuing and Forwarding	3
2.1. CSQF Basic Concepts	3
2.2. CSQF Queuing Model	5
2.3. CSQF Timing Model	7
2.4. Congestion Protection and Resource Reservation	8
2.5. An Example of CSQF	9
3. Segment Routing Extensions for CSQF	10
4. IANA Considerations	11
5. Security Considerations	11
6. Acknowledgements	11
7. References	11
7.1. Normative References	11
7.2. Informative References	11
Authors' Addresses	12

1. Introduction

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] is defined to provide end-to-end bounded latency and extremely low packet loss rates for critical flows. For a specific path, the end-to-end latency consists of two parts: 1) the accumulated latency on the wire, 2) the accumulated latency of nodes along the path. The former can be considered as constant once the path has been determined. The latter is contributed by the latency within each node along the path. So, to guarantee the end-to-end bounded latency, control the bounded latency within a node is the key. If every node along the path can guarantee bounded latency, then end-to-end bounded latency can be achieved.

[I-D.finn-detnet-bounded-latency] gives a framework that describes how bounded latency and zero congestion loss are achieved. It introduces a parameterized timing model that can be used by DetNet solutions by selecting a corresponding Quality of Service (QoS) algorithm and resource reservation algorithm to achieve the bounded latency and zero congestion loss goal.

This document defines how to leverage Segment Routing (SR) [RFC8402] to implement bounded latency. Specifically, the SR Identifier (SID) is used to carry and specify the "sending time" (cycle) of a packet, and ensure that the packet will be transmitted in that specified sending cycle in order to achieve the bounded latency. This is called Cycle Specified Queuing and Forwarding (CSQF) in this document.

2. Cycle Specified Queuing and Forwarding

2.1. CSQF Basic Concepts

By specifying the sending cycle of a packet at a node and making sure that the packet will be transmitted in that cycle, CSQF can achieve bounded latency within the node. By specifying the sending cycle at every node along a path, the end-to-end bounded latency can be achieved.

To support CSQF, similar to Cyclic Queuing and Forwarding (CQF) [IEEE802.1Qch], the sending time of an output interface of a node is divided into a series of equal time intervals with the duration of T . Each time interval is called a "cycle", and each cycle corresponds to a queue. During a cycle, only the corresponding queue is open and all the packets in that queue will be transmitted. CSQF can not only control the bounded latency at every node along a path, but regulate the traffic at each node as planned. Therefore, no congestion will occur.

Figure 1 provides an overview of CSQF.

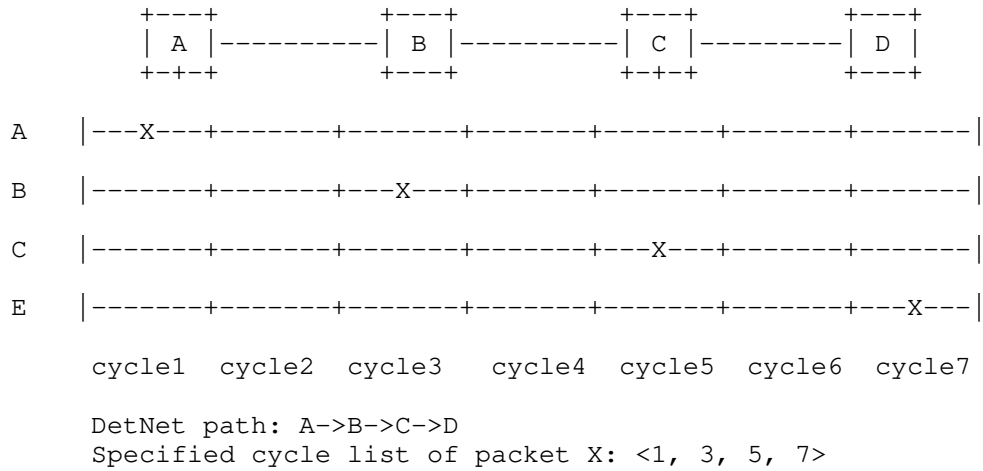


Figure 1: CSQF Overview

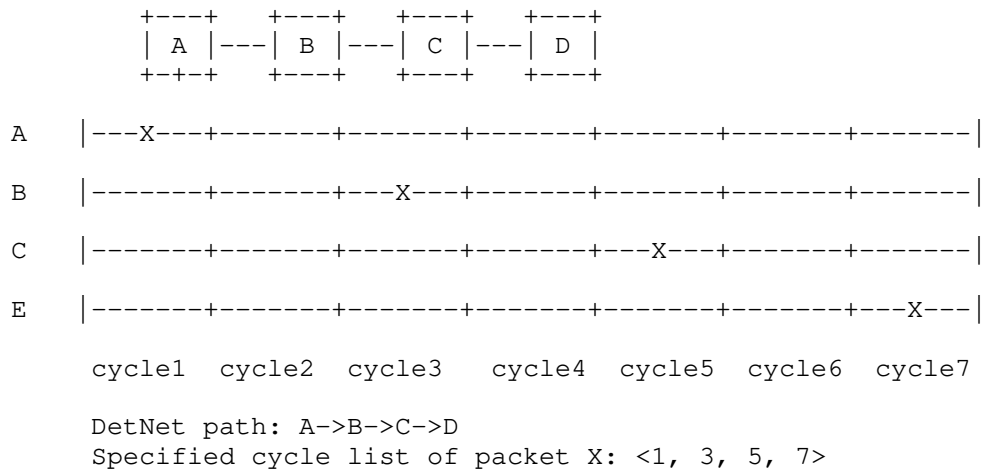


Figure 1: CSQF Overview

CSQF has the following characteristics:

- o The sending time (cycle) of a packet at each node along a path is specified so that the packet will be transmitted in the specified cycles, hence to guarantee the end-to-end bounded latency.
- o The specified cycles are calculated by fully considering the link delay, processing delay and the available cycle resources,

resulting in no bandwidth waste and no congestion (cycle-based traffic regulation).

- o Segment routing (SR) is used. Specifically, a SID is used to indicate in which cycle and to which output interface that a packet is specified to transmit, and an SR SID list is used to carry the specified cycles along a path. With SR, there is no per-flow states maintained at the intermediate and egress node. As a result, scalability is greatly improved compared to a solution that maintains flow state at each hop.
- o Flow aggregation is naturally supported by introducing SR and cycle-based scheduling.

2.2. CSQF Queuing Model

In Cyclic Queuing and Forwarding (CQF) [IEEE802.1Qch], time is divided into numbered time intervals, and each time interval is called a cycle; the critical traffic is then transmitted and queued for transmission along a path in a cyclic manner. With CQF, the delays experienced by a given packet are as follows:

- o The maximum end-to-end delay = $(N+1) * T$;
- o The minimum end-to-end delay = $(N-1) * T$;
- o Where the N is the number of hops and T is the duration of the cycle.

CQF assumes that a packet is transmitted from an upstream node in a cycle and the packet must be received at the downstream node in the same cycle, and it must be transmitted in the next cycle to the nexthop node. This assumption leads to very low bandwidth utilization when the link delay, processing delay, etc., factors cannot be considered as trivial. To guarantee this assumption, more bandwidth has to be reserved as a guard band for each cycle, and the effective bandwidth for DetNet service will be greatly reduced.

CSQF improves on CQF by explicitly specifying the sending cycles at every node along the path. This relieves the limitation that the sending (at the upstream node) and receiving (at the downstream node) have to be in the same cycle. For CSQF, the cycle to use depends on traffic planning and path calculation. The path calculation will consider the available cycle resources, bandwidth, and delay constraints.

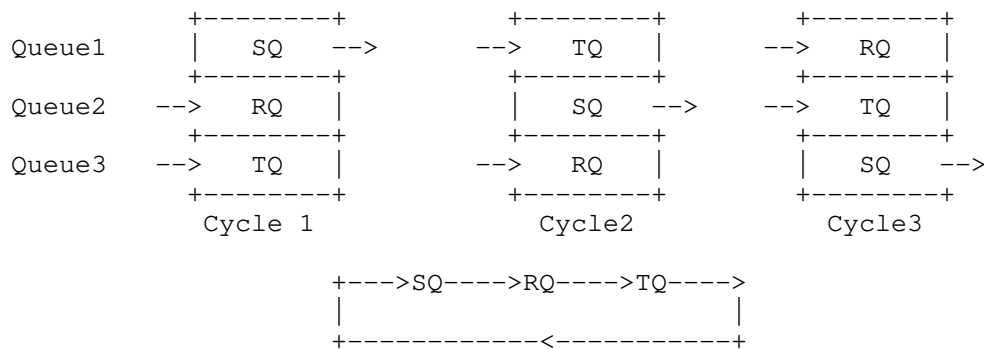


Figure 2: CSQF Queuing Model

For CSQF, three queues (in theory, two or more queues work as well) for each output interface are used. During a particular cycle, only one queue is open and the packets in that queue will be transmitted. This queue is called the sending queue (SQ). The other two queues are closed and can enqueue packets. One of them is called the receiving queue (RQ). The third queue is called the tolerating queue (TQ).

The RQ is used for receiving the packets that are expected to be transmitted in the next cycle. The TQ is used for tolerating the packets that come a bit early due to processing delay variation (processing jitter) or other reasons (e.g., packets are not transmitted as required by the traffic specification). Both RQ and TQ can have the capability to absorb a certain amount of processing jitter and traffic bursts. The upper bound of the absorbing capacity is $2T$. In order to increase the jitter/burst absorbing capacity, a four or more-queue model can be used. If the processing delay and traffic bursts are small, two-queue model works as well.

The roles of the three queues are not fixed, and on the contrary, they rotate with each cycle change. As showed in Figure 2, during cycle 1, queue 1 is SQ, queue 2 is RG and queue 3 is TQ; during cycle 2, queue 1 is TQ, queue 2 is SQ and queue 3 is RQ, during cycle 3, queue 1 is RQ, queue 2 is TQ and queue 3 is SQ. That means, for a particular queue, its role will rotate as "...->SQ->RQ->TQ->SQ->...", the starting role of a queue can be any one of the three roles.

In CSQF, a cycle corresponds to a queue. There are several ways to do cycle to queue mapping. The simplest mapping between cycles and queues is 1:1 mapping. There could be N:1 mapping, but that requires more identifiers, which in the case of segment routing, would require

more SIDs. This document does not specify which mapping should be used. The mapping choice is left to the operator.

2.3. CSQF Timing Model

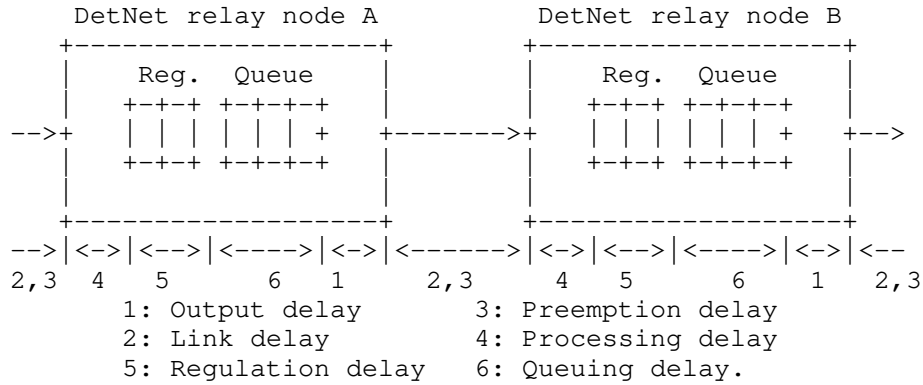


Figure 3: Timing model for DetNet

The DetNet timing model in Figure 3 is defined in [I-D.finn-detnet-bounded-latency]. It details the delays that a packet can experience from hop to hop. There are six delays, the detailed explanation of which can be found in [I-D.finn-detnet-bounded-latency]. This document simplifies the above model as follows:

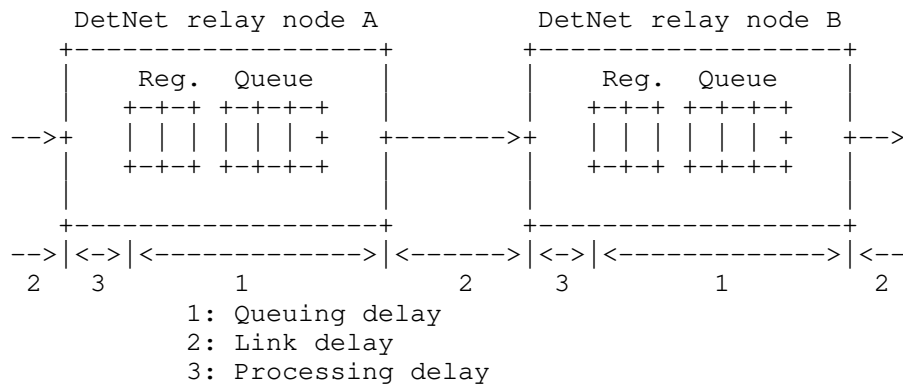


Figure 4: Simplified Timing model for DetNet

In this simplified timing model, only three delays are defined. The queuing delay in this new model includes the output delay, regulation delay, and queuing delay that are defined in the DetNet timing model (Figure 3). The link delay defined in this document includes the

link delay and the preemption delay defined in [I-D.finn-detnet-bounded-latency]. The processing delay is the same as defined in [I-D.finn-detnet-bounded-latency].

To further simplify the model, it assumes that the link delay only depends on the distance of the link. Once the DetNet path has been determined, the link delay can be considered as constant. The processing delay and queuing delay are variable but have their upper bounds.

For the processing delay, there are two bounds: minimum processing delay (Min-P-Delay) and maximum processing delay (Max-P-Delay).

- o Thus, the maximum processing jitter (Max-P-Jitter) = Max-P-Delay - Min-P-Delay.

As described in Section 2.2, both the RQ and TQ can be used for absorbing processing jitter, and the upper bound of the absorbing capacity is $2T$. So, if the processing jitter is less than $2T$, the three-queue model can work. Otherwise, more buffer is needed to absorb the jitter, through increasing the duration of the cycle or by adding more queues. Increasing the duration of the cycles is equivalent to increasing the depth of the queues (adding more buffer for each queue).

With above, for CSQF, the delays experienced by a given packet are as follows:

- o The maximum end-to-end delay = Link delay + $N * (Max-P-Delay + 2T)$;
- o The maximum end-to-end jitter = $2T$;
- o Where N is the number of hops and T is the duration of a cycle.

2.4. Congestion Protection and Resource Reservation

Congestion protection is the key for bounded latency and zero congestion loss. An essential component of DetNet is Traffic Engineering (TE), so that dedicated resources can be reserved for the exclusive use of DetNet flows. To avoid congestion, two or more flows must be prevented from contending for the same resource. For normal TE, the critical resource is bandwidth, but in the case of CSQF, the critical resource is interface occupation time. Bandwidth is an average value, which can generally guarantee the quality of service generally, but bursts and congestion may still occur. By comparison, the interface occupation time is an absolute value, which can avoid packet packets conflicting for the same resource by

controller computation and time allocation for different flows. The unit of time allocation is the cycle, and a Traffic Specification, the flow transmission description, is necessary for the computation.

CSQF uses segment routing SIDs to carry the time allocation information (the cycle), and it ensures that a node can schedule different packets without conflict and forward the packets at the proper time. The resource reservation is not explicitly implemented by a control plane protocol, such as Resource Reservation Protocol - Traffic Engineering (RSVP-TE) or Stream Reservation Protocol (SRP). Rather, it is guaranteed by the SR controller, which maintains the status of different flows and time occupation of all the network devices in the domain. This is called the Virtual Resource Reservation (VRR) in this document.

2.5. An Example of CSQF

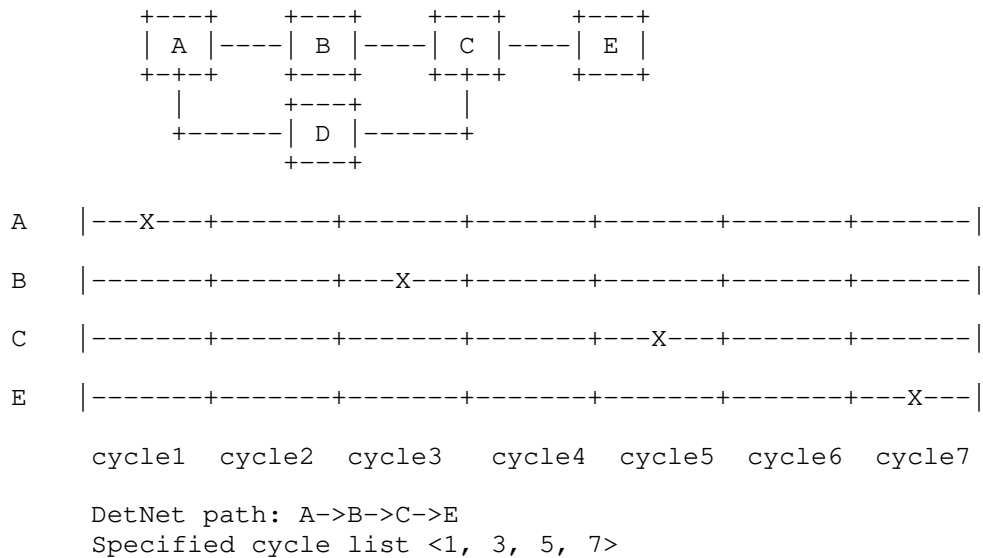


Figure 5: CSQF Example

As showed in Figure 5, there is a DetNet path (A->B->C->E), and a packet (X) is expected to be transmitted in cycle 1 at node A, in cycle 3 at node B, in cycle 5 at node C and in cycle 7 at node E. A cycle list <1, 3, 5, 7> is attached to the packet, and the packet will be transmitted along the path as the specified cycles.

Given the topology as above, assume the duration of a cycle is 10us; the link delays between nodes are the same (e.g., 100us); the minimum

processing delay at each node = 10us, the maximum processing delay at each node is 20us, so the maximum processing jitter is 10us.

For a given packet that is transmitted along the path(A->B->C->D->E), the experienced maximum end-to-end delay is:

$$\begin{aligned} & (N-1) * \text{link delay} + N * (\text{maximum processing delay} + 2T) \\ &= 3*100 + 4* 40 \\ &= 460 \text{ (us)} \end{aligned}$$

The maximum end-to-end jitter is always 2T (20us).

3. Segment Routing Extensions for CSQF

This document defines a new segment that is called a Cycle Segment, which is used to identify a cycle. A Cycle Segment is a local segment and is allocated from the Segment Routing Local Block (SRLB) [RFC8402].

A Cycle Segment has two meanings: 1) identify an interface/link, just like the adjacency segment does; 2) identify a cycle of the interface/link. To specify to which interface and in which cycle a packet should be transmitted, it just needs to attach a Cycle Segment to the packet. By attaching a list of Cycle Segments to a packet, it can not only implement the explicit route of the packet that is required by DetNet [I-D.ietf-detnet-architecture], but also specify the sending cycle at each node along the path without maintaining per-flow states at the intermediate and egress nodes. Hence, it naturally supports flow aggregation, and that allows DetNet to support large number of DetNet flows and scale to large networks.

Normally, several SR SIDs are required to be allocated for each CSQF capable interface. How many SIDs are allocated depends on how many cycles are used. Given a three-queue model and a 1:1 cycle to queue mapping is used, three SIDs will be allocated for each CSQF capable interface. For example, given node A, SR-MPLS SIDs 1001, 1002, and 1003 are allocated to one of its interfaces. SID 1001 identifies cycle 1, SID 1002 identifies cycle 2, SID 1003 identifies cycle 3.

The SR [RFC8402] can be instantiated on various data planes. There are two data-plane instantiations of SR: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). Both SR-MPLS and SRv6 SIDs can be used for CSQF cycle identification. The mapping (IGP extensions) between a cycle and a SID will be defined in a separate document.

4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

6. Acknowledgements

The authors would like to thank Andrew G. Malis, Norman Finn for his review, suggestion and comments to this document.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

7.2. Informative References

[I-D.finn-detnet-bounded-latency]
Finn, N., Boudec, J., Mohammadpour, E., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-finn-detnet-bounded-latency-01 (work in progress), July 2018.

[I-D.geng-detnet-conf-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "DetNet Configuration YANG Model", draft-geng-detnet-conf-yang-05 (work in progress), October 2018.

[I-D.geng-detnet-info-distribution]
Geng, X., Chen, M., and Z. Li, "IGP-TE Extensions for DetNet Information Distribution", draft-geng-detnet-info-distribution-02 (work in progress), March 2018.

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-08 (work in progress), September 2018.

- [IEEE802.1Qch]
"IEEE, "Cyclic Queuing and Forwarding (IEEE Draft P802.1Qch)", 2017,
<<http://www.ieee802.org/1/files/private/ch-drafts/>>.",
2016.
- [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>>.

Authors' Addresses

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Zhenqiang Li
China Mobile

Email: lizhenqiang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 8, 2019

M. Chen
X. Geng
Huawei
Z. Li
China Mobile
May 7, 2019

Segment Routing (SR) Based Bounded Latency
draft-chen-detnet-sr-based-bounded-latency-01

Abstract

One of the goals of DetNet is to provide bounded end-to-end latency for critical flows. This document defines how to leverage Segment Routing (SR) to implement bounded latency. Specifically, the SR Identifier (SID) is used to specify transmission time (cycles) of a packet. When forwarding devices along the path follow the instructions carried in the packet, the bounded latency is achieved. This is called Cycle Specified Queuing and Forwarding (CSQF) in this document.

Since SR is a source routing technology, no per-flow state is maintained at intermediate and egress nodes, SR-based CSQF naturally supports flow aggregation that is deemed to be a key capability to allow DetNet to scale to large networks.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 8, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Cycle Specified Queuing and Forwarding	4
3.1. CSQF Basic Concepts	4
3.2. CSQF Queuing Model	5
3.3. CSQF Timing Model	7
3.4. Congestion Protection and Resource Reservation	8
3.5. An Example of CSQF	9
4. Segment Routing Extensions for CSQF	10
4.1. Time Aware Adjacency Segment (TA-Adj-SID)	11
5. IANA Considerations	11
6. Security Considerations	11
7. Acknowledgements	11
8. References	12
8.1. Normative References	12
8.2. Informative References	12
Authors' Addresses	12

1. Introduction

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] is defined to provide end-to-end bounded latency and extremely low packet loss rates for critical flows. For a specific path, the end-to-end latency consists of two parts: 1) the accumulated latency on the wire, 2) the accumulated latency of nodes along the path. The former can be considered as constant once the path has been determined. The latter is contributed by the latency within each node along the path. So, to guarantee the end-to-end bounded latency, control the bounded latency within a node is the key. If

every node along the path can guarantee bounded latency, then end-to-end bounded latency can be achieved.

[I-D.finn-detnet-bounded-latency] gives a framework that describes how bounded latency and zero congestion loss are achieved. It introduces a parameterized timing model that can be used by DetNet solutions by selecting a corresponding Quality of Service (QoS) algorithm and resource reservation algorithm to achieve the bounded latency and zero congestion loss goal.

This document defines how to leverage Segment Routing (SR) [RFC8402] to implement bounded latency, which is called Time Aware Segment Routing(TA-SR). A segment is associated with a topological instruction, which instruct a node to forward the packet via a specific outgoing interface, as it is defined in [RFC8402]. At the same time, the segment is also associated with DetNet bounded latency service. Specifically, the segment ID(SID) is used to carry and specify the "sending time" of a packet, and some mechanisms can be used to ensure that the packet will be transmitted in that specified period of sending time, which is called Time Aware Segment Routing(TA-SR).

The TA-SR architecture supports any type of control plane: distributed (IS-IS or OSPF or BGP), centralized (NETCONF or PCEP or BGP), or hybrid (PCEP or BGP).

The TA-SR architecture can be instantiated on various data planes, including TA-SR over MPLS (TA-SR MPLS) or TA-SR over IPv6 (TA-SRv6).

2. Terminology

All the terminologies used in this document are extensions of [RFC8402].

Time Aware Segment:

Time Aware SID:

TA-SR MPLS SID:

TA-SRv6 SID:

TA-SR Domain:

TA-SR Globle Block (SRGB):

TA-SR Local Block (SRGB):

TA-Adjacency Segment:

Forwarding Time Base: besides: the node uses the SID as an entry to get the Egress interface with Forwarding Information Base(FIB); Similarly, the node can use the SID as an entry to get the sending time of the packet, with Forwarding Time Base.

3. Cycle Specified Queuing and Forwarding

3.1. CSQF Basic Concepts

By specifying the sending cycle of a packet at a node and making sure that the packet will be transmitted in that cycle, CSQF can achieve bounded latency within the node. By specifying the sending cycle at every node along a path, the end-to-end bounded latency can be achieved.

To support CSQF, similar to Cyclic Queuing and Forwarding (CQF) [IEEE802.1Qch], the sending time of an output interface of a node is divided into a series of equal time intervals with the duration of T. Each time interval is called a "cycle", and each cycle corresponds to a queue. During a cycle, only the corresponding queue is open and all the packets in that queue will be transmitted. CSQF can not only control the bounded latency at every node along a path, but regulate the traffic at each node as planned. Therefore, no congestion will occur.

Figure 1 provides an overview of CSQF.

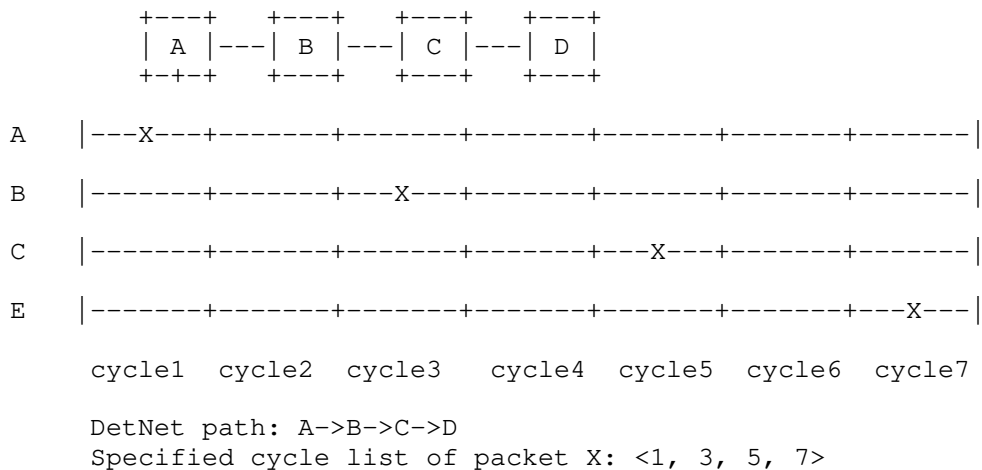


Figure 1: CSQF Overview

CSQF has the following characteristics:

- o The sending time (cycle) of a packet at each node along a path is specified so that the packet will be transmitted in the specified cycles, hence to guarantee the end-to-end bounded latency.
- o The specified cycles are calculated by fully considering the link delay, processing delay and the available cycle resources, resulting in no bandwidth waste and no congestion (cycle-based traffic regulation).
- o Segment routing (SR) is used. Specifically, a SID is used to indicate in which cycle and to which output interface that a packet is specified to transmit, and an SR SID list is used to carry the specified cycles along a path. With SR, there is no per-flow states maintained at the intermediate and egress node. As a result, scalability is greatly improved compared to a solution that maintains flow state at each hop.
- o Flow aggregation is naturally supported by introducing SR and cycle-based scheduling.

3.2. CSQF Queuing Model

In Cyclic Queuing and Forwarding (CQF) [IEEE802.1Qch], time is divided into numbered time intervals, and each time interval is called a cycle; the critical traffic is then transmitted and queued for transmission along a path in a cyclic manner. With CQF, the delays experienced by a given packet are as follows:

- o The maximum end-to-end delay = $(N+1) * T$;
- o The minimum end-to-end delay = $(N-1) * T$;
- o Where the N is the number of hops and T is the duration of the cycle.

CQF assumes that a packet is transmitted from an upstream node in a cycle and the packet must be received at the downstream node in the same cycle, and it must be transmitted in the next cycle to the nexthop node. This assumption leads to very low bandwidth utilization when the link delay, processing delay, etc., factors cannot be considered as trivial. To guarantee this assumption, more bandwidth has to be reserved as a guard band for each cycle, and the effective bandwidth for DetNet service will be greatly reduced.

CSQF improves on CQF by explicitly specifying the sending cycles at every node along the path. This relieves the limitation that the

sending (at the upstream node) and receiving (at the downstream node) have to be in the same cycle. For CSQF, the cycle to use depends on traffic planning and path calculation. The path calculation will consider the available cycle resources, bandwidth, and delay constraints.

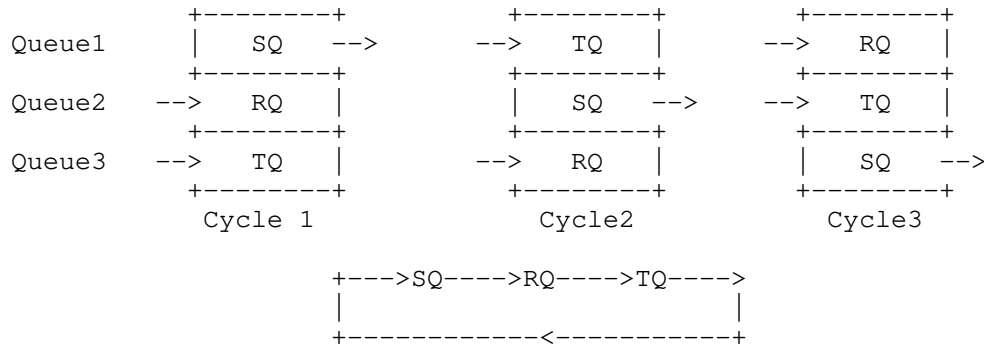


Figure 2: CSQF Queuing Model

For CSQF, three queues (in theory, two or more queues work as well) for each output interface are used. During a particular cycle, only one queue is open and the packets in that queue will be transmitted. This queue is called the sending queue (SQ). The other two queues are closed and can enqueue packets. One of them is called the receiving queue (RQ). The third queue is called the tolerating queue (TQ).

The RQ is used for receiving the packets that are expected to be transmitted in the next cycle. The TQ is used for tolerating the packets that come a bit early due to processing delay variation (processing jitter) or other reasons (e.g., packets are not transmitted as required by the traffic specification). Both RQ and TQ can have the capability to absorb a certain amount of processing jitter and traffic bursts. The upper bound of the absorbing capacity is $2T$. In order to increase the jitter/burst absorbing capacity, a four or more-queue model can be used. If the processing delay and traffic bursts are small, two-queue model works as well.

The roles of the three queues are not fixed, and on the contrary, they rotate with each cycle change. As showed in Figure 2, during cycle 1, queue 1 is SQ, queue 2 is RG and queue 3 is TQ; during cycle 2, queue 1 is TQ, queue 2 is SQ and queue 3 is RQ, during cycle 3, queue 1 is RQ, queue 2 is TQ and queue 3 is SQ. That means, for a particular queue, its role will rotate as "...->SQ->RQ->TQ->SQ->...", the starting role of a queue can be any one of the three roles.

In CSQF, a cycle corresponds to a queue. There are several ways to do cycle to queue mapping. The simplest mapping between cycles and queues is 1:1 mapping. There could be N:1 mapping, but that requires more identifiers, which in the case of segment routing, would require more SIDs. This document does not specify which mapping should be used. The mapping choice is left to the operator.

3.3. CSQF Timing Model

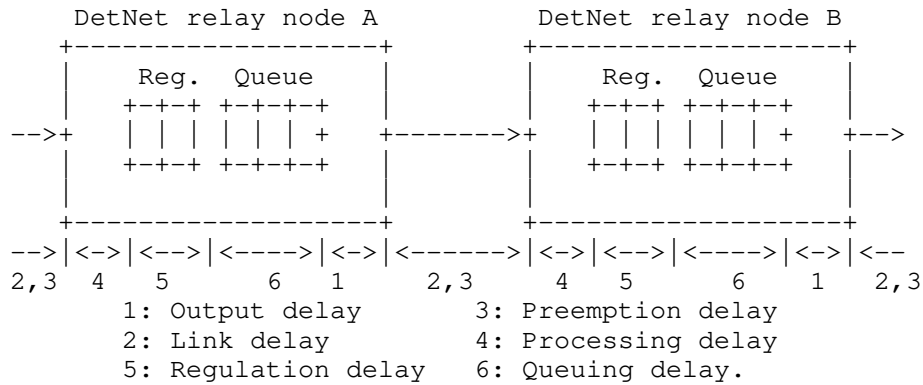


Figure 3: Timing model for DetNet

The DetNet timing model in Figure 3 is defined in [I-D.finn-detnet-bounded-latency]. It details the delays that a packet can experience from hop to hop. There are six delays, the detailed explanation of which can be found in [I-D.finn-detnet-bounded-latency]. This document simplifies the above model as follows:

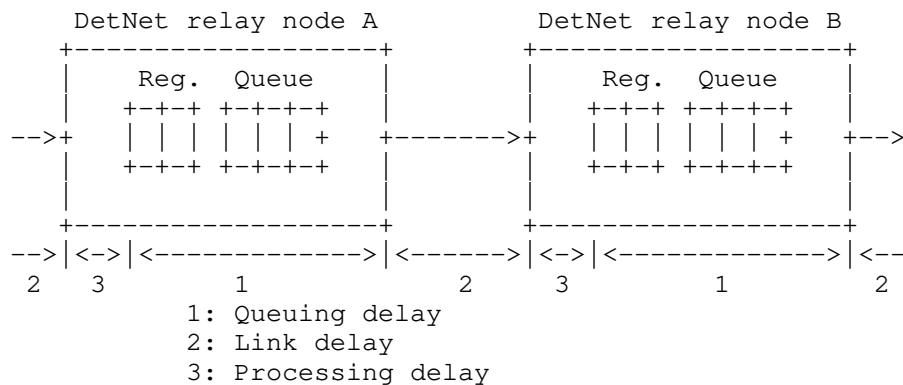


Figure 4: Simplified Timing model for DetNet

In this simplified timing model, only three delays are defined. The queuing delay in this new model includes the output delay, regulation delay, and queuing delay that are defined in the DetNet timing model (Figure 3). The link delay defined in this document includes the link delay and the preemption delay defined in [I-D.finn-detnet-bounded-latency]. The processing delay is the same as defined in [I-D.finn-detnet-bounded-latency].

To further simplify the model, it assumes that the link delay only depends on the distance of the link. Once the DetNet path has been determined, the link delay can be considered as constant. The processing delay and queuing delay are variable but have their upper bounds.

For the processing delay, there are two bounds: minimum processing delay (Min-P-Delay) and maximum processing delay (Max-P-Delay).

- o Thus, the maximum processing jitter (Max-P-Jitter) = Max-P-Delay - Min-P-Delay.

As described in Section 2.2, both the RQ and TQ can be used for absorbing processing jitter, and the upper bound of the absorbing capacity is $2T$. So, if the processing jitter is less than $2T$, the three-queue model can work. Otherwise, more buffer is needed to absorb the jitter, through increasing the duration of the cycle or by adding more queues. Increasing the duration of the cycles is equivalent to increasing the depth of the queues (adding more buffer for each queue).

With above, for CSQF, the delays experienced by a given packet are as follows:

- o The maximum end-to-end delay = Link delay + $N * (Max-P-Delay + 2T)$;
- o The maximum end-to-end jitter = $2T$;
- o Where N is the number of hops and T is the duration of a cycle.

3.4. Congestion Protection and Resource Reservation

Congestion protection is the key for bounded latency and zero congestion loss. An essential component of DetNet is Traffic Engineering (TE), so that dedicated resources can be reserved for the exclusive use of DetNet flows. To avoid congestion, two or more flows must be prevented from contending for the same resource. For normal TE, the critical resource is bandwidth, but in the case of CSQF, the critical resource is interface occupation time. Bandwidth

is an average value, which can generally guarantee the quality of service generally, but bursts and congestion may still occur. By comparison, the interface occupation time is an absolute value, which can avoid packet packets conflicting for the same resource by controller computation and time allocation for different flows. The unit of time allocation is the cycle, and a Traffic Specification, the flow transmission description, is necessary for the computation.

CSQF uses segment routing SIDs to carry the time allocation information (the cycle), and it ensures that a node can schedule different packets without conflict and forward the packets at the proper time. The resource reservation is not explicitly implemented by a control plane protocol, such as Resource Reservation Protocol - Traffic Engineering (RSVP-TE) or Stream Reservation Protocol (SRP). Rather, it is guaranteed by the SR controller, which maintains the status of different flows and time occupation of all the network devices in the domain. This is called the Virtual Resource Reservation (VRR) in this document.

3.5. An Example of CSQF

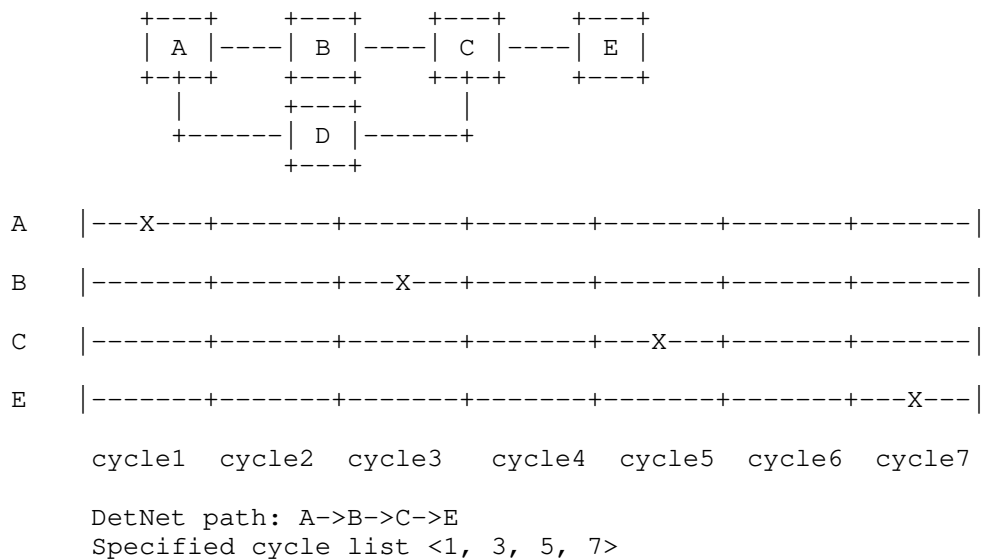


Figure 5: CSQF Example

As showed in Figure 5, there is a DetNet path (A->B->C->E), and a packet (X) is expected to be transmitted in cycle 1 at node A, in cycle 3 at node B, in cycle 5 at node C and in cycle 7 at node E. A

cycle list <1, 3, 5, 7> is attached to the packet, and the packet will be transmitted along the path as the specified cycles.

Given the topology as above, assume the duration of a cycle is 10us; the link delays between nodes are the same (e.g., 100us); the minimum processing delay at each node = 10us, the maximum processing delay at each node is 20us, so the maximum processing jitter is 10us.

For a given packet that is transmitted along the path(A->B->C->D->E), the experienced maximum end-to-end delay is:

$$\begin{aligned} & (N-1) * \text{link delay} + N * (\text{maximum processing delay} + 2T) \\ & = 3*100 + 4* 40 \\ & = 460 \text{ (us)} \end{aligned}$$

The maximum end-to-end jitter is always 2T (20us).

4. Segment Routing Extensions for CSQF

This document defines a new segment that is called a Cycle Segment, which is used to identify a cycle. A Cycle Segment is a local segment and is allocated from the Segment Routing Local Block (SRLB) [RFC8402].

A Cycle Segment has two meanings: 1) identify an interface/link, just like the adjacency segment does; 2) identify a cycle of the interface/link. To specify to which interface and in which cycle a packet should be transmitted, it just needs to attach a Cycle Segment to the packet. By attaching a list of Cycle Segments to a packet, it can not only implement the explicit route of the packet that is required by DetNet [I-D.ietf-detnet-architecture], but also specify the sending cycle at each node along the path without maintaining per-flow states at the intermediate and egress nodes. Hence, it naturally supports flow aggregation, and that allows DetNet to support large number of DetNet flows and scale to large networks.

Normally, several SR SIDs are required to be allocated for each CSQF capable interface. How many SIDs are allocated depends on how many cycles are used. Given a three-queue model and a 1:1 cycle to queue mapping is used, three SIDs will be allocated for each CSQF capable interface. For example, given node A, SR-MPLS SIDs 1001, 1002, and 1003 are allocated to one of its interfaces. SID 1001 identifies cycle 1, SID 1002 identifies cycle 2, SID 1003 identifies cycle 3.

The SR [RFC8402] can be instantiated on various data planes. There are two data-plane instantiations of SR: SR over MPLS (SR-MPLS) and

SR over IPv6 (SRv6). Both SR-MPLS and SRv6 SIDs can be used for CSQF cycle identification. The mapping (IGP extensions) between a cycle and a SID will be defined in a separate document.

4.1. Time Aware Adjacency Segment (TA-Adj-SID)

An Time Aware Adjacency segment is an IGP segment attached to a specified sending time of a unidirectional adjacency, which inheriting all the definitions of Adjacency segment defined in [RFC8402], adding new capability:

When a node binds a group of AT-Adj-SIDs V1-Vn to a local data-link L, the node MUST install the following FIB entry:

Incoming Active Segment: V1-Vn

Ingress Operation: NEXT

Egress Interface: L

When a node binds an TA-Adj-SID V1 to sending time: Cycle 1, the node MUST install the following Forwarding Time Base (FTB) entry:

Incoming Active Segment: V1

Sending Time: Cycle 1

Output Queue: Queue 1

So a packet with TA-Adj-SID V1 will be transmitted go through output queue 1 of egress interface L within cycle 1.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

7. Acknowledgements

The authors would like to thank Andrew G. Malis, Norman Finn for his review, suggestion and comments to this document.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

- [I-D.finn-detnet-bounded-latency]
Finn, N., Boudec, J., Mohammadpour, E., Zhang, J., Varga, B., and J. Farkas, "DetNet Bounded Latency", draft-finn-detnet-bounded-latency-03 (work in progress), March 2019.
- [I-D.geng-detnet-conf-yang]
Geng, X., Chen, M., Li, Z., and R. Rahman, "DetNet Configuration YANG Model", draft-geng-detnet-conf-yang-06 (work in progress), October 2018.
- [I-D.geng-detnet-info-distribution]
Geng, X., Chen, M., and Z. Li, "IGP-TE Extensions for DetNet Information Distribution", draft-geng-detnet-info-distribution-03 (work in progress), October 2018.
- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-12 (work in progress), March 2019.
- [IEEE802.1Qch]
IEEE, "IEEE, "Cyclic Queuing and Forwarding (IEEE Draft P802.1Qch)", 2017, <<http://www.ieee802.org/1/files/private/ch-drafts/>>.", 2016.
- [RFC8402] Filss, 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>>.

Authors' Addresses

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Zhenqiang Li
China Mobile

Email: lizhenqiang@chinamobile.com

SPRING Working Group
Internet-Draft
Intended Status: Informational
Expires: August 18, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
P. L. Ventre
CNIT
M. Chen
Huawei
February 14, 2019

In-band Performance Measurement for
Segment Routing Networks with MPLS Data Plane
draft-gandhi-spring-rfc6374-srpm-mpls-00

Abstract

RFC 6374 specifies protocol mechanisms to enable the efficient and accurate measurement of packet loss, one-way and two-way delay, as well as related metrics such as delay variation in MPLS networks using probe messages. This document reviews how these mechanisms can be used for Delay and Loss Performance Measurements (PM) in Segment Routing (SR) networks with MPLS data plane (SR-MPLS), for both SR links and end-to-end SR Policies. The performance measurements for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and are advertised in the network using the routing protocol extensions.

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 <http://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."

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	3
2.1. Abbreviations	3
2.2. Reference Topology	4
2.3. In-band Probe Messages	5
3. Probe Query and Response Packets	5
3.1. Probe Packet Header for SR-MPLS Policies	5
3.2. Probe Packet Header for SR-MPLS Links	6
3.3. Probe Response Message for SR-MPLS Links and Policies	6
3.3.1. One-way Measurement Probe Response Message	6
3.3.2. Two-way Measurement Probe Response Message	6
4. Performance Delay Measurement	7
4.1. Delay Measurement Message Format	7
4.2. Timestamps	7
5. Performance Loss Measurement	7
5.1. Loss Measurement Message Format	8
6. Performance Measurement for P2MP SR Policies	8
7. ECMP for SR-MPLS Policies	8
8. SR Link Extended TE Metrics Advertisements	8
9. Security Considerations	9
10. IANA Considerations	9
11. References	9
11.1. Normative References	9
11.2. Informative References	9
Acknowledgments	11
Contributors	11
Authors' Addresses	11

1. Introduction

Service provider's ability to satisfy Service Level Agreements (SLAs) depend on the ability to measure and monitor performance metrics for packet loss and one-way and two-way delay, as well as related metrics such as delay variation. The ability to monitor these performance metrics also provides operators with greater visibility into the performance characteristics of their networks, thereby facilitating planning, troubleshooting, and network performance evaluation.

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics in MPLS networks using probe messages. The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. However, mechanisms defined in [RFC6374] are more suitable for Segment Routing (SR) when using MPLS data plane (SR-MPLS). The [RFC6374] also supports IEEE 1588 timestamps [IEEE1588] and "direct mode" Loss Measurement (LM), which are required in SR networks.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe replies over an UDP return path when receiving RFC 6374 based probe queries. These procedures can be used to send out-of-band PM replies for both SR links and SR Policies [I-D.spring-segment-routing-policy] for one-way measurement.

This document reviews how probe based mechanisms defined in [RFC6374] can be used for Delay and Loss Performance Measurements (PM) in SR networks with MPLS data plane, for both SR links and end-to-end SR Policies. The performance measurements for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and are advertised in the network using routing protocol extensions.

2. Conventions Used in This Document

2.1. Abbreviations

ACH: Associated Channel Header.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

G-ACh: Generic Associated Channel (G-ACh).

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

TC: Traffic Class.

TE: Traffic Engineering.

URO: UDP Return Object.

2.2. Reference Topology

In the reference topology shown in Figure 1, the querier node R1 initiates a performance measurement probe query and the responder node R5 sends a probe response for the query message received. The probe response is typically sent to the querier node R1. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].

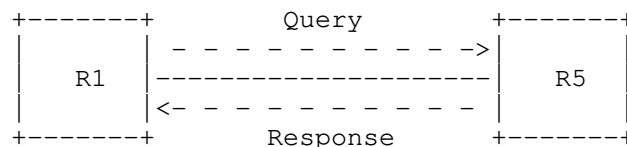


Figure 1: Reference Topology

Both delay and loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in Section 2.4 of [RFC6374]. Transmit and Receive packet loss measurements are defined in Section 2.2 and Section 2.6 of [RFC6374]. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

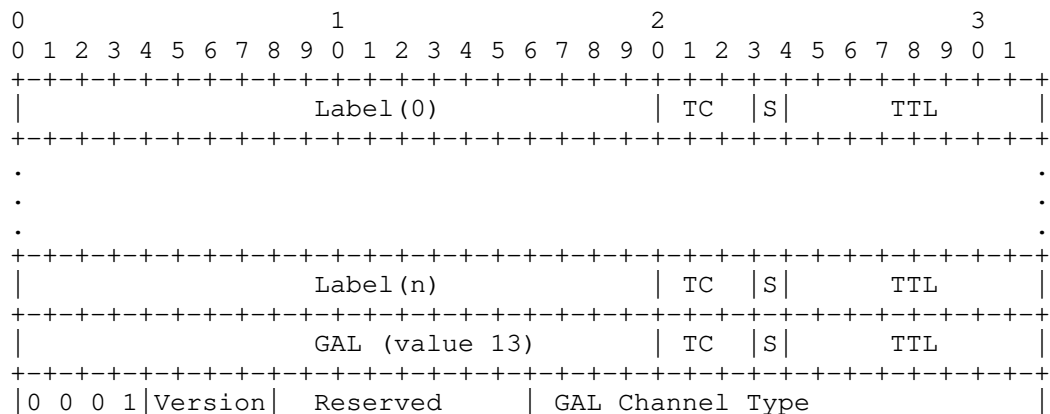
2.3. In-band Probe Messages

For both Delay and Loss measurements for links and SR Policies, no PM session is created on the responder node. The probe messages for Delay measurement are sent in-band by the querier node to measure the delay experienced by the actual traffic flowing on the links and SR Policies. For Loss measurement, in-band probe messages are used to collect the traffic counter for the incoming link or incoming SID on which the probe query message is received at the responder node R5 (as it has no PM session state present on the node).

3. Probe Query and Response Packets

3.1. Probe Packet Header for SR-MPLS Policies

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for an end-to-end measurement for SR Policy contains SR-MPLS label stack [I-D.spring-segment-routing-policy], with the G-ACh Label (GAL) at the bottom of the stack. The GAL is followed by an Associated Channel Header (ACH), which identifies the message type and the message payload following the ACH as shown in Figure 2.



```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 2: Probe Packet Header for an End-to-end SR-MPLS Policy

The SR-MPLS label stack can be empty to indicate Implicit NULL label case.

3.2. Probe Packet Header for SR-MPLS Links

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for SR-MPLS links contains G-ACh Label (GAL). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 3.

```

0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               GAL (value 13)               | TC | S |               TTL               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 0 0 1 | Version |   Reserved   | GAL Channel Type |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 3: Probe Packet Header for an SR-MPLS Link

3.3. Probe Response Message for SR-MPLS Links and Policies

3.3.1. One-way Measurement Probe Response Message

For one-way performance measurement [RFC7679], the PM querier node can receive "out-of-band" probe replies by properly setting the UDP Return Object (URO) TLV in the probe query message. The URO TLV (Type=131) is defined in [RFC7876] and includes the UDP-Destination-Port and IP Address. In particular, if the querier sets its own IP address in the URO TLV, the probe response is sent back by the responder node to the querier node. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

3.3.2. Two-way Measurement Probe Response Message

For two-way performance measurement [RFC6374], when using a bidirectional channel, the probe response message is sent back to the querier node in-band on the reverse direction SR Link or SR Policy

using a message with format similar to their probe query message. In this case, the "control code" in the probe query message is set to "in-band response requested".

A Path Segment Identifier [I-D.spring-mpls-path-segment] of the forward SR Policy can be used to find the reverse SR Policy and to send back the probe response message.

4. Performance Delay Measurement

4.1. Delay Measurement Message Format

As defined in [RFC6374], MPLS DM probe query and response messages use Associated Channel Header (ACH) (value 0x000C for delay measurement) [RFC6374], which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR Policies, the same MPLS DM ACH value is used.

The DM message payload as defined in Section 3.2 of [RFC6374] is used for SR-MPLS delay measurement, for both SR links and end-to-end SR Policies.

4.2. Timestamps

The Section 3.4 of [RFC6374] defines timestamp format that can be used for delay measurement. The IEEE 1588 Precision Time Protocol (PTP) timestamp format [IEEE1588] is used by default as described in Appendix A of [RFC6374], but it may require hardware support. As an alternative, Network Time Protocol (NTP) timestamp format can also be used [RFC6374].

Note that for one-way delay measurement, clock synchronization between the querier and responder nodes using the methods detailed in [RFC6374] is required. The two-way delay measurement does not require clock synchronization between the querier and responder nodes.

5. Performance Loss Measurement

The LM protocol can perform two distinct kinds of loss measurement as described in Section 2.9.8 of [RFC6374].

- o In inferred mode, LM will measure the loss of specially generated test messages in order to infer the approximate data plane loss level. Inferred mode LM provides only approximate loss accounting.

- o In direct mode, LM will directly measure data plane packet loss. Direct mode LM provides perfect loss accounting, but may require hardware support.

For both of these modes of LM, Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] is used for accounting received traffic on the egress node of the SR-MPLS Policy.

5.1. Loss Measurement Message Format

As defined in [RFC6374], MPLS LM probe query and response messages use Associated Channel Header (ACH) (value 0x000A for direct loss measurement or value 0x000B for inferred loss measurement), which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR Policies, the same MPLS LM ACH value is used.

The LM message payload as defined in Section 3.1 of [RFC6374] is used for SR-MPLS loss measurement, for both SR links and end-to-end SR Policies.

6. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement reviewed in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

The responder node adds the "Source Address" TLV (Type 130) [RFC6374] in the probe response message. This TLV allows the querier node to identify the responder nodes of the P2MP SR Policy.

7. ECMP for SR-MPLS Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM messages using [RFC6374] can not traverse all ECMP paths to measure performance delay of all paths of an SR Policy.

8. SR Link Extended TE Metrics Advertisements

The extended TE metrics for SR link delay and loss computed using the performance measurement procedures reviewed in this document can be advertised in the routing domain as follows:

- o For OSPF, ISIS, and BGP-LS, protocol extensions defined in [RFC7471], [RFC7810] [I-D.lsr-isis-rfc7810bis], and [I-D.idr-te-pm-bgp] are used, respectively for advertising the extended TE link metrics in the network.
- o The extended TE link delay metrics advertised are minimum-delay, maximum-delay, average-delay, and delay-variance for one-way.
- o The delay-variance metric is computed as specified in Section 4.2 of [RFC5481].
- o The one-way delay metrics can be computed using two-way measurement by dividing the measured delay values by 2.
- o The extended TE link loss metric advertised is one-way percentage packet loss.

9. Security Considerations

This document reviews the procedures for performance delay and loss measurement for SR-MPLS networks, for both links and end-to-end SR Policies using the mechanisms defined in [RFC6374]. This document does not introduce any additional security considerations other than those covered in [RFC6374], [RFC7471], [RFC7810], and [RFC7876].

10. IANA Considerations

This document does not require any IANA actions.

11. References

11.1. Normative References

- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks", RFC 6374, September 2011.
- [RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.

11.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC7679] Almes, G., et al., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", RFC 7679, January 2016.
- [RFC7471] Giacalone, S., et al., "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.
- [RFC7810] Previdi, S., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, May 2016.
- [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>>.
- [I-D.lsr-isis-rfc7810bis] Ginsberg, L., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", draft-ietf-lsr-isis-rfc7810bis, work in progress.
- [I-D.idr-te-pm-bgp] Ginsberg, L. Ed., et al., "BGP-LS Advertisement of IGP Traffic Engineering Performance Metric Extensions", draft-ietf-idr-te-pm-bgp, work in progress.
- [I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment, work in progress.

Acknowledgments

The authors would like to thank Greg Mirsky for providing many useful comments and suggestions.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

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

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

SPRING Working Group
Internet-Draft
Intended Status: Standards Track
Expires: April 6, 2020

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
M. Chen
Huawei
October 4, 2019

Performance Measurement for
Segment Routing Networks with MPLS Data Plane
draft-gandhi-spring-rfc6374-srpm-mpls-02

Abstract

Segment Routing (SR) leverages the source routing paradigm. RFC 6374 specifies protocol mechanisms to enable the efficient and accurate measurement of packet loss, one-way and two-way delay, as well as related metrics such as delay variation in MPLS networks using probe messages. This document utilizes these mechanisms for Performance Delay and Loss Measurements in Segment Routing (SR) networks with MPLS data plane (SR-MPLS), for both SR links and end-to-end SR Policies. In addition, this document defines Return Path TLV for two-way performance measurement and Block Number TLV for loss measurement.

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 <http://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."

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
3. Overview	5
4. Probe Query and Response Packets	6
4.1. Probe Packet Header for SR-MPLS Policies	6
4.2. Probe Packet Header for SR-MPLS Links	6
4.3. Probe Response Message for SR-MPLS Links and Policies	7
4.3.1. One-way Measurement Mode	7
4.3.2. Two-way Measurement Mode	7
4.3.2.1. Return Path TLV	7
4.3.3. Loopback Measurement Mode	9
5. Performance Delay Measurement	9
5.1. Delay Measurement Message Format	9
5.2. Timestamps	9
6. Performance Loss Measurement	10
6.1. Loss Measurement Message Format	10
6.1.1. Block Number TLV	11
7. Performance Measurement for P2MP SR Policies	11
8. ECMP for SR-MPLS Policies	12
9. SR Link Extended TE Metrics Advertisements	12
10. Security Considerations	13
11. IANA Considerations	13
12. References	14
12.1. Normative References	14
12.2. Informative References	14
Acknowledgments	17
Contributors	17
Authors' Addresses	17

1. Introduction

Service provider's ability to satisfy Service Level Agreements (SLAs) depend on the ability to measure and monitor performance metrics for packet loss and one-way and two-way delay, as well as related metrics such as delay variation. The ability to monitor these performance metrics also provides operators with greater visibility into the performance characteristics of their networks, thereby facilitating planning, troubleshooting, and network performance evaluation.

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics in MPLS networks using probe messages. The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. However, mechanisms defined in [RFC6374] are more suitable for Segment Routing (SR) when using MPLS data plane (SR-MPLS). [RFC6374] also supports IEEE 1588 timestamps [IEEE1588] and "direct mode" Loss Measurement (LM), which are required in SR networks.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe replies over an UDP return path when receiving RFC 6374 based probe queries. These procedures can be used to send out-of-band PM replies for both SR-MPLS links and Policies [I-D.spring-segment-routing-policy] for one-way measurement.

This document utilizes the probe-based mechanisms defined in [RFC6374] for Performance Delay and Loss Measurements in SR networks with MPLS data plane, for both SR links and end-to-end SR Policies. In addition, this document defines Return Path TLV for two-way performance measurement and Block Number TLV for loss measurement. The Performance Measurements (PM) for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and can be advertised in the network using the routing protocol extensions.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

ACH: Associated Channel Header.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

G-ACh: Generic Associated Channel (G-ACh).

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

TC: Traffic Class.

TE: Traffic Engineering.

URO: UDP Return Object.

2.3. Reference Topology

In the reference topology shown in Figure 1, the sender node R1 initiates a performance measurement probe query and the responder node R5 sends a probe response for the query message received. The probe response is typically sent back to the sender node R1. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].

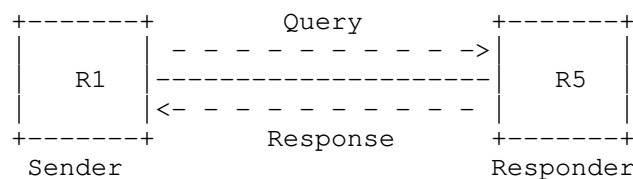


Figure 1: Reference Topology

3. Overview

One-way delay and two-way delay measurement procedure defined in Section 2.4 of [RFC6374] are used. Transmit and Receive packet loss measurement procedures defined in Section 2.2 and Section 2.6 of [RFC6374] are used. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss. For both links and end-to-end SR Policies, no PM session for delay or loss measurement is created on the responder node R5 [RFC6374].

For Performance Measurement, probe query and response messages are sent as following:

- o For Delay Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the links and SR Policies.
- o For Loss Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the responder node (incoming link or incoming SID needed since the

responder node does not have PM session state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.spring-ioam-sr-mpls] are used to carry PM information in-band as part of the data traffic, and are outside the scope of this document.

4. Probe Query and Response Packets

4.1. Probe Packet Header for SR-MPLS Policies

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for an end-to-end measurement for SR Policy contains SR-MPLS label stack [I-D.spring-segment-routing-policy], with the G-ACh Label (GAL) at the bottom of the stack (with S=1). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 2.

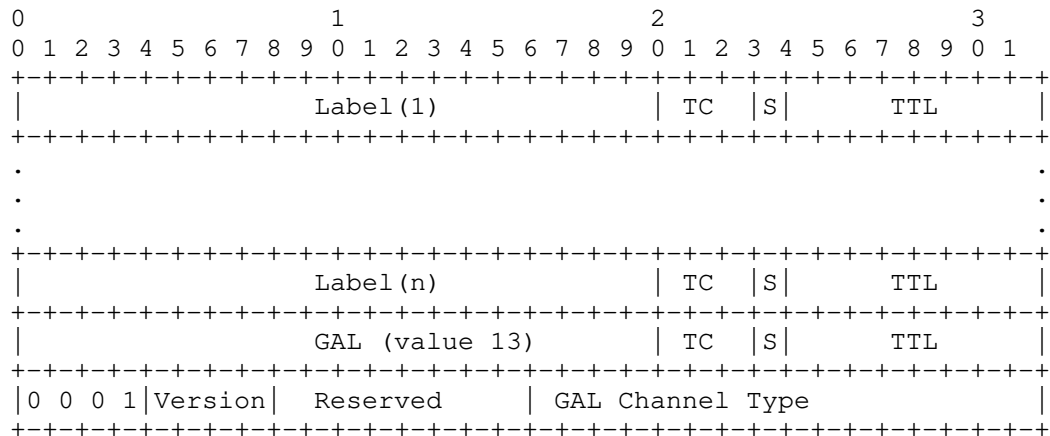


Figure 2: Probe Packet Header for an End-to-end SR-MPLS Policy

The SR-MPLS label stack can be empty (as shown in Figure 3) to indicate Implicit NULL label case.

4.2. Probe Packet Header for SR-MPLS Links

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for SR-MPLS links contains G-ACh Label (GAL)

(with S=1). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 3.

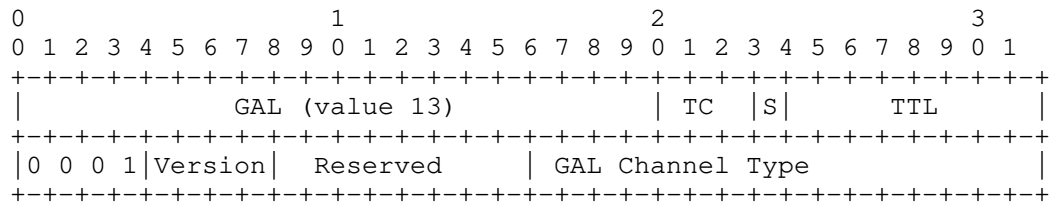


Figure 3: Probe Packet Header for an SR-MPLS Link

4.3. Probe Response Message for SR-MPLS Links and Policies

4.3.1. One-way Measurement Mode

In one-way performance measurement mode [RFC7679], the PM sender node can receive "out-of-band" probe replies by properly setting the UDP Return Object (URO) TLV in the probe query message. The URO TLV (Type=131) is defined in [RFC7876] and includes the UDP-Destination-Port and IP Address. In particular, if the sender sets its own IP address in the URO TLV, the probe response is sent back by the responder node to the sender node. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

4.3.2. Two-way Measurement Mode

In two-way performance measurement mode [RFC6374], when using a bidirectional path, the probe response message is sent back to the sender node on the congruent path of the data traffic on the reverse direction SR Link or SR Policy using a message with format similar to their probe query message. In this case, the "control code" in the probe query message is set to "in-band response requested".

A Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] of the forward SR-MPLS Policy can be used to find the reverse SR-MPLS Policy and to send back the probe response message for two-way measurement.

4.3.2.1. Return Path TLV

For two-way performance measurement, the responder node needs to send

the probe response message on a specific reverse path. This way the destination node does not require any additional SR Policy state. The sender node can request the responder node to send a response message back on a given reverse path (e.g. co-routed path for two-way measurement).

[RFC6374] defines DM and LM probe query messages that can include one or more optional TLVs. New TLV Type (TBA1) is defined in this document for Return Path to carry reverse path for probe response messages (in the payload of the message). The format of the Return Path TLV is shown in Figure 7A and 7B:

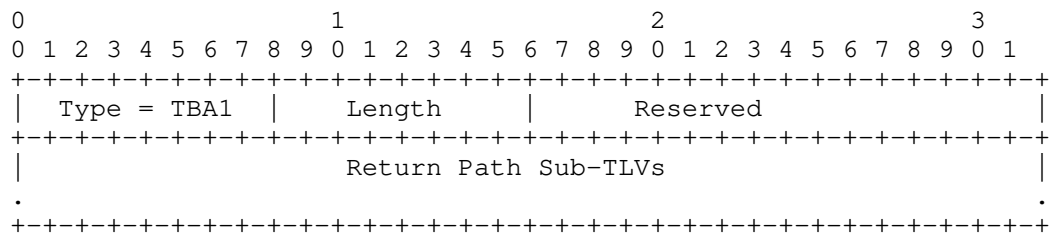


Figure 7A: Return Path TLV

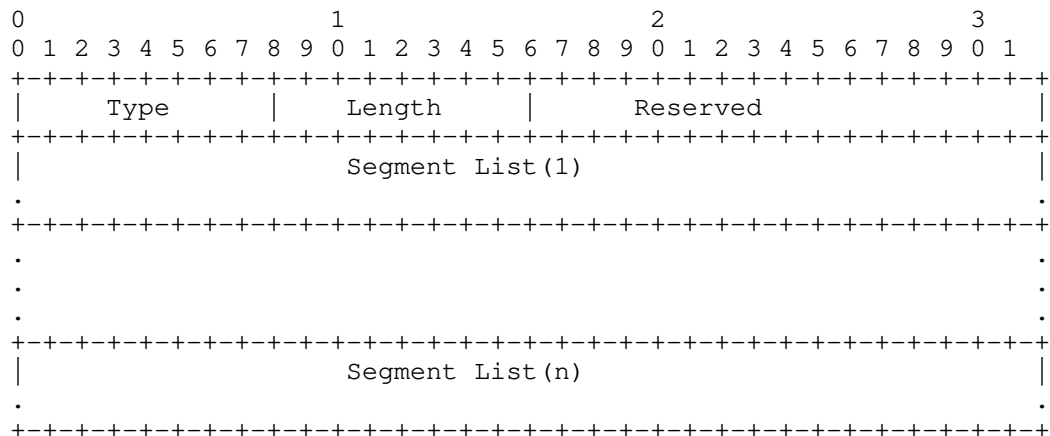


Figure 7B: Segment List Sub-TLV in Return Path TLV

The Segment List Sub-TLV in the Return Path TLV can be one of the following Types:

- o Type (value 1): Respond back on Incoming Interface (Layer-3 and

Layer-2) (Segment List is Empty)

- o Type (value 2): SR-MPLS Segment List (Label Stack) of the Reverse SR Path
- o Type (value 3): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

The Return Path TLV is optional. The PM sender node MUST only insert one Return Path TLV in the probe query message and the responder node MUST only process the first Return Path TLV in the probe query message and ignore other Return Path TLVs if present. The responder node MUST send probe response message back on the reverse path specified in the Return Path TLV and MUST NOT add Return Path TLV in the probe response message.

4.3.3. Loopback Measurement Mode

The Loopback measurement mode defined in Section 2.8 of [RFC6374] can be used to measure round-trip delay for a bidirectional SR Path. The probe query messages in this case carries the reverse SR Path label stack as part of the MPLS header. The GAL is still carried at the bottom of the label stack (with S=1). The responder node does not process the PM probe messages and generate response messages.

5. Performance Delay Measurement

5.1. Delay Measurement Message Format

As defined in [RFC6374], MPLS DM probe query and response messages use Associated Channel Header (ACH) (value 0x000C for delay measurement) [RFC6374], which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR-MPLS Policies, the same MPLS DM ACH value is used.

The DM message payload as defined in Section 3.2 of [RFC6374] is used for SR-MPLS delay measurement, for both SR links and end-to-end SR Policies.

5.2. Timestamps

The Section 3.4 of [RFC6374] defines timestamp format that can be used for delay measurement. The IEEE 1588 Precision Time Protocol (PTP) timestamp format [IEEE1588] is used by default as described in Appendix A of [RFC6374], preferred with hardware support. As an alternative, Network Time Protocol (NTP) timestamp format can also be

measurement or value 0x000B for inferred loss measurement), which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR-MPLS Policies, the same MPLS LM ACH value is used.

The LM message payload as defined in Section 3.1 of [RFC6374] is used for SR-MPLS loss measurement, for both SR links and end-to-end SR Policies.

6.1.1. Block Number TLV

The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (or color) of the traffic counters carried by the probe query and response messages. Probe query and response messages specified in [RFC6374] for Loss Measurement do not define any means to carry the Block Number.

[RFC6374] defines probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA2) is defined in this document to carry Block Number (16-bit) for the traffic counters in the probe query and response messages for loss measurement. The format of the Block Number TLV is shown in Figure 5:

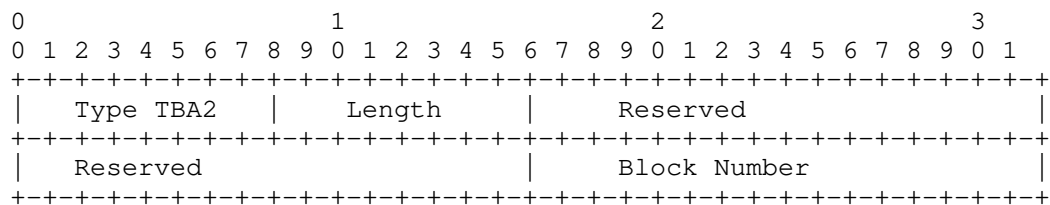


Figure 5: Block Number TLV

The Block Number TLV is optional. The PM sender node SHOULD only insert one Block Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Block Number TLV from the probe query messages and ignore other Block Number TLVs if present. In both probe query and response messages, the counters MUST belong to the same Block Number.

7. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement reviewed in this document for Point-to-Point (P2P) SR-MPLS Policies [I-D.spring-segment-routing-policy] are also equally applicable to the Point-to-Multipoint (P2MP) SR-MPLS Policies [I-D.spring-sr-p2mp-policy] as following:

- o The sender root node sends probe query messages using the either Spray P2MP segment or TreeSID P2MP segment defined in [I-D.spring-sr-p2mp-policy] over the P2MP SR Policy as shown in Figure 6.
- o Each responder leaf node adds the "Source Address" TLV (Type 130) [RFC6374] with its IP address in the probe response messages. This TLV allows the sender root node to identify the responder leaf nodes of the P2MP SR Policy.
- o The P2MP root node measures the end-to-end delay and loss performance for each P2MP leaf node.

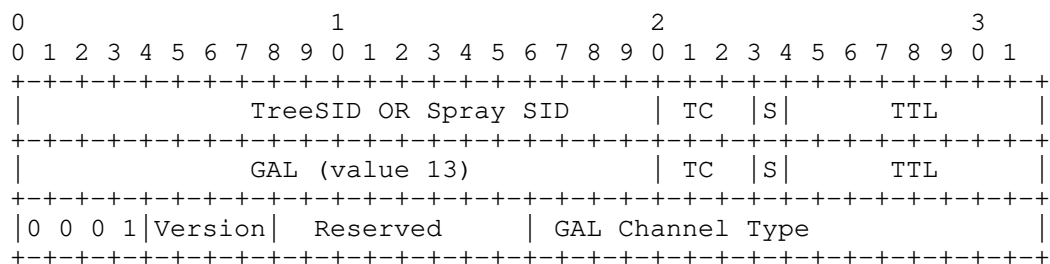


Figure 6: With P2MP Segment Identifier for SR-MPLS Policy

8. ECMP for SR-MPLS Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. For SR-MPLS Policy, entropy label [RFC6790] can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the ECMP path taken by them.

9. SR Link Extended TE Metrics Advertisements

The extended TE metrics for SR link delay and loss computed using the performance measurement procedures reviewed in this document can be advertised in the routing domain as follows:

- o For OSPF, ISIS, and BGP-LS, protocol extensions defined in [RFC7471], [RFC8570], and [RFC8571] are used, respectively for advertising the extended TE link metrics in the network.
- o The extended TE link delay metrics advertised are minimum-delay, maximum-delay, average-delay, and delay-variance for one-way.
- o The delay-variance metric is computed as specified in Section 4.2 of [RFC5481].
- o The one-way delay metrics can be computed using two-way delay measurement or round-trip delay measurement from loopback mode by dividing the measured delay values by 2.
- o The extended TE link loss metric advertised is one-way percentage packet loss.
- o Similarly, the extended TE link delay and loss metrics are advertised for Layer 2 bundle members in ISIS [I-D.lsr-ospf-l2bundles] and OSPF [I-D.isis-l2bundles] using the same mechanisms defined in [RFC8570] and [RFC7471], respectively.

10. Security Considerations

This document reviews the procedures for performance delay and loss measurement for SR-MPLS networks, for both links and end-to-end SR Policies using the mechanisms defined in [RFC6374] and [RFC7876]. This document does not introduce any additional security considerations other than those covered in [RFC6374], [RFC7471], [RFC8570], [RFC8571], and [RFC7876].

11. IANA Considerations

IANA is requested to allocate a value for the following Return Path TLV Type for RFC 6374 to be carried in PM probe query messages:

- o Type TBA1: Return Path TLV

IANA is requested to allocate the values for the following Sub-TLV Types for the Return Path TLV for RFC 6374.

- o Type (value 1): Respond back on Incoming Interface (Layer-3 and Layer-2) (Segment List is Empty)
- o Type (value 2): SR-MPLS Segment List (Label Stack) of the Reverse

SR Path

- o Type (value 3): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

IANA is also requested to allocate a value for the following Block Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for loss measurement:

- o Type TBA2: Block Number TLV

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks", RFC 6374, September 2011.
- [RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

12.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and

- L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [RFC7679] Almes, G., et al., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", RFC 7679, January 2016.
- [RFC7471] Giacalone, S., et al., "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.
- [RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, July 2018.
- [RFC8570] Ginsberg, L. Ed., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, March 2019.
- [RFC8571] Ginsberg, L. Ed., et al., "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, March 2019.
- [I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.pce-binding-label-sid] Filsfils, C., et al., "Carrying Binding Label/Segment-ID in PCE-based Networks", draft-ietf-pce-binding-label-sid, work in progress.
- [I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment, work in progress.
- [I-D.spring-ioam-sr-mpls] Gandhi, R. Ed., et al., "Segment Routing with MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-spring-ioam-sr-mpls, work in progress.
- [I-D.lsr-ospf-l2bundles] Talaulikar, K., et al., "Advertising L2 Bundle Member Link Attributes in OSPF", draft-ketant-lsr-ospf-l2bundles, work in progress.

[I-D.isis-l2bundles] Ginsberg, L., et al., "Advertising L2 Bundle
Member Link Attributes in IS-IS",
draft-ietf-isis-l2bundles, work in progress.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for the performance measurement in segment routing networks. The authors would like to thank Greg Mirsky for providing many useful comments and suggestions. The authors would also like to thank Stewart Bryant, Sam Aldrin, and Rajiv Asati for their review comments.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

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

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

SPRING Working Group
Internet-Draft
Intended Status: Standards Track
Expires: August 18, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
P. L. Ventre
CNIT
M. Chen
Huawei
February 14, 2019

In-band Performance Measurement Using UDP Path
for Segment Routing Networks
draft-gandhi-spring-rfc6374-srpm-udp-00

Abstract

Segment Routing (SR) is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedures for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement. The procedure uses the RFC 6374 defined mechanisms for Delay and Loss performance measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both links and end-to-end measurement for SR Policies. In addition, this document defines Return Path TLV for two-way performance measurement and Block Number TLV for loss measurement.

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 <http://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."

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
2.4. In-band Probe Messages	6
3. Probe Messages	6
3.1. Probe Query Message	6
3.1.1. Delay Measurement Probe Query Message	7
3.1.2. Loss Measurement Probe Query Message	7
3.1.2.1. Block Number TLV	8
3.1.3. Probe Query for SR Links	9
3.1.4. Probe Query for End-to-end Measurement for SR Policy	9
3.1.4.1. Probe Query Message for SR-MPLS Policy	9
3.1.4.2. Probe Query Message for SRv6 Policy	9
3.2. Probe Response Message	10
3.2.1. One-way Measurement for SR Link and end-to-end SR Policy	11
3.2.1.1. Probe Response Message to Controller	11
3.2.2. Two-way Measurement for SR Links	11
3.2.3. Two-way End-to-end Measurement for SR Policy	12
3.2.3.1. Return Path TLV	12
3.2.3.2. Probe Response Message for SR-MPLS Policy	13
3.2.3.3. Probe Response Message for SRv6 Policy	14
4. Performance Measurement for P2MP SR Policies	14
5. ECMP Support	14
6. Sequence Numbers	15
6.1. Sequence Number TLV in Unauthenticated Mode	15
6.2. Sequence Number TLV in Authenticated Mode	16
7. Security Considerations	17
8. IANA Considerations	17
9. References	18

9.1. Normative References	18
9.2. Informative References	19
Acknowledgments	21
Contributors	21
Authors' Addresses	21

1. Introduction

Segment Routing (SR) technology greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source, transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined path using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. These protocols rely on control channel signaling to establish a test channel over an UDP path. These protocols lack support for IEEE 1588 timestamp [IEEE1588] format and direct-mode Loss Measurement (LM), which are required in SR networks [RFC6374]. The Simple Two-way Active Measurement Protocol (STAMP) [I-D.ippm-stamp] alleviates the control channel signaling by using configuration data model to provision test channels. In addition, the STAMP supports IEEE 1588 timestamp format for Delay Measurement (DM). The TWAMP Light from broadband forum [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer Edge IP networks. [Y1731] specifies the mechanisms to carry OAM messages specifically for Ethernet networks that include Ethernet Frame Delay and Loss measurements.

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics and can be used in SR networks with MPLS data plane [I-D.spring-rfc6374-srpm-mpls]. [RFC6374] addresses the limitations of the IP based performance measurement protocols as specified in Section 1 of [RFC6374]. The [RFC6374] requires data plane to support MPLS Generic Associated Channel Label (GAL) and Generic Associated Channel (G-Ach), which may not be supported on all nodes in the network.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe response

messages over an UDP return path for RFC 6374 based probe queries. [RFC7876] can be used to send out-of-band PM probe responses in both SR-MPLS and SRv6 networks for one-way performance measurement.

For SR Policies, there are ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Existing PM protocols (e.g. RFC 6374) do not define handling for ECMP forwarding paths in SR networks.

For two-way measurements for SR Policies, there is a need to specify a return path in the form of a Segment List in PM probe query messages without requiring any SR Policy state on the destination node. Existing protocols do not have such mechanisms to specify return path in the PM probe query messages.

This document specifies a procedure for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement that does not require to bootstrap PM sessions. The procedure uses RFC 6374 defined mechanisms for Delay and Loss PM and unless otherwise specified, the procedures from RFC 6374 are not modified. The procedure specified is applicable to both SR-MPLS and SRv6 data planes. The procedure can be used for both SR links and end-to-end performance measurement for SR Policies. This document also defines mechanisms for handling Equal Cost Multipaths (ECMPs) of SR Policies for performance delay measurement. In addition, this document defines Return Path TLV for two-way performance measurement, Block Number TLV for loss measurement and Sequence Number TLV.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

ACH: Associated Channel Header.

BSID: Binding Segment ID.

DFlag: Data Format Flag.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

G-ACh: Generic Associated Channel (G-ACh).

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

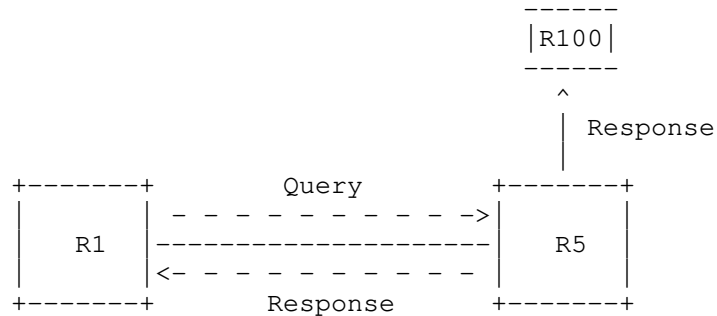
TWAMP: Two-Way Active Measurement Protocol.

URO: UDP Return Object.

2.3. Reference Topology

In the reference topology, the querier node R1 initiates a probe query for performance measurement and the responder node R5 sends a probe response for the query message received. The probe response may be sent to the querier node R1 or to a controller node R100. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to

node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].



Reference Topology

Both Delay and Loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in Section 2.4 of [RFC6374]. Transmit and Receive packet loss measurements are defined in Section 2.2 and Section 2.6 of [RFC6374]. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

2.4. In-band Probe Messages

For both Delay and Loss measurements for links and SR Policies, no PM session is created on the responder node. The probe messages for Delay measurement are sent in-band by the querier node to measure the delay experienced by the actual traffic flowing on the links and SR Policies. For Loss measurement, in-band probe messages are used to collect the traffic counter for the incoming link or incoming SID on which the probe query message is received at the responder node R5 as it has no PM session state present on the node. The performance measurement for Delay and Loss using out-of-band probe query messages are outside the scope of this document.

3. Probe Messages

3.1. Probe Query Message

In this document, UDP path is used for Delay and Loss measurements for SR links and end-to-end SR Policies. A user-configured UDP port is used for identifying PM probe packets that does not require to bootstrap PM sessions. A UDP port number from the Dynamic and/or

Private Ports range 49152-65535 is used as the destination UDP port. This approach is similar to the one defined in STAMP protocol [I-D.ippm-stamp]. The IPv4 TTL or IPv6 Hop Limit field of the IP header MUST be set to 255.

3.1.1. Delay Measurement Probe Query Message

The message content for Delay Measurement probe query message using UDP header [RFC768] is shown in Figure 1. The DM probe query message is sent with user-configured Destination UDP port number. The Source UDP port can be set to the same value for two-way delay measurement as indication of query and response is present in the message. The DM probe query message contains the payload for delay measurement defined in Section 3.2 of [RFC6374].

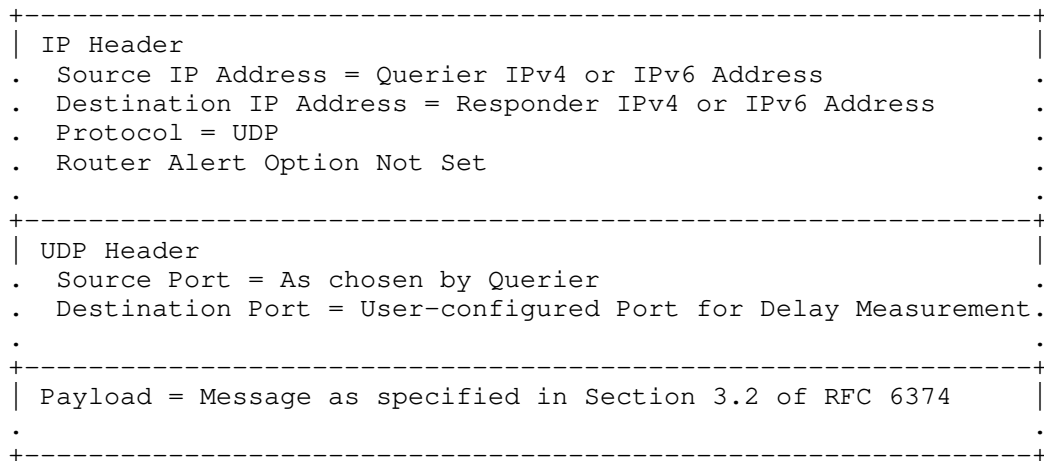
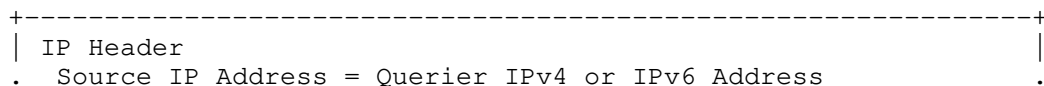


Figure 1: DM Probe Query Message

3.1.2. Loss Measurement Probe Query Message

The message content for Loss measurement probe query message using UDP header [RFC768] is shown in Figure 2. As shown, the LM probe query message is sent with user-configured Destination UDP port number. The Source UDP port can be set to the same value for two-way loss measurement as indication of query and response is present in the message. The LM probe query message contains the payload for loss measurement defined in Section 3.1 of [RFC6374].



```

. Destination IP Address = Responder IPv4 or IPv6 Address      .
. Protocol = UDP                                              .
. Router Alert Option Not Set                                .
.                                                            .
+-----+
| UDP Header                                                    |
. Source Port = As chosen by Querier                          .
. Destination Port = User-configured Port for Loss Measurement .
.                                                            .
+-----+
| Payload = Message as specified in Section 3.1 of RFC 6374    |
.                                                            .
+-----+

```

Figure 2: LM Probe Query Message

The Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] of the SR Policy is used for accounting received traffic on the egress node for loss measurement.

3.1.2.1. Block Number TLV

The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (or color) of the traffic counters carried by the probe query and response messages. Probe query and response messages specified in [RFC6374] for Loss Measurement do not define any means to carry the Block Number.

[RFC6374] defines probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA2) is defined in this document to carry Block Number (32-bit) for the traffic counters in the probe query and response messages for loss measurement. The format of the Block Number TLV is shown in Figure 11:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type TBA2  |   Length   |   Reserved   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Block Number                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 11: Block Number TLV

The Block Number TLV is optional. The PM querier node SHOULD only insert one Block Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Block Number TLV from the probe query messages and ignore other Block

Number TLVs if present. In both probe query and response messages, the counters MUST belong to the same Block Number.

3.1.3. Probe Query for SR Links

The probe query message as defined in Figure 1 is sent in-band for Delay measurement. The probe query message as defined in Figure 2 is sent in-band for Loss measurement.

3.1.4. Probe Query for End-to-end Measurement for SR Policy

3.1.4.1. Probe Query Message for SR-MPLS Policy

The message content for in-band probe query message using UDP header for end-to-end performance measurement of SR-MPLS Policy is shown in Figure 3.

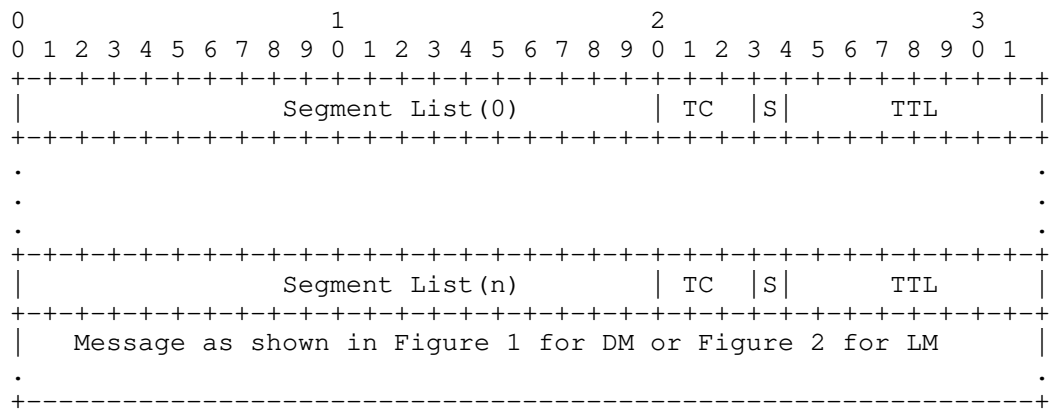
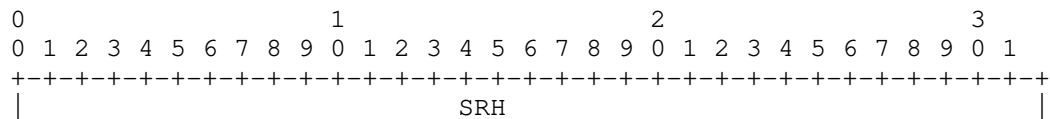


Figure 3: Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case.

3.1.4.2. Probe Query Message for SRv6 Policy

The in-band probe query messages using UDP header for end-to-end performance measurement of an SRv6 Policy is sent using SRv6 Segment Routing Header (SRH) and Segment List of the SRv6 Policy as defined in [I-D.6man-segment-routing-header] and is shown in Figure 4.



```

.   END.OTP (DM) or END.OP (LM) with Target SRv6 SID   .
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   Message as shown in Figure 1 for DM or Figure 2 for LM   |
.   (Using IPv6 Addresses)                                .
.
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 4: Probe Query Message for SRv6 Policy

For delay measurement of SRv6 Policy, END function END.OTP [I-D.spring-srv6-oam] is used with the target SRv6 SID to punt probe messages on the target node, as shown in Figure 4. Similarly, for loss measurement of SRv6 Policy, END function END.OP [I-D.spring-srv6-oam] is used with target SRv6 SID to punt probe messages on the target node.

3.2. Probe Response Message

When the received probe query message does not contain any UDP Return Object (URO) TLV [RFC7876], the probe response message is sent using the IP/UDP information from the probe query message. The content of the probe response message is shown in Figure 5.

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   IP Header                                               |
.   Source IP Address = Responder IPv4 or IPv6 Address     .
.   Destination IP Address = Source IP Address from Query  .
.   Protocol = UDP                                          .
.   Router Alert Option Not Set                            .
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   UDP Header                                              |
.   Source Port = As chosen by Responder                    .
.   Destination Port = Source Port from Query              .
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   Message as specified in Section 3.2 of RFC 6374 for DM, or |
.   Message as specified in Section 3.1 of RFC 6374 for LM  .
.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 5: Probe Response Message

When the received probe query message contains UDP Return Object (URO) TLV [RFC7876], the probe response message uses the IP/UDP information from the URO in the probe query message. The content of the probe response message is shown in Figure 6.

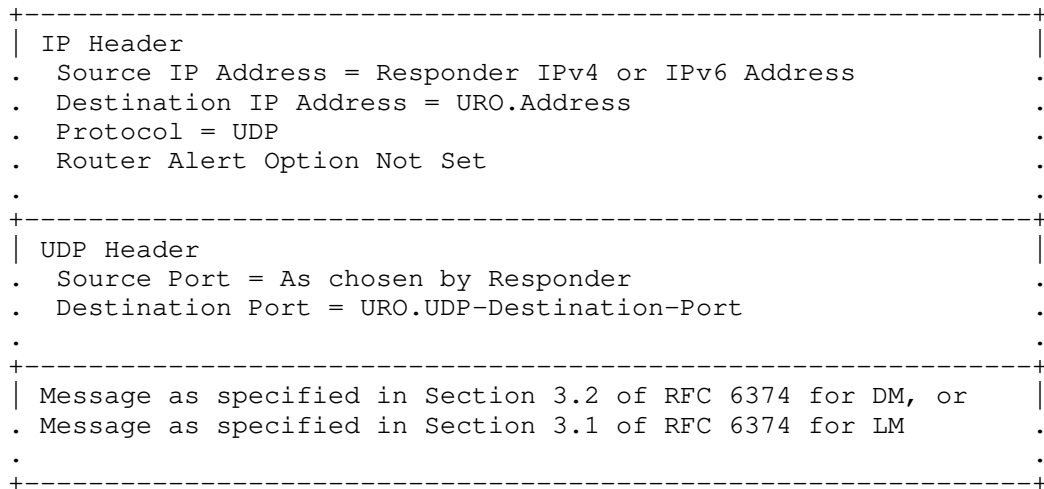


Figure 6: Probe Response Message Using URO from Probe Query Message

3.2.1. One-way Measurement for SR Link and end-to-end SR Policy

For one-way performance measurement, the probe response message as defined in Figure 5 or Figure 6 is sent out-of-band for both SR links and SR Policies.

The PM querier node can receive probe response message back by properly setting its own IP address as Source Address of the header or by adding URO TLV in the probe query message and setting its own IP address in the IP Address in the URO TLV (Type=131) [RFC7876]. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

3.2.1.1. Probe Response Message to Controller

As shown in the Reference Topology, if the querier node requires the probe response message to be sent to the controller R100, it adds URO TLV in the probe query message and sets the IP address of R100 in the IP Address field and user-configured UDP port for DM and for LM in the UDP-Destination-Port field of the URO TLV (Type=131) [RFC7876].

3.2.2. Two-way Measurement for SR Links

For two-way performance measurement, when using a bidirectional channel, the probe response message as defined in Figure 5 or Figure 6 is sent back in-band to the querier node for SR links. In this

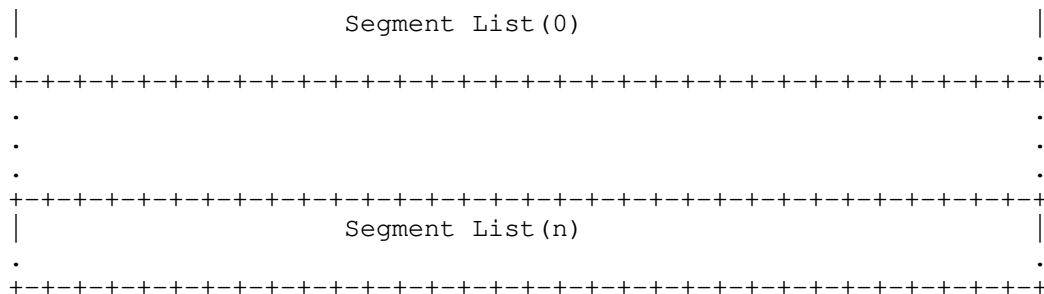


Figure 7B: Segment List Sub-TLV in Return Path TLV

The Sub-TLV in the Return Path TLV can be one of the following Types:

- o Type (value 1): SR-MPLS Label Stack of the Reverse SR Policy
- o Type (value 2): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy
- o Type (value 3): SRv6 Segment List of the Reverse SR Policy
- o Type (value 4): SRv6 Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

With sub-TLV Type 1, the Segment List(0) can be used by the responder node to compute the next-hop IP address and outgoing interface to send the probe response messages.

The Return Path TLV is optional. The PM querier node MUST only insert one Return Path TLV in the probe query message and the responder node MUST only process the first Return Path TLV in the probe query message and ignore other Return Path TLVs if present. The responder node MUST send probe response message back on the reverse path specified in the Return Path TLV and MUST NOT add Return Path TLV in the probe response message.

3.2.3.2. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 8. The SR-MPLS label stack in the packet header is built using the Segment List received in the Return Path TLV in the probe query message.

0 1 2 3

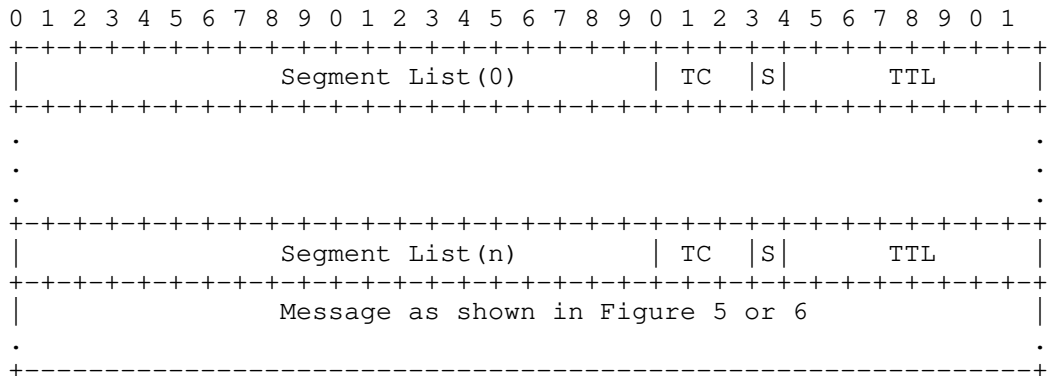


Figure 8: Probe Response Message for SR-MPLS Policy

3.2.3.3. Probe Response Message for SRv6 Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SRv6 Policy is shown in Figure 9. For SRv6 Policy, the SRv6 SID list in the SRH of the probe response message is built using the SRv6 Segment List received in the Return Path TLV in the probe query message.

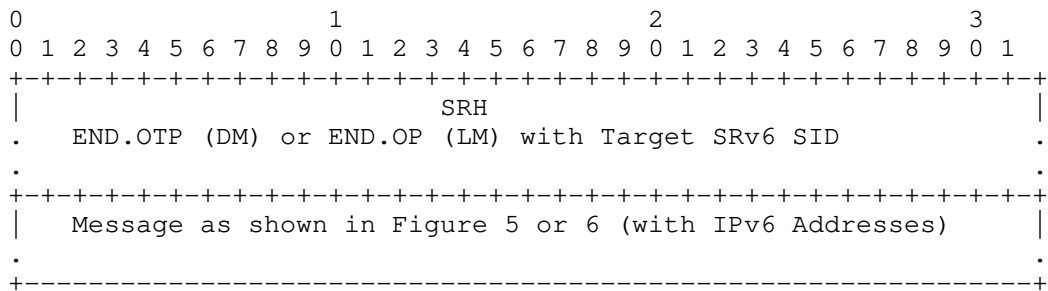


Figure 9: Probe Response Message for SRv6 Policy

4. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

5. ECMP Support

An SR Policy can have ECMPs between the source and transit nodes,

between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. Following mechanisms can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the path taken by them.

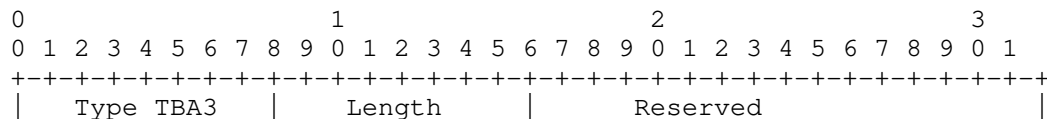
- o The mechanisms described in [RFC8029] [RFC5884] for handling ECMPs are also applicable to the performance measurement. In the IP/UDP header of the PM probe messages, Destination Addresses in 127/8 range for IPv4 or 0:0:0:0:0:FFFF:7F00/104 range for IPv6 can be used to exercise a particular ECMP path. As specified in [RFC6437], 3-tuple of Flow Label, Source Address and Destination Address fields in the IPv6 header can also be used.
- o For SR-MPLS, entropy label [RFC6790] in the PM probe messages can be used.
- o For SRv6, Flow Label in SRH [I-D.6man-segment-routing-header] of the PM probe messages can be used.

6. Sequence Numbers

The message formats for DM and LM [RFC6374] can carry either timestamp or sequence number but not both. There are case where both timestamp and sequence number are desired for both DM and LM. Sequence numbers can be useful when some probe query messages are lost or they arrive out of order. In addition, the sequence numbers can be useful for detecting denial-of-service (DoS) attacks on UDP ports.

6.1. Sequence Number TLV in Unauthenticated Mode

[RFC6374] defines DM and LM probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA3) is defined in this document to carry sequence number for probe query and response messages for delay and loss measurement. The format of the Sequence Number TLV is shown in Figure 10:



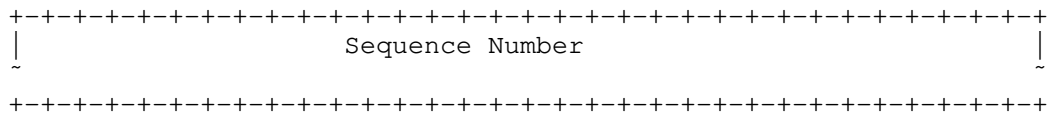


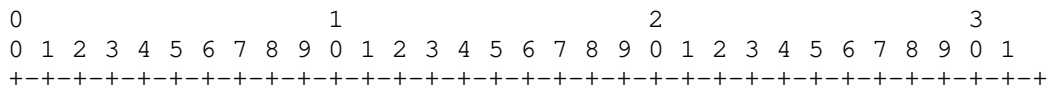
Figure 10: Sequence Number TLV - Unauthenticated Mode

- o The sequence numbers start with 0 and are incremented by one for each subsequent probe query packet.
- o The sequence number are independent for DM and LM messages.
- o The sequence number can be of any length determined by the querier node.
- o The Sequence Number TLV is optional.
- o The PM querier node SHOULD only insert one Sequence Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Sequence Number TLV from the probe query message and ignore the other Sequence Number TLVs if present.
- o When Sequence Number TLV is added, the DM and LM messages SHOULD NOT carry sequence number in the timestamp field of the message.

6.2. Sequence Number TLV in Authenticated Mode

The PM probe query and reply packet format in authenticated mode includes a key Hashed Message Authentication Code (HMAC) ([RFC2104]) hash. Each probe query and reply messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It uses HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPSec defined in [RFC4868]); hence the length of the HMAC field is 16 octets. HMAC uses own key and the definition of the mechanism to distribute the HMAC key is outside the scope of this document.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe packet MAY include Comp.MBZ (Must Be Zero) variable length field to align the packet on 16 octets boundary.



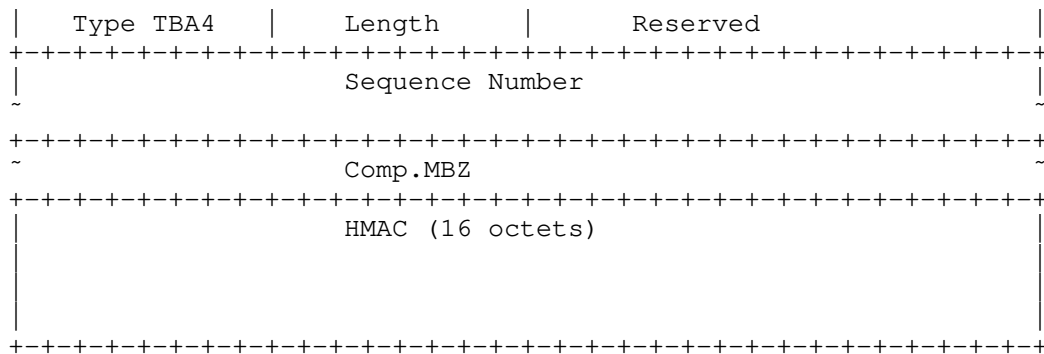


Figure 11: Sequence Number TLV - Authenticated Mode

- o This TLV is mandatory in the authenticated mode.
- o The node MUST discard the probe message if HMAC is invalid.
- o The Sequence Number follows the same processing rule as defined in the unauthenticated mode.

7. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far end responder node. The security considerations described in Section 8 of [RFC6374] are applicable to this specification, and particular attention should be paid to the last three paragraphs.

Use of HMAC-SHA-256 in the authenticated mode defined in this document protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [I-D.6man-segment-routing-header]. Hence, PM probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

8. IANA Considerations

IANA is requested to allocate values for the following Return Path TLV Type for RFC 6374 to be carried in PM probe query messages:

- o Type TBA1: Return Path TLV

IANA is requested to allocate the values for the following Sub-TLV Types for the Return Path TLV.

- o Type 1: SR-MPLS Label Stack of the Reverse SR Policy
- o Type 2: SR-MPLS Binding SID of the Reverse SR Policy
- o Type 3: SRv6 Segment List of the Reverse SR Policy
- o Type 4: SRv6 Binding SID of the Reverse SR Policy

IANA is also requested to allocate a value for the following Block Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for loss measurement:

- o Type TBA2: Block Number TLV

IANA is also requested to allocate a value for the following Sequence Number TLV Types for RFC 6374 to be carried in the PM probe query and response messages for delay and loss measurement:

- o Type TBA3: Sequence Number TLV in Unauthenticated Mode
- o Type TBA4: Sequence Number TLV in Authenticated Mode

9. References

9.1. Normative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks", RFC 6374, September 2011.
- [RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

[I-D.spring-srv6-oam] Ali, Z., et al., "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ali-spring-srv6-oam.

9.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [Y1731] ITU-T Recommendation Y.1731 (02/08), "OAM functions and mechanisms for Ethernet based networks", February 2008.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <<https://www.rfc-editor.org/info/rfc4868>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Kumar, N., Aldrin, S. and M. Chen, "Detecting Multiprotocol Label

- Switched (MPLS) Data-Plane Failures", RFC 8029, March 2017.
- [RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.
- [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>>.
- [I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.6man-segment-routing-header] Filsfils, C., et al., "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.
- [I-D.spring-rfc6374-srpm-mpls] Filsfils, C., Gandhi, R. Ed., et al. "Performance Measurement in Segment Routing Networks with MPLS Data Plane", draft-gandhi-spring-rfc6374-srpm-mpls, work in progress.
- [I-D.pce-binding-label-sid] Filsfils, C., et al., "Carrying Binding Label Segment-ID in PCE-based Networks", draft-sivabalan-pce-binding-label-sid, work in progress.
- [I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment, work in progress.
- [I-D.ippm-stamp] Mirsky, G. et al. "Simple Two-way Active Measurement Protocol", draft-ietf-ippm-stamp, work in progress.
- [BBF.TR-390] "Performance Measurement from IP Edge to Customer Equipment using TWAMP Light", BBF TR-390, May 2017.

Acknowledgments

The authors would like to thank Nagendra Kumar and Carlos Pignataro for the discussion on SRv6 Performance Measurement. The authors would also like to thank Stewart Bryant for the discussion on UDP port allocation for Performance Measurement and Greg Mirsky for providing many useful comments and suggestions.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

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

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

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

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
M. Chen
Huawei
August 6, 2020

Performance Measurement Using RFC 6374 with UDP Path for Segment Routing
Networks
draft-gandhi-spring-rfc6374-srpm-udp-05

Abstract

Segment Routing (SR) leverages the source routing paradigm. Segment Routing (SR) is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedures for using UDP path for sending and processing probe query and response messages for Performance Measurement (PM). The procedure uses the mechanisms defined in RFC 6374 for Performance Delay and Loss Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both Links and end-to-end SR Paths including SR Policies measurements.

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 February 7, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
3. Overview	5
3.1. Example Provisioning Model	6
4. Probe Query Message	7
4.1. Delay Measurement Probe Query Message	7
4.2. Loss Measurement Probe Query Message	7
4.3. Combined Loss/Delay Measurement Probe Query Message	8
4.4. Probe Query Message for Links	9
4.5. Probe Query Message for SR Policies	9
4.5.1. Probe Query Message for SR-MPLS Policy	9
4.5.2. Probe Query Message for SRv6 Policy	10
5. Probe Response Message	11
5.1. One-way Measurement Mode	13
5.1.1. Links and SR Policies	13
5.1.2. Probe Response Message to Controller	13
5.2. Two-way Measurement Mode	13
5.2.1. Links	13
5.2.2. SR Policies	13
5.2.3. Return Path TLV Extensions	14
5.2.4. Probe Response Message for SR-MPLS Policy	14
5.2.5. Probe Response Message for SRv6 Policy	15
5.3. Loopback Measurement Mode	15
6. Performance Measurement for P2MP SR Policies	16
7. ECMP Support for SR Policies	16
8. Additional Probe Message Processing Rules	16
9. Sequence Numbers	16
9.1. Sequence Number TLV Extension in Unauthenticated Mode	16

9.2. Sequence Number TLV Extension in Authenticated Mode . . .	17
10. Performance Delay and Liveness Monitoring	18
11. Security Considerations	19
12. IANA Considerations	19
13. References	20
13.1. Normative References	20
13.2. Informative References	20
Acknowledgments	22
Contributors	22
Authors' Addresses	23

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics and can be used in SR networks with MPLS data plane [I-D.ietf-mpls-rfc6374-sr]. [RFC6374] addresses the limitations of the IP based performance measurement protocols as specified in Section 1 of [RFC6374]. [RFC6374] requires data plane to support MPLS Generic Associated Channel Label (GAL) and Generic Associated Channel (G-ACh), which may not be supported on all nodes in the segment routing network.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe response messages over an UDP return path for RFC 6374 based probe queries. [RFC7876] can be used to send out-of-band probe response messages in both SR-MPLS and SRv6 networks for one-way performance measurement.

For SR Policies, there are ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. RFC 6374 does not define handling for ECMP forwarding paths when used in SR networks.

For two-way measurements for SR Policies, there is a requirement to specify a return path in the form of a Segment List in probe query messages that does not require on any SR Policy state information on the destination node.

This document specifies a procedure for sending and processing probe query and response messages using UDP paths for Performance Measurement in SR networks. The procedure uses RFC 6374 defined mechanisms for Performance Delay and Loss Measurement and unless otherwise specified, the procedures from RFC 6374 are not modified. The procedure specified is applicable to both SR-MPLS and SRv6 data planes. The procedure can be used for both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

G-ACh: Generic Associated Channel (G-ACh).

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

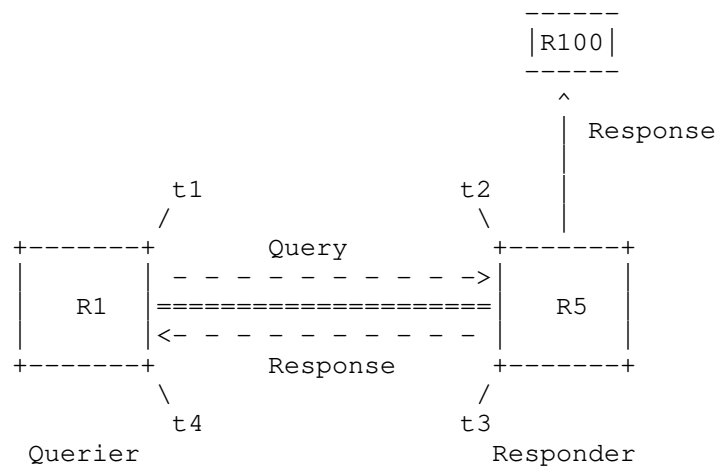
TC: Traffic Class.

URO: UDP Return Object.

2.3. Reference Topology

In the reference topology shown below, the querier node R1 initiates a probe query for performance measurement and the responder node R5 sends a probe response message for the probe query message received. The probe response message may be sent to the querier node R1 or to a controller node R100.

SR is enabled on nodes R1 and R5. The nodes R1 and R5 may be directly connected via a Link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R5 (called head-end).



Reference Topology

3. Overview

For one-way, two-way and round-trip delay measurements in Segment Routing networks, the procedures defined in Section 2.4 and Section 2.6 of [RFC6374] are used. For transmit and receive packet

loss measurements, the procedures defined in Section 2.2 and Section 2.6 of [RFC6374] are used. The procedures use probe messages with IP/UDP path and do not use MPLS GAL. For both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths, no PM state for delay or loss measurement is created on the responder node R5 [RFC6374].

Separate UDP destination port numbers are user-configured for delay and loss measurements from the range specified in [RFC8762]. The querier and responder nodes use the destination UDP port number following the guidelines specified in Section 6 in [RFC6335]. The same destination UDP port is used for Links and SR Paths and the responder is unaware if the query is for the Links or SR Paths. The number of UDP ports with PM functionality needs to be minimized due to limited hardware resources.

For Performance Measurement, probe query and response messages are sent as following:

- o For delay measurement, the probe messages are sent on the congruent path of the data traffic by the querier node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Policies.
- o For loss measurement, the probe messages are sent on the congruent path of the data traffic by the querier node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the responder node (incoming link or incoming SID needed since the responder node does not have PM state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example provisioning model described in [I-D.gandhi-spring-stamp-srpm] is also applicable to the procedures defined in this document, albeit using the Measurement Protocol as [RFC6374]. The querier node is the sender node and the responder node is the reflector node when using [RFC6374]. The provisioning model is not used for signaling PM parameters between the responder and querier nodes in SR networks.

4. Probe Query Message

In this document, UDP path is used for delay and loss measurements for Links and end-to-end SR Policies for the probe messages defined in [RFC6374]. The user-configured destination UDP ports (separate UDP ports for different delay and loss message formats) are used for identifying the probe messages.

4.1. Delay Measurement Probe Query Message

The message content for delay measurement for probe query message using UDP header [RFC0768] is shown in Figure 1. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port can also be used as Source port for two-way delay measurement, since the message has a flag to distinguish between query and response. The DM probe query message contains the payload format for delay measurement defined in Section 3.2 of [RFC6374].

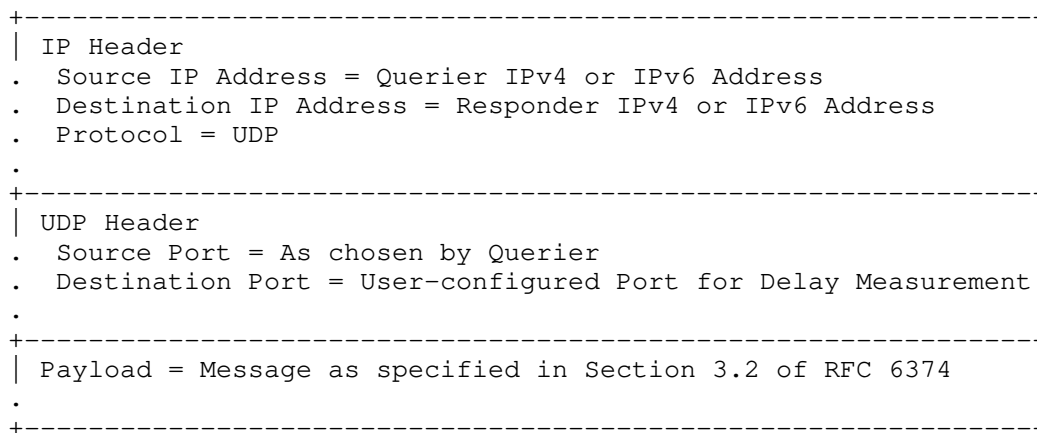


Figure 1: DM Probe Query Message

It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format as a default format as specified in Appendix A of [RFC6374], with hardware support. As an alternative, Network Time Protocol (NTP) timestamp format can also be used [RFC6374].

4.2. Loss Measurement Probe Query Message

The message content for loss measurement probe query message using UDP header [RFC0768] is shown in Figure 2. As shown, the LM probe query message is sent with user-configured Destination UDP port

number for LM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port can also be used as Source port for two-way loss measurement, since the message has a flag to distinguish between query and response. The LM probe query message contains the payload format for loss measurement defined in Section 3.1 of [RFC6374].

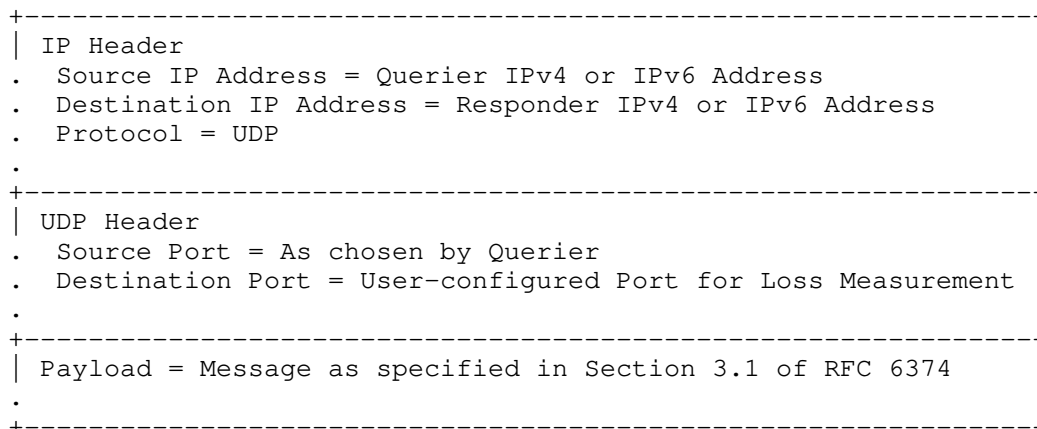


Figure 2: LM Probe Query Message

4.3. Combined Loss/Delay Measurement Probe Query Message

The message content for combined Loss/Delay measurement probe query message using UDP header [RFC0768] is shown in Figure 3. As shown, the probe query message is sent with user-configured Destination UDP port number for combined LM/DM message format. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port can also be used as Source port for two-way loss/delay measurement, since the message has a flag to distinguish between query and response. The probe query message contains the payload format for combined loss/delay measurement defined in Section 3.3 of [RFC6374].

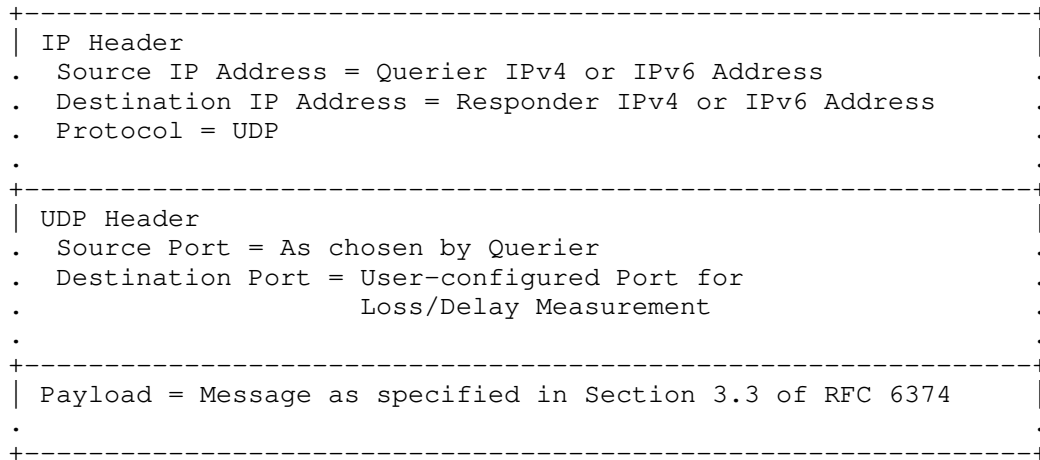


Figure 3: LM/DM Probe Query Message

4.4. Probe Query Message for Links

The probe query message as defined in Figure 1 for delay measurement and Figure 2 for loss measurement are used for Links which may be physical, virtual or LAG (bundle), LAG (bundle) member, numbered/unnumbered Links. The probe messages are pre-routed over the Link for both delay and loss measurement.

4.5. Probe Query Message for SR Policies

The performance delay and loss measurement for segment routing is applicable to both end-to-end SR-MPLS and SRv6 Policies.

4.5.1. Probe Query Message for SR-MPLS Policy

The probe query message for performance measurement of end-to-end SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

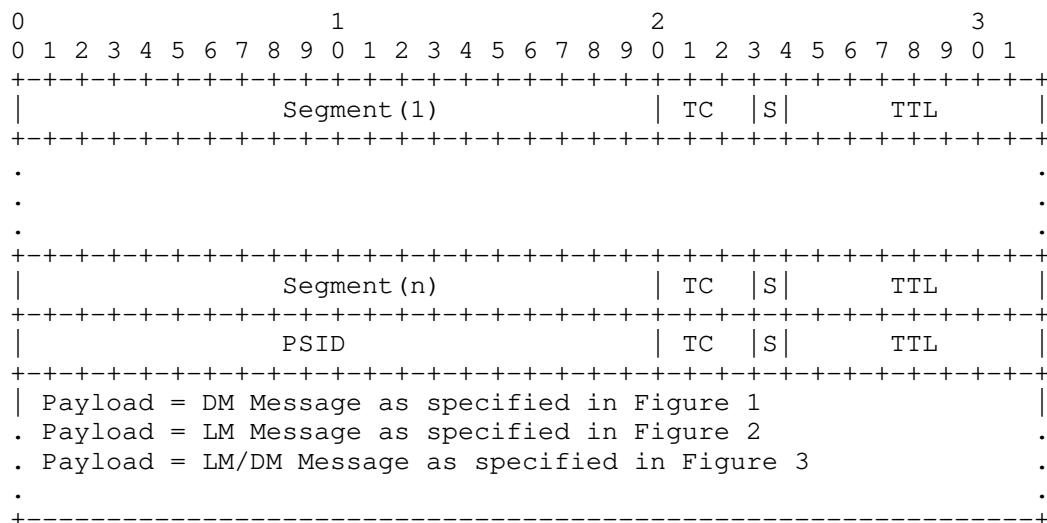


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID)

[I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.5.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List is defined in [RFC8754]. The SRv6 network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages using UDP header for performance measurement of end-to-end SRv6 Policy is sent using its SRv6 Segment Routing Header (SRH) with Segment List as shown in Figure 5. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe query messages.

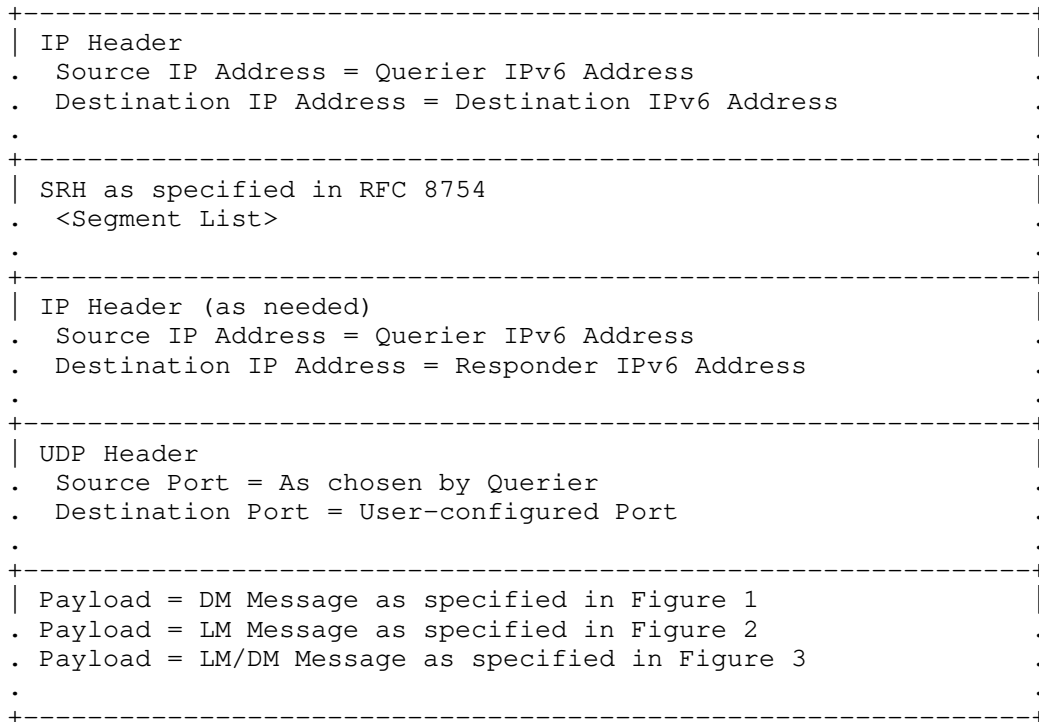


Figure 5: Example Probe Query Message for SRv6 Policy

5. Probe Response Message

When the received probe query message does not contain any UDP Return Object (URO) TLV [RFC7876], the probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 6.

```

+-----+
| IP Header |
. Source IP Address = Responder IPv4 or IPv6 Address .
. Destination IP Address = Source IP Address from Query .
. Protocol = UDP .
. .
+-----+
| UDP Header |
. Source Port = As chosen by Responder .
. Destination Port = Source Port from Query .
. .
+-----+
| Message as specified in Section 3.2 of RFC 6374 for DM, or |
. Message as specified in Section 3.1 of RFC 6374 for LM, or .
. Message as specified in Section 3.3 of RFC 6374 for LM/DM .
. .
+-----+

```

Figure 6: Probe Response Message

When the received probe query message contains UDP Return Object (URO) TLV [RFC7876], the probe response message uses the IP/UDP information from the URO in the probe query message. The content of the probe response message is shown in Figure 7.

```

+-----+
| IP Header |
. Source IP Address = Responder IPv4 or IPv6 Address .
. Destination IP Address = URO.Address .
. Protocol = UDP .
. .
+-----+
| UDP Header |
. Source Port = As chosen by Responder .
. Destination Port = URO.UDP-Destination-Port .
. .
+-----+
| Message as specified in Section 3.2 of RFC 6374 for DM, or |
. Message as specified in Section 3.1 of RFC 6374 for LM, or .
. Message as specified in Section 3.3 of RFC 6374 for LM/DM .
. .
+-----+

```

Figure 7: Probe Response Message Using URO from Probe Query

5.1. One-way Measurement Mode

5.1.1. Links and SR Policies

In one-way measurement mode, the probe response message as defined in Figure 6 or Figure 7 is sent out-of-band to the querier node, for both Links and SR Policies.

The querier node can receive probe response message back by setting its own IP address as Source Address of the header or by adding URO TLV in the probe query message and setting its own IP address in the IP Address in the URO TLV (Type=131) [RFC7876]. The "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message. In this delay measurement mode, as per Reference Topology, timestamps $t1$ and $t2$ are collected by the probes to measure one-way delay as $(t2 - t1)$.

5.1.2. Probe Response Message to Controller

As shown in the Reference Topology, if the querier node requires the probe response message to be sent to the controller R100, it adds URO TLV in the probe query message and sets the IP address of R100 in the IP Address field and user-configured UDP port for DM and for LM in the UDP-Destination-Port field of the URO TLV (Type=131) [RFC7876].

5.2. Two-way Measurement Mode

5.2.1. Links

In two-way measurement mode, when using a bidirectional link, the probe response message as defined in Figure 6 or Figure 7 is sent back on the congruent path of the data traffic to the querier node for Links. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374]. In this delay measurement mode, as per Reference Topology, timestamps $t1$, $t2$, $t3$ and $t4$ are collected by the probes to measure two-way delay as $((t4 - t1) - (t3 - t2))$.

5.2.2. SR Policies

In two-way measurement mode, when using a bidirectional path, the probe response message is sent back on the congruent path of the data traffic to the querier node for end-to-end SR Policies measurements. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374].

5.2.3. Return Path TLV Extensions

For two-way measurement, the querier node can request the responder node to send a response message back on a given reverse path (e.g. co-routed path for two-way measurement). Return Path TLV defined in [I-D.ietf-mpls-rfc6374-sr] is used to carry reverse SR path information as part of the payload of the probe query message. This way the responder node does not require any additional SR state for PM (recall that in SR networks, the state is in the probe packet and signaling of the parameters is avoided).

Additional Sub-TLVs are defined in this document for the Return Path TLV for the following Types:

- o Type (value TBA1): SRv6 Segment List of the Reverse Path
- o Type (value TBA2): SRv6 Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

5.2.4. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way end-to-end SR-MPLS Policy performance measurement is shown in Figure 8. The SR-MPLS label stack in the probe packet header is built using the Segment List received in the Return Path TLV in the probe query message.

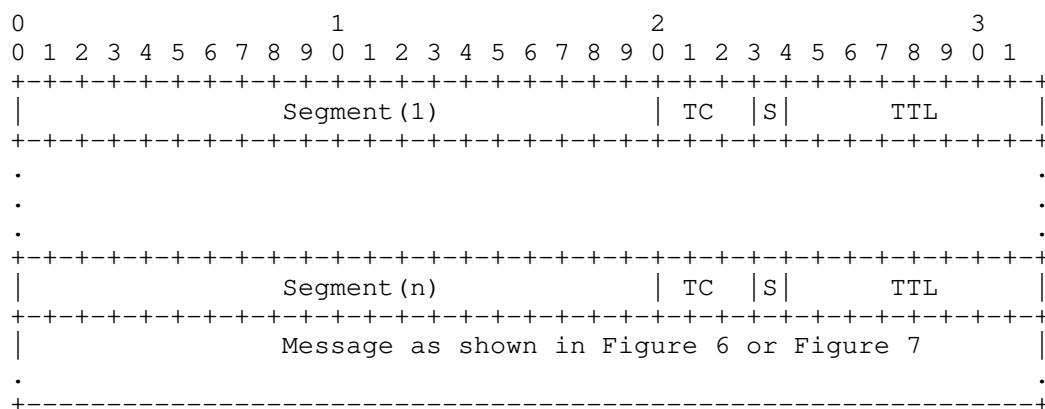


Figure 8: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR-MPLS Policy can be used to find the reverse SR-MPLS Policy to send the probe response message for two-way measurement in the absence of Return Path TLV.

5.2.5. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way end-to-end SRv6 Policy performance measurement is shown in Figure 9. For SRv6 Policy using SRH, the SRv6 SID list in the SRH of the probe response message is built using the SRv6 Segment List received in the Return Path TLV in the probe query message. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe response messages.

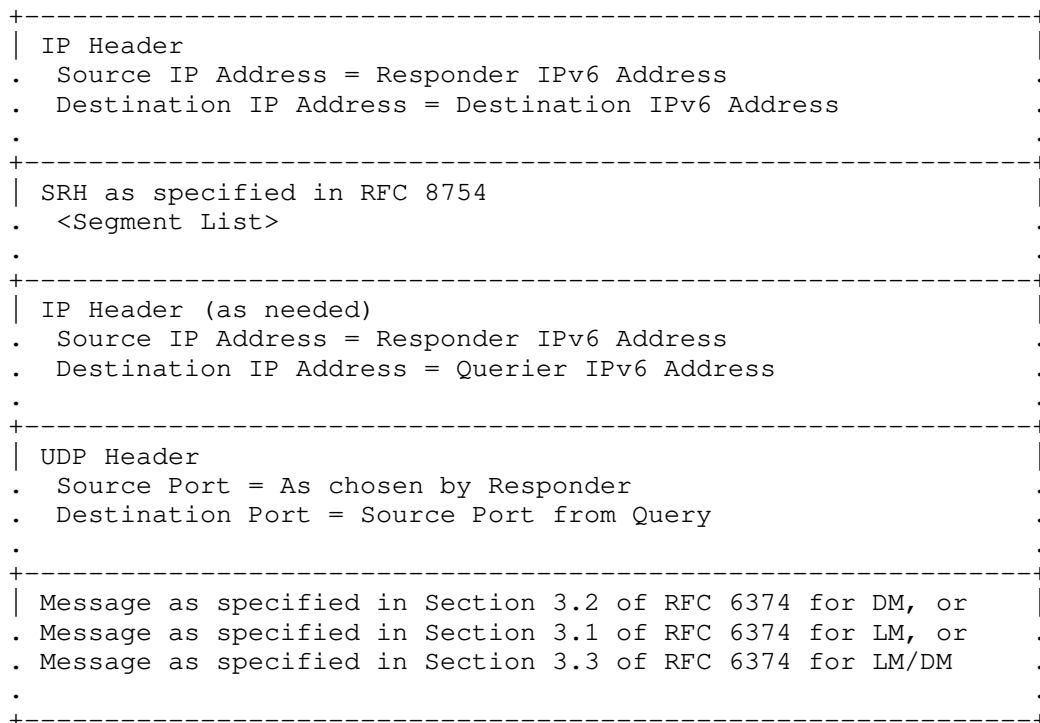


Figure 9: Example Probe Response Message for SRv6 Policy

5.3. Loopback Measurement Mode

The Loopback measurement mode defined in Section 2.8 of [RFC6374] can be used to measure round-trip delay of a bidirectional SR Path. The IP header of the probe query message contains the destination address equals to the querier node address and the source address equals to the responder address. Optionally, the probe query message can carry the reverse path information (e.g. reverse path label stack for SR-

MPLS) as part of the SR header. The responder node does not process the probe messages and generate response messages, and hence Loopback Request object (Type 3) is not required for SR. In this delay measurement mode, as per Reference Topology, timestamps t1 and t4 are collected by the probes to measure round-trip delay.

6. Performance Measurement for P2MP SR Policies

The procedure defined for P2MP SR Policies [I-D.ietf-pim-sr-p2mp-policy] in [I-D.gandhi-spring-stamp-srpm] is also applicable using the RFC 6374 defined messages in the payload.

7. ECMP Support for SR Policies

The procedure defined for handling ECMP for SR Policies in [I-D.gandhi-spring-stamp-srpm] is also applicable to the procedure defined in this document.

8. Additional Probe Message Processing Rules

The additional probe message processing rules defined in [I-D.gandhi-spring-stamp-srpm] are also applicable to the procedures defined in this document.

9. Sequence Numbers

The message formats for DM and LM [RFC6374] can carry either timestamp or sequence number but not both. There are case where both timestamp and sequence number are desired for both DM and LM. Sequence numbers can be useful when some probe query messages are lost or they arrive out of order. In addition, the sequence numbers can be useful for detecting denial-of-service (DoS) attacks on UDP ports.

9.1. Sequence Number TLV Extension in Unauthenticated Mode

[RFC6374] defines DM and LM probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA3) is defined in this document to carry sequence number for probe query and response messages for delay and loss measurement. The format of the Sequence Number TLV in unauthenticated mode is shown in Figure 10.

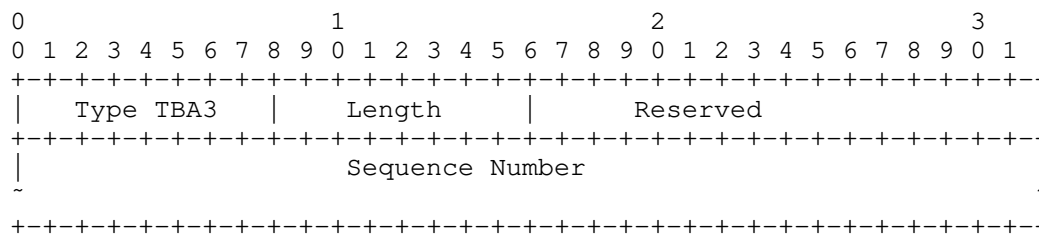


Figure 10: Sequence Number TLV - Unauthenticated Mode

- o The sequence numbers start with 0 and are incremented by one for each subsequent probe query message.
- o The sequence number are independent for DM and LM messages.
- o The sequence number can be of any length determined by the querier node.
- o The Sequence Number TLV is optional.
- o The querier node SHOULD only insert one Sequence Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Sequence Number TLV from the probe query message and ignore the other Sequence Number TLVs if present.
- o When Sequence Number TLV is added, the DM and LM messages SHOULD NOT carry sequence number in the timestamp field of the message.

9.2. Sequence Number TLV Extension in Authenticated Mode

The probe query and response message format in authenticated mode includes a key Hashed Message Authentication Code (HMAC) ([RFC2104]) hash. Each probe query and response messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It can use HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPsec defined in [RFC4868]); hence the length of the HMAC field is 16 octets.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe message MAY include Comp.MBZ (Must Be Zero) variable length field to align the message on 16 octets boundary.

The computation of HMAC field using HMAC-SHA1 can be used with the procedure defined in this document. HMAC uses own key and the definition of the mechanism to distribute the HMAC key is outside the

scope of this document. Both the authentication type and key can be user-configured on both the querier and responder nodes.

The format of the Sequence Number TLV in authentication mode is shown in Figure 11.

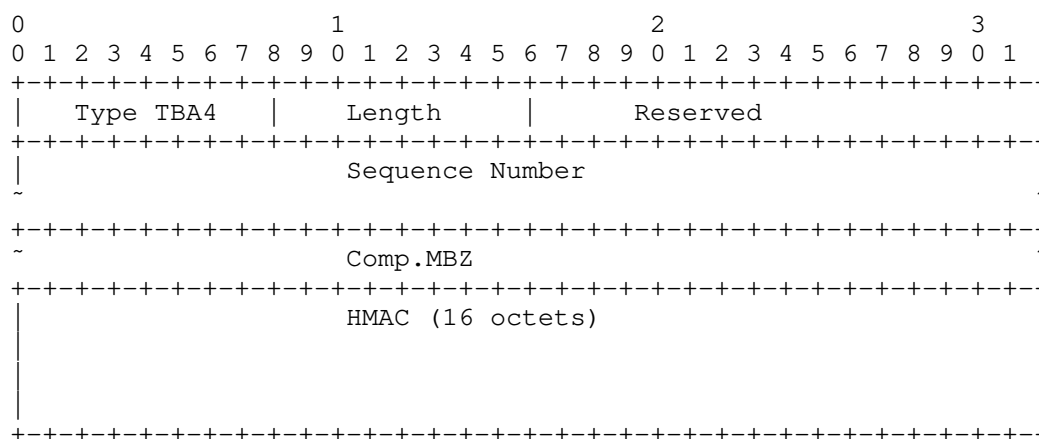


Figure 11: Sequence Number TLV - Authenticated Mode

- o This TLV is mandatory in the authenticated mode.
- o The node MUST discard the probe message if HMAC is invalid.
- o The Sequence Number follows the same processing rule as defined in the unauthenticated mode.

10. Performance Delay and Liveness Monitoring

Liveness monitoring is required for connectivity verification and continuity check in an SR network. The procedure defined in this document for one-way, two-way and loopback mode for delay measurement can also be applied to liveness monitoring of Links and SR Paths. Liveness failure is notified when consecutive N number of probe response messages are not received back at the querier node, where N is locally provisioned value. Note that for one-way and two-way modes, the failure detection interval and scale for number of probe messages need to account for the processing of the probe query messages which need to be punted from the forwarding fast path (to slow path or control plane), and response messages need to be injected on the responder node. Hence, loopback mode is more suitable for liveness monitoring.

11. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far end responder node. The security considerations described in Section 8 of [RFC6374] are applicable to this specification, and particular attention should be paid to the last three paragraphs.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the querier node, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode defined in this document protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

12. IANA Considerations

IANA is requested to allocate the values for the following Sub-TLV Types for the Return Path TLV for RFC 6374 from the sub-registry "Return Path Sub-TLV Type" of the "MPLS Loss/Delay Measurement TLV Object" registry contained within the "Generic Associated Channel (G-ACh) Parameters" registry set:

- o Type TBA1: SRv6 Segment List of the Reverse Path
- o Type TBA2: SRv6 Binding SID of the Reverse SR Policy

IANA is also requested to allocate the values for the following Sequence Number TLV Types for RFC 6374 to be carried in the probe query and response messages for delay and loss measurement from the "MPLS Loss/Delay Measurement TLV Object" registry contained within the "Generic Associated Channel (G-ACh) Parameters" registry set:

- o Type TBA3: Sequence Number TLV in Unauthenticated Mode
- o Type TBA4: Sequence Number TLV in Authenticated Mode

13. References

13.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [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>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC7876] Bryant, S., Sivabalan, S., and S. Soni, "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, DOI 10.17487/RFC7876, July 2016, <<https://www.rfc-editor.org/info/rfc7876>>.
- [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>>.
- [I-D.ietf-mpls-rfc6374-sr]
Gandhi, R., Filsfils, C., Voyer, D., Salsano, S., and M. Chen, "Performance Measurement Using RFC 6374 for Segment Routing Networks with MPLS Data Plane", draft-ietf-mpls-rfc6374-sr-00 (work in progress), July 2020.
- [I-D.gandhi-spring-stamp-srpm]
Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Performance Measurement Using STAMP for Segment Routing Networks", draft-gandhi-spring-stamp-srpm-02 (work in progress), August 2020.

13.2. Informative References

- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.

- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <<https://www.rfc-editor.org/info/rfc4868>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [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>>.
- [RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <<https://www.rfc-editor.org/info/rfc8762>>.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.
- [I-D.ietf-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.
- [I-D.ietf-pce-binding-label-sid]
Filsfils, C., Sivabalan, S., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-ietf-pce-binding-label-sid-03 (work in progress), June 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-16 (work in progress), June 2020.

[I-D.ietf-spring-mpls-path-segment]

Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler,
"Path Segment in MPLS Based Segment Routing Network",
draft-ietf-spring-mpls-path-segment-02 (work in progress),
February 2020.

[I-D.gandhi-mpls-ioam-sr]

Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B.,
and V. Kozak, "MPLS Data Plane Encapsulation for In-situ
OAM Data", draft-gandhi-mpls-ioam-sr-02 (work in
progress), March 2020.

[I-D.ali-spring-ioam-srv6]

Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar,
N., Pignataro, C., Li, C., Chen, M., and G. Dawra,
"Segment Routing Header encapsulation for In-situ OAM
Data", draft-ali-spring-ioam-srv6-02 (work in progress),
November 2019.

Acknowledgments

The authors would like to thank Patrick Khordoc for the discussions on RFC 6374; Nagendra Kumar and Carlos Pignataro for the discussion on SRv6 Performance Measurement. The authors would like to thank Thierry Couture for the discussions on the use-cases for the performance measurement in segment routing networks. The authors would also like to thank Stewart Bryant for the discussion on UDP port allocation for Performance Measurement and Greg Mirsky for providing useful comments and suggestions.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

SPRING Working Group
Internet-Draft
Intended Status: Standards Track
Expires: August 13, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
February 9, 2019

In-band Performance Measurement Using TWAMP
for Segment Routing Networks
draft-gandhi-spring-twamp-srpm-00

Abstract

Segment Routing (SR) is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedures for sending and processing in-band probe query and response messages for Performance Measurement. The procedure uses the mechanisms defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP)) for Delay Measurement, and also uses the mechanisms for direct-mode loss measurement defined in this document. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both links and end-to-end measurement for SR Policies.

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 <http://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."

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions Used in This Document	3
2.1. Requirements Language	3
2.2. Abbreviations	3
2.3. Reference Topology	4
2.4. In-band Probe Messages	5
3. Probe Messages	5
3.1. Probe Query Message	5
3.1.1. Delay Measurement Probe Query Message	5
3.1.2. Loss Measurement Probe Query Message	6
3.1.3. Probe Query for SR Links	10
3.1.4. Probe Query for End-to-end Measurement for SR Policy	11
3.1.4.1. Probe Query Message for SR-MPLS Policy	11
3.1.4.2. Probe Query Message for SRv6 Policy	11
3.2. Probe Response Message	12
3.2.1. One-way Measurement	12
3.2.2. Two-way Measurement	12
3.2.2.1. Probe Response Message for SR-MPLS Policy	13
3.2.2.2. Probe Response Message for SRv6 Policy	13
4. Packet Loss Calculation	13
5. Performance Measurement for P2MP SR Policies	14
6. ECMP Support	14
7. Security Considerations	14
8. IANA Considerations	15
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Acknowledgments	18
Authors' Addresses	18

1. Introduction

Segment Routing (SR) technology greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes.

SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source, transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined path using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. These protocols rely on control channel signaling to establish a test channel over an UDP path. These protocols lack support for direct-mode Loss Measurement (LM) to detect actual data traffic loss which is required in SR networks. The Simple Two-way Active Measurement Protocol (STAMP) [I-D.ippm-stamp] alleviates the control channel signaling by using configuration data model to provision test channels and required UDP ports. The TWAMP Light from broadband forum [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer Edge IP networks.

This document specifies procedures for sending and processing in-band probe query and response messages for Performance Measurement. The procedure uses the mechanisms defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP)) for Delay Measurement, and also uses the mechanisms for direct-mode loss measurement defined in this document. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both links and end-to-end measurement for SR Policies. For SR Policies, there are ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. This document also defines mechanisms for handling ECMPs of SR Policies for performance delay measurement.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

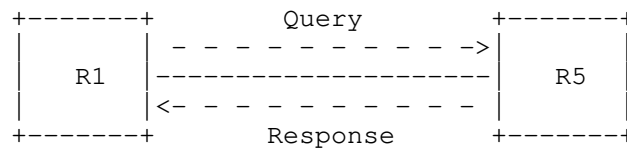
STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

TWAMP: Two-Way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology, the querier node R1 initiates a probe query for performance measurement and the responder node R5 sends a probe response for the query message received. The probe response may be sent to the querier node R1. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].



Reference Topology

Both Delay and Loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in [RFC4656] and [RFC5357], respectively. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

2.4. In-band Probe Messages

For both Delay and Loss measurements for links and SR Policies, no PM session is created on the responder node. The probe messages for Delay measurement are sent in-band by the querier node to measure the delay experienced by the actual traffic flowing on the links and SR Policies. For Loss measurement, in-band probe messages are used to collect the traffic counter for the incoming link or incoming SID on which the probe query message is received at the responder node R5 as it has no PM session state present on the node. The performance measurement for Delay and Loss using out-of-band probe query messages are outside the scope of this document.

3. Probe Messages

3.1. Probe Query Message

In this document, procedures using [RFC5357] is used for Delay and Loss measurements for SR links and end-to-end SR Policies. A user-configured UDP port is used for identifying PM probe packets that does not require to bootstrap PM sessions. A UDP port number from the Dynamic and/or Private Ports range 49152-65535 is used as the destination UDP port. This approach is similar to the one defined in STAMP protocol [I-D.ippm-stamp]. The IPv4 TTL or IPv6 Hop Limit field of the IP header MUST be set to 255.

3.1.1. Delay Measurement Probe Query Message

The message content for Delay Measurement probe query message using UDP header [RFC768] is shown in Figure 1. The DM probe query message is sent with user-configured Destination UDP port number [I-D.ippm-stamp]. The Source UDP port is set to the same value for two-way

delay measurement. The DM probe query message contains the payload for delay measurement defined in Section 4.2.1 of [RFC5357] for TWAMP or in Section 4.1.2 of [RFC4656] for OWAMP.

```

+-----+
| IP Header                                     |
. Source IP Address = Querier IPv4 or IPv6 Address      .
. Destination IP Address = Responder IPv4 or IPv6 Address .
. Protocol = UDP                                         .
. Router Alert Option Not Set                           .
.                                                       .
+-----+
| UDP Header                                       |
. Source Port = As chosen by Querier                    .
. Destination Port = User-configured Port for Delay Measurement.
.                                                       .
+-----+
| Payload = Message as specified in Section 4.2.1 of RFC 5357 |
| Payload = Message as specified in Section 4.1.2 of RFC 4656 |
.                                                       .
+-----+

```

Figure 1: DM Probe Query Message

Timestamp field is eight bytes and by default uses the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588].

3.1.2. Loss Measurement Probe Query Message

The message content for Loss Measurement probe query message using UDP header [RFC768] is shown in Figure 2. The LM probe query message is sent with user-configured Destination UDP port number [I-D.ippm-stamp]. The Source UDP port is set to the same value for two-way loss measurement. The LM probe query message contains the payload for loss measurement defined below.

```

+-----+
| IP Header                                     |
. Source IP Address = Querier IPv4 or IPv6 Address      .
. Destination IP Address = Responder IPv4 or IPv6 Address .
. Protocol = UDP                                         .
. Router Alert Option Not Set                           .
.                                                       .
+-----+
| UDP Header                                       |
. Source Port = As chosen by Querier                    .

```

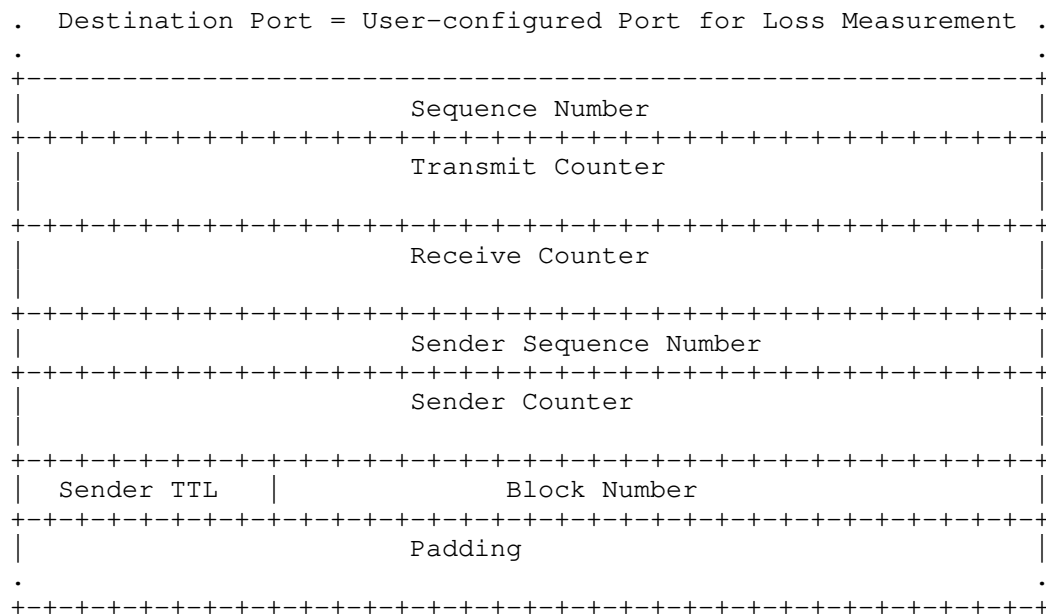
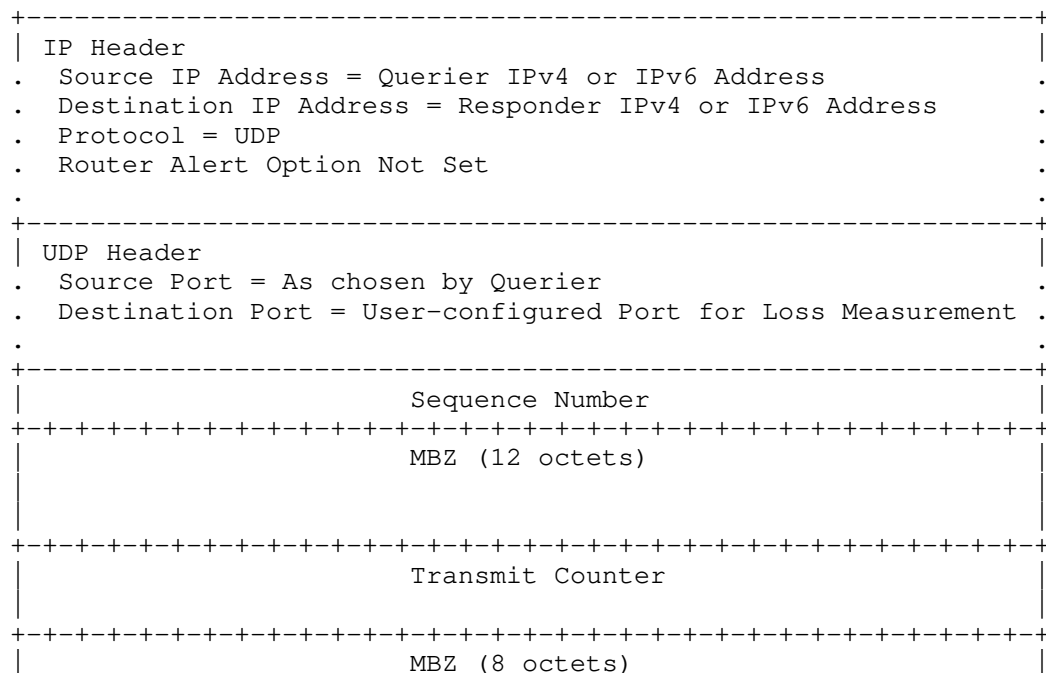


Figure 2A: LM Probe Query Message for TWAMP



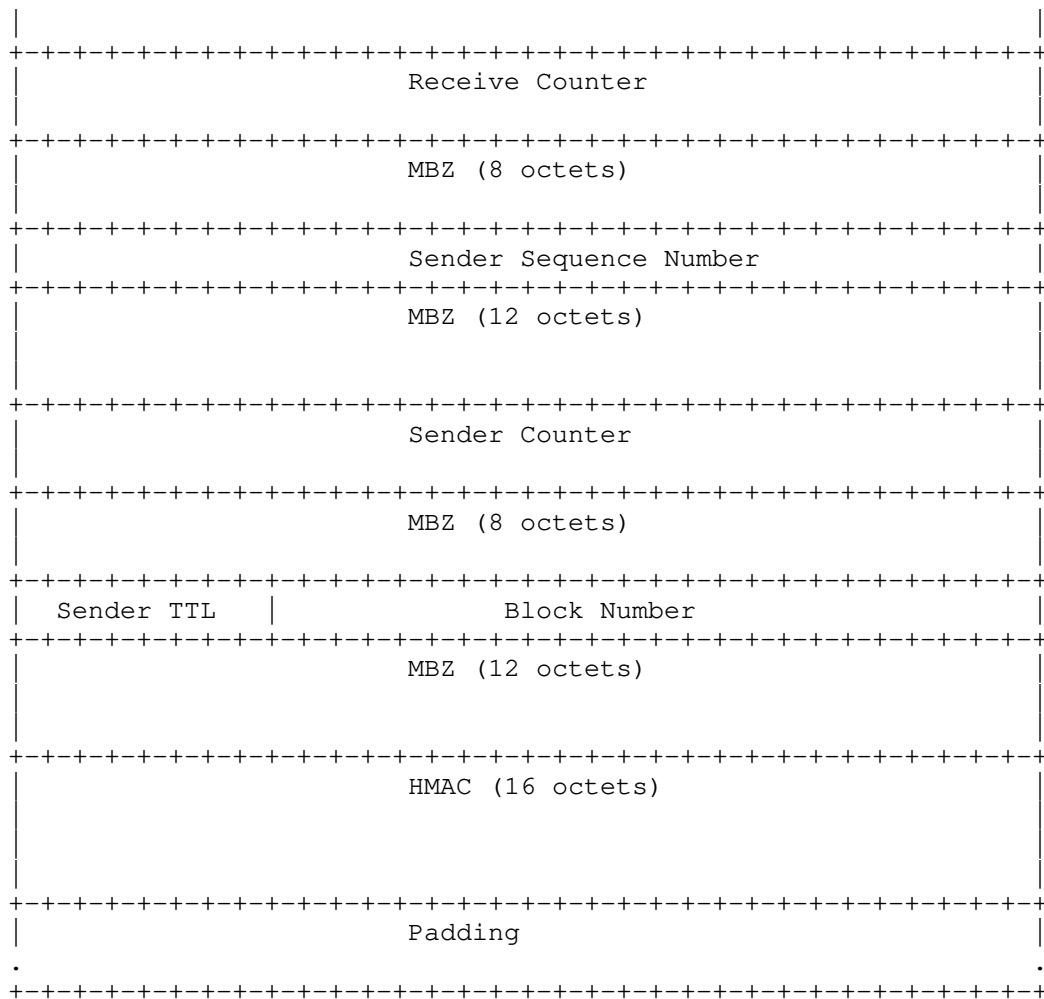
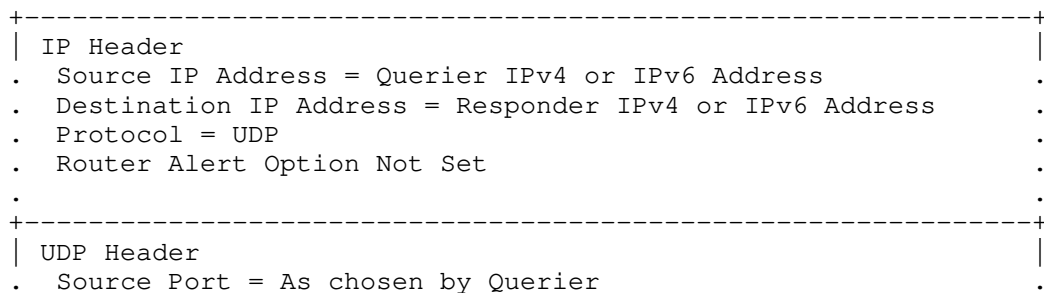


Figure 2B: LM Probe Query Message for TWAMP - Authenticated Mode



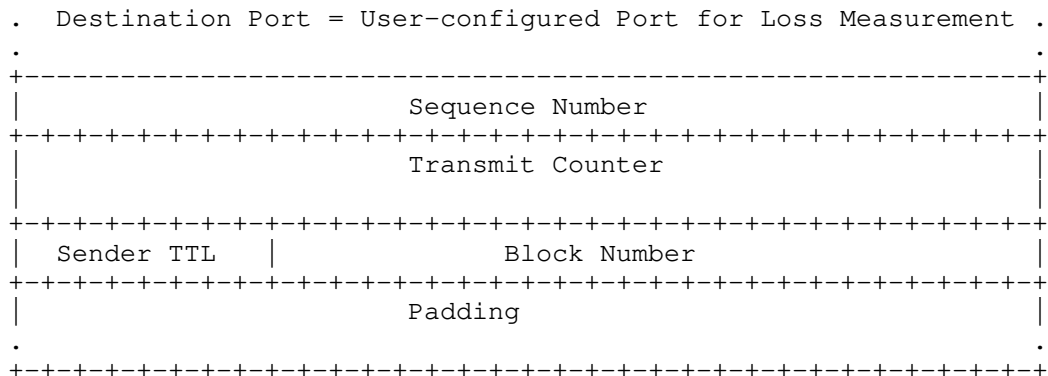
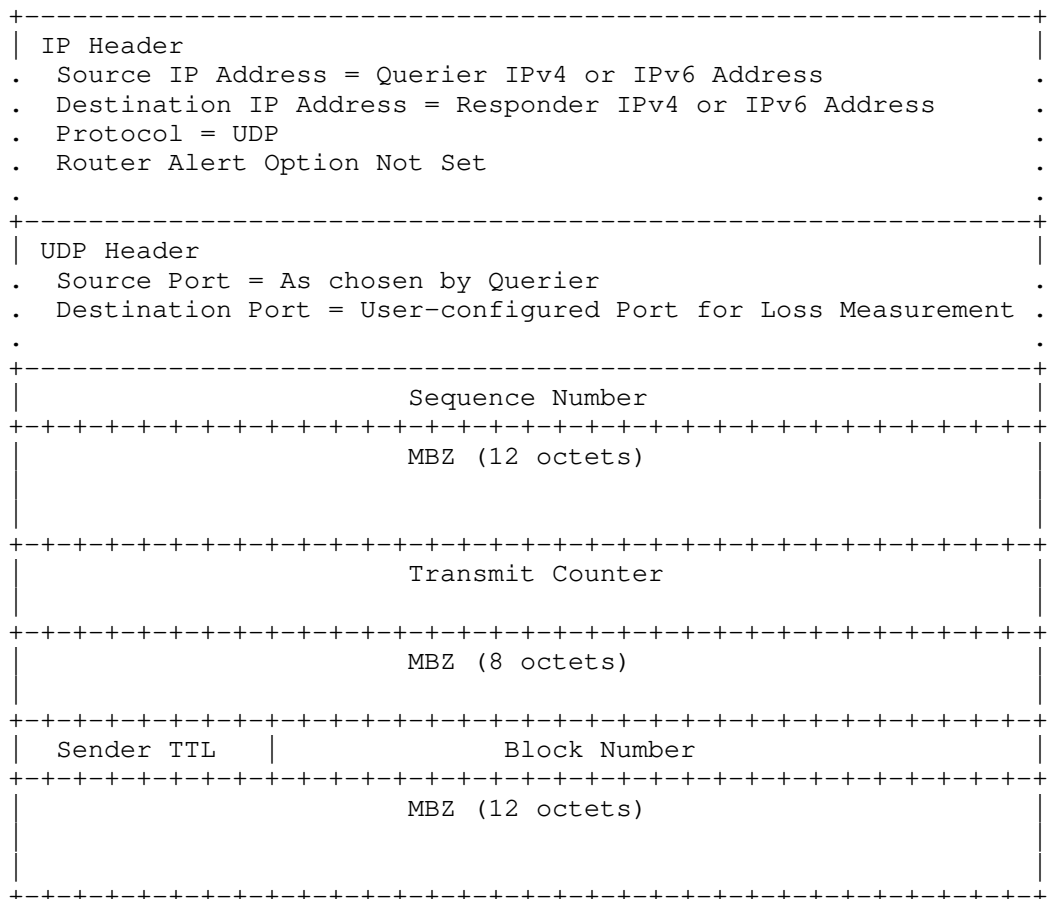


Figure 2C: LM Probe Query Message for OWAMP



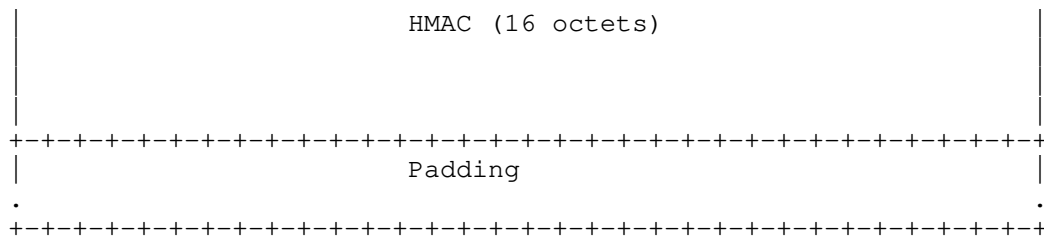


Figure 2D: LM Probe Query Message for OWAMP - Authenticated Mode

Sequence Number (32-bit): As defined in [RFC5357].

Transmit Counter (64-bit): The number of packets sent by the querier node in the query message and by the responder node in the response message. The counter is always written at fixed location in the probe query and response messages.

Receive Counter (64-bit): The number of packets received at the responder node. It is written by the responder node in the probe response message.

Sender Counter (64-bit): This is the exact copy of the transmit counter from the received query message. It is written by the responder node in the probe response message.

Sender Sequence Number (32-bit): As defined in [RFC5357].

Sender TTL: As defined in [RFC5357].

Block Number (24-bit): The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (or color) of the traffic counters. The probe query and response messages carry Block Number for the traffic counters for loss measurement. In both probe query and response messages, the counters MUST belong to the same Block Number.

The Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

3.1.3. Probe Query for SR Links

The probe query message as defined in Figure 1 is sent in-band for Delay measurement. The probe query message as defined in Figure 2 is sent in-band for Loss measurement.

3.1.4. Probe Query for End-to-end Measurement for SR Policy

3.1.4.1. Probe Query Message for SR-MPLS Policy

The message content for in-band probe query message using UDP header for end-to-end performance measurement of SR-MPLS Policy is shown in Figure 3.

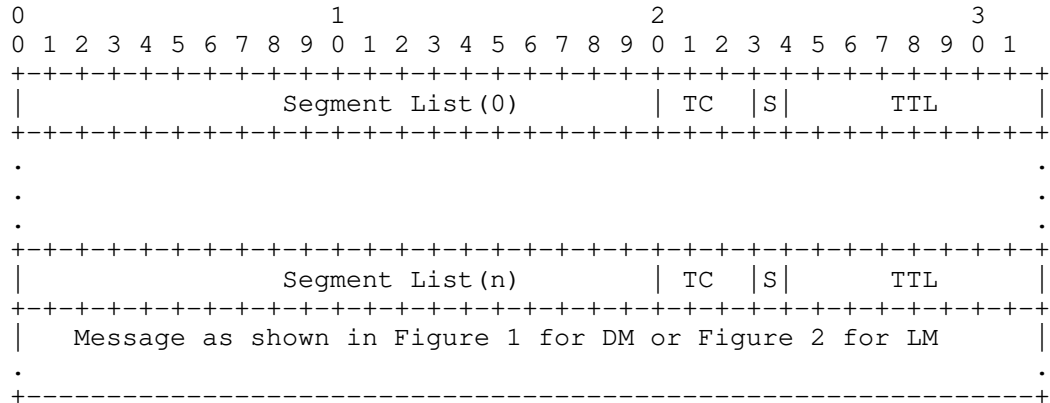


Figure 3: Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case.

3.1.4.2. Probe Query Message for SRv6 Policy

The in-band probe query messages using UDP header for end-to-end performance measurement of an SRv6 Policy is sent using SRv6 Segment Routing Header (SRH) and Segment List of the SRv6 Policy as defined in [I-D.6man-segment-routing-header] and is shown in Figure 4.

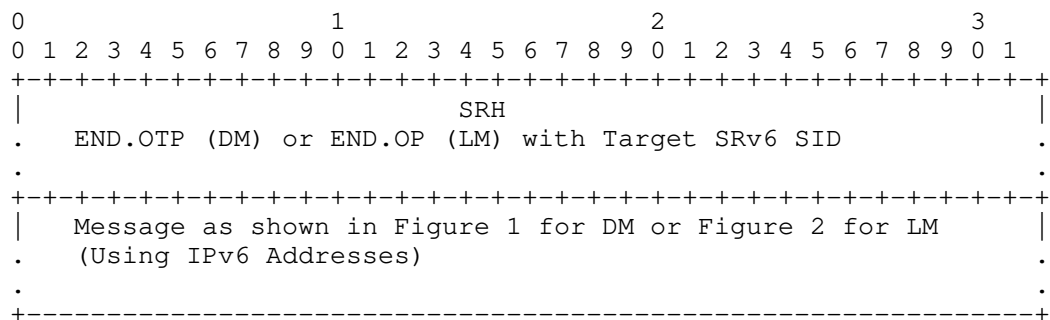


Figure 4: Probe Query Message for SRv6 Policy

For delay measurement of SRv6 Policy, END function END.OTP [I-D.spring-srv6-oam] is used with the target SRv6 SID to punt probe messages on the target node, as shown in Figure 4. Similarly, for loss measurement of SRv6 Policy, END function END.OP [I-D.spring-srv6-oam] is used with target SRv6 SID to punt probe messages on the target node.

3.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the probe query message. The content of the probe response message is shown in Figure 5.

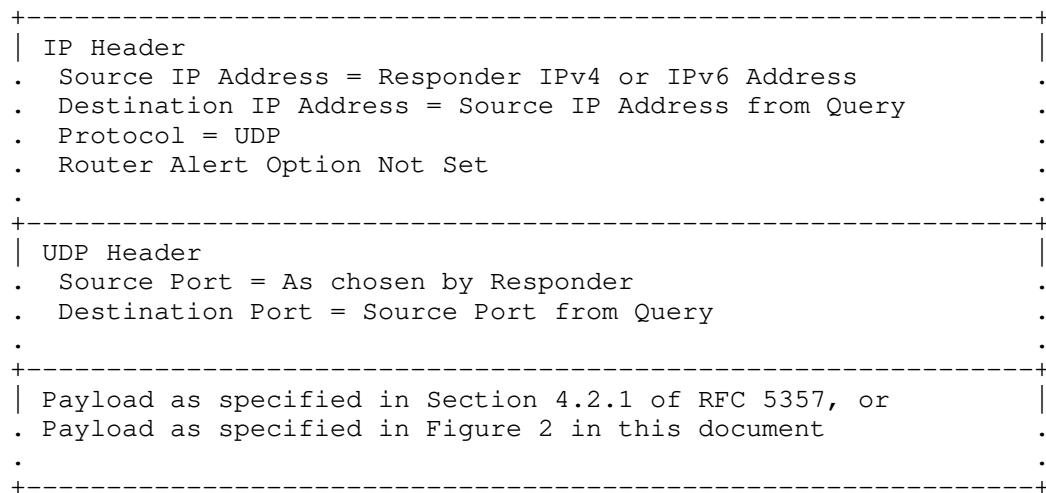


Figure 5: Probe Response Message

3.2.1. One-way Measurement

For one-way performance measurement, the probe response message as defined in Figure 5 is sent for both SR links and SR Policies.

3.2.2. Two-way Measurement

For two-way performance measurement, when using a bidirectional channel, the probe response message as defined in Figure 5 is sent back in-band to the querier node.

The Path Segment Identifier (PSID) [I-D.spring-mpis-path-segment] of the forward SR Policy can be used to find the reverse SR Policy to send the probe response message for two-way measurement of SR Policy.

3.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 6.

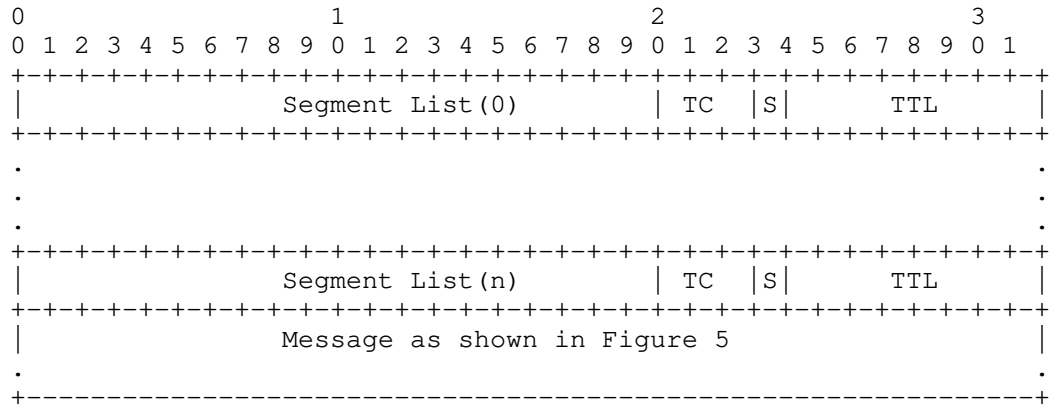


Figure 6: Probe Response Message for SR-MPLS Policy

3.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SRv6 Policy is shown in Figure 7.

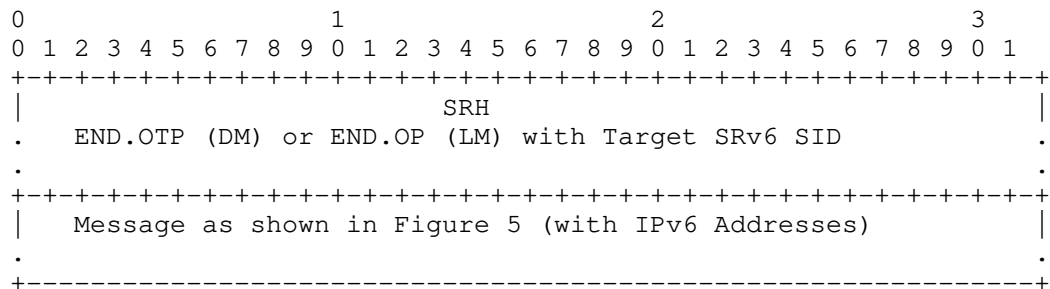


Figure 7: Probe Response Message for SRv6 Policy

4. Packet Loss Calculation

The formula for calculating the one-way packet loss using counters for a given block number is as following:

- o One-way Packet_Loss[n-1, n] = (Sender_Counter[n] - Sender_Counter[n-1]) - (Receive_Counter[n] - Receive_Counter[n-1])

5. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

6. ECMP Support

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. Following mechanisms can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the path taken by them.

- o The mechanisms described in [RFC8029] [RFC5884] for handling ECMPs are also applicable to the performance measurement. In the IP/UDP header of the PM probe messages, Destination Addresses in 127/8 range for IPv4 or 0:0:0:0:0:FFFF:7F00/104 range for IPv6 can be used to exercise a particular ECMP path. As specified in [RFC6437], 3-tuple of Flow Label, Source Address and Destination Address fields in the IPv6 header can also be used.
- o For SR-MPLS, entropy label [RFC6790] in the PM probe messages can be used.
- o For SRv6, Flow Label in SRH [I-D.6man-segment-routing-header] of the PM probe messages can be used.

7. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far end responder node.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the querier, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode defined in this document protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [I-D.6man-segment-routing-header]. Hence, PM probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

8. IANA Considerations

This document does not require any IANA actions.

9. References

9.1. Normative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.
- [I-D.spring-srv6-oam] Ali, Z., et al., "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ali-spring-srv6-oam.

9.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Kumar, N., Aldrin, S. and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, March 2017.
- [RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.
- [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>>.
- [I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment, work in progress.
- [I-D.6man-segment-routing-header] Filsfils, C., et al., "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.
- [I-D.ippm-stamp] Mirsky, G. et al. "Simple Two-way Active

Measurement Protocol", draft-ietf-ippm-stamp, work in progress.

[BBF.TR-390] "Performance Measurement from IP Edge to Customer Equipment using TWAMP Light", BBF TR-390, May 2017.

Acknowledgments

TBA

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

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

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
M. Chen
Huawei
B. Janssens
Colt
October 21, 2020

Performance Measurement Using TWAMP Light for Segment Routing Networks
draft-gandhi-spring-twamp-srpm-11

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the mechanisms defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP) Light) and its extensions for Performance Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	3
2.1. Requirements Language	3
2.2. Abbreviations	3
2.3. Reference Topology	4
3. Overview	5
3.1. Example Provisioning Model	6
4. Probe Messages	7
4.1. Probe Query Message	7
4.1.1. Delay Measurement Query Message	7
4.1.2. Loss Measurement Query Message	8
4.1.3. Probe Query for Links	9
4.1.4. Probe Query for SR Policy	9
4.2. Probe Response Message	11
4.2.1. One-way Measurement Mode	11
4.2.2. Two-way Measurement Mode	11
4.2.3. Loopback Measurement Mode	13
4.3. Additional Probe Message Processing Rules	14
4.3.1. TTL and Hop Limit	14
4.3.2. Router Alert Option	14
4.3.3. UDP Checksum	14
5. Performance Measurement for P2MP SR Policies	14
6. ECMP Support for SR Policies	16
7. Performance Delay and Liveness Monitoring	16
8. Security Considerations	16
9. IANA Considerations	17
10. References	17
10.1. Normative References	17
10.2. Informative References	17
Acknowledgments	20
Authors' Addresses	21

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks using probe messages. These protocols rely on control-channel signaling to establish a test-channel over an UDP path. The TWAMP Light [Appendix I in RFC5357] [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer IP networks by provisioning UDP paths and eliminates the need for control-channel signaling.

This document specifies procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the mechanisms defined in [RFC5357] (TWAMP Light) and its extensions for Performance Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies and Flex- Algo IGP Paths. Unless otherwise specified, the mechanisms defined in [RFC5357] are not modified by this document.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

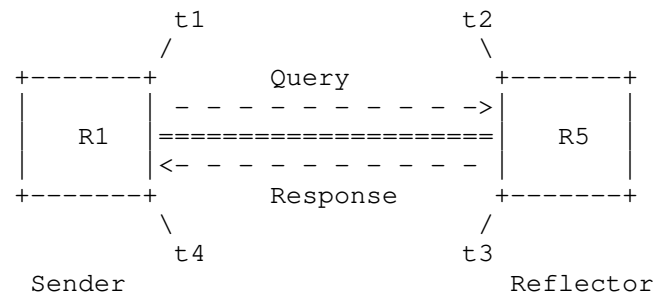
TC: Traffic Class.

TWAMP: Two-Way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a performance measurement probe query message and the reflector node R5 sends a probe response message for the query message received. The probe response message is typically sent to the sender node R1.

SR is enabled on nodes R1 and R5. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R5 (called tail-end).



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC5357] are used. For direct-mode and inferred-mode loss measurements, the probe messages defined in [I-D.gandhi-ippm-twamp-srpm] are used. For both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths, no PM state for delay or loss measurement need to be created on the reflector node R5.

Separate UDP destination port numbers are user-configured for delay and loss measurements. As specified in [RFC8545], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. The same destination UDP port is used for Links and SR Paths and the reflector is unaware if the query is for the Links or SR Paths. The number of UDP ports with PM functionality needs to be minimized due to limited hardware resources.

For Performance Measurement, probe query and response messages are sent as following:

- o For delay measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Paths.
- o For loss measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

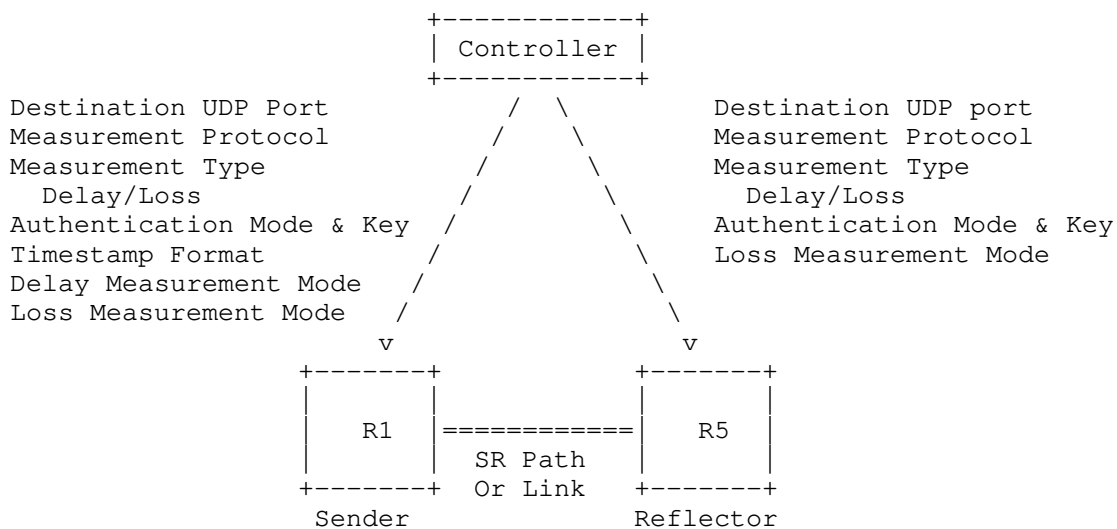


Figure 1: Example Provisioning Model

Example of Measurement Protocol is TWAMP Light, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

The mechanisms to provision the sender and reflector nodes are outside the scope of this document. The provisioning model is not used for signaling the PM parameters between the reflector and sender nodes in SR networks.

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC5357] are used for delay measurement for Links and end-to-end SR Paths including SR Policies. For loss measurement, the probe messages defined in [I-D.gandhi-ippm-twamp-srpm] are used.

4.1.1. Delay Measurement Query Message

The message content for delay measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in Section 4.1.2 of [RFC5357]. For symmetrical size query and response messages as defined in [RFC6038], the DM probe query message contains the payload format defined in Section 4.2.1 of [RFC5357].

```

+-----+
| IP Header                                     |
. Source IP Address = Sender IPv4 or IPv6 Address .
. Destination IP Address = Reflector IPv4 or IPv6 Address .
. Protocol = UDP .
. .
+-----+
| UDP Header                                   |
. Source Port = As chosen by Sender .
. Destination Port = User-configured Port for Delay Measurement.
. .
+-----+
| Payload = DM Message as specified in Section 4.2.1 of RFC 5357 |
. Payload = DM Message as specified in Section 4.1.2 of RFC 5357.
. .
+-----+

```

Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC5357]. It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for loss measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in [I-D.gandhi-ippm-twamp-srpm].

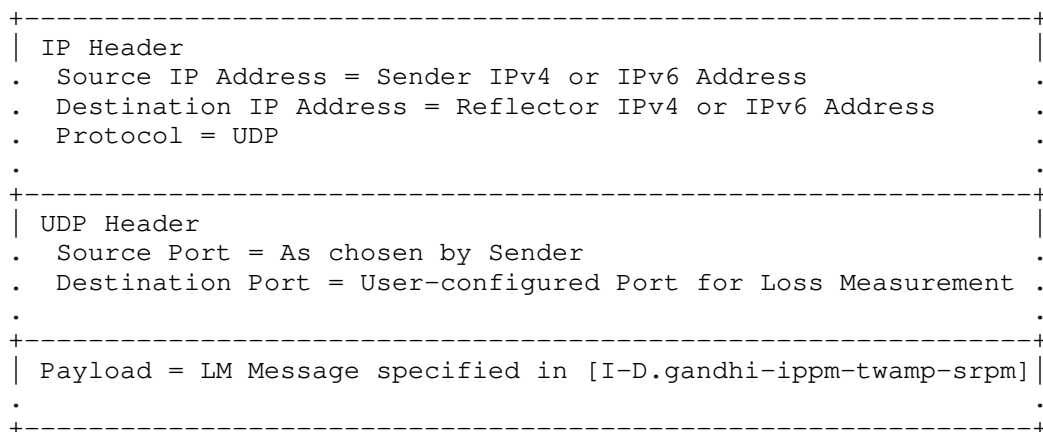


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement are used for Links which may be physical, virtual or LAG (bundle), LAG (bundle) member, numbered/unnumbered Links. The probe messages are pre-routed over the Link for both delay and loss measurement. The local and remote IP addresses of the link are used as Source and Destination Addresses. They can also be IPv6 link local address as probe messages are pre-routed.

4.1.4. Probe Query for SR Policy

The performance delay and loss measurement for segment routing is applicable to both end-to-end SR-MPLS and SRv6 Policies.

The sender IPv4 or IPv6 address is used as the source address. The endpoint IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the loopback address from range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 is used as the destination address, respectively.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for performance measurement of an end-to-end SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

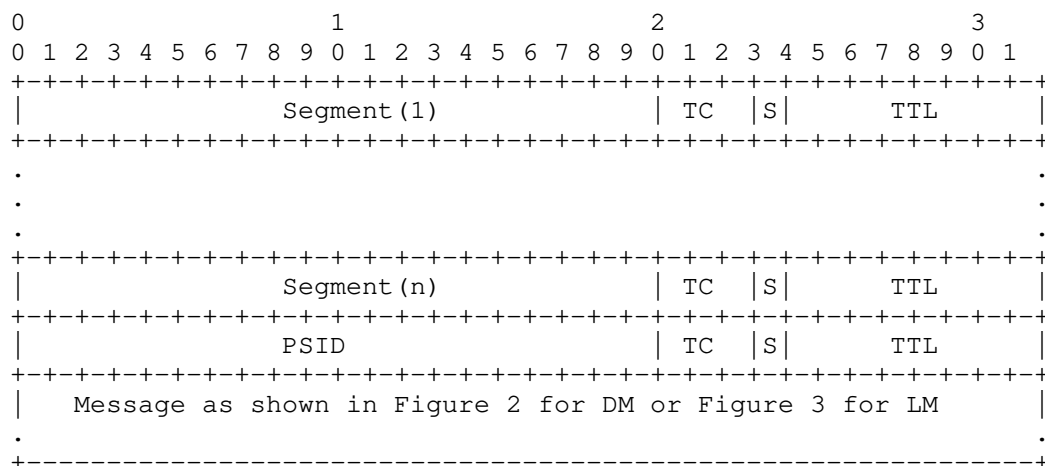


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. The SRv6 network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for performance measurement of an end-to-end SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe query messages.

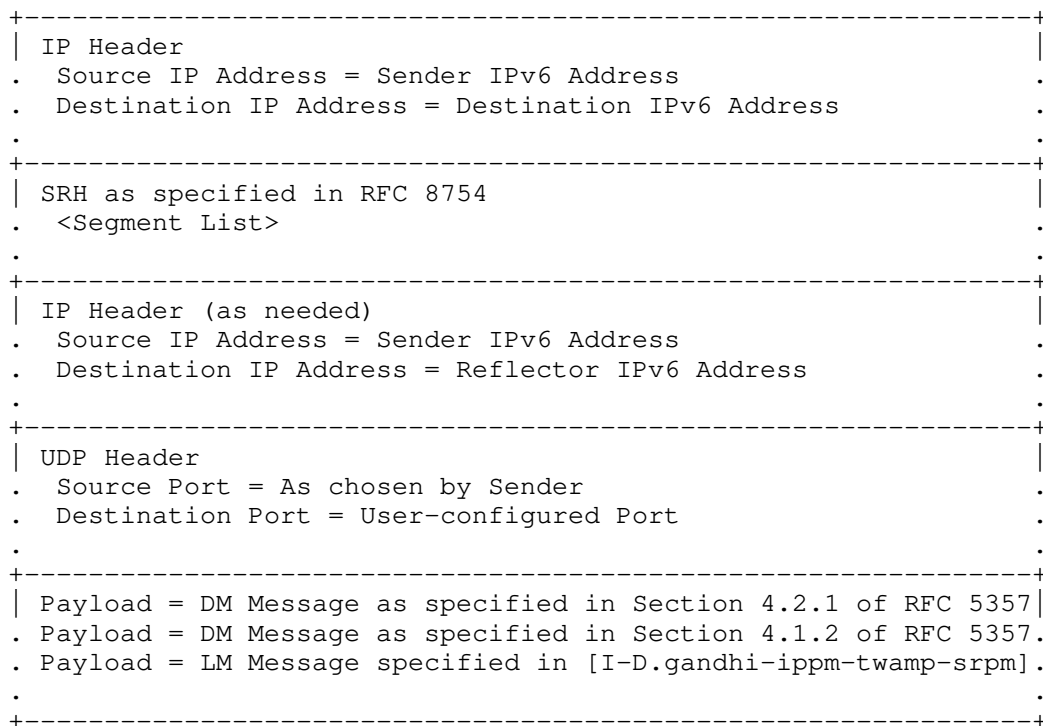


Figure 5: Example Probe Query Message for SRv6 Policy

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 6.

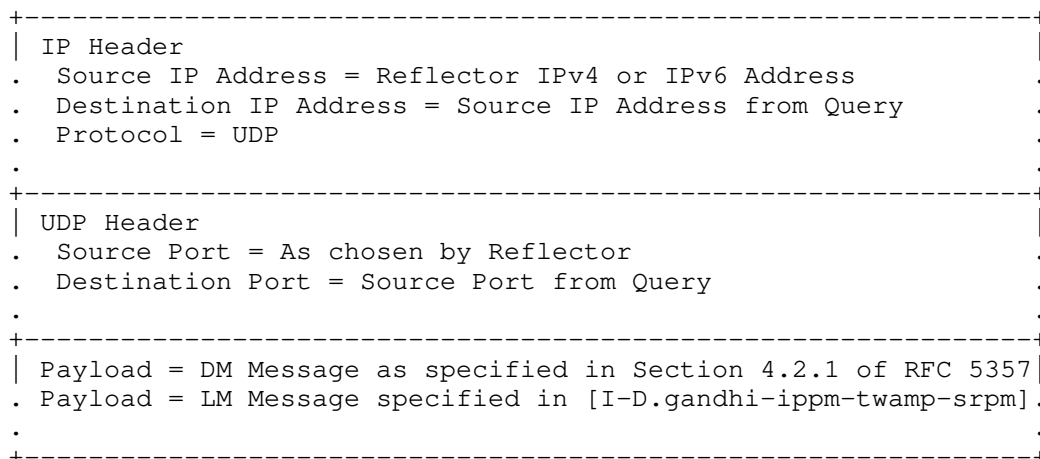


Figure 6: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way measurement mode, the probe response message as defined in Figure 6 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t_1 , t_2 , t_3 , and t_4 are collected by the probes. However, only timestamps t_1 and t_2 are used to measure one-way delay as $(t_2 - t_1)$.

4.2.2. Two-way Measurement Mode

In two-way measurement mode, when using a bidirectional path, the probe response message as defined in Figure 6 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t_1 , t_2 , t_3 , and t_4 are collected by the probes. All four timestamps are used to measure two-way delay as $((t_4 - t_1) - (t_3 - t_2))$.


```

+-----+
| IP Header                                     |
. Source IP Address = Reflector IPv6 Address   .
. Destination IP Address = Destination IPv6 Address .
.                                             .
+-----+
| SRH as specified in RFC 8754                 |
. <Segment List>                             .
.                                             .
+-----+
| IP Header (as needed)                       |
. Source IP Address = Reflector IPv6 Address   .
. Destination IP Address = Source IPv6 Address from Query .
.                                             .
+-----+
| UDP Header                                   |
. Source Port = As chosen by Sender           .
. Destination Port = User-configured Port     .
.                                             .
+-----+
| Payload = DM Message as specified in Section 4.2.1 of RFC 5357 |
. Payload = LM Message specified in [I-D.gandhi-ippm-twamp-srpm].
.                                             .
+-----+

```

Figure 8: Example Probe Response Message for SRv6 Policy

4.2.3. Loopback Measurement Mode

The Loopback measurement mode can be used to measure round-trip delay for a bidirectional SR Path. The IP header of the probe query message contains the destination address equals to the sender address and the source address equals to the reflector address. Optionally, the probe query message can carry the reverse path information (e.g. reverse path label stack for SR-MPLS) as part of the SR header. The probe messages are not punted at the reflector node and it does not process them and generate response messages. The Sender Control Code is set to the default value of 0. In this mode, as the probe packet is not punted on the reflector node for processing, the querier copies the 'Sequence Number' in 'Session-Sender Sequence Number' directly. In this delay measurement mode, as per Reference Topology, the timestamps t1 and t4 are collected by the probes. Both these timestamps are used to measure round-trip delay as (t4 - t1).

4.3. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to TWAMP Light messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.3.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC5357]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC5357].

When using the Destination IPv4 Address from range 127/8, the TTL field in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.3.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.3.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR Path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy] or P2MP Transport [I-D.shen-spring-p2mp-transport-chain]).

The procedures for delay and loss measurement described in this document for P2P SR Policies are also equally applicable to the P2MP SR Policies. The procedure for one-way measurement is defined as following:

- o The sender root node sends probe query messages using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 9.
- o The probe query messages can contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].
- o The Destination Address is set to the loopback address from range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 address.
- o Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 9. This allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.
- o The P2MP root node measures the delay and loss performance for each P2MP leaf node of the end-to-end P2MP SR Policy.

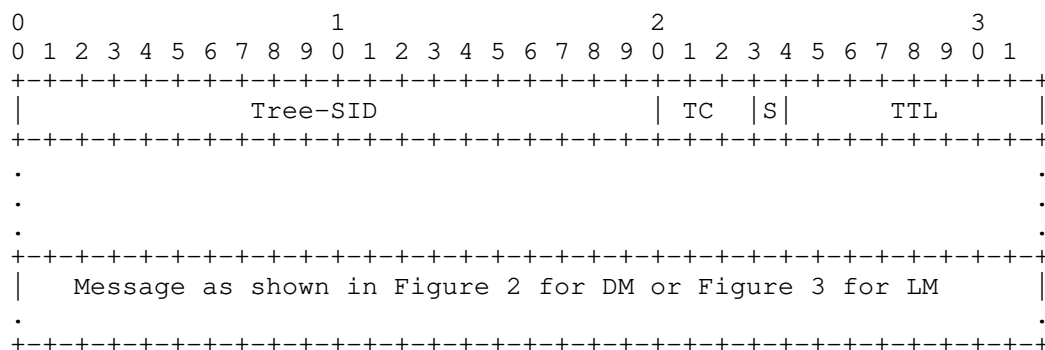


Figure 9: Example Probe Query with Tree-SID for SR-MPLS Policy

The probe query messages can also be sent using the scheme defined for P2MP Transport using Chain Replication that may contain Bud SID as defined in [I-D.shen-spring-p2mp-transport-chain].

The considerations for two-way mode for performance measurement for P2MP SR Policy (e.g. for bidirectional SR Path) are outside the scope of this document.

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the probe messages, sweeping of Destination Address from range 127/8 can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

Liveness monitoring is required for connectivity verification and continuity check in an SR network. The procedure defined in this document for delay measurement using the TWAMP Light probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that for one-way and two-way modes, the failure detection interval and scale for number of probe messages need to account for the processing of the probe query messages which need to be punted from the forwarding fast path (to slow path or control plane) and response messages need to be injected on the reflector node. This is improved by using the probes in loopback mode.

8. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far-end reflector node.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state

associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

This document does not require any IANA action.

10. References

10.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [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>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [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>>.
- [I-D.gandhi-ippm-twamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "TWAMP Light Extensions for Segment Routing", draft-gandhi-ippm-twamp-srpm-00 (work in progress), October 2020.

10.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

- [RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <<https://www.rfc-editor.org/info/rfc2113>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<https://www.rfc-editor.org/info/rfc5884>>.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <<https://www.rfc-editor.org/info/rfc6038>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <<https://www.rfc-editor.org/info/rfc6936>>.
- [RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <<https://www.rfc-editor.org/info/rfc7820>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.

- [RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <<https://www.rfc-editor.org/info/rfc8186>>.
- [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>>.
- [RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019, <<https://www.rfc-editor.org/info/rfc8545>>.
- [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>>.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.
- [I-D.ietf-spring-sr-replication-segment]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-00 (work in progress), July 2020.
- [I-D.shen-spring-p2mp-transport-chain]
Shen, Y., Zhang, Z., Parekh, R., Bidgoli, H., and Y. Kamite, "Point-to-Multipoint Transport Using Chain Replication in Segment Routing", draft-shen-spring-p2mp-transport-chain-02 (work in progress), April 2020.
- [I-D.ietf-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.

[I-D.ietf-spring-mpls-path-segment]

Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler,
"Path Segment in MPLS Based Segment Routing Network",
draft-ietf-spring-mpls-path-segment-03 (work in progress),
September 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-24 (work in
progress), October 2020.

[BBF.TR-390]

"Performance Measurement from IP Edge to Customer
Equipment using TWAMP Light", BBF TR-390, May 2017.

[I-D.gandhi-mpls-ioam-sr]

Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B.,
and V. Kozak, "MPLS Data Plane Encapsulation for In-situ
OAM Data", draft-gandhi-mpls-ioam-sr-03 (work in
progress), September 2020.

[I-D.ali-spring-ioam-srv6]

Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar,
N., Pignataro, C., Li, C., Chen, M., and G. Dawra,
"Segment Routing Header encapsulation for In-situ OAM
Data", draft-ali-spring-ioam-srv6-02 (work in progress),
November 2019.

[I-D.ietf-pce-sr-bidir-path]

Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong,
"PCEP Extensions for Associated Bidirectional Segment
Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-03 (work
in progress), September 2020.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING
Internet-Draft
Intended status: Standards Track
Expires: March 31, 2019

J. Guichard, Ed.
H. Song
Huawei
J. Tantsura
Nuage Networks
J. Halpern
Ericsson
W. Henderickx
Nokia
M. Boucadair
Orange
September 27, 2018

NSH and Segment Routing Integration for Service Function Chaining (SFC)
draft-guichard-spring-nsh-sr-00

Abstract

This document describes two application scenarios where Network Service Header (NSH) and Segment Routing (SR) techniques can be deployed together to support Service Function Chaining (SFC) in an efficient manner while maintaining separation of the service and transport planes as originally intended by the SFC architecture.

In the first scenario, an NSH-based SFC is created using SR as the transport between SFFs. SR in this case is just one of many encapsulations that could be used to maintain the transport-independent nature of NSH-based service chains.

In the second scenario, SR is used to represent each service hop of the NSH-based SFC as a segment within the segment-list. SR and NSH in this case are integrated.

In both scenarios SR is responsible for steering packets between SFFs along a given SFP while NSH is responsible for maintaining the integrity of the service plane, the SFC instance context, and any associated metadata.

These application scenarios demonstrate that NSH and SR can work jointly and complement each other leaving the network operator with the flexibility to use whichever transport technology makes sense in specific areas of their network infrastructure, and still maintain an end-to-end service plane using NSH.

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 March 31, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. SFC Overview and Rationale	3
1.2. SFC within SR Networks	4
2. NSH-based SFC with SR-based transport tunnel	5
3. SR-based SFC with Integrated NSH Service Plane	9
4. Encapsulation Details	11
4.1. NSH using MPLS-SR Transport	11
4.2. NSH using SRv6 Transport	12
5. Security Considerations	13
6. IANA Considerations	13
6.1. UDP Port Number for NSH	13
6.2. Protocol Number for NSH	14
7. Acknowledgments	14
8. References	14

8.1. Normative References	14
8.2. Informative References	15
Authors' Addresses	15

1. Introduction

1.1. SFC Overview and Rationale

The dynamic enforcement of a service-derived, adequate forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators and service providers. Particularly, cascading SFs, for example at the Gi interface in the context of mobile network infrastructure, have shown their limits, such as the same redundant classification features must be supported by many SFs in order to execute their function, some SFs are receiving traffic that they are not supposed to process (e.g., TCP proxies receiving UDP traffic), which inevitably affects their dimensioning and performance, an increased design complexity related to the properly ordered invocation of several SFs, etc.

In order to solve those problems and to avoid the adherence with the underlying physical network topology while allowing for simplified service delivery, Service Function Chaining (SFC) techniques have been introduced.

SFC techniques are meant to rationalize the service delivery logic and master the companion complexity while optimizing service activation time cycles for operators that need more agile service delivery procedures to better accommodate ever-demanding customer requirements. Indeed, SFC allows to dynamically create service planes that can be used by specific traffic flows. Each service plane is realized by invoking and chaining the relevant service functions in the right sequence. [RFC7498] provides an overview of the SFC problem space and [RFC7665] specifies an SFC architecture. The SFC architecture has the merit to not make assumptions on how advanced features (e.g., load-balancing, loose or strict service paths) have to be enabled with a domain. Various deployment options are made available to operators with the SFC architecture and this approach is fundamental to accommodate various and heterogeneous deployment contexts.

Many approaches can be considered for encoding the information required for SFC purposes (e.g., communicate a service chain pointer, encode a list of loose/explicit paths, disseminate a service chain identifier together with a set of context information, etc.). Likewise, many approaches can also be considered for the channel to be used to carry SFC-specific information (e.g., define a new header, re-use existing fields, define an IPv6 extension header, etc.).

Among all these approaches, the IETF endorsed a transport-independent SFC encapsulation scheme: NSH [RFC8300]; which is the most mature SFC encapsulation solution. This design is pragmatic as it does not require replicating the same specification effort as a function of underlying transport encapsulation. Moreover, this design approach encourages consistent SFC-based service delivery in networks enabling distinct transport protocols in various segments of the network or even between SFFs vs SF-SFF hops.

1.2. SFC within SR Networks

As described in [I-D.ietf-spring-segment-routing], Segment Routing (SR) leverages the source routing technique. Concretely, a node steers a packet through an SR policy instantiated as an ordered list of instructions called segments. While initially designed for policy-based source routing, SR also finds its application in supporting SFC [I-D.xu-clad-spring-sr-service-chaining]. The two SR flavors, namely MPLS-SR [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.ietf-6man-segment-routing-header], can both encode a Service Function (SF) as a segment so that an SFC can be specified as a segment list. Nevertheless, and as discussed in [RFC7498], traffic steering is only a subset of the issues that motivated the design of the SFC architecture. Further considerations such as simplifying classification at intermediate SFs and allowing for coordinated behaviors among SFs by means of supplying context information should be taken into account when designing an SFC data plane solution.

While each scheme (i.e., NSH-based SFC and SR-based SFC) can work independently, this document describes how the two can be used together in concert and complement each other through two representative application scenarios. Both application scenarios may be supported using either MPLS-SR or SRv6:

- o NSH-based SFC with SR-based transport plane: in this scenario segment routing provides the transport encapsulation between SFFs while NSH is used to convey and trigger SFC policies.
- o SR-based SFC with integrated NSH service plane: in this scenario each service hop of the SFC is represented as a segment of the SR segment-list. SR is responsible for steering traffic through the necessary SFFs as part of the segment routing path and NSH is responsible for maintaining the service plane, and holding the SFC instance context and associated metadata.

It is of course possible to combine both of these two scenarios so as to support specific deployment requirements and use cases.

2. NSH-based SFC with SR-based transport tunnel

Because of the transport-independent nature of NSH-based service chains, it is expected that the NSH has broad applicability across different domains of a network. By way of illustration the various SFs involved in a service chain are available in a single data center, or spread throughout multiple locations (e.g., data centers, different POPs), depending upon the operator preference and/or availability of service resources. Regardless of where the service resources are deployed it is necessary to provide traffic steering through a set of SFFs and NSH-based service chains provide the flexibility for the network operator to choose which particular transport encapsulation to use between SFFs, which may be different depending upon which area of the network the SFFs/SFs are currently deployed. Therefore from an SFC architecture perspective, segment routing is simply one of multiple available transport encapsulations that can be used for traffic steering between SFFs. Concretely, NSH does not require to use a unique transport encapsulation when traversing a service chain. NSH-based service forwarding relies upon underlying service node capabilities.

The following three figures provide an example of an SFC established for flow F that has SF instances located in different data centers, DC1 and DC2. For the purpose of illustration, let the SFC's Service Path Identifier (SPI) be 100 and the initial Service Index (SI) be 255.

Referring to Figure 1, packets of flow F in DC1 are classified into an NSH-based SFC and encapsulated after classification as <Inner Pkt><NSH: SPI 100, SI 255><Outer-transport> and forwarded to SFF1 (which is the first SFF hop for this service chain).

After removing the outer transport encapsulation, that may or may not be MPLS-SR or SRv6, SFF1 uses the SPI and SI carried within the NSH encapsulation to determine that it should forward the packet to SF1. SF1 applies its service, decrements the SI by 1, and returns the packet to SFF1. SFF1 therefore has <SPI 100, SI 254> when the packet comes back from SF1. SFF1 does a lookup on <SPI 100, SI 254> which results in <next-hop: DC1-GW1> and forwards the packet to DC1-GW1.

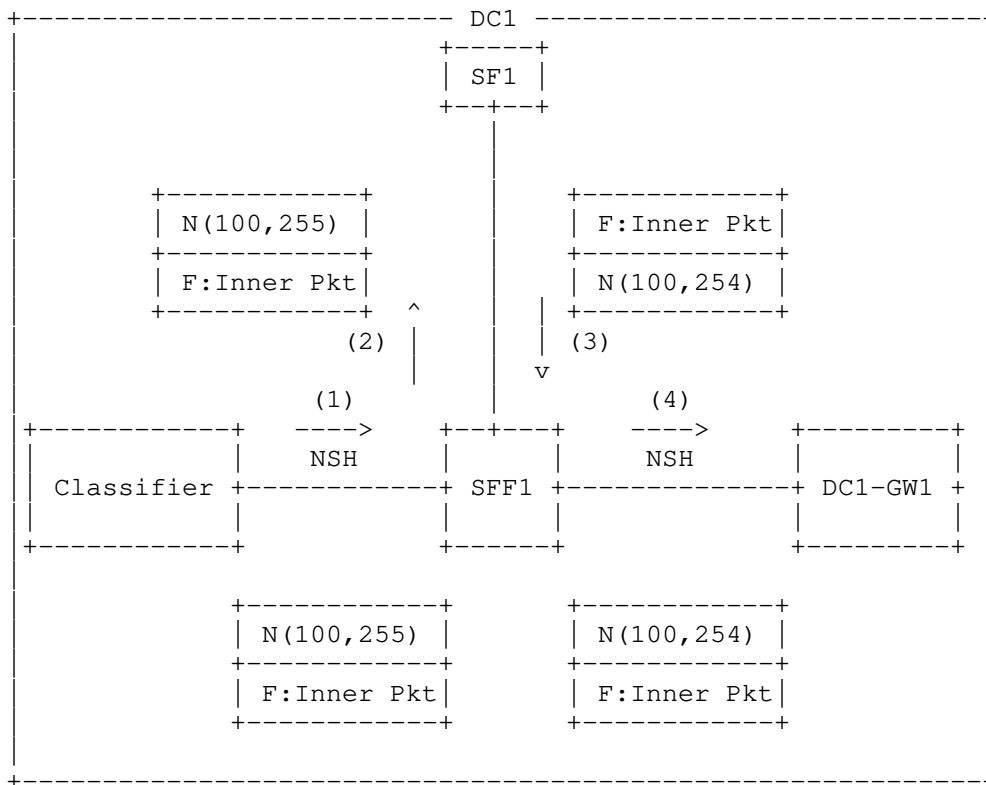


Figure 1: SR for inter-DC SFC - Part 1

Referring now to Figure 2, DC1-GW1 performs a lookup on the information conveyed in the NSH which results in <next-hop: DC2-GW1, encapsulation: SR>. The SR encapsulation has the SR segment-list to forward the packet across the inter-DC network to DC2.

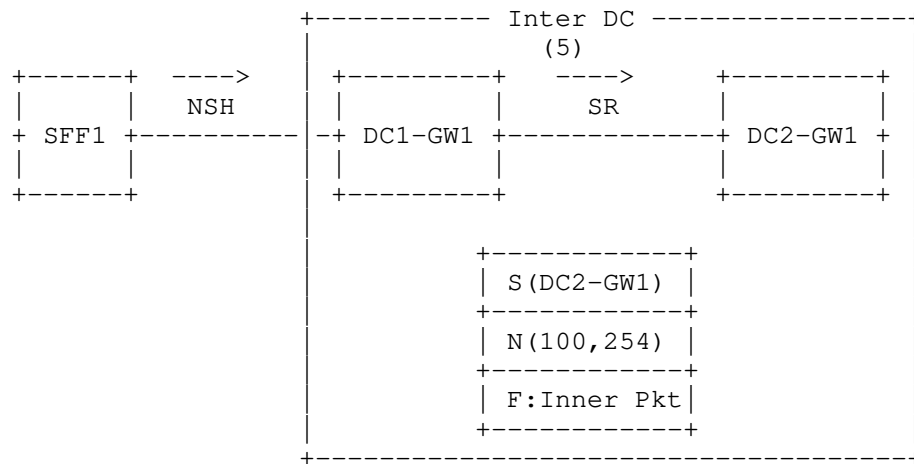


Figure 2: SR for inter-DC SFC - Part 2

When the packet arrives at DC2, as shown in Figure 3, the SR encapsulation is removed and DC2-GW1 performs a lookup on the NSH which results in next-hop: SFF2. The outer transport encapsulation may be any transport that is able to identify NSH as the next protocol.

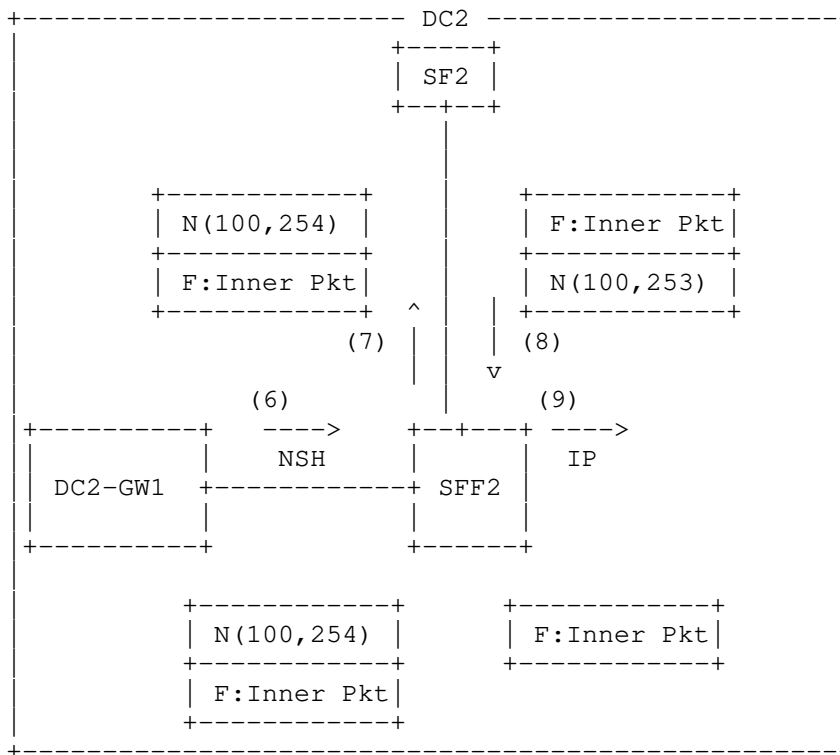


Figure 3: SR for inter-DC SFC - Part 3

The benefits of this scheme are listed hereafter:

- o The network operator is able to take advantage of the transport-independent nature of the NSH encapsulation.
- o The network operator is able to take advantage of the traffic steering capability of SR where appropriate.
- o Light-weight NSH is used in the data center for SFC and avoids more complex hierarchical SFC schemes between data centers.
- o Clear responsibility division and scope between NSH and SR.

Note that this scenario is applicable to any case where multiple segments of a service chain are distributed into multiple domains or where traffic-engineered paths are necessary between SFFs (strict forwarding paths for example). Further note that the above example can also be implemented using end to end segment routing between SFF1

and SFF2. (As such DC-GW1 and DC-GW2 are forwarding the packets based on segment routing instructions and are not looking at the NSH header for forwarding).

3. SR-based SFC with Integrated NSH Service Plane

In this scenario we assume that the SFs are NSH-aware and therefore it should not be necessary to implement an SFC proxy to achieve Service Function Chaining. The operation relies upon SR to perform SFF-SFF transport and NSH to provide the service plane between SFs thereby maintaining SFC context and metadata.

When a service chain is established, a packet associated with that chain will first encapsulate an NSH that will be used to maintain the end-to-end service plane through use of the SFC context. The SFC context (e.g., the service plane path referenced by the SPI) is used by an SFF to determine the SR segment list for forwarding the packet to the next-hop SFFs. The packet is then encapsulated using the (transport-specific) SR header and forwarded in the SR domain following normal SR operation.

When a packet has to be forwarded to an SF attached to an SFF, the SFF performs a lookup on the prefix SID associated with the SF to retrieve the next-hop context between the SFF and SF. E.g. to retrieve the destination MAC address in case native ethernet encapsulation is used between SFF and SF. How the next-hop context is populated is out of the scope of this document. The SFF strips the SR information of the packet, updates the SR information, and saves it to a cache indexed by the NSH SPI. This saved SR information is used to encapsulate and forward the packet(s) coming back from the SF.

When the SF receives the packet, it processes it as usual and sends it back to the SFF. Once the SFF receives this packet, it extracts the SR information using the NSH SPI as the index into the cache. The SFF then pushes the SR header on top of the NSH header, and forwards the packet to the next segment in the segment list.

Figure 4 illustrates an example of this scenario.

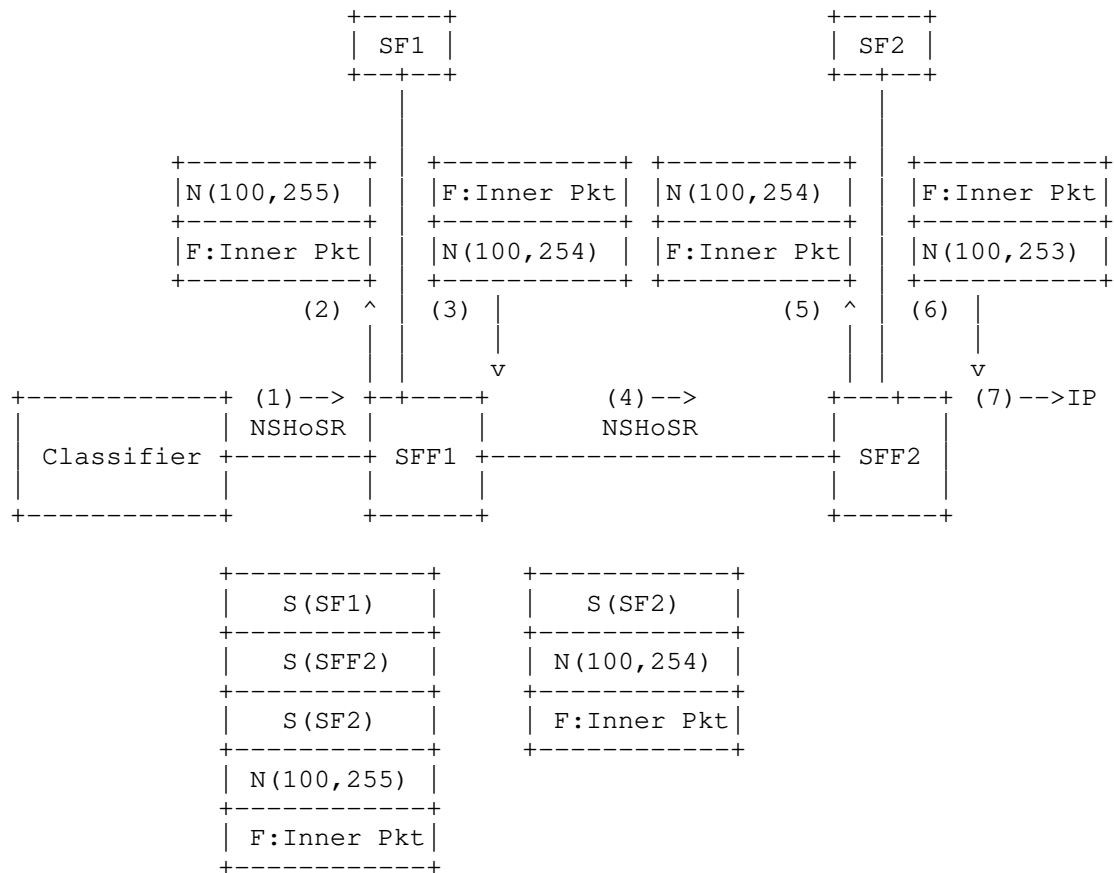


Figure 4: NSH over SR for SFC

The benefits of this scheme include:

- o It is economically sound for SF vendors to only support one unified SFC solution. The SF is unaware of the SR.
- o It simplifies the SFF (i.e., the SR router) by nullifying the needs for re-classification and SR proxy.
- o It provides a unique and standard way to pass metadata to SFs. Note that currently there is no solution for MPLS-SR to carry metadata and there is no solution to pass metadata to SR-unaware SFs.
- o SR is also used for forwarding purposes including between SFFs.

- o It takes advantage of SR to eliminate the NSH forwarding state in SFFs. This applies each time strict or loose SFFs are in use.
- o It requires no interworking as would be the case if MPLS-SR based SFC and NSH-based SFC were deployed as independent mechanisms in different parts of the network.

4. Encapsulation Details

4.1. NSH using MPLS-SR Transport

MPLS-SR instantiates Segment IDs (SIDs) as MPLS labels and therefore the segment routing header is a stack of MPLS labels.

When carrying NSH within an MPLS-SR transport, the full encapsulation headers are as illustrated in Figure 5.

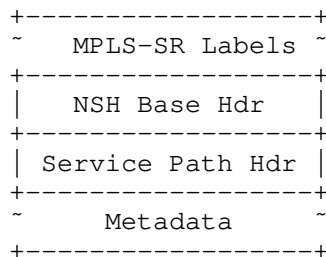


Figure 5: NSH using MPLS-SR Transport

As described in [I-D.ietf-spring-segment-routing] the IGP signaling extension for IGP-Prefix segment includes a flag to indicate whether directly connected neighbors of the node on which the prefix is attached should perform the NEXT operation or the CONTINUE operation when processing the SID. When NSH is carried beneath MPLS-SR it is necessary to terminate the NSH-based SFC at the tail-end node of the MPLS-SR label stack. This is the equivalent of MPLS Ultimate Hop Popping (UHP) and therefore the prefix-SID associated with the tail-end of the SFC MUST be advertised with the CONTINUE operation so that the penultimate hop node does not pop the top label of the MPLS-SR label stack and thereby expose NSH to the wrong SFF. It is RECOMMENDED that a specific prefix-SID be allocated at each node for use by the SFC application for this purpose.

At the end of the MPLS-SR path it is necessary to provide an indication to the tail-end that NSH follows the MPLS-SR label stack.

There are several ways to achieve this but its specification is outside the scope of this document.

4.2. NSH using SRv6 Transport

When carrying NSH within an SRv6 transport the full encapsulation is as illustrated in Figure 6.

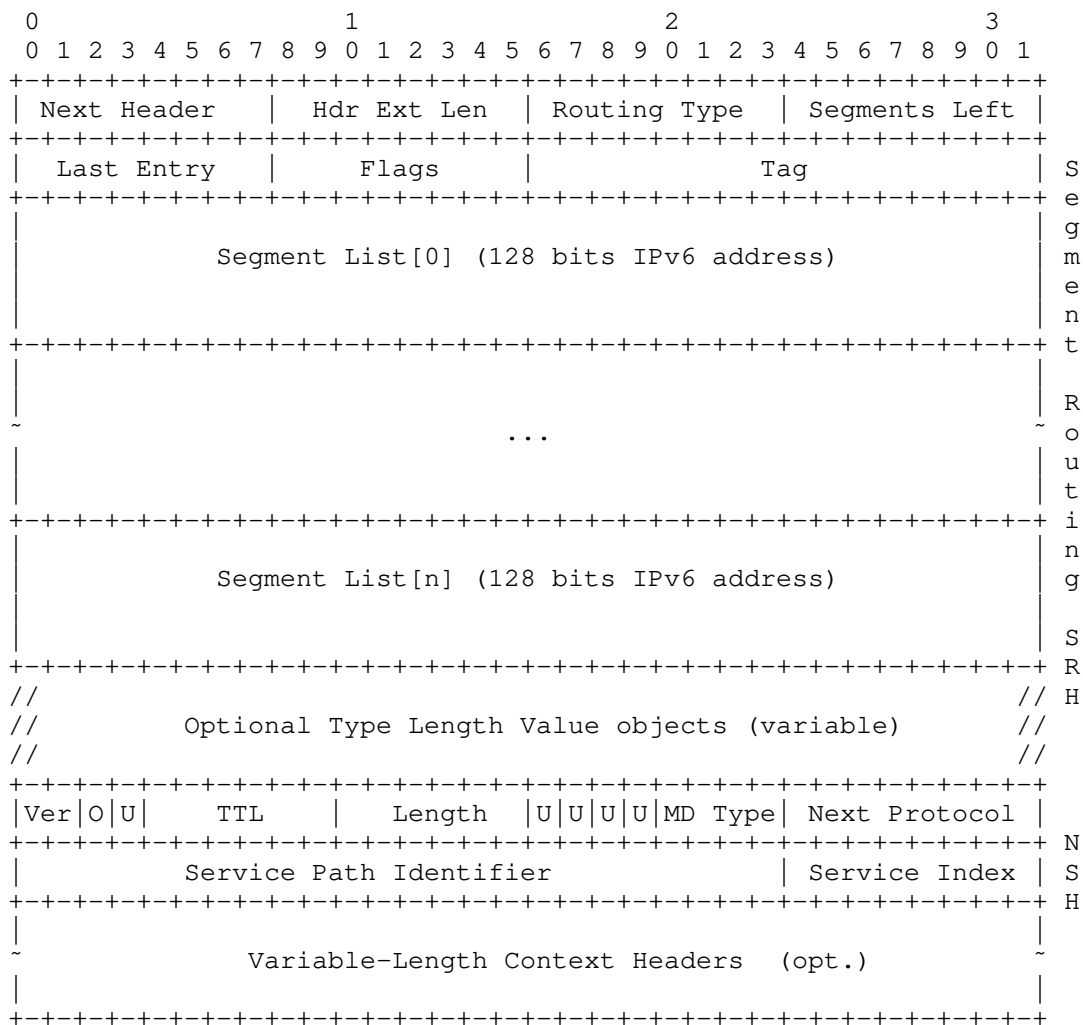


Figure 6: NSH using SRv6 Transport

Encapsulation of NSH following SRv6 may be indicated either by encapsulating NSH in UDP (UDP port TBA1) and indicating UDP in the Next Header field of the SRH, or by indicating an IP protocol number for NSH in the Next Header of the SRH. The behavior for encapsulating NSH over UDP, including the selection of the source port number in particular, adheres to similar considerations as those discussed in [RFC8086].

5. Security Considerations

Generic SFC-related security considerations are discussed in [RFC7665]. NSH-specific security considerations are discussed in [RFC8300]. NSH-in-UDP with DTLS [RFC6347] should follow the considerations discussed in Section 5 of [RFC8086], with a destination port number set to TBA2

6. IANA Considerations

6.1. UDP Port Number for NSH

IANA is requested to assign the UDP port numbers TBA1 and TBA2 to the NSH from the "Service Name and Transport Protocol Port Number Registry" available at <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>:

Service Name: NSH-in-UDP
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP Encapsulation
Reference: [ThisDocument]
Port Number: TBA1
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

Service Name: NSH-UDP-DTLS
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP with DTLS Encapsulation
Reference: [ThisDocument]
Port Number: TBA2
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

6.2. Protocol Number for NSH

IANA is requested to assign a protocol number TBA3 for the NSH from the "Assigned Internet Protocol Numbers" registry available at <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

Decimal	Keyword	Protocol	IPv6 Extension Header	Reference
TBA3	NSH	Network Service Header	N	[ThisDocument]

7. Acknowledgments

TBD.

8. References

8.1. Normative References

- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.
- [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-12 (work in progress), February 2018.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

8.2. Informative References

- [I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J.,
Field, B., daniel.voyer@bell.ca, d.,
daniel.bernier@bell.ca, d., Matsushima, S., Leung, I.,
Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun,
D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing
Header (SRH)", draft-ietf-6man-segment-routing-header-09
(work in progress), March 2018.
- [I-D.xu-clad-spring-sr-service-chaining]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca,
d., Decraene, B., Yadlapalli, C., Henderickx, W., Salsano,
S., and S. Ma, "Segment Routing for Service Chaining",
draft-xu-clad-spring-sr-service-chaining-00 (work in
progress), December 2017.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for
Service Function Chaining", RFC 7498,
DOI 10.17487/RFC7498, April 2015,
<<https://www.rfc-editor.org/info/rfc7498>>.

Authors' Addresses

James N Guichard (editor)
Huawei
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@huawei.com

Haoyu Song
Huawei
2330 Central Express Way
Santa Clara
USA

Email: haoyu.song@huawei.com

Jeff Tantsura
Nuage Networks
USA

Email: jefftant.ietf@gmail.com

Joel Halpern
Ericsson
USA

Email: joel.halpern@ericsson.com

Wim Henderickx
Nokia
USA

Email: wim.henderickx@nokia.com

Mohamed Boucadair
Orange
USA

Email: mohamed.boucadair@orange.com

SPRING
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

J. Guichard, Ed.
H. Song
Huawei
J. Tantsura
Nuage Networks
J. Halpern
Ericsson
W. Henderickx
Nokia
M. Boucadair
Orange
S. Hassan
Cisco Systems
March 11, 2019

NSH and Segment Routing Integration for Service Function Chaining (SFC)
draft-guichard-spring-nsh-sr-01

Abstract

This document describes two application scenarios where Network Service Header (NSH) and Segment Routing (SR) techniques can be deployed together to support Service Function Chaining (SFC) in an efficient manner while maintaining separation of the service and transport planes as originally intended by the SFC architecture.

In the first scenario, an NSH-based SFC is created using SR as the transport between SFFs. SR in this case is just one of many encapsulations that could be used to maintain the transport-independent nature of NSH-based service chains.

In the second scenario, SR is used to represent each service hop of the NSH-based SFC as a segment within the segment-list. SR and NSH in this case are integrated.

In both scenarios SR is responsible for steering packets between SFFs along a given SFP while NSH is responsible for maintaining the integrity of the service plane, the SFC instance context, and any associated metadata.

These application scenarios demonstrate that NSH and SR can work jointly and complement each other leaving the network operator with the flexibility to use whichever transport technology makes sense in specific areas of their network infrastructure, and still maintain an end-to-end service plane using NSH.

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 September 12, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. SFC Overview and Rationale	3
1.2. SFC within SR Networks	4
2. NSH-based SFC with SR-based transport tunnel	5
3. SR-based SFC with Integrated NSH Service Plane	9
4. Encapsulation Details	11
4.1. NSH using MPLS-SR Transport	11
4.2. NSH using SRv6 Transport	12
5. Security Considerations	13
6. IANA Considerations	13
6.1. UDP Port Number for NSH	13
6.2. Protocol Number for NSH	14
7. Acknowledgments	14
8. References	14

8.1. Normative References	14
8.2. Informative References	15
Authors' Addresses	15

1. Introduction

1.1. SFC Overview and Rationale

The dynamic enforcement of a service-derived, adequate forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators and service providers. Particularly, cascading SFs, for example at the Gi interface in the context of mobile network infrastructure, have shown their limits, such as the same redundant classification features must be supported by many SFs in order to execute their function, some SFs are receiving traffic that they are not supposed to process (e.g., TCP proxies receiving UDP traffic), which inevitably affects their dimensioning and performance, an increased design complexity related to the properly ordered invocation of several SFs, etc.

In order to solve those problems and to avoid the adherence with the underlying physical network topology while allowing for simplified service delivery, Service Function Chaining (SFC) techniques have been introduced.

SFC techniques are meant to rationalize the service delivery logic and master the companion complexity while optimizing service activation time cycles for operators that need more agile service delivery procedures to better accommodate ever-demanding customer requirements. Indeed, SFC allows to dynamically create service planes that can be used by specific traffic flows. Each service plane is realized by invoking and chaining the relevant service functions in the right sequence. [RFC7498] provides an overview of the SFC problem space and [RFC7665] specifies an SFC architecture. The SFC architecture has the merit to not make assumptions on how advanced features (e.g., load-balancing, loose or strict service paths) have to be enabled with a domain. Various deployment options are made available to operators with the SFC architecture and this approach is fundamental to accommodate various and heterogeneous deployment contexts.

Many approaches can be considered for encoding the information required for SFC purposes (e.g., communicate a service chain pointer, encode a list of loose/explicit paths, disseminate a service chain identifier together with a set of context information, etc.). Likewise, many approaches can also be considered for the channel to be used to carry SFC-specific information (e.g., define a new header, re-use existing fields, define an IPv6 extension header, etc.).

Among all these approaches, the IETF endorsed a transport-independent SFC encapsulation scheme: NSH [RFC8300]; which is the most mature SFC encapsulation solution. This design is pragmatic as it does not require replicating the same specification effort as a function of underlying transport encapsulation. Moreover, this design approach encourages consistent SFC-based service delivery in networks enabling distinct transport protocols in various segments of the network or even between SFFs vs SF-SFF hops.

1.2. SFC within SR Networks

As described in [I-D.ietf-spring-segment-routing], Segment Routing (SR) leverages the source routing technique. Concretely, a node steers a packet through an SR policy instantiated as an ordered list of instructions called segments. While initially designed for policy-based source routing, SR also finds its application in supporting SFC [I-D.xu-clad-spring-sr-service-chaining]. The two SR flavors, namely MPLS-SR [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.ietf-6man-segment-routing-header], can both encode a Service Function (SF) as a segment so that an SFC can be specified as a segment list. Nevertheless, and as discussed in [RFC7498], traffic steering is only a subset of the issues that motivated the design of the SFC architecture. Further considerations such as simplifying classification at intermediate SFs and allowing for coordinated behaviors among SFs by means of supplying context information should be taken into account when designing an SFC data plane solution.

While each scheme (i.e., NSH-based SFC and SR-based SFC) can work independently, this document describes how the two can be used together in concert and complement each other through two representative application scenarios. Both application scenarios may be supported using either MPLS-SR or SRv6:

- o NSH-based SFC with SR-based transport plane: in this scenario segment routing provides the transport encapsulation between SFFs while NSH is used to convey and trigger SFC policies.
- o SR-based SFC with integrated NSH service plane: in this scenario each service hop of the SFC is represented as a segment of the SR segment-list. SR is responsible for steering traffic through the necessary SFFs as part of the segment routing path and NSH is responsible for maintaining the service plane, and holding the SFC instance context and associated metadata.

It is of course possible to combine both of these two scenarios so as to support specific deployment requirements and use cases.

2. NSH-based SFC with SR-based transport tunnel

Because of the transport-independent nature of NSH-based service chains, it is expected that the NSH has broad applicability across different domains of a network. By way of illustration the various SFs involved in a service chain are available in a single data center, or spread throughout multiple locations (e.g., data centers, different POPs), depending upon the operator preference and/or availability of service resources. Regardless of where the service resources are deployed it is necessary to provide traffic steering through a set of SFFs and NSH-based service chains provide the flexibility for the network operator to choose which particular transport encapsulation to use between SFFs, which may be different depending upon which area of the network the SFFs/SFs are currently deployed. Therefore from an SFC architecture perspective, segment routing is simply one of multiple available transport encapsulations that can be used for traffic steering between SFFs. Concretely, NSH does not require to use a unique transport encapsulation when traversing a service chain. NSH-based service forwarding relies upon underlying service node capabilities.

The following three figures provide an example of an SFC established for flow F that has SF instances located in different data centers, DC1 and DC2. For the purpose of illustration, let the SFC's Service Path Identifier (SPI) be 100 and the initial Service Index (SI) be 255.

Referring to Figure 1, packets of flow F in DC1 are classified into an NSH-based SFC and encapsulated after classification as <Inner Pkt><NSH: SPI 100, SI 255><Outer-transport> and forwarded to SFF1 (which is the first SFF hop for this service chain).

After removing the outer transport encapsulation, that may or may not be MPLS-SR or SRv6, SFF1 uses the SPI and SI carried within the NSH encapsulation to determine that it should forward the packet to SF1. SF1 applies its service, decrements the SI by 1, and returns the packet to SFF1. SFF1 therefore has <SPI 100, SI 254> when the packet comes back from SF1. SFF1 does a lookup on <SPI 100, SI 254> which results in <next-hop: DC1-GW1> and forwards the packet to DC1-GW1.

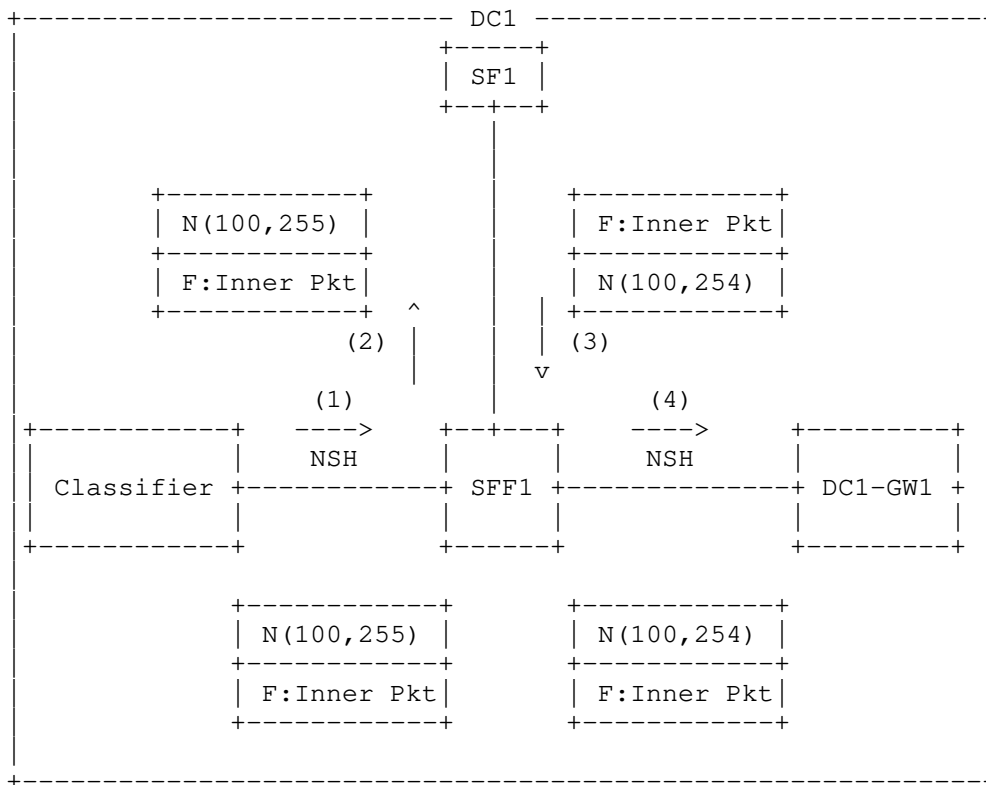


Figure 1: SR for inter-DC SFC - Part 1

Referring now to Figure 2, DC1-GW1 performs a lookup on the information conveyed in the NSH which results in <next-hop: DC2-GW1, encapsulation: SR>. The SR encapsulation has the SR segment-list to forward the packet across the inter-DC network to DC2.

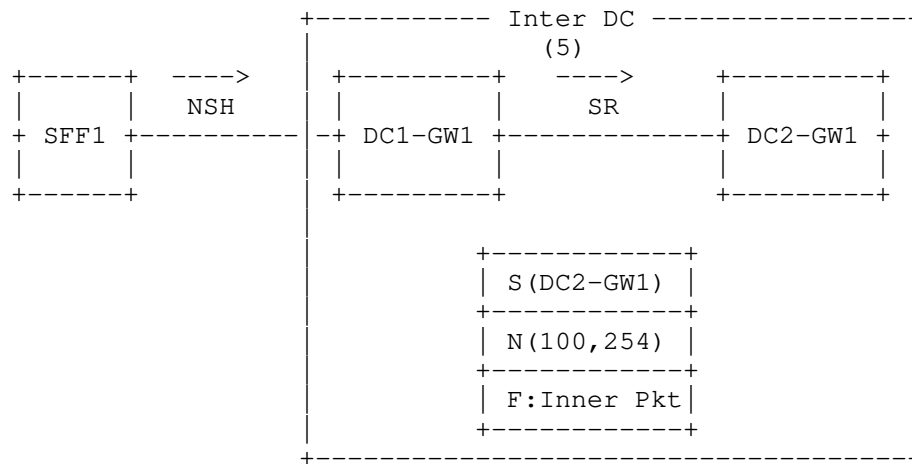


Figure 2: SR for inter-DC SFC - Part 2

When the packet arrives at DC2, as shown in Figure 3, the SR encapsulation is removed and DC2-GW1 performs a lookup on the NSH which results in next-hop: SFF2. The outer transport encapsulation may be any transport that is able to identify NSH as the next protocol.

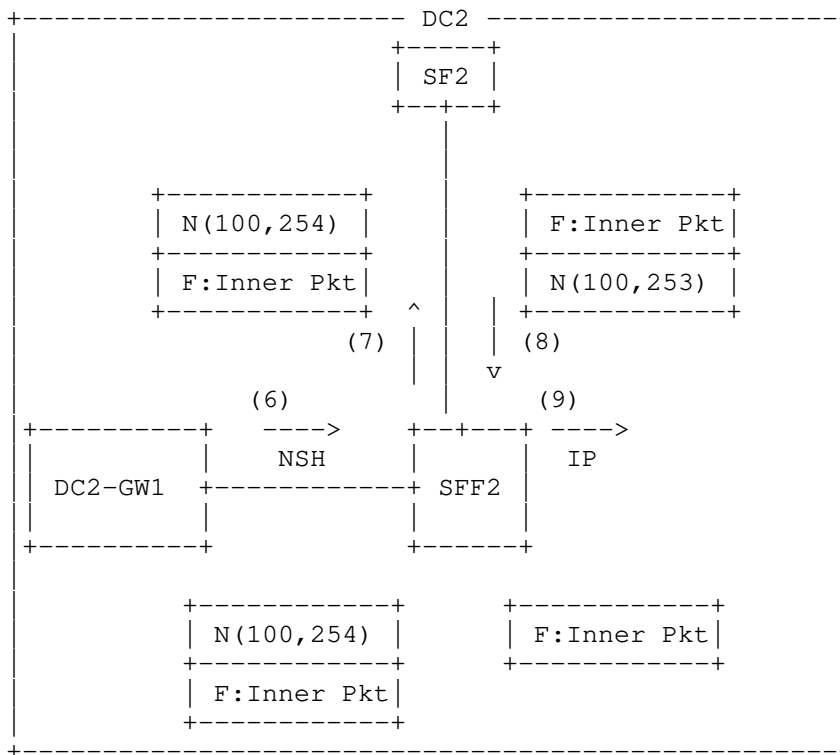


Figure 3: SR for inter-DC SFC - Part 3

The benefits of this scheme are listed hereafter:

- o The network operator is able to take advantage of the transport-independent nature of the NSH encapsulation.
- o The network operator is able to take advantage of the traffic steering capability of SR where appropriate.
- o Light-weight NSH is used in the data center for SFC and avoids more complex hierarchical SFC schemes between data centers.
- o Clear responsibility division and scope between NSH and SR.

Note that this scenario is applicable to any case where multiple segments of a service chain are distributed into multiple domains or where traffic-engineered paths are necessary between SFFs (strict forwarding paths for example). Further note that the above example can also be implemented using end to end segment routing between SFF1

and SFF2. (As such DC-GW1 and DC-GW2 are forwarding the packets based on segment routing instructions and are not looking at the NSH header for forwarding).

3. SR-based SFC with Integrated NSH Service Plane

In this scenario we assume that the SFs are NSH-aware and therefore it should not be necessary to implement an SFC proxy to achieve Service Function Chaining. The operation relies upon SR to perform SFF-SFF transport and NSH to provide the service plane between SFs thereby maintaining SFC context and metadata.

When a service chain is established, a packet associated with that chain will first encapsulate an NSH that will be used to maintain the end-to-end service plane through use of the SFC context. The SFC context (e.g., the service plane path referenced by the SPI) is used by an SFF to determine the SR segment list for forwarding the packet to the next-hop SFFs. The packet is then encapsulated using the (transport-specific) SR header and forwarded in the SR domain following normal SR operation.

When a packet has to be forwarded to an SF attached to an SFF, the SFF performs a lookup on the prefix SID associated with the SF to retrieve the next-hop context between the SFF and SF. E.g. to retrieve the destination MAC address in case native ethernet encapsulation is used between SFF and SF. How the next-hop context is populated is out of the scope of this document. The SFF strips the SR information of the packet, updates the SR information, and saves it to a cache indexed by the NSH SPI. This saved SR information is used to encapsulate and forward the packet(s) coming back from the SF.

When the SF receives the packet, it processes it as usual and sends it back to the SFF. Once the SFF receives this packet, it extracts the SR information using the NSH SPI as the index into the cache. The SFF then pushes the SR header on top of the NSH header, and forwards the packet to the next segment in the segment list.

Figure 4 illustrates an example of this scenario.

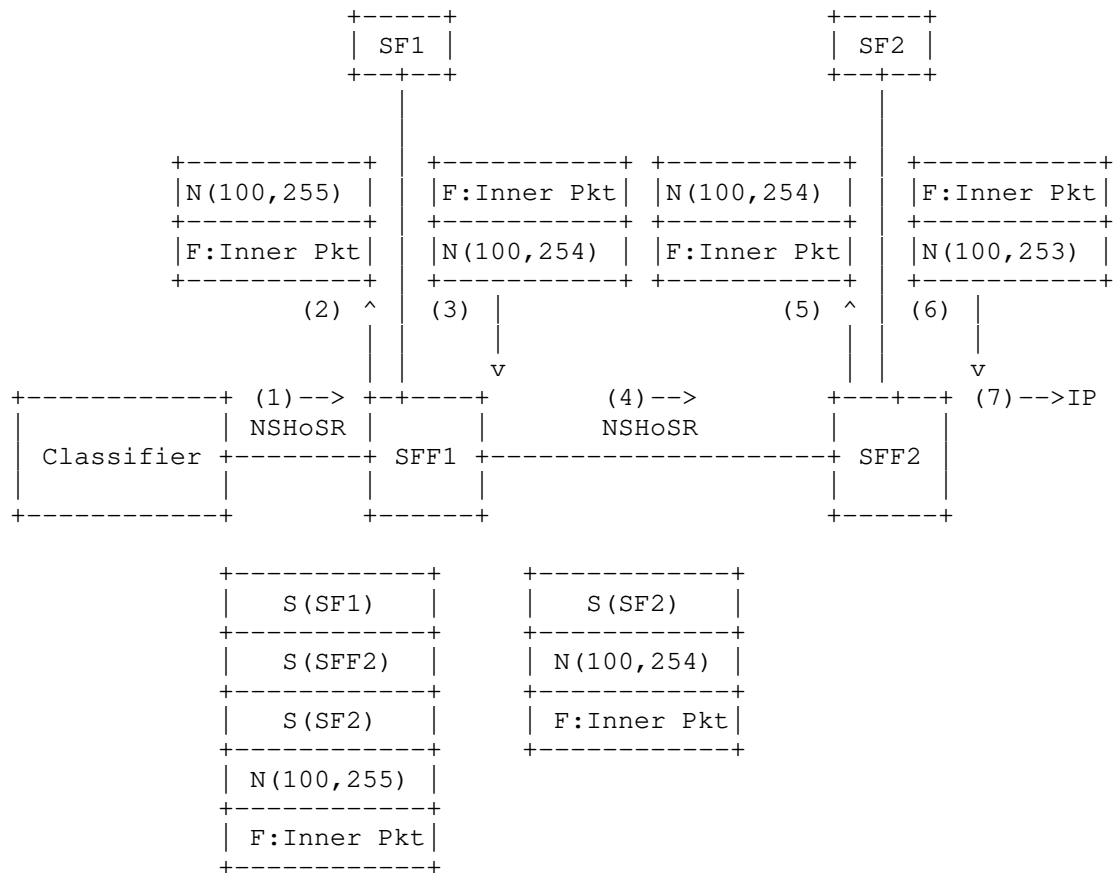


Figure 4: NSH over SR for SFC

The benefits of this scheme include:

- o It is economically sound for SF vendors to only support one unified SFC solution. The SF is unaware of the SR.
- o It simplifies the SFF (i.e., the SR router) by nullifying the needs for re-classification and SR proxy.
- o It provides a unique and standard way to pass metadata to SFs. Note that currently there is no solution for MPLS-SR to carry metadata and there is no solution to pass metadata to SR-unaware SFs.
- o SR is also used for forwarding purposes including between SFFs.

- o It takes advantage of SR to eliminate the NSH forwarding state in SFFs. This applies each time strict or loose SFFs are in use.
- o It requires no interworking as would be the case if MPLS-SR based SFC and NSH-based SFC were deployed as independent mechanisms in different parts of the network.

4. Encapsulation Details

4.1. NSH using MPLS-SR Transport

MPLS-SR instantiates Segment IDs (SIDs) as MPLS labels and therefore the segment routing header is a stack of MPLS labels.

When carrying NSH within an MPLS-SR transport, the full encapsulation headers are as illustrated in Figure 5.

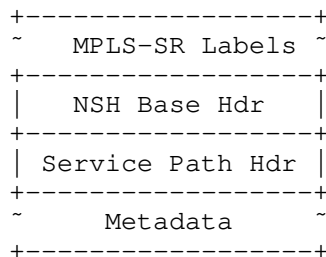


Figure 5: NSH using MPLS-SR Transport

As described in [I-D.ietf-spring-segment-routing] the IGP signaling extension for IGP-Prefix segment includes a flag to indicate whether directly connected neighbors of the node on which the prefix is attached should perform the NEXT operation or the CONTINUE operation when processing the SID. When NSH is carried beneath MPLS-SR it is necessary to terminate the NSH-based SFC at the tail-end node of the MPLS-SR label stack. This is the equivalent of MPLS Ultimate Hop Popping (UHP) and therefore the prefix-SID associated with the tail-end of the SFC MUST be advertised with the CONTINUE operation so that the penultimate hop node does not pop the top label of the MPLS-SR label stack and thereby expose NSH to the wrong SFF. It is RECOMMENDED that a specific prefix-SID be allocated at each node for use by the SFC application for this purpose.

At the end of the MPLS-SR path it is necessary to provide an indication to the tail-end that NSH follows the MPLS-SR label stack.

There are several ways to achieve this but its specification is outside the scope of this document.

4.2. NSH using SRv6 Transport

When carrying NSH within an SRv6 transport the full encapsulation is as illustrated in Figure 6.

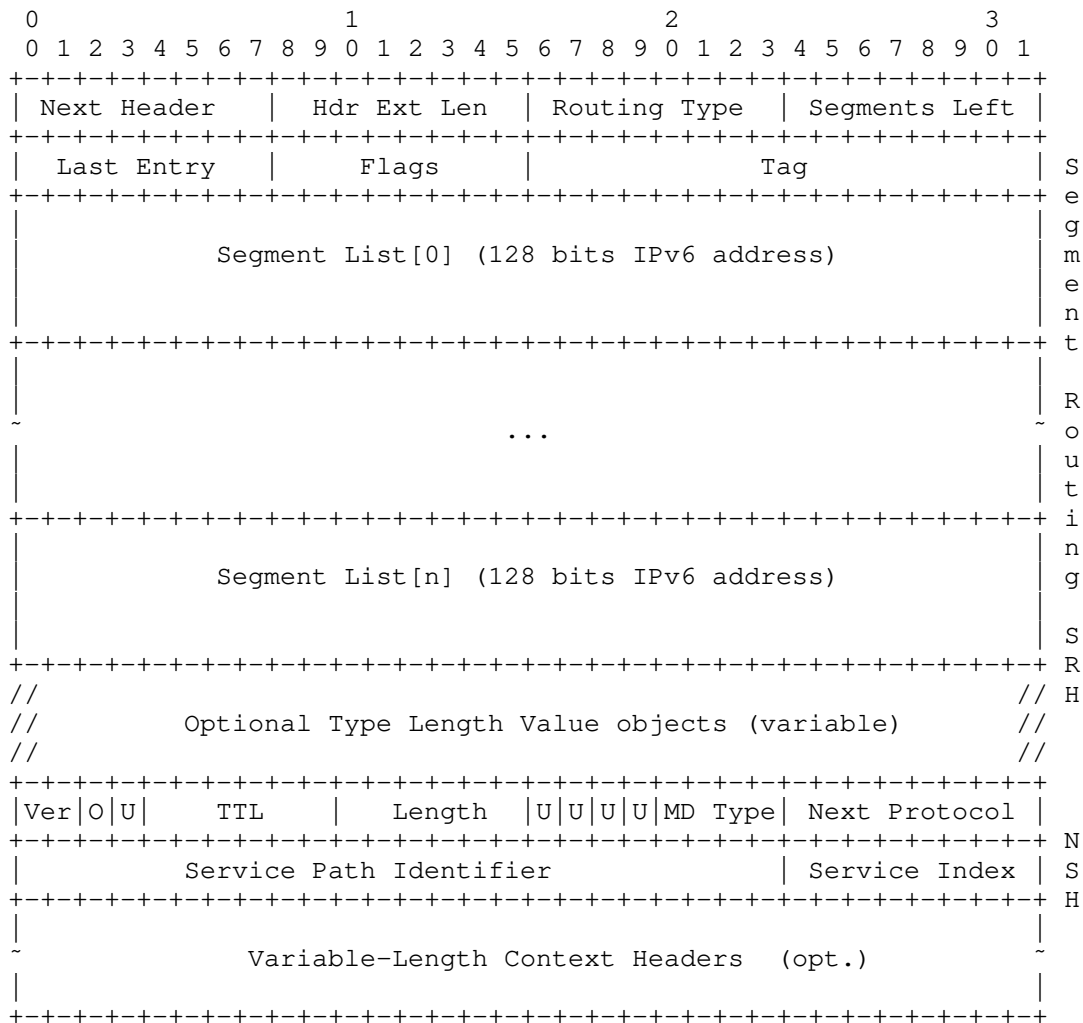


Figure 6: NSH using SRv6 Transport

Encapsulation of NSH following SRv6 may be indicated either by encapsulating NSH in UDP (UDP port TBA1) and indicating UDP in the Next Header field of the SRH, or by indicating an IP protocol number for NSH in the Next Header of the SRH. The behavior for encapsulating NSH over UDP, including the selection of the source port number in particular, adheres to similar considerations as those discussed in [RFC8086].

5. Security Considerations

Generic SFC-related security considerations are discussed in [RFC7665]. NSH-specific security considerations are discussed in [RFC8300]. NSH-in-UDP with DTLS [RFC6347] should follow the considerations discussed in Section 5 of [RFC8086], with a destination port number set to TBA2

6. IANA Considerations

6.1. UDP Port Number for NSH

IANA is requested to assign the UDP port numbers TBA1 and TBA2 to the NSH from the "Service Name and Transport Protocol Port Number Registry" available at <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>:

Service Name: NSH-in-UDP
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP Encapsulation
Reference: [ThisDocument]
Port Number: TBA1
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

Service Name: NSH-UDP-DTLS
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP with DTLS Encapsulation
Reference: [ThisDocument]
Port Number: TBA2
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

6.2. Protocol Number for NSH

IANA is requested to assign a protocol number TBA3 for the NSH from the "Assigned Internet Protocol Numbers" registry available at <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

Decimal	Keyword	Protocol	IPv6 Extension Header	Reference
TBA3	NSH	Network Service Header	N	[ThisDocument]

7. Acknowledgments

TBD.

8. References

8.1. Normative References

- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.
- [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-12 (work in progress), February 2018.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

8.2. Informative References

- [I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J.,
Field, B., daniel.voyer@bell.ca, d.,
daniel.bernier@bell.ca, d., Matsushima, S., Leung, I.,
Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun,
D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing
Header (SRH)", draft-ietf-6man-segment-routing-header-09
(work in progress), March 2018.
- [I-D.xu-clad-spring-sr-service-chaining]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca,
d., Decraene, B., Yadlapalli, C., Henderickx, W., Salsano,
S., and S. Ma, "Segment Routing for Service Chaining",
draft-xu-clad-spring-sr-service-chaining-00 (work in
progress), December 2017.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for
Service Function Chaining", RFC 7498,
DOI 10.17487/RFC7498, April 2015,
<<https://www.rfc-editor.org/info/rfc7498>>.

Authors' Addresses

James N Guichard (editor)
Huawei
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@huawei.com

Haoyu Song
Huawei
2330 Central Express Way
Santa Clara
USA

Email: haoyu.song@huawei.com

Jeff Tantsura
Nuage Networks
USA

Email: jefftant.ietf@gmail.com

Joel Halpern
Ericsson
USA

Email: joel.halpern@ericsson.com

Wim Henderickx
Nokia
USA

Email: wim.henderickx@nokia.com

Mohamed Boucadair
Orange
USA

Email: mohamed.boucadair@orange.com

Syed Hassan
Cisco Systems
USA

Email: shassan@cisco.com

MPLS Workgroup
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2019

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Weiqiang Cheng
China Mobile
March 10, 2019

Inter-domain Use Cases of Segment Routing with MPLS Data Plane for
Transport Network
draft-hu-mpls-sr-inter-domain-use-cases-01

Abstract

This document discusses the inter-domain scenarios for Transport Profile of SR-MPLS (SR-MPLS-TP), including SR-MPLS-TP inter-working with MPLS-TP network.

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 September 11, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Transport Profile in SR-MPLS	3
4. SR-MPLS-TP Inter-domain	4
4.1. SR-MPLS-TP Stitching Inter-domain	4
4.1.1. Inter-domain Path Segment	4
4.1.2. Border Node Inter-domain Scenario	5
4.1.3. Border Link Inter-domain Scenario	5
4.2. SR-MPLS-TP Nesting Inter-domain	7
5. SR-MPLS-TP Inter-working with MPLS-TP	8
6. Security Considerations	10
7. Acknowledgements	10
8. IANA Considerations	10
9. Normative References	10
Authors' Addresses	11

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an SR Policy instantiated as an ordered list of instructions called "segments". A segment can represent any instruction, topological or service based. A segment can have a semantic local to an SR node or global within an SR domain. SR supports per-flow explicit routing while maintaining per-flow state only at the ingress nodes of the SR domain. Segment Routing can be instantiated on MPLS data plane or IPv6 data plane. The former is referred to as SR-MPLS [I-D.ietf-spring-segment-routing-mpls], the latter is SRv6 [I-D.ietf-6man-segment-routing-header]. SR-MPLS leverages the MPLS label stack to construct the SR path, and SRv6 uses the Segment Routing Header to construct the SR path.

[I-D.cheng-spring-mpls-path-segment] defines a Path Segment identifier to support bidirectional path correlation for transport network. This document defines an inter-domain path segment and discusses the inter-domain use cases in the context of the Transport Profile of SR-MPLS, referred to in this document as SR-MPLS-TP, and describes the use case related to SR-MPLS-TP inter-working with the MPLS-TP network.

2. Conventions used in this document

2.1. Terminology

A->B SID list: The SID List from SR node A to SR node B.

B-SID: Binding SID.

e-Path: End-to-end Path segment.

MPLS-TP: MPLS Transport Profile.

s-Path: Sub-path Path Segment.

i-Path/i-PSID: Inter-domain Path Segment.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SR-MPLS-TP: Transport Profile of SR-MPLS.

Border node inter-domain: Two domains interconnects with an edge node which belongs to both domains.

Border link inter-domain: Two domains interconnects with an inter-link which connects the edge node in each domain.

2.2. 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. Transport Profile in SR-MPLS

In the SR-MPLS network, an SR path is a unidirectional path. [I-D.cheng-spring-mpls-path-segment] defines a Path Segment identifier to support SR bidirectional path correlation for transport network. In the context of the Transport Profile of SR-MPLS, referred to in this document as SR-MPLS-TP, a Path Segment uniquely identifies an SR path in a specific context. For example, the Path Segment is used to indicate the intra-domain path in a single domain and correlate the two unidirectional SR paths at both ends of the paths.

In multi-domain scenario, the SR bidirectional end-to-end path MUST to be established in transport network. The SR-MPLS-TP inter-domain models include the stitching inter-domain model and the nesting inter-domain model. Path Segment MAY also be used to indicate the inter-domain path or the end-to-end path and correlate the inter-domain paths or end-to-end unidirectional paths.

4. SR-MPLS-TP Inter-domain

Two SR-MPLS-TP inter-domain models are discussed in this document including the stitching inter-domain model and the nesting inter-domain model. Two use cases of stitching SR-MPLS-TP domains, using a border node inter-domain and a border link inter-domain, are described in Section 4.1.1 and Section 4.1.2 respectively.

4.1. SR-MPLS-TP Stitching Inter-domain

4.1.1. Inter-domain Path Segment

In the stitching inter-domain model, the end-to-end SR path being split into multiple segments. And each segment can be identified by an inter-domain path segment (i-Path or i-PSID). The inter-domain path segment is valid in the corresponding domain and the border nodes maintain the forwarding entries of that i-Path segment mapping to the next i-Path. In the headend node, the i-Path can be mapped to the inter-domain path of reverse direction and correlates the two unidirectional paths. The border nodes should install the following MPLS data entries for Path segments:

incoming label: i-Path
outgoing label: the SID list of the next domain or link + next i-Path

Taking Figure 1 as an example, the border node X installs the MPLS data entries:

incoming label: i-Path(A->X)
outgoing label: X->Y SID list + i-Path(X->Y)

The i-Path can be a locally unique label and assigned from the Segment Routing Local Block (SRLB). It is required that the controller(e.g., PCE) assigns the label to ensure the ingress and the egress node can recognize it and it also can be assigned from egress node of each domain. PCEP based i-Path allocation and procedure is defined in [I-D.xiong-pce-stateful-pce-sr-inter-domain].

4.1.2. Border Node Inter-domain Scenario

The Figure 1 displays the border node inter-domain scenario. SR node X and SR node Y are the border nodes of two different domains. The i-Paths from A->X, X->Y, and Y->Z are used for the inter-domain path segment. The ingress SR node A encapsulates the data packet with i-Path (A->X) and A->X SID list. The data packet is forwarded to SR node X according to the A->X SID list. Node X pushes the i-Path (X->Y) and X->Y SID list based on the above mentioned forwarding entry. The data packet is forwarded to node Y and then to the SR node Z based on the same forwarding procedure. In node Z, the i-Path (Y->Z) can be mapped to the path from Z to Y of reverse direction and correlates the two unidirectional paths. The packet transmission of the reverse direction is the same with the forwarding direction with different i-Paths.

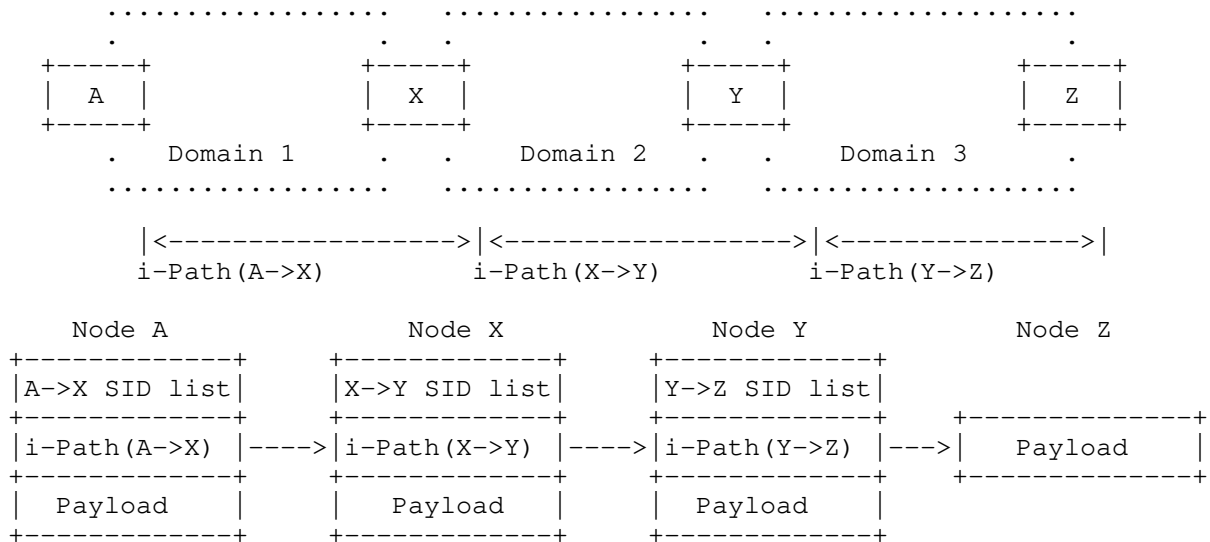


Figure 1: Stitching Border Node Inter-Domain Scenario

4.1.3. Border Link Inter-domain Scenario

Figure 2 illustrates the border link inter-domain scenario. All the SR nodes belong to a single domain. Neighboring border nodes of different domains are interconnected by direct physical or logical links. Ingress SR node A encapsulates the data packet with i-Path (A->B) and A->B SID list. The data packet is forwarded to SR node B

according to the A->B SID list. Node B pushes i-Path (B->C) and the inter-domain link label(B->C SID) based on the forwarding entry, and forwards the packet to node C. SR node C forwards the packet to node X, then node X forwards the packets to node Y. The data packet reaches the destination SR node Z according to the same forwarding procedure. In node Z, the i-Path (Y->Z) can be mapped to the path from Z to Y of reverse direction and correlates the two unidirectional paths. The packet transmission of the reverse direction is the same with the forwarding direction with different i-Paths.

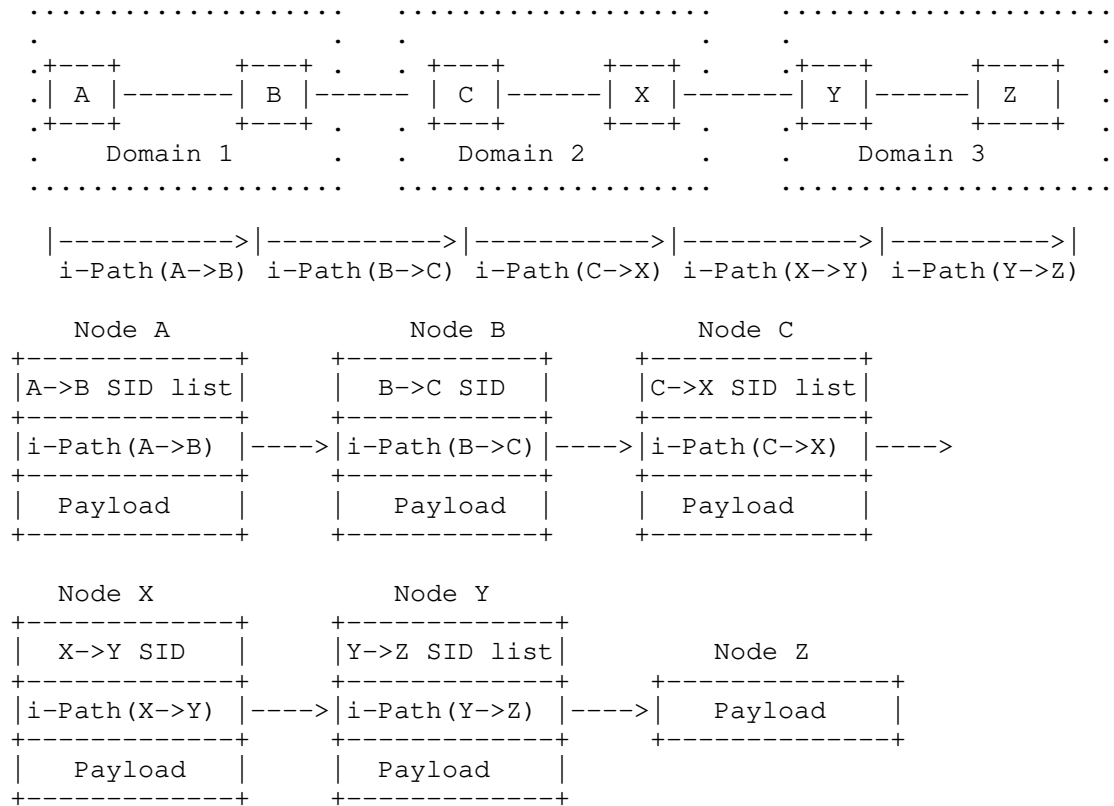


Figure 2: Stitching Border Link Inter-Domain Scenario

4.2. SR-MPLS-TP Nesting Inter-domain

The nesting inter-domain model is described in [I-D.cheng-spring-mpls-path-segment], an end-to-end path segment, also referred to as e-Path, is used to indicate the end-to-end path, and an s-Path is used to indicate the intra-domain path. The e-Path is encapsulated at the ingress nodes and decapsulated at the egress nodes. The transit nodes, even the border nodes of domains, are not aware of the e-Path segment. Only the s-Path is pushed and popped at the border nodes of the corresponding domain.

Figure 3 shows the SR-MPLS-TP nesting inter-domain scenario. The e-Path(A->Z) is used to indicate the end-to-end path. The s-Path is used to identify the domain's sub-path. The e-Path, s-Path and SR list are pushed by the ingress node. To reduce the size of the label stacks, the use of the binding SID [RFC8402] is recommended to replace the SR list of each domain. As shown in Figure 3, the B-SID(X->Y) is used to replace the X->Y SID list. Ingress node A pushes e-Path(A->Z), B-SID(Y->Z), B-SID(X->Y), s-Path(A->X) and A->X SID list in turn. When the packet is received at node X, the s-Path(A->X) and X->Y SID list are popped, and the new s-Path(X->Y) is pushed. Also, X->Y SID list replaces B-SID(X->Y) to indicate that packet to be forwarded from node X to node Y. The data packet reaches the SR node Z according to the same forwarding procedure. In SR node Z, the e-Path (A->Z) is used to correlate the two unidirectional end-to-end paths.

The e-Path can be a globally unique or local label. If the e-Path is globally unique, it MUST be assigned from the SRGB block of each domain. If the e-Path is a local label, it is required that the controller(e.g., PCE) or a super controller (e.g., hierarchical PCE) assigns the label to ensure the ingress(A) and the egress node(Z) can recognize it and there is no SID collision in the ingress and egress domains.

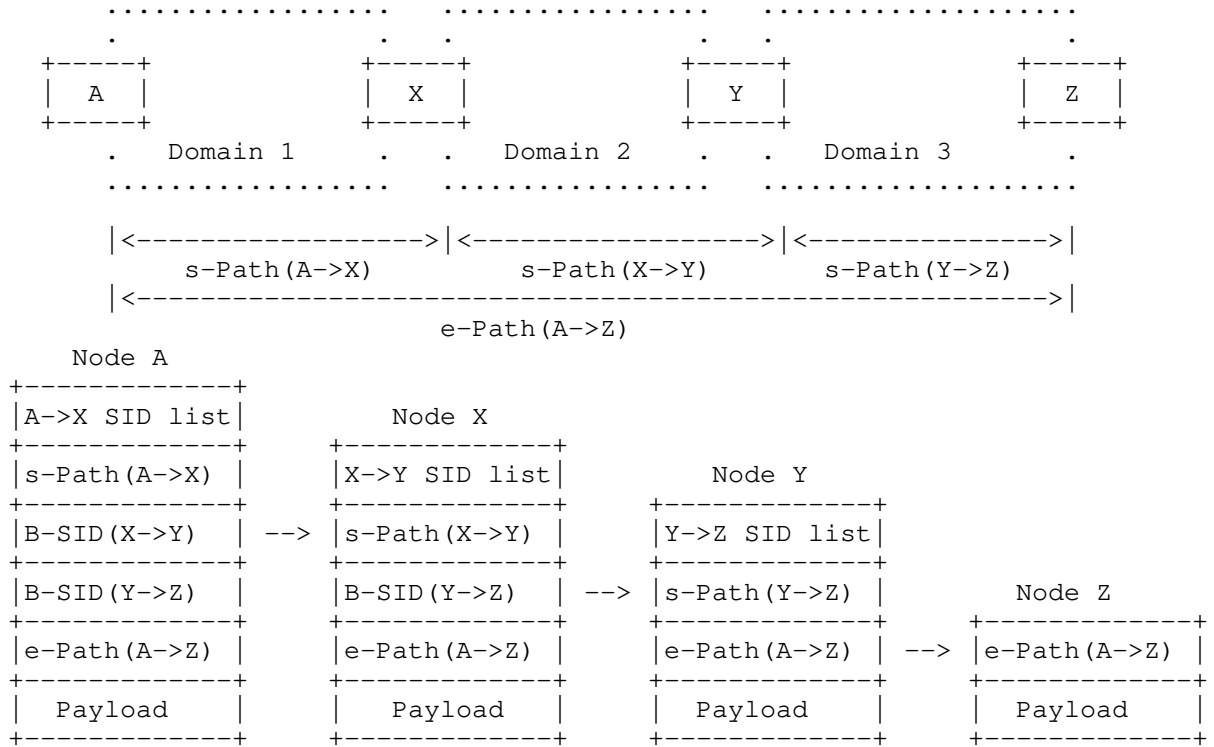


Figure 3: Nesting Inter-Domain Scenario

5. SR-MPLS-TP Inter-working with MPLS-TP

It is a common requirement that SR-MPLS-TP needs to inter-work with MPLS-TP when SR is incrementally being deployed in the MPLS-TP domain.

Figure 4 shows the stitching model of SR-MPLS-TP inter-working with MPLS-TP. The left is the SR-MPLS-TP network, and the right is the MPLS-TP network. The path segment which is i-Path is used for the bidirectional tunnel correlation in SR-MPLS-TP network. The edge nodes of the SR-MPLS-TP network should map the path segment to the corresponding MPLS-TP label for bidirectional tunnel indication in the MPLS-TP network.

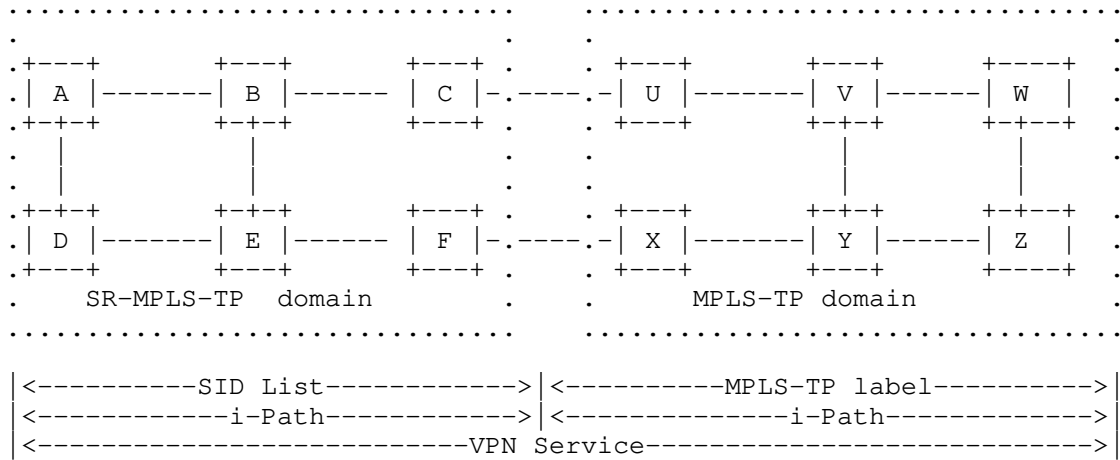


Figure 4: Stitching SR-MPLS-TP inter-working with MPLS-TP

Figure 5 displays the nesting model of SR-MPLS-TP and MPLS-TP inter-working. Compared with stitching SR-MPLS-TP inter-working with MPLS-TP, the path segment is e-Path and presents end-to-end encapsulation in the packet from SR-MPLS-TP domain to MPLS-TP domain.

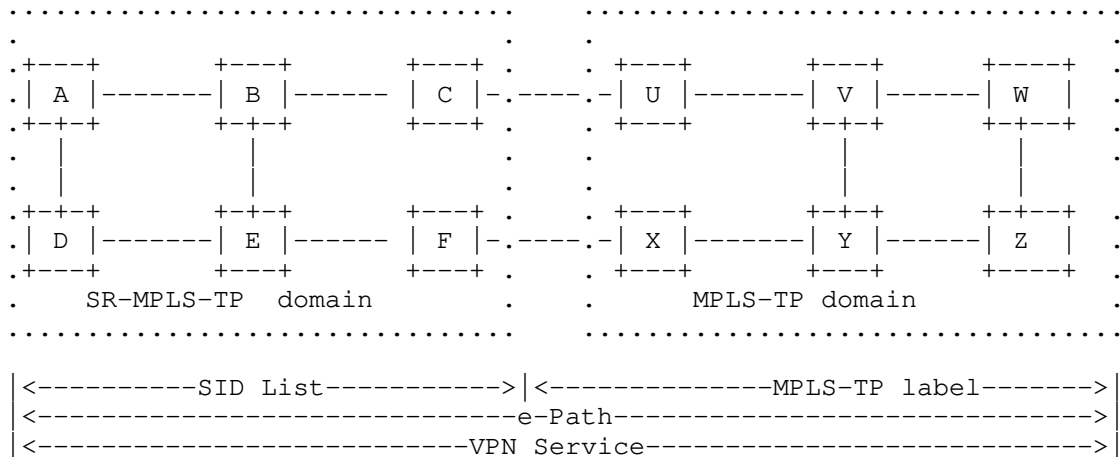


Figure 5: Nesting SR-MPLS-TP inter-working with MPLS-TP

The requirements for the SR-MPLS-TP inter-working with MPLS-TP are as follows:

- o It is required to establish the end-to-end VPN service through the SR-MPLS-TP domain and the MPLS-TP domain;
- o The path segment MUST be carried through SR-MPLS-TP and MPLS-TP domains in the nesting SR-MPLS-TP inter-working with MPLS-TP model.
- o The edge nodes of the MPLS-TP network SHOULD process the path segment from the SR-MPLS-TP network.
- o The edge nodes in the MPLS-TP SHOULD process MPLS SID sent from the MPLS-SR-TP domain
- o The edge nodes in the SR-MPLS-TP network SHOULD process the MPLS-TP labels sent from the MPLS-TP domain;

6. Security Considerations

TBA

7. Acknowledgements

TBA

8. IANA Considerations

TBA

9. Normative References

[I-D.cheng-spring-mpls-path-segment]

Cheng, W., Wang, L., Li, H., Chen, M., Gandhi, R., Zigler, R., and S. Zhan, "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment-03 (work in progress), October 2018.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.

[I-D.ietf-pce-association-group]

Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-07 (work in progress), December 2018.

- [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-18 (work in progress), December 2018.
- [I-D.xiong-pce-stateful-pce-sr-inter-domain]
Xiong, Q., hu, f., Mirsky, G., and W. Cheng, "Stateful PCE for SR-MPLS-TP Inter-domain", draft-xiong-pce-stateful-pce-sr-inter-domain-00 (work in progress), December 2018.
- [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>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [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>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [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>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual
China

Email: hufwei@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 5, 2019

Z. Hu
H. Chen
J. Yao
Huawei Technologies
C. Bowers
Juniper Networks
March 4, 2019

Segment Routing Proxy Forwarding
draft-hu-spring-segment-routing-proxy-forwarding-01

Abstract

Segment Routing Traffic Engineering (SR-TE) supports the creation of explicit paths using segment lists containing adjacency-sids, node-sids, anycast-sids, and binding-sids. When the segment list defining an SR-TE path contains a node-sid, and the node fails, the network may no longer be able to properly forward traffic on that SR-TE path. [I-D.bashandy-rtgwg-segment-routing-ti-lfa] and [I-D.hegde-spring-node-protection-for-sr-te-paths] describe a mechanism that allows local repair actions on the direct neighbors of the failed node to temporarily route traffic to the node immediately following the failed node on the SR-TE path segment list. However, once the IGP shortest paths have converged, the local repair mechanism is no longer sufficient to continue forwarding traffic using the original segment list of the SR-TE path, since the non-neighbors of the failed node will no longer have a route to reach the failed node. This document describes a mechanism that allows traffic to continue to be forwarded on an SR-TE path for an extended period of time after the failure of a node used in the path's segment list.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 5, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Extensions to IGP for Proxy Forwarding	4
2.1. Extensions to OSPF	4
2.1.1. Advertising Proxy Forwarding	4
2.1.2. Advertising Binding Segment	7
2.2. Extensions to IS-IS	9
2.2.1. Advertising Proxy Forwarding	9
2.2.2. Advertising Binding Segment	11
3. Building Proxy Forwarding Table	13
3.1. Advertising Proxy Forwarding	15
3.2. Building Proxy Forwarding Table	15
4. Node Protection for Segment List	15
4.1. Next Segment is an Adjacency Segment	16
4.2. Next Segment is a Node Segment	16
4.3. Next Segment is a Binding Segment	17
5. Security Considerations	18
6. IANA Considerations	18
7. Acknowledgements	18
8. References	18
8.1. Normative References	18
8.2. Informative References	18
Authors' Addresses	19

1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using Segment Routing. SR-TE supports the creation of explicit paths using adjacency-sids, node-sids, anycast-sids, and binding-sids. A node-sid in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When a particular node fails, it would be useful to be able to continue to send traffic on an SR-TE path that uses the node-sid of the failed node for an extended period of time, without having to immediately modify the segment list used at the ingress to the SR-TE path.

The first step to achieve this objective is to make the rest of the routers in the network continue to forward traffic using the node-sid of the failed node. If we don't do anything special, once the IGP converges to take into account the failed node, a given router will no longer maintain a route corresponding to the node-sid. Any traffic that arrives at the router with the node-sid of the failed node as the active segment will be dropped. This document addresses this problem by having each neighbor of the failed node advertise its SR proxy forwarding capability. This indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from neighbors of a failed node will send traffic using the node-sid of the failed node to the nearest Proxy Forwarder.

Once the affected traffic reaches a Proxy Forwarder, the Proxy Forwarder sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list. [I-D.bashandy-rtgwg-segment-routing-ti-lfa] and [I-D.hegde-spring-node-protection-for-sr-te-paths] describe how the immediate neighbors of a failed node can accomplish this by forwarding based on the first two segments in the segment list. The forwarding described in these drafts was originally intended to be used for only a short period of time, to provide fast-reroute protection until the IGP converges. The current document proposes to extend this behavior on the Proxy Forwarder until well after the IGP has converged.

If the faulty node is a label adhesion node, the Binding-sids cannot be exchanged to the label stack for its identity, and the traffic will be lost before it reaches the faulty node.

In this document, the proxy mechanism is provided in the neighbor node of the faulty node of the forwarding path to implement traffic forwarding after the node with the label adhesion fails on the SR-TE loose path.

2. Extensions to IGP for Proxy Forwarding

When a node has segment routing proxy forwarding capability, it advertises this capability. The capability indicates that the node has the ability to proxy forward the global sid of each of its neighbors. When an neighbor who advertises its global sid fails, the traffic can be forwarded to the proxy node.

2.1. Extensions to OSPF

2.1.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for all its neighboring nodes for protecting the failures of these nodes, node P advertises its SR proxy forwarding capability in its router information opaque LSA, which contains a Router Information Capabilities TLV of the format as shown in Figure 1.

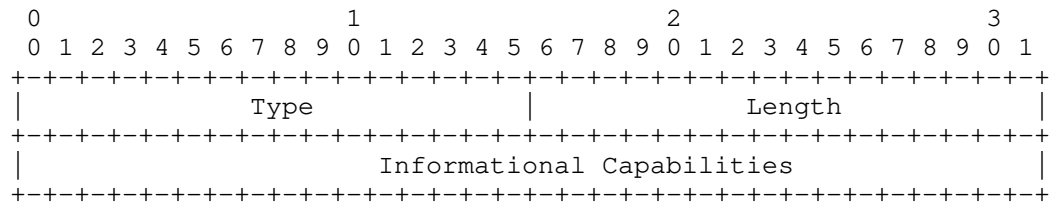


Figure 1: Router Information Capabilities TLV

One bit (called PF bit) in the Informational Capabilities field of the TLV is used to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/node SID of node N. Thus node X will forward the traffic targeting to node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forwarding the traffic to its destination without going through node N. After node N fails, node X will keep the proxy prefix/node SID of node N for a given period of time.

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

A new TLV, called Proxy Node SIDs TLV, is defined for node P to advertise the node SIDs of some of its neighboring nodes. It has the format as shown in Figure 2.

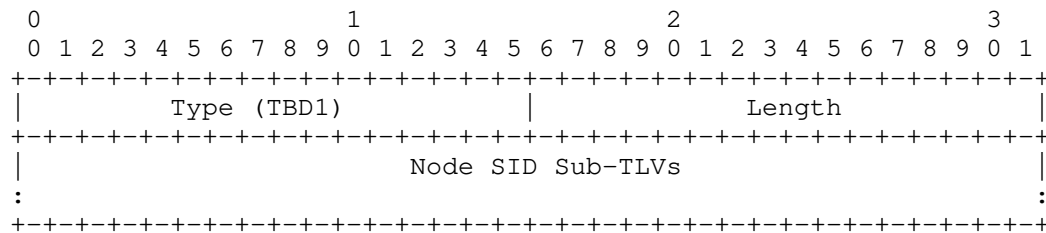


Figure 2: OSPF Proxy Node SIDs TLV

The Type (TBD1) is to be assigned by IANA. The TLV contains a number of Node SID Sub-TLVs. The Length is the total size of the Node SID Sub-TLVs included in the TLV. A Node SID Sub-TLV is the Prefix SID Sub-TLV defined in [I-D.ietf-ospf-segment-routing-extensions].

A proxy forwarding node P originates an Extended Prefix Opaque LSA containing this new TLV. The TLV includes the Node SID Sub-TLVs for the node SIDs of some of P's neighboring nodes. For each of some of P's neighboring nodes, the Node SID Sub-TLV for its prefix/node SID is included in the TLV. This prefix/node SID is called a proxy prefix/node SID.

A proxy forwarding node will originate an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The format of the LSA is shown in Figure 3.

For a proxy forwarding node P, having a number of neighboring nodes, P originates and maintains an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The TLV contains the Prefix/Node SID Sub-TLV for each of some of the neighboring nodes after node P creates the corresponding proxy forwarding entries for protecting the failure of some of the neighboring nodes.

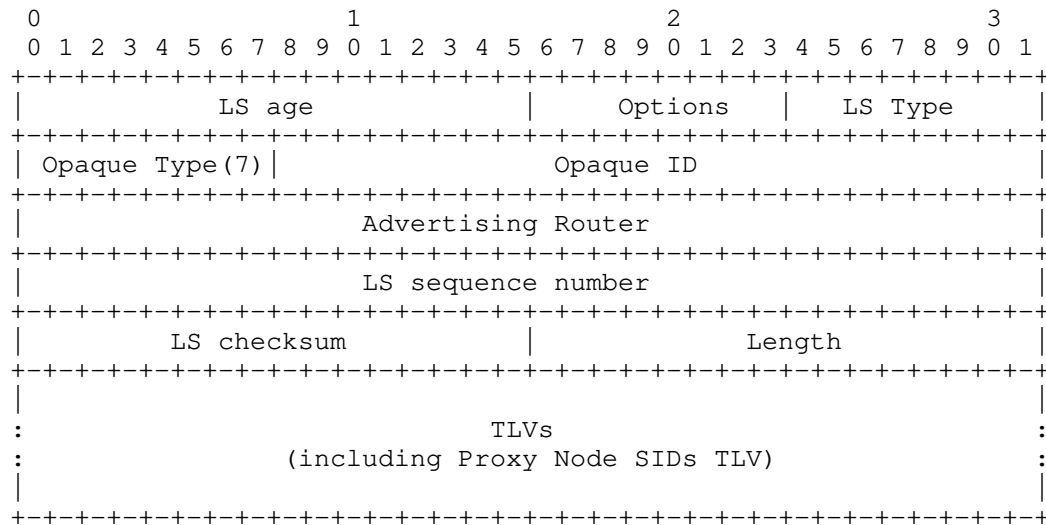


Figure 3: OSPFv2 Extended Prefix Opaque LSA

When an neighboring node fails, P maintains the LSA with the TLV containing the Prefix/Node SID Sub-TLV for the neighboring node for a given period of time. After the given period of time, the Prefix/Node SID Sub-TLV for the neighboring node is removed from the TLV in the LSA and then after a given time the corresponding proxy forwarding entries for protecting the failure of the neighboring node is removed.

For a node X in the network, it learns the prefix/node SID of node N and the proxy prefix/node SID of node N. The former is originated and advertised by node N, and the latter is originated and advertised by the proxy forwarding node P of node N. Note that the proxy Prefix/Node SID Sub-TLV for node N does not contain a prefix of node N, and the prefix is the prefix associated with the prefix/node SID of node N originated by node N.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/node SID of node N. Thus node X will forward the traffic targeting to node N to node P when node N fails, and node P will do a proxy forwarding for node N and forwarding the traffic to its destination without going through node N.

2.1.2. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of segments, node A advertises an LSA containing the binding (i.e., the binding SID and the list of the segments). The LSA is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LSA is a opaque LSA of LS type 9 (i.e., a link local scope LSA).

A binding segment is represented by binding segment TLV of the format as shown in Figure 4.

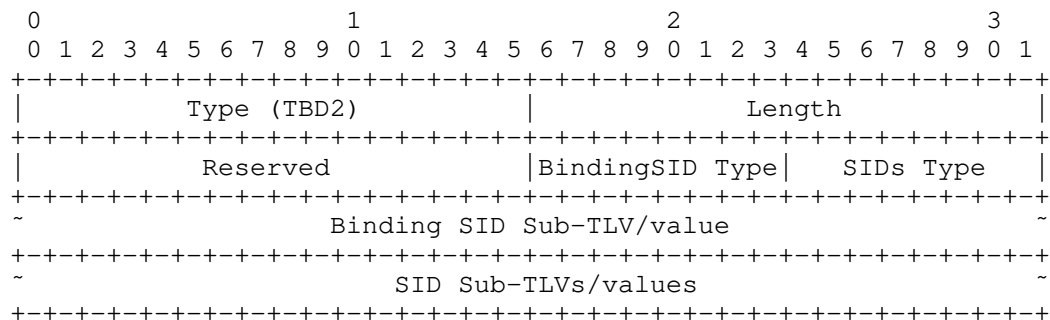


Figure 4: OSPF Binding Segment TLV

It comprises a binding SID and a list of segments (SIDs). The fields of this TLV are defined as follows:

Type: 2 octets, its value (TBD2) is to be assigned by IANA.

Length: 2 octets, its value is (4 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

- o BT = 0: The binding SID is represented by a Sub-TLV (i.e., Binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [I-D.ietf-ospf-segment-routing-extensions]. BT != 0 indicates that the binding SID is represented by a value.

- o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

- o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

SIDs Type (ST): 1 octet indicates whether the list of segments (SIDs) are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

- o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [I-D.ietf-ospf-segment-routing-extensions]. ST != 0 indicates that the SIDs are represented by values.
- o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.
- o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

The opaque LSA of LS Type 9 containing the binding segment (i.e., the binding SID and the list of the segments) has the format as shown in Figure 5. It may have Opaque Type of x (the exact type is to be assigned by IANA) for Binding Segment Opaque LSA.

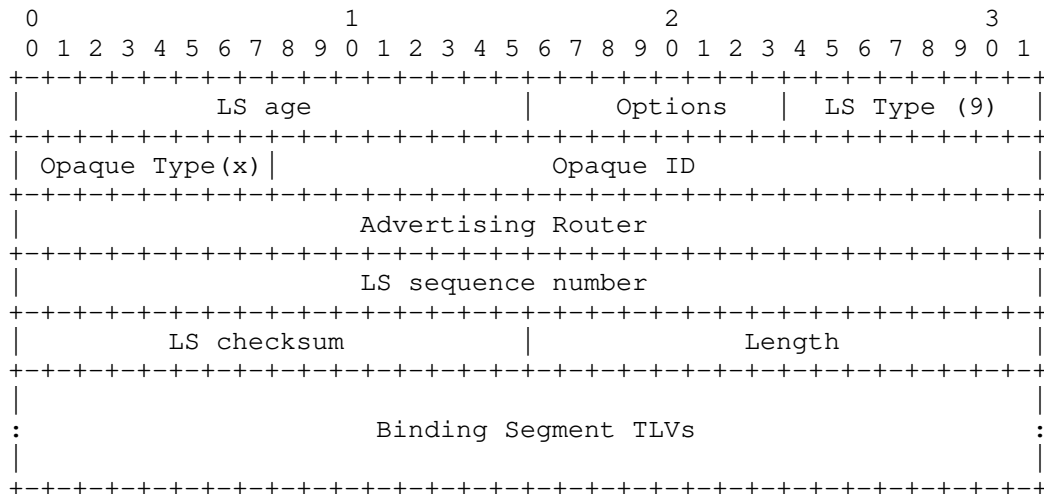


Figure 5: OSPFv2 Binding Segment Opaque LSA

For every binding on a node A, the LSA originated by A contains a binding segment TLV for it.

For node A running OSPFv3, it originates a link-local scoping LSA of a new LSA function code (TBD3) containing binding segment TLVs for

the bindings on it. The format of the LSA is illustrated in Figure 6.

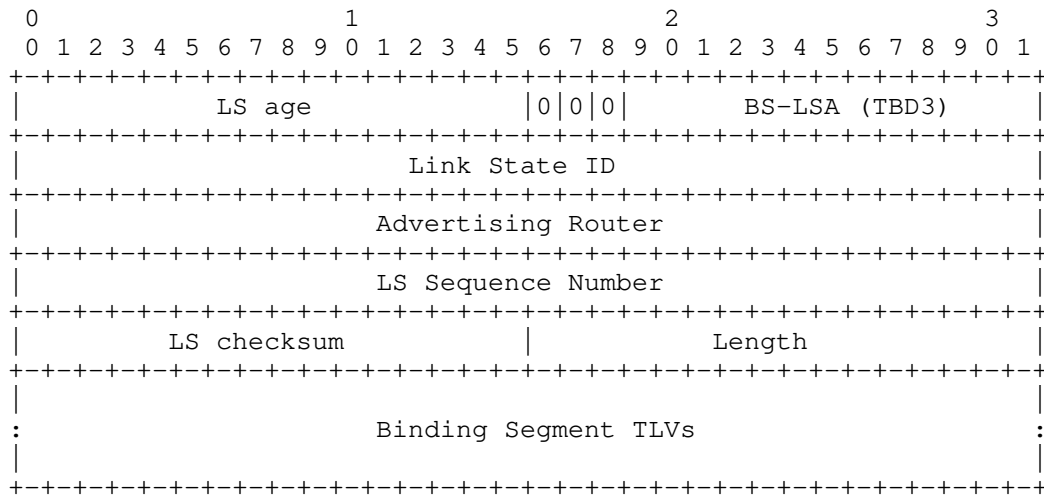


Figure 6: OSPFv3 Binding Segment Opaque LSA

The U-bit is set to 0, and the scope is set to 00 for link-local scoping.

2.2. Extensions to IS-IS

2.2.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for its neighboring nodes for protecting the failures of them, node P advertises its SR proxy forwarding capability in its LSP, which contains a Router Capability TLV of Type 242 including a SR capabilities sub-TLV of sub-Type 2.

One bit (called PF bit as shown in Figure 7) in the Flags field of the SR capabilities sub-TLV is defined to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

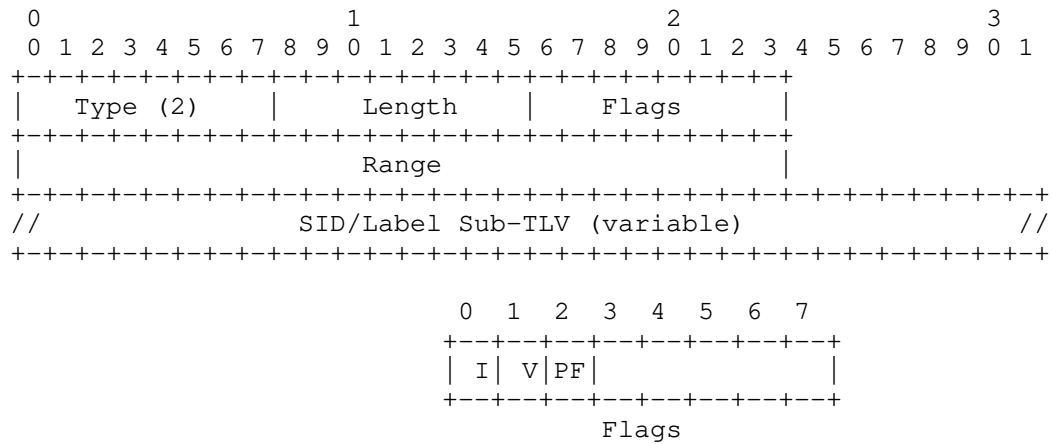


Figure 7: SR Capabilities sub-TLV

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

The IS-IS SID/Label Binding TLV (suggested value 149) is defined in [I-D.ietf-isis-segment-routing-extensions]. A Proxy Forwarder uses the SID/Label Binding TLV to advertise the node Sid of its neighboring node. The Flags field of the SID/Label Binding TLV is extended to include a P flag as shown in Figure 8. The prefix/node SID in prefix/node Sid Sub-TLV included in SID/Label Binding TLV is identified as a proxy forwarding prefix/node SID.

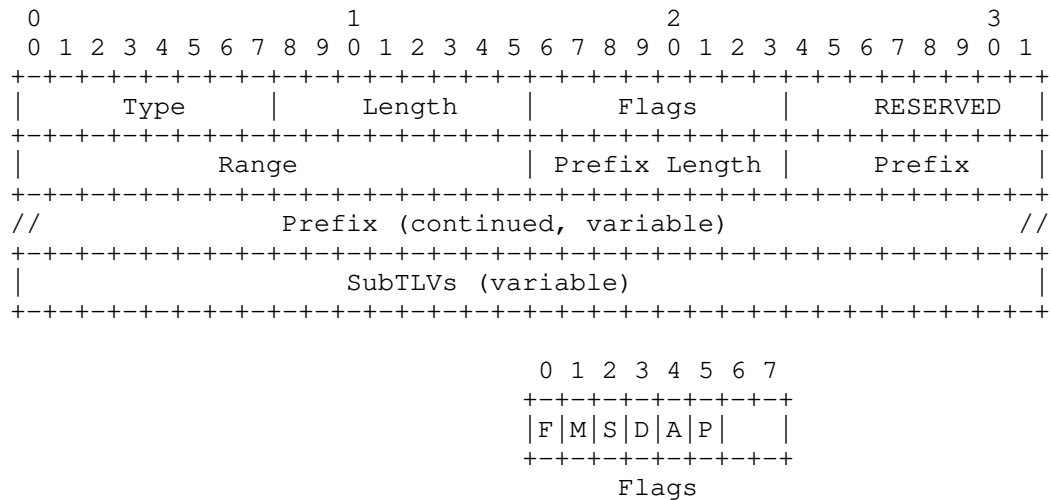


Figure 8: SID/Label Binding TLV

Where:

P-Flag: Proxy forwarding flag. If set, this prefix/node Sid is advertised by the proxy node. This TLV is used to announce that the node has the ability to proxy forward the prefix/node Sid.

When the P-flag is set in the SID/Label Binding TLV, the following usage rules apply.

The Range, Prefix Length and Prefix field are not used. They should be set to zero on transmission and ignored on receipt.

SID/Label Binding TLV contains a number of prefix/node SID Sub-TLVs. The TLV advertised by a proxy forwarding node P contains prefix/node SID Sub-TLVs for the node SIDs of P's neighbor nodes. Each of the Sub-TLVs is a prefix/node SID Sub-TLV defined in [I-D.ietf-isis-segment-routing-extensions]. From the SID in a prefix/node SID Sub-TLV advertised by the Proxy Forwarding node, its prefix can be obtained through matching corresponding prefix/node SID advertised by the neighbor/protected node using TLV-135 (or 235, 236, or 237) together with the prefix/node SID Sub-TLV.

2.2.2. Advertising Binding Segment

[I-D.ietf-spring-segment-routing-policy] has defined the usage of binding-sid. For supporting binding sid proxy forwarding, a new IS-IS TLV, called Binding Segment TLV, is defined. It contains a binding SID and a list of segments (SIDs). This TLV may be

advertised in IS-IS Hello (IIH) PDUs, LSPs, or in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. Its format is shown in Figure 9.

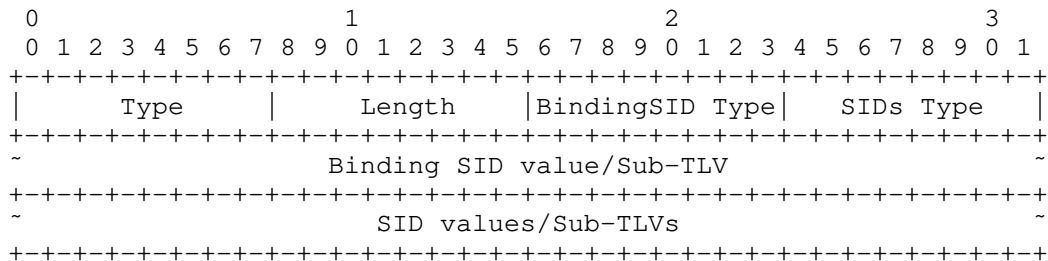


Figure 9: IS-IS Binding Segment TLV

The fields of this TLV are defined as follows:

Type: 1 octet Suggested value 152 (to be assigned by IANA)

Length: 1 octet (2 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

- o BT = 0: The binding SID is represented by a Sub-TLV (i.e., binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [I-D.ietf-isis-segment-routing-extensions]. BT != 0 indicates that the binding SID is represented by a value.

- o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

- o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

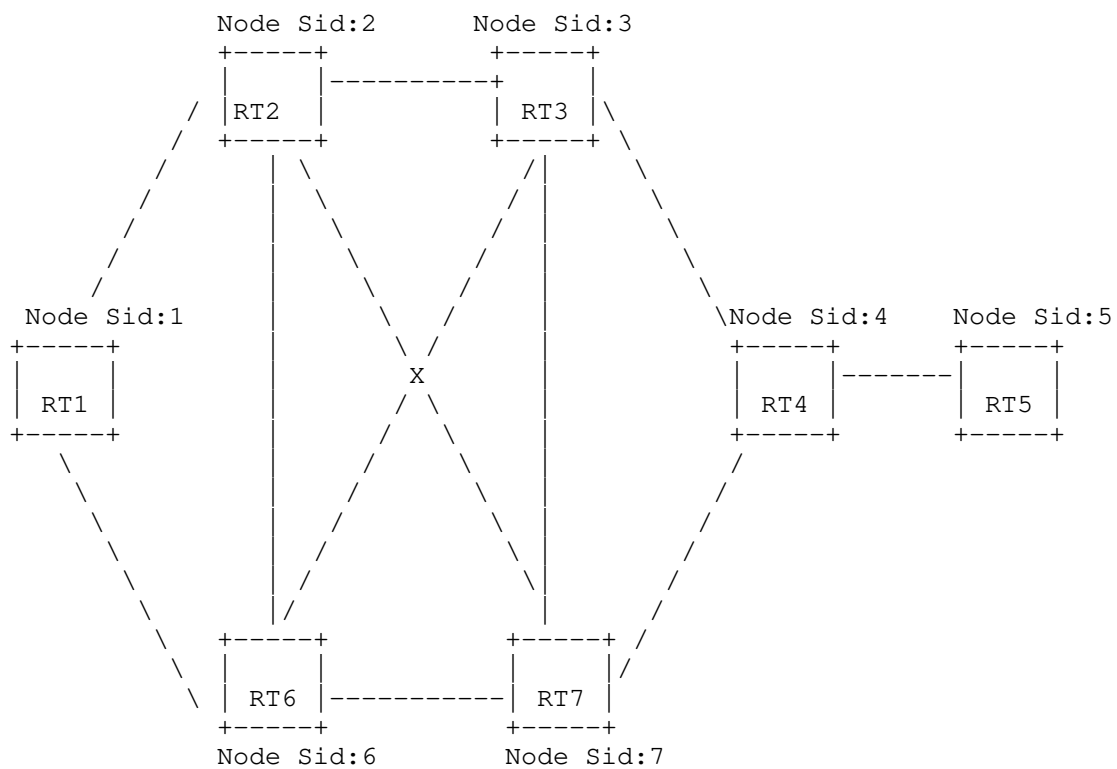
SIDs Type (ST): 1 octet indicates whether the SIDs are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

- o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [I-D.ietf-isis-segment-routing-extensions]. ST != 0 indicates that the SIDs are represented by values.

- o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.
- o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

3. Building Proxy Forwarding Table

Figure 10 is used to illustrate the SR proxy forwarding approach. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Three label stacks are shown in the figure. Label Stack 1 uses only adjacency-SIDs and represents the path RT1->RT2->RT3->RT4->RT5. Label Stack 2 uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5. Label Stack 3 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 3, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.



Node SRGB	Adj-Sid			
RT1:[1000-1999]	RT1->RT2:10012	Label Stack 1	Label Stack 2	Label Stack 3
RT2:[2000-2999]	RT2->RT3:20023	10012	1003	1003
RT3:[3000-3999]	RT3->RT6:30036	20023	3004	100
RT4:[4000-4999]	RT3->RT7:30037	30034	4005	100 is binding SID to {30034,40045}
RT5:[5000-5999]	RT3->RT4:30034	40045		
RT6:[6000-6999]	RT7->RT4:70074			
RT7:[7000-7999]	RT4->RT5:40045			

Figure 10: Topology of SR-TE Path

3.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do a SR proxy forwarding for all its neighboring nodes, it must advertise this capability. If the PLR can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, for example, RT3, then it uses proxy Node SIDs TLV to advertise the prefix-sid learned from RT3. The TLV contains the Sub-TLV/value for the prefix/node sid of RT3 as a proxy SID. When RT3 fails, RT2 needs to maintain the Sub-TLV/value for a period of time. When the proxy forwarding table corresponding to the fault node is deleted (see section 3.2), the Sub-TLV/value is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node Sid TLV advertised by RT3 and the proxy Node SIDs TLV advertised by RT2. When RT3 is normal, the nodes prefer prefix/node Sid TLV. When the RT3 fails, the proxy prefix/node SIDs TLV advertised by RT2 is preferred.

3.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

- 1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;
- 2: All adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.
- 3: The binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adj-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in the topology of Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. The structure of the table and how to build the table is a local implementation issue.

4. Node Protection for Segment List

Segment Routing Traffic Engineering supports the creation of explicit paths using adjacency-sids, node-sids, and binding-sids. The label stack is a combination of one or more of adjacency-sids, node-sids,

and binding-sids. This Section shows how a proxy node uses the SR proxy forwarding mechanism to protect traffic to the destination node when the next segment of label stack is adjacency-sids, node-sids, or binding-sids, respectively.

4.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 1 {10012, 20023, 30034, 40045} represents SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adj-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 pops top adj-SID 10012, and forwards the packet to RT2;
- b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The Proxy Forwarding Table corresponding to RT3 returns an outgoing interface and label stack representing a path to RT4 that does not pass through RT3. In this case, outgoing interface to RT7 with label stack 7004, satisfies this requirement.
- c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT4 forwards it to RT4, where the original path is rejoined.
- d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

4.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} represents SR-TE loose path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.

- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.
- c. RT2 uses 2003 as the in-label to lookup Proxy Forwarding table, and the query result is forwarding the packet to RT4.
- d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.
- e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.
- f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

4.3. Next Segment is a Binding Segment

As shown in Figure 1, Label Stack 3 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-Sid, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.
- c. RT2 uses Binding-sid:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.
- d. RT2 swaps Binding-sid:100 to Segment list {30034, 40045}, and uses the 3034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.
- e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.
- f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

5. Security Considerations

TBD

6. IANA Considerations

TBD

7. Acknowledgements

The authors would like to thank Peter Psenak and Les Ginsberg for their comments to this work.

8. References

8.1. Normative References

- [I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-22 (work in progress), December 2018.
- [I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-27 (work in progress), December 2018.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.

8.2. Informative References

- [I-D.bashandy-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-05 (work in progress), October 2018.

- [I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu,
"Node Protection for SR-TE Paths", draft-hegde-spring-
node-protection-for-sr-te-paths-04 (work in progress),
October 2018.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d.,
bogdanov@google.com, b., and P. Mattes, "Segment Routing
Policy Architecture", draft-ietf-spring-segment-routing-
policy-02 (work in progress), October 2018.
- [I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J.,
Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID
in PCE-based Networks.", draft-sivabalan-pce-binding-
label-sid-06 (work in progress), February 2019.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching
(MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic
Class" Field", RFC 5462, DOI 10.17487/RFC5462, February
2009, <<https://www.rfc-editor.org/info/rfc5462>>.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: huzhibo@huawei.com

Huaimo Chen
Huawei Technologies
Boston, MA
USA

Email: Huaimo.chen@huawei.com

Junda Yao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: yaojunda@huawei.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: cbowers@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 October 2022

Z. Hu
Huawei Technologies
H. Chen
Futurewei
J. Yao
Huawei Technologies
C. Bowers
Juniper Networks
Y. Zhu
China Telecom
Y. Liu
China Mobile
11 April 2022

SR-TE Path Midpoint Restoration
draft-hu-spring-segment-routing-proxy-forwarding-19

Abstract

Segment Routing Traffic Engineering (SR-TE) supports explicit paths using segment lists containing adjacency-SIDs, node-SIDs and binding-SIDs. The current SR FRR such as TI-LFA provides fast re-route protection for the failure of a node along a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node. This document describes a mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment of the path. With the restoration of the routes to the failure, the traffic is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

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 13 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. Proxy Forwarding	4
3. Protocol Extensions/Re-uses for Proxy Forwarding	4
3.1. Advertising Binding Segment	4
3.2. Advertising Proxy Forwarding	5
4. Proxy Forwarding Example	6
4.1. Advertising Proxy Forwarding	8
4.2. Building Proxy Forwarding Table	8
4.3. Proxy Forwarding for Binding Segment	9
5. Security Considerations	10
6. Acknowledgements	10
7. References	10
7.1. Normative References	10
7.2. Informative References	11
Appendix A. Proxy Forwarding for Adjacency and Node Segment	11
A.1. Next Segment is an Adjacency Segment	11
A.2. Next Segment is a Node Segment	12
Authors' Addresses	13

1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using a segment list. SR-TE supports the creation of explicit paths using adjacency-SIDs, node-SIDs, anycast-SIDs, and binding-SIDs. A node-SID in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When the node fails, the network may no longer be able to properly forward traffic on that SR-TE path.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes an SR FRR mechanism that provides fast re-route protection for the failure of a node on a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node and drop the traffic.

To solve this problem,

[I-D.ietf-spring-segment-protection-sr-te-paths] proposes that a hold timer should be configured on every router in a network. After the IGP converges on the event of a node failure, if the node-SID of the failed node becomes unreachable, the forwarding changes should not be communicated to the forwarding planes on all configured routers (including PLRs for the failed node) until the hold timer expires. This solution may not work for some cases such as some of nodes in the network not supporting this solution.

This document describes a proxy forwarding mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment on a failed node along the path. With the restoration of the routes to the failure, the traffic for the SR-MPLS TE path is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

1.1. Terminology

SR: Segment Routing.

PLR: Point of Local Repair.

LSP: Link State Protocol Data Unit (PDU) in IS-IS.

LSA: Link State Advertisement in OSPF.

LS: Link State, which is LSP or LSA.

2. Proxy Forwarding

In the proxy forwarding mechanism, each neighbor of a possible failed node advertises its SR proxy forwarding capability in its network domain when it has the capability. This capability indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from the neighbors of a failed node will send traffic using the node-SID of the failed node to the nearest Proxy Forwarder after the IGP converges on the event of the failure.

Once the affected traffic reaches a Proxy Forwarder, it sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list.

For a binding segment of a possible failed node, the node advertises the information about the binding segment, including the binding SID and the list of SIDs/segments associated with the binding SID, to its direct neighbors only. Note that the information is not advertised in the network domain.

After the node fails and the IGP converges on the failure, the traffic with the binding SID of the failed node will reach its neighbor having SR Proxy Forwarding capability. Once receiving the traffic, the neighbor swaps the binding SID with the list of SIDs/segments associated with the binding SID and sends the traffic along the post-failure shortest path to the first node in the segment list.

3. Protocol Extensions/Re-uses for Proxy Forwarding

This section describes the semantic of protocol extensions/re-uses for advertising the information about each binding segment (including its binding SID and the list of SIDs/segments associated with the binding SID) of a node to its direct neighbors and the SR proxy forwarding capability of a node in a network domain.

3.1. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of SIDs/segments, node A advertises an LS containing the binding (i.e., the binding SID and the list of the SIDs/segments) in a binding segment TLV. The LS is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LS is a opaque LSA of LS type 9 (i.e., a link local scope LSA). For IS-IS, the TLV is advertised in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356].

Alternatively, when a protocol (such as PCE or BGP running on a controller) supports sending a binding on a node A to A, this protocol may be extended to send the binding with node A to A's neighbors if the controller knows the neighbors and there are protocol (PCE or BGP) sessions between the controller and the neighbors.

Note: how to send bindings of node A to A's neighbors via which protocol is out of the scope of this document.

3.2. Advertising Proxy Forwarding

When a node P is able to do SR proxy forwarding for its neighboring nodes for protecting the failures of these nodes, P advertises its SR proxy forwarding capability for these nodes. The mirror SID [RFC8402] for a node N (Neighbor of P) advertised by P using IS-IS extensions [RFC8667] indicates the capability of P for N.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P. The route to the proxy prefix/node SID is through proxy forwarding capable nodes.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/node SID of node N. Thus node X will forward the traffic targeting to the prefix/node SID of node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forward the traffic towards its final destination without going through node N.

Note that the behaviors of normal IP forwarding and routing convergences in a network are not changed at all by the SR proxy forwarding. For example, the next hop used by BGP is an IP address (or prefix). The IGP and BGP converge in normal ways for changes in the network. The packet with its IP destination to this next hop is forwarded according to the IP forwarding table (FIB) derived from IGP and BGP routes.

Similar to IS-IS [RFC8667], OSPF should be extended for advertising mirror SID to indicate the capability. Note that OSPF extensions is out of the scope of this document.

4. Proxy Forwarding Example

This section illustrates the proxy forwarding for a binding SID through an example. The proxy forwarding for a node SID and an adjacency SID can refer to [I-D.ietf-spring-segment-protection-sr-te-paths] or Appendix. Figure 1 is an example network topology used to illustrate the proxy forwarding mechanism for a binding SID. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Label Stack 1 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 1, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.

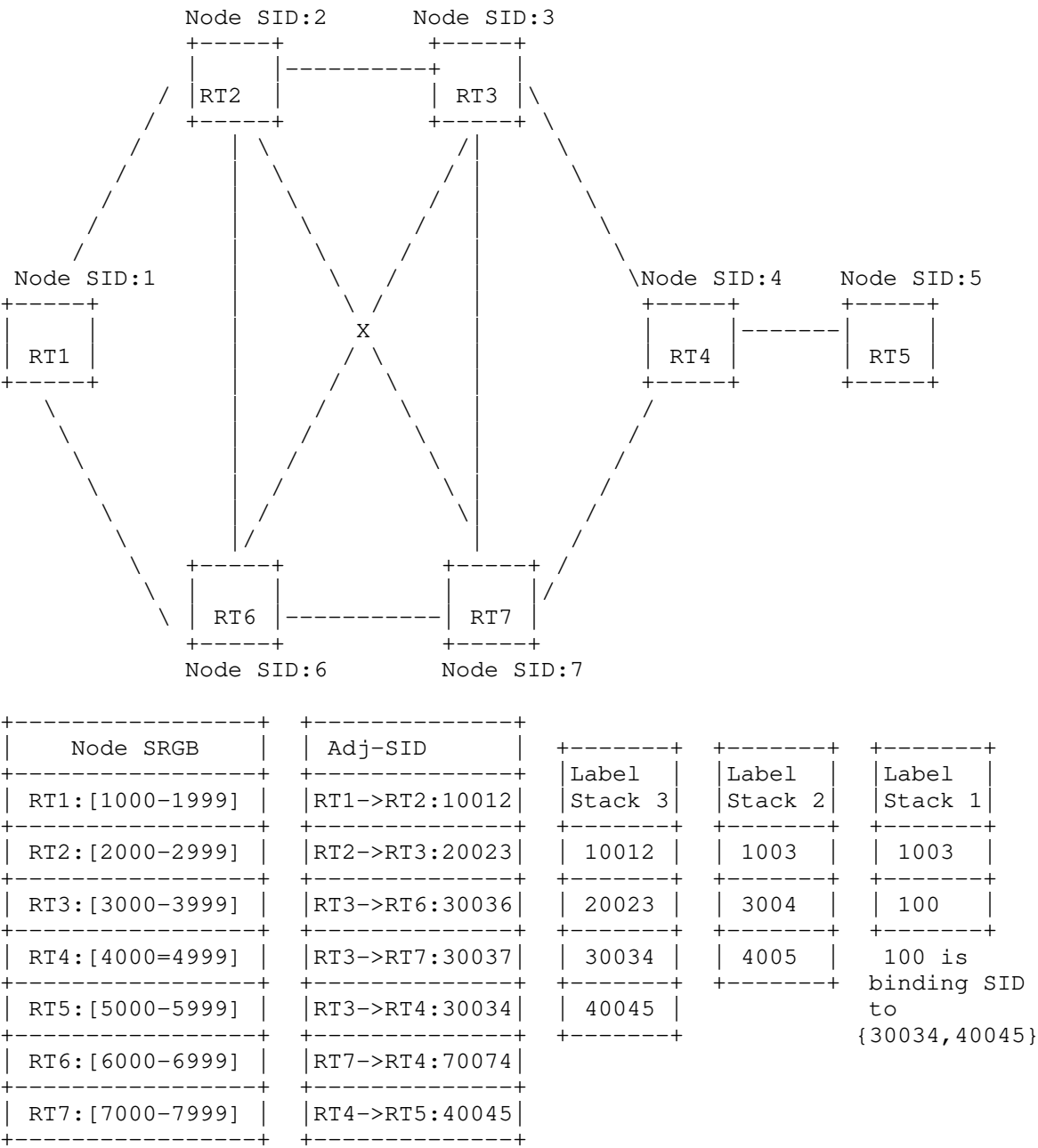


Figure 1: Topology of SR-TE Path

4.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do SR proxy forwarding for its neighboring nodes such as RT3, it must advertise this capability. When RT3 fails, RT2 needs to maintain its SR proxy forwarding capability for a period of time. When the proxy forwarding table corresponding to the fault node is deleted, the capability is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node SID advertised by RT3 and the proxy forwarding capability for RT3 advertised by RT2. When RT3 is normal, the nodes prefer prefix/node SID. When the RT3 fails, the proxy prefix/node SIDs of RT3 for RT2 is preferred.

For binding-SID 100, which is associated with segment list {30034, 40045}, RT3 advertises the binding (i.e., 100 bond to {30034, 40045}) to its neighbors RT2, RT4 and RT7. RT2 as PLR uses the binding to build an entry for proxy forwarding for binding-SID 100 in its Proxy Forwarding Table for RT3. The entry is used when RT3 fails.

4.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

- 1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;
- 2: Every adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.
- 3: Every binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adjacency-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. RT2 calculates the proxy forwarding table for RT3, as shown in Figure 2.

In-label	SRGBDiffValue	Next Label	Action	Map Label
2003	-1000	30034	Fwd to RT4	2004
		30036	Fwd to RT6	2006
		30037	Fwd to RT7	2007
		100	Swap to { 30034, 40045 }	

Figure 2: RT2's Proxy Forwarding Table for RT3

4.3. Proxy Forwarding for Binding Segment

This Section shows through example how a proxy node uses the SR proxy forwarding mechanism to forward traffic to the destination node when a node fails and the next segment of label stack is a binding-SID.

As shown in Figure 1, Label Stack 1 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-SID, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.
- c. RT2 uses Binding-SID:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.
- d. RT2 swaps Binding-SID:100 to Segment list {30034, 40045}, and uses the 30034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.
- e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.

f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

5. Security Considerations

The extensions to OSPF and IS-IS described in this document result in two types of behaviors in data plane when a node in a network fails. One is that for a node, which is a upstream (except for the direct upstream) node of the failed node along a SR-TE path, it continues to send the traffic to the failed node along the SR-TE path for an extended period of time. The other is that for a node, which is the direct upstream node of the failed node, it fast re-routes the traffic around the failed node to the direct downstream node of the failed node along the SR-TE path. These behaviors are internal to a network and should not cause extra security issues.

6. Acknowledgements

The authors would like to thank Peter Psenak, Acee Lindem, Les Ginsberg, Bruno Decraene and Jeff Tantsura for their comments to this work.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [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>>.
- [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>>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

7.2. Informative References

[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-08, 21 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-08.txt>>.

[I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", Work in Progress, Internet-Draft, draft-ietf-spring-segment-protection-sr-te-paths-03, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-protection-sr-te-paths-03.txt>>.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-22, 22 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-22.txt>>.

Appendix A. Proxy Forwarding for Adjacency and Node Segment

This Section shows through example how a proxy node forward traffic to the destination node when a node fails and the next segment of label stack is an adjacency-SID or node-SID.

A.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 3 {10012, 20023, 30034, 40045} uses only adjacency-SIDs and represents the SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adjacency-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 pops top adjacency-SID 10012, and forwards the packet to RT2;
- b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The query result is 2004. RT2 uses 2004 as the incoming label to query the label forwarding table. The next hop is RT7, and the incoming label is changed to 7004.
- c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT7 forwards it to RT4, where the original path is rejoined.
- d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

A.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure, RT2 pops label 2003.
- c. RT2 uses 3004 as the in-label to lookup Proxy Forwarding table, The value of Map Label calculated based on SRGBDiffValue is 2004. and the query result is forwarding the packet to RT4.
- d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.
- e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.
- f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China
Email: huzhibo@huawei.com

Huaimo Chen
Futurewei
Boston, MA,
United States of America
Email: Huaimo.chen@futurewei.com

Junda Yao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China
Email: yaojunda@huawei.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA, 94089
United States of America
Email: cbowers@juniper.net

Yongqing
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou
510000
China
Email: zhuyq8@chinatelecom.cn

Yisong
China Mobile
510000
China

Email: liuyisong@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 3, 2020

D. Voyer, Ed.
Bell Canada
C. Filsfils
R. Parekh
Cisco Systems, Inc.
H. Bidgoli
Nokia
Z. Zhang
Juniper Networks
July 2, 2019

SR Replication Policy for P2MP Service Delivery
draft-voyer-spring-sr-p2mp-policy-03

Abstract

This document describes the SR policy architecture for P2MP service delivery.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 15, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. SR Replication Policy	3
3. SR P2MP Policy	4
4. Using Controller to build a P2MP Segment	5
4.1. SR P2MP Policy Creation	6
4.1.1. API	6
4.1.2. Invoking API	6
4.2. P2MP Segment Computation	7
4.2.1. Topology Discovery	7
4.2.2. Capability and Attribute Discovery	7
4.3. Instantiating P2MP segment nodes	7
4.3.1. PCEP	8
4.3.2. BGP	8
4.3.3. NetConf	8
4.4. Protection	8
4.4.1. Local Protection	8
4.4.2. Path Protection	8
5. IANA Considerations	8
6. Security Considerations	9
7. Acknowledgements	9
8. Contributors	9
9. Normative References	10
Authors' Addresses	10

1. Introduction

This document defines variants of the SR Policy [I-D. ietf-spring-segment-routing-policy] to support Point-to-Multipoint service delivery.

We define a Point-to-Multipoint (P2MP) segment, which connects a Root node to a set of Leaf nodes in a Segment Routing Domain.

We also define a Replication Segment, which corresponds to the state of a P2MP segment on a particular node.

A P2MP segment consists of replication segments for the root, leaves, and optionally intermediate replication nodes. Note that a node may forward only one copy to a downstream node (be it a leaf or another intermediate node) or even just forward traffic off the p2mp segment (i.e. as a leaf), but we still call the forwarding behavior on the node a replication segment.

For a P2MP segment, a controller may be used to compute paths from a Root node to a set of Leaf nodes, optionally via a set of replication nodes. A packet is replicated at the root node and optionally on Replication nodes towards each Leaf node.

A Point-to-Multipoint service delivery could be via Ingress Replication (aka Spray in some SR context), i.e., the root unicasts individual copies of traffic to each leaf. The corresponding P2MP segment consists of replication segments only for the root and the leaves.

A Point-to-Multipoint service delivery could also be via Downstream Replication (aka TreeSID in some SR context), i.e., the root and some downstream replication nodes replicate the traffic along the way as it traverses closer to the leaves.

Notice that Spray is actually a special form of TreeSID. Also notice that, the explicit path from the root or a replication node to a leaf or a downstream replication node can optionally be partially or completely specified by the controller or determined locally.

2. SR Replication Policy

An SR Replication policy is a variant of an SR policy [I-D.ietf-spring-segment-routing-policy]. A replication policy corresponds to a replication segment, which defines the forwarding behavior on a particular node on a particular P2MP segment.

An SR Replication Policy can be either provisioned locally or programmed by a controller.

An SR Replication Policy is identified through the tuple <Node-ID, Root, Tree-ID>.

An SR Replication Policy is defined by following elements:

- o Node-ID: The node that the replication segment is for.
- o Root: The root of the P2MP segment that the replication segment is for.

- o Tree-ID: Tree that the replication segment is part of.
- o Replication-SID: Segment ID for this Replication Segment.
- o Candidate Paths: See below.

The Replication-SID is instantiated into the forwarding plane at the node. An incoming packet with the SID is forwarded according to the replication branches. The Replication-SID may be the same on all nodes of the tree, and referred to as Tree-SID.

A SR Replication Policy may comprise of multiple candidate paths. The active candidate path is selected based on the tie breaking rules amongst the valid candidate-paths.

Each candidate path includes a list of replication branches. In this document, each branch is abstracted to a <Downstream Node, Downstream Replication-SID> tuple. For the signaling from a controller to a tree node, the Downstream Node in the tuple could be represented by its Node-SID (i.e. it does not matter how traffic gets to the downstream node, whether it's directly connected or not), or in case of a directly connected Downstream Node it could be represented by one of this node's Adjacency-SIDs (for the interface connecting to the directly connected Downstream Node). Alternatively, the Downstream Node could also be expanded to a SID-list that partially/fully specify the explicit path to it. In all cases, the node converts the signaled SIDs to its local forwarding representation (e.g., a Node/Adjacency-SID of a directly connected Downstream Node is translated to a local interface).

Each replication branch may also include one or more backup branches for protection purpose. Details will be added in a future revision.

3. SR P2MP Policy

The SR P2MP policy is a variant of an SR policy [I-D.ietf-spring-segment-routing-policy]. It correspond to an SR P2MP Segment.

A SR P2MP Policy is defined by following elements:

- o Root node: This is the headend of the P2MP segment.
- o Leaf nodes: A set of nodes that terminate the P2MP segment.
- o Constraints/Objectives: Optional set of topological/resource constraints and optimization objectives to be satisfied by the P2MP segment.

A SR P2MP Policy is identified through the tuple <Root node, Tree-ID>.

An SR P2MP Policy has a BSID [I-D.ietf-spring-segment-routing-policy] instantiated into the forwarding plane. The BSID is applicable only at the Root node.

An SR P2MP policy can be either provisioned locally or programmed by a controller onto the root node of the segment, for the purpose of steering traffic into the segment. A controller calculates the tree and program corresponding replication segments on root, leaves and optional replication nodes.

Traffic is steered into a SR P2MP Policy in two ways:

- o Based on a local policy-based routing at the Root node.
- o Based on remote classification and steering via the BSID of the SR P2MP Policy at the Root node.

Traffic is then forwarded toward the leaves following the replication segments.

4. Using Controller to build a P2MP Segment

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

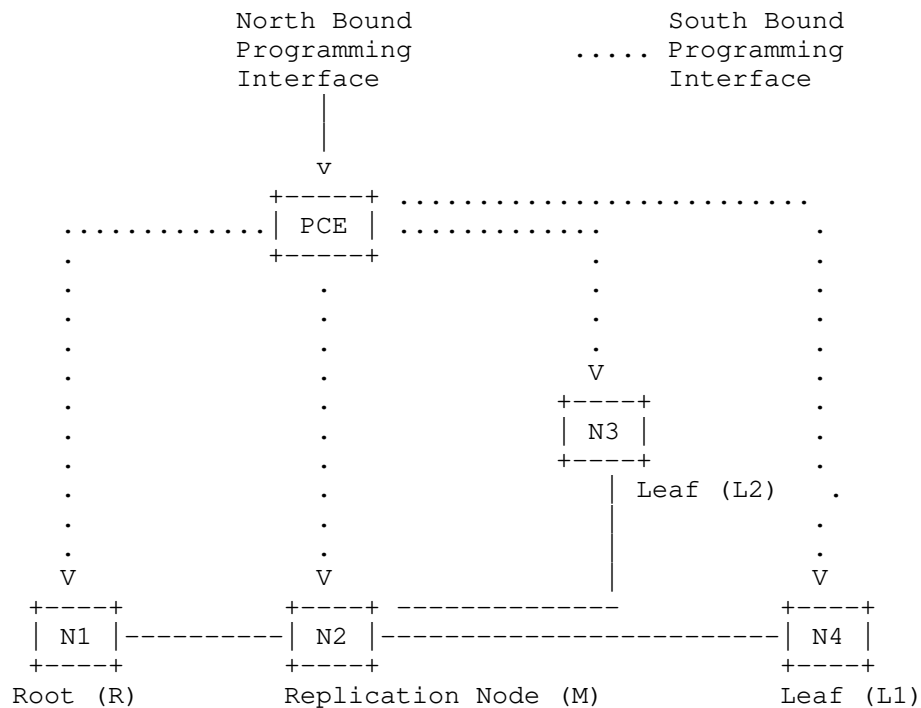


Figure 1: Centralized Control Plane Model

4.1. SR P2MP Policy Creation

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

4.1.1. API

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

1. Create P2MP SR policy
2. Delete P2MP SR policy
3. Update P2MP SR policy

4.1.2. Invoking API

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

4.2. P2MP Segment Computation

Network operator passes the addresses of the root (R) and set of leaves {L} as well as Traffic Engineering (TE) attributes (e.g., constraints such as link color, optimization criteria such as latency) of the P2MP segment to PCE via a suitable North-Bound API. The PCE computes the tree instantiates the P2MP segment on Root, Replication, and Leaf nodes.

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

Ideally, same P2MP SID SHOULD be used for forwarding entries at Root, Mid, and Leaf nodes. Different P2MP SIDs MAY be used at different node(s) if it is not feasible to use same P2MP SID. SIDs (BSID as well as P2MP SID) can also be assigned by operator.

A PCE can modify a P2MP segment following network element failure or in case a better path can be found based on the new network state. In this case, the PCE may want to setup the new tree and remove the old tree from the network in order to minimize traffic loss. As such, a separate P2MP SID can be used for the new tree.

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

4.2.1. Topology Discovery

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

4.2.2. Capability and Attribute Discovery

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

4.3. Instantiating P2MP segment nodes

Once a PCE computes a tree for P2MP segment, it needs to instantiate the segment on the relevant network nodes. The PCE can use various

protocols to program the forwarding entries, and these protocols are described below.

4.3.1. PCEP

PCE Protocol (PCEP) has been traditionally used:

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

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

4.3.2. BGP

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

4.3.3. NetConf

TBD

4.4. Protection

4.4.1. Local Protection

A network link/node on the tree of a P2MP segment can be protected using SR policies computed by PCE. The backup SR policies shall be programmed in forwarding plane in order to minimize traffic loss when the protected link/node fails.

4.4.2. Path Protection

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

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

There are no additional security risks introduced by this design.

7. Acknowledgements

The authors would like to acknowledge Siva Sivabalan.

8. Contributors

Clayton Hassen
Bell Canada
Vancouver
Canada

Email: clayton.hassen@bell.ca

Kurtis Gillis
Bell Canada
Halifax
Canada

Email: kurtis.gillis@bell.ca

Arvind Venkateswaran
Cisco Systems, Inc.
San Jose
US

Email: arvvenka@cisco.com

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Swadesh Agrawal
Cisco Systems, Inc.
San Jose
US

Email: swaagraw@cisco.com

Jayant Kotalwar
Nokia
Mountain View
US

Email: jayant.kotalwar@nokia.com

Tanmoy Kundu
Nokia
Mountain View
US

Email: tanmoy.kundu@nokia.com

9. Normative References

- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-03 (work in progress), May 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>>.

Authors' Addresses

Daniel Voyer (editor)
Bell Canada
Montreal
CA

Email: daniel.voyer@bell.ca

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Rishabh Parekh
Cisco Systems, Inc.
San Jose
US

Email: riparekh@cisco.com

Hooman Bidgoli
Nokia
Ottawa
CA

Email: hooman.bidgoli@nokia.com

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net

SPRING
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

F. Clad, Ed.
Cisco Systems, Inc.
X. Xu, Ed.
Alibaba
C. Filsfils
Cisco Systems, Inc.
D. Bernier
Bell Canada
C. Li
Huawei
B. Decraene
Orange
S. Ma
Juniper
C. Yadlapalli
AT&T
W. Henderickx
Nokia
S. Salsano
Universita di Roma "Tor Vergata"
October 22, 2018

Service Programming with Segment Routing
draft-xuclad-spring-sr-service-programming-01

Abstract

This document defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Classification and steering	4
4. Service segments	5
4.1. SR-aware services	5
4.2. SR-unaware services	6
5. SR service policies	7
5.1. SR-MPLS data plane	8
5.2. SRv6 data plane	10
6. SR proxy behaviors	11
6.1. Static SR proxy	14
6.1.1. SR-MPLS pseudocode	16
6.1.2. SRv6 pseudocode	17
6.2. Dynamic SR proxy	19
6.2.1. SR-MPLS pseudocode	19
6.2.2. SRv6 pseudocode	20
6.3. Shared memory SR proxy	21
6.4. Masquerading SR proxy	21
6.4.1. SRv6 masquerading proxy pseudocode	22
6.4.2. Variant 1: Destination NAT	23
6.4.3. Variant 2: Caching	23
7. Metadata	23
7.1. MPLS data plane	23
7.2. IPv6 data plane	24
7.2.1. SRH TLV objects	24
7.2.2. SRH tag	25
8. Implementation status	25
8.1. SR-aware services	26
8.2. Proxy behaviors	26
9. Related works	26
10. IANA Considerations	27

10.1. SRv6 Endpoint Behaviors	27
10.2. Segment Routing Header TLVs	27
11. Security Considerations	27
12. Acknowledgements	27
13. Contributors	28
14. References	28
14.1. Normative References	28
14.2. Informative References	29
Authors' Addresses	29

1. Introduction

Segment Routing (SR) is an architecture based on the source routing paradigm that seeks the right balance between distributed intelligence and centralized programmability. SR can be used with an MPLS or an IPv6 data plane to steer packets through an ordered list of instructions, called segments. These segments may encode simple routing instructions for forwarding packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of these services, running either on a physical appliance or in a virtual environment, are associated with a segment identifier (SID). These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined together in a SID-list to achieve service programming, but also with other types of segments as defined in [RFC8402]. SR thus provides a fully integrated solution for overlay, underlay and service programming. Furthermore, the IPv6 instantiation of SR (SRv6) supports metadata transportation in the Segment Routing header [I-D.ietf-6man-segment-routing-header], either natively in the tag field or with extensions such as TLVs.

This document describes how a service can be associated with a SID, including legacy services with no SR capabilities, and how these service SIDs are integrated within an SR policy. The definition of an SR Policy and the traffic steering mechanisms are covered in [I-D.ietf-spring-segment-routing-policy] and hence outside the scope of this document.

The definition of control plane components, such as service segment discovery, is outside the scope of this data plane document. For reference, the option of using BGP extensions to support SR service programming is proposed in [I-D.dawra-idr-bgp-sr-service-chaining].

2. Terminology

This document leverages the terminology proposed in [RFC8402] and [I-D.ietf-spring-segment-routing-policy]. It also introduces the following new terms.

Service segment: A segment associated with a service. The service may either run on a physical appliance or in a virtual environment such as a virtual machine or container.

SR-aware service: A service that is fully capable of processing SR traffic. An SR-aware service can be directly associated with a service segment.

SR-unaware service: A service that is unable to process SR traffic or may behave incorrectly due to presence of SR information in the packet headers. An SR-unaware service can be associated with a service segment through an SR proxy function.

3. Classification and steering

Classification and steering mechanisms are defined in section 8 of [I-D.ietf-spring-segment-routing-policy] and are independent from the purpose of the SR policy. From the perspective of a headend node classifying and steering traffic into an SR policy, there is no difference whether this policy contains IGP, BGP, peering, VPN or service segments, or any combination of these.

As documented in the above reference, traffic is classified when entering an SR domain. The SR policy headend may, depending on its capabilities, classify the packets on a per-destination basis, via simple FIB entries, or apply more complex policy routing rules requiring to look deeper into the packet. These rules are expected to support basic policy routing such as 5-tuple matching. In addition, the IPv6 SRH tag field defined in [I-D.ietf-6man-segment-routing-header] can be used to identify and classify packets sharing the same set of properties. Classified traffic is then steered into the appropriate SR policy and forwarded as per the SID-list(s) of the active candidate path.

SR traffic can be re-classified by an SR endpoint along the original SR policy (e.g., DPI service) or a transit node intercepting the traffic. This node is the head-end of a new SR policy that is imposed onto the packet, either as a stack of MPLS labels or as an IPv6 SRH.

4. Service segments

In the context of this document, the term service refers to a physical appliance running on dedicated hardware, a virtualized service inside an isolated environment such as a VM, container or namespace, or any process running on a compute element. A service may also comprise multiple sub-components running in different processes or containers. Unless otherwise stated, this document does not make any assumption on the type or execution environment of a service.

The execution of a service can be integrated as part of an SR policy by assigning a segment identifier, or SID, to the service and including this service SID in the SR policy SID-list. Such a service SID may be of local or global significance. In the former case, other segments, such as prefix or adjacency segments, can be used to steer the traffic up to the node where the service segment is instantiated. In the latter case, the service is directly reachable from anywhere in the routing domain. This is realized with SR-MPLS by assigning a SID from the global label block ([I-D.ietf-spring-segment-routing-mpls]), or with SRv6 by advertising the SID locator in the routing protocol ([I-D.filsfils-spring-srv6-network-programming]). It is up to the network operator to define the scope and reachability of each service SID. This decision can be based on various considerations such as infrastructure dynamicity, available control plane or orchestration system capabilities.

This document categorizes services in two types, depending on whether they are able to behave properly in the presence of SR information or not. These are respectively named SR-aware and SR-unaware services.

4.1. SR-aware services

An SR-aware service can process the SR information in the packets it receives. This means being able to identify the active segment as a local instruction and move forward in the segment list, but also that the service's own behavior is not hindered due to the presence of SR information. For example, an SR-aware firewall filtering SRv6 traffic based on its final destination must retrieve that information from the last entry in the SRH rather than the Destination Address field of the IPv6 header.

An SR-aware service is associated with a locally instantiated service segment, which is used to steer traffic through it.

If the service is configured to intercept all the packets passing through the appliance, the underlying routing system only has to

implement a default SR endpoint behavior (SR-MPLS node segment or SRv6 End function), and the corresponding SID will be used to steer traffic through the service.

If the service requires the packets to be directed to a specific virtual interface, networking queue or process, a dedicated SR behavior may be required to steer the packets to the appropriate location. The definition of such service-specific functions is out of the scope of this document.

SR-aware services also enable advanced network programming functionalities such as conditional branching and jumping to arbitrary SIDs in the segment list. In addition, SRv6 provides several ways of passing and exchanging information between services (e.g., SID arguments, tag field and TLVs). An example scenario involving these features is described in [IFIP18], which discusses the implementation of an SR-aware Intrusion Detection System.

Examples of SR-aware services are provided in section Section 8.1.

4.2. SR-unaware services

Any service that does not meet the above criteria for SR-awareness is considered as SR-unaware.

An SR-unaware service is not able to process the SR information in the traffic that it receives. It may either drop the traffic or take erroneous decisions due to the unrecognized routing information. In order to include such services in an SR policy, it is thus required to remove the SR information as well as any other encapsulation header before the service receives the packet, or to alter it in such a way that the service can correctly process the packet.

In this document, we define the concept of an SR proxy as an entity, separate from the service, that performs these modifications and handle the SR processing on behalf of a service. The SR proxy can run as a separate process on the service appliance, on a virtual switch or router on the compute node or on a different host.

An SR-unaware service is associated with a service segment instantiated on the SR proxy, which is used to steer traffic through the service. Section 6 describes several SR proxy behaviors to handle the encapsulation headers and SR information under various circumstances.

5. SR service policies

An SR service policy is an SR policy, as defined in [I-D.ietf-spring-segment-routing-policy], that includes at least one service. This service is represented in the SID-list by its associated service SID. In case the policy should include several services, the service traversal order is indicated by the relative position of each service SID in the SID-list. Using the mechanisms described in [I-D.ietf-spring-segment-routing-policy], it is possible to load balance the traffic over several services, or instances of the same service, by associating with the SR service policy a weighted set of SID-lists, each containing a possible sequence of service SIDs to be traversed. Similarly, several candidate paths can be specified for the SR service policy, each with its own set of SID-lists, for resiliency purposes.

Furthermore, binding SIDs (BSIDs) can be leveraged in the context of service policies to reduce the number of SIDs imposed by the headend, provide opacity between domains and improve scalability, as described in [I-D.filsfils-spring-sr-policy-considerations]. For example, a network operator may want a policy in its core domain to include services that are running in one of its datacenters. One option is to define an SR policy at ingress edge of the core domain that explicitly includes all the SIDs needed to steer the traffic through the core and in the DC, but that may result in a long SID-list and requires to update the ingress edge configuration every time the DC part of the policy is modified. Alternatively, a separate policy can be defined at the ingress edge of the datacenter with only the SIDs that needs to be executed there and its BSID included in the core domain policy. That BSID remains stable when the DC policy is modified and can even be shared among several core domain policies that would require the same type of processing in the DC.

This section describes how services can be integrated within an SR-MPLS or SRv6 service policy.

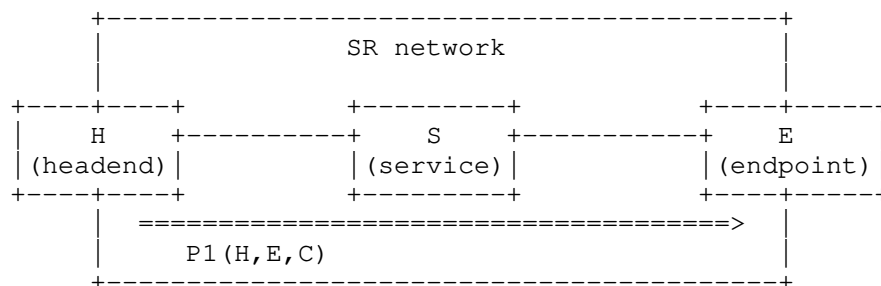


Figure 1: SR service policy

Figure 1 illustrates a basic SR service policy instantiated on a headend node H towards an endpoint E and traversing a service S. The SR policy may also include additional requirements, such as traffic engineering or VPN. On the head-end H, the SR policy P1 is created with a color C and endpoint E and associated with an SR path that can either be explicitly configured, dynamically computed on H or provisioned by a network controller.

In its most basic form, the SR policy P1 would be resolved into the SID-list $\langle \text{SID}(S), \text{SID}(E) \rangle$. This is assuming that $\text{SID}(S)$ and $\text{SID}(E)$ are directly reachable from H and S, respectively, and that the forwarding path meets the policy requirement. However, depending on the dataplane and the segments available in the network, additional SIDs may be required to enforce the SR policy.

This model applies regardless of the SR-awareness of the service. If it is SR-unaware, then S simply represents the proxy that takes care of transmitting the packet to the actual service.

Traffic can then be steered into this policy using any of the mechanisms described in [I-D.ietf-spring-segment-routing-policy].

The following subsections describe the specificities of each SR dataplane.

5.1. SR-MPLS data plane

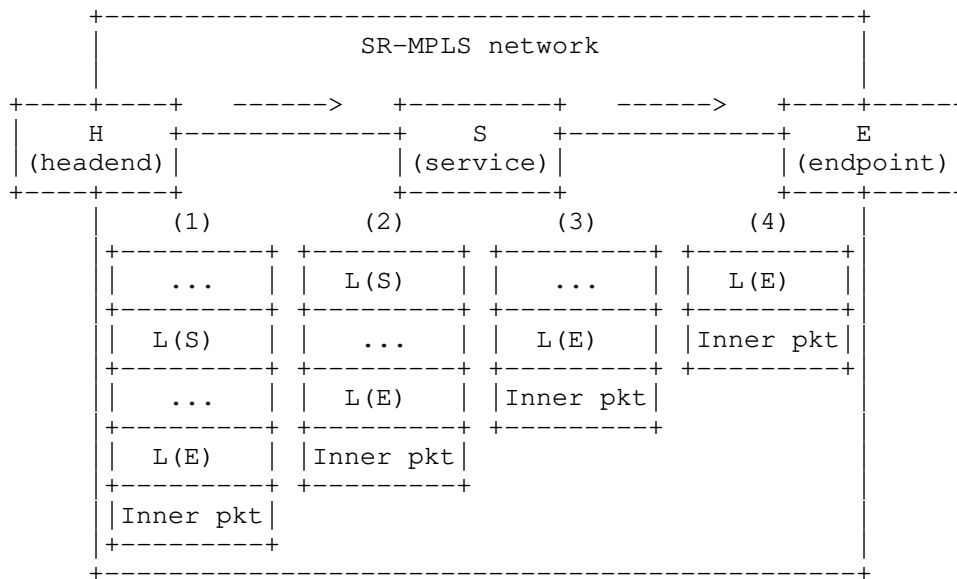


Figure 2: Packet walk in an SR-MPLS network

In an SR-MPLS network, the SR policy SID-list is encoded as a stack of MPLS labels[I-D.ietf-spring-segment-routing-mpls] and pushed on top of the packet.

In the example shown on Figure 2, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an MPLS label stack that includes at least a label L(S) associated to service S and a label L(E) associated to the endpoint E. The label stack may also include additional intermediate segments if these are required for traffic engineering (e.g., to encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID L(S) may be taken from the global or local SID block of node S and, in the latter case, one or more SIDs might be needed before L(S) in order for the packet to reach node S (e.g., a prefix-SID of S), where L(S) can be interpreted. The same applies for the segment L(E) at the SR policy endpoint.

Special consideration must be taken into account when using Local SIDs for service identification due to increased label stack depth and the associated impacts.

When the packet arrives at S, this node determines how to process the packet based on the semantic locally associated to the top label L(S). If S is an SR-aware service, the SID L(S) may provide

additional context or indication on how to process the packet (e.g., payload type or a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, L(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, L(S) is also popped from the label stack in order to expose the next SID, which may be L(E) or another intermediate segment.

5.2. SRv6 data plane

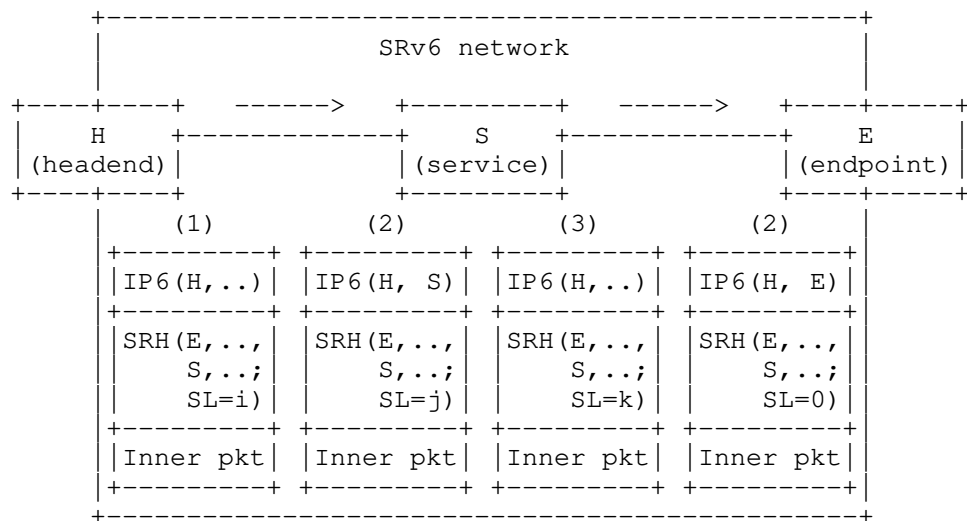


Figure 3: Packet walk in an SRv6 network

In an SRv6 network, the SR Policy is encoded into the packet as an IPv6 header possibly followed by a Segment Routing header (SRH) [I-D.ietf-6man-segment-routing-header].

In the example shown on Figure 3, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an SRH that includes at least a segment SID(S) to the service, or service proxy, S and a segment SID(E) to the endpoint E. The SRH may also include additional intermediate segments if these are required for traffic engineering (e.g., to encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service segment locator may or may not be advertised in the routing protocol and, in the latter case, one or more SIDs might be needed before SID(S) in order to bring the packet up to node S, where SID(S) can be interpreted. The same applies for the segment SID(E) at the SR policy endpoint.

When the packet arrives at S, this node determines how to process the packet based on the semantic locally associated to the active segment SID(S). If S is an SR-aware service, then SID(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, SID(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, the SRv6 End function is also applied in order to make the next SID, which may be SID(E) or another intermediate segment, active.

The "Inner pkt" on Figure 3 represents the SRv6 payload, which may be an encapsulated IP packet, an Ethernet frame or a transport-layer payload, for example.

6. SR proxy behaviors

This section describes several SR proxy behaviors designed to enable SR service programming through SR-unaware services. A system implementing one of these functions may handle the SR processing on behalf of an SR-unaware service and allows the service to properly process the traffic that is steered through it.

A service may be located at any hop in an SR policy, including the last segment. However, the SR proxy behaviors defined in this section are dedicated to supporting SR-unaware services at intermediate hops in the segment list. In case an SR-unaware service is at the last segment, it is sufficient to ensure that the SR information is ignored (IPv6 routing extension header with Segments Left equal to 0) or removed before the packet reaches the service (MPLS PHP, SRv6 End.D or PSP).

As illustrated on Figure 4, the generic behavior of an SR proxy has two parts. The first part is in charge of passing traffic from the network to the service. It intercepts the SR traffic destined for the service via a locally instantiated service segment, modifies it in such a way that it appears as non-SR traffic to the service, then sends it out on a given interface, IFACE-OUT, connected to the service. The second part receives the traffic coming back from the service on IFACE-IN, restores the SR information and forwards it according to the next segment in the list. IFACE-OUT and IFACE-IN are respectively the proxy interface used for sending traffic to the service and the proxy interface that receives the traffic coming back from the service. These can be physical interfaces or sub-interfaces (VLANs) and, unless otherwise stated, IFACE-OUT and IFACE-IN can represent the same interface.

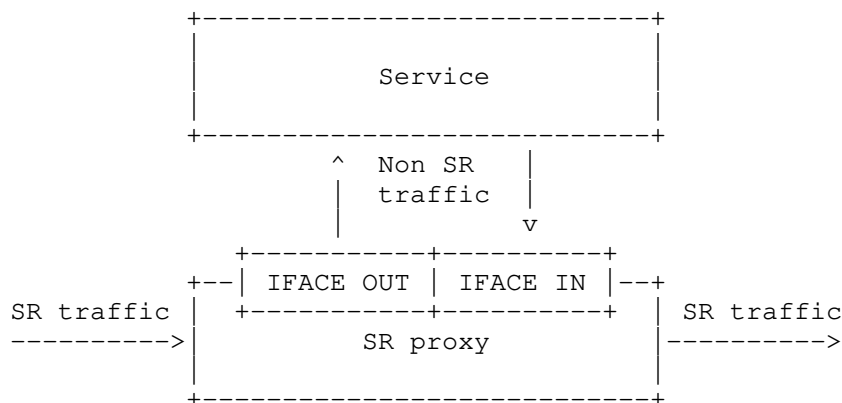


Figure 4: Generic SR proxy

In the next subsections, the following SR proxy mechanisms are defined:

- o Static proxy
- o Dynamic proxy
- o Shared-memory proxy
- o Masquerading proxy

Each mechanism has its own characteristics and constraints, which are summarized in the below table. It is up to the operator to select the best one based on the proxy node capabilities, the service behavior and the traffic type. It is also possible to use different proxy mechanisms within the same service policy.

		S t a t i c	D y n a m i c	S h a r e d m e m .	M a s q u e r a d i n g
SR flavors	SR-MPLS	Y	Y	Y	-
	SRv6 insertion	P	P	P	Y
	SRv6 encapsulation	Y	Y	Y	-
Chain agnostic configuration		N	N	Y	Y
Transparent to chain changes		N	Y	Y	Y
Service support	DA modification	Y	Y	Y	NAT
	Payload modification	Y	Y	Y	Y
	Packet generation	Y	Y	cache	cache
	Packet deletion	Y	Y	Y	Y
	Transport endpoint	Y	Y	cache	cache
Supported traffic	Ethernet	Y	Y	Y	-
	IPv4	Y	Y	Y	-
	IPv6	Y	Y	Y	Y

Figure 5: SR proxy summary

Note: The use of a shared memory proxy requires both the service (VNF) and the proxy to be running on the same node.

6.1. Static SR proxy

The static proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an MPLS label stack or an IPv6 header on top of an inner packet, which can be Ethernet, IPv4 or IPv6.

A static SR proxy segment is associated with the following mandatory parameters

- o INNER-TYPE: Inner packet type
- o NH-ADDR: Next hop Ethernet address (only for inner type IPv4 and IPv6)
- o IFACE-OUT: Local interface for sending traffic towards the service
- o IFACE-IN: Local interface receiving the traffic coming back from the service
- o CACHE: SR information to be attached on the traffic coming back from the service, including at least
 - * CACHE.SA: IPv6 source address (SRv6 only)
 - * CACHE.LIST: Segment list expressed as MPLS labels or IPv6 address

A static SR proxy segment is thus defined for a specific service, inner packet type and cached SR information. It is also bound to a pair of directed interfaces on the proxy. These may be both directions of a single interface, or opposite directions of two different interfaces. The latter is recommended in case the service is to be used as part of a bi-directional SR SC policy. If the proxy and the service both support 802.1Q, IFACE-OUT and IFACE-IN can also represent sub-interfaces.

The first part of this behavior is triggered when the proxy node receives a packet whose active segment matches a segment associated with the static proxy behavior. It removes the SR information from the packet then sends it on a specific interface towards the associated service. This SR information corresponds to the full label stack for SR-MPLS or to the encapsulation IPv6 header with any attached extension header in the case of SRv6.

The second part is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This

policy attaches to the incoming traffic the cached SR information associated with the SR proxy segment. If the proxy segment uses the SR-MPLS data plane, CACHE contains a stack of labels to be pushed on top of the packets. With the SRv6 data plane, CACHE is defined as a source address, an active segment and an optional SRH (tag, segments left, segment list and metadata). The proxy encapsulates the packets with an IPv6 header that has the source address, the active segment as destination address and the SRH as a routing extension header. After the SR information has been attached, the packets are forwarded according to the active segment, which is represented by the top MPLS label or the IPv6 Destination Address. An MPLS TTL or IPv6 Hop Limit value may also be configured in CACHE. If it is not, the proxy should set these values according to the node's default setting for MPLS or IPv6 encapsulation.

In this scenario, there are no restrictions on the operations that can be performed by the service on the stream of packets. It may operate at all protocol layers, terminate transport layer connections, generate new packets and initiate transport layer connections. This behavior may also be used to integrate an IPv4-only service into an SRv6 policy. However, a static SR proxy segment can be used in only one service policy at a time. As opposed to most other segment types, a static SR proxy segment is bound to a unique list of segments, which represents a directed SR SC policy. This is due to the cached SR information being defined in the segment configuration. This limitation only prevents multiple segment lists from using the same static SR proxy segment at the same time, but a single segment list can be shared by any number of traffic flows. Besides, since the returning traffic from the service is re-classified based on the incoming interface, an interface can be used as receiving interface (IFACE-IN) only for a single SR proxy segment at a time. In the case of a bi-directional SR SC policy, a different SR proxy segment and receiving interface are required for the return direction.

The static proxy behavior may also be used for sending traffic through "bump in the wire" services that are transparent to the IP and Ethernet layers. This type of processing is assumed when the inner traffic type is Ethernet, since the original destination address of the Ethernet frame is preserved when the packet is steered into the SR Policy and likely associated with a node downstream of the policy tail-end. In case the inner type is IP (IPv4 or IPv6), the NH-ADDR parameter may be set to a dummy or broadcast Ethernet address, or simply to the address of the proxy receiving interface (IFACE-IN).

6.1.1.1. SR-MPLS pseudocode

6.1.1.1.1. Static proxy for inner type Ethernet

Upon receiving an MPLS packet with top label L, where L is an MPLS L2 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is Ethernet THEN
3. Forward the exposed frame on IFACE-OUT
4. ELSE
5. Drop the packet

Upon receiving on IFACE-IN an Ethernet frame with a destination address different than the interface address, a node N does:

1. Push labels in CACHE on top of the frame Ethernet header
2. Lookup the top label and proceed accordingly

The receiving interface must be configured in promiscuous mode in order to accept those Ethernet frames.

6.1.1.1.2. Static proxy for inner type IPv4

Upon receiving an MPLS packet with top label L, where L is an MPLS IPv4 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is IPv4 THEN
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Upon receiving a non-link-local IPv4 packet on IFACE-IN, a node N does:

1. Decrement TTL and update checksum
2. Push labels in CACHE on top of the packet IPv4 header
3. Lookup the top label and proceed accordingly

6.1.1.1.3. Static proxy for inner type IPv6

Upon receiving an MPLS packet with top label L, where L is an MPLS IPv6 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is IPv6 THEN
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N does:

1. Decrement Hop Limit
2. Push labels in CACHE on top of the packet IPv6 header
3. Lookup the top label and proceed accordingly

6.1.2. SRv6 pseudocode

6.1.2.1. Static proxy for inner type Ethernet

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for Ethernet traffic, a node N does:

1. IF ENH == 59 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed frame on IFACE-OUT
4. ELSE
5. Drop the packet

Ref1: 59 refers to "no next header" as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving on IFACE-IN an Ethernet frame with a destination address different than the interface address, a node N does:

1. Retrieve CACHE entry matching IFACE-IN and traffic type
2. Push SRH with CACHE.LIST on top of the Ethernet header ;; Ref2
3. Push IPv6 header with
 - SA = CACHE.SA
 - DA = CACHE.LIST[0] ;; Ref3
 - Next Header = 43 ;; Ref4
4. Set outer payload length and flow label
5. Lookup outer DA in appropriate table and proceed accordingly

Ref2: Unless otherwise specified, the segments in CACHE.LIST should be encoded in reversed order, Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1, and Next Header should be set to 59.

Ref3: CACHE.LIST[0] represents the first IPv6 SID in CACHE.LIST.

Ref4: If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header value must be set to 59.

The receiving interface must be configured in promiscuous mode in order to accept those Ethernet frames.

6.1.2.2. Static proxy for inner type IPv4

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for IPv4 traffic, a node N does:

1. IF ENH == 4 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Ref1: 4 refers to IPv4 encapsulation as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving a non-link-local IPv4 packet on IFACE-IN, a node N does:

1. Decrement TTL and update checksum
2. IF CACHE.SRH THEN ;; Ref2
3. Push CACHE.SRH on top of the existing IPv4 header
4. Set NH value of the pushed SRH to 4
5. Push outer IPv6 header with SA, DA and traffic class from CACHE
6. Set outer payload length and flow label
7. Set NH value to 43 if an SRH was added, or 4 otherwise
8. Lookup outer DA in appropriate table and proceed accordingly

Ref2: CACHE.SRH represents the SRH defined in CACHE, if any, for the static SR proxy segment associated with IFACE-IN.

6.1.2.3. Static proxy for inner type IPv6

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for IPv6 traffic, a node N does:

1. IF ENH == 41 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Ref1: 41 refers to IPv6 encapsulation as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N does:

1. Decrement Hop Limit
2. IF CACHE.SRH THEN ;; Ref2
3. Push CACHE.SRH on top of the existing IPv6 header
4. Set NH value of the pushed SRH to 41
5. Push outer IPv6 header with SA, DA and traffic class from CACHE
6. Set outer payload length and flow label
7. Set NH value to 43 if an SRH was added, or 41 otherwise
8. Lookup outer DA in appropriate table and proceed accordingly

Ref2: CACHE.SRH represents the SRH defined in CACHE, if any, for the static SR proxy segment associated with IFACE-IN.

6.2. Dynamic SR proxy

The dynamic proxy is an improvement over the static proxy that dynamically learns the SR information before removing it from the incoming traffic. The same information can then be re-attached to the traffic returning from the service. As opposed to the static SR proxy, no CACHE information needs to be configured. Instead, the dynamic SR proxy relies on a local caching mechanism on the node instantiating this segment.

Upon receiving a packet whose active segment matches a dynamic SR proxy function, the proxy node pops the top MPLS label or applies the SRv6 End behavior, then compares the updated SR information with the cache entry for the current segment. If the cache is empty or different, it is updated with the new SR information. The SR information is then removed and the inner packet is sent towards the service.

The cache entry is not mapped to any particular packet, but instead to an SR SC policy identified by the receiving interface (IFACE-IN). Any non-link-local IP packet or non-local Ethernet frame received on that interface will be re-encapsulated with the cached headers as described in Section 6.1. The service may thus drop, modify or generate new packets without affecting the proxy.

6.2.1. SR-MPLS pseudocode

The dynamic proxy SR-MPLS pseudocode is obtained by inserting the following instructions at the beginning of the static SR-MPLS pseudocode (Section 6.1.1).

```

1.  IF top label S bit is 0 THEN                                ;; Ref1
2.      Pop top label
3.      IF C(IFACE-IN) different from remaining labels THEN    ;; Ref2
4.          Copy all remaining labels into C(IFACE-IN)         ;; Ref3
5.  ELSE
6.      Drop the packet

```

Ref1: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

Ref2: A TTL margin can be configured for the top label stack entry to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a TTL difference smaller than the configured margin should not trigger a cache update (provided that the labels are the same).

Ref3: C(IFACE-IN) represents the cache entry associated to the dynamic SR proxy segment. It is identified with IFACE-IN in order to efficiently retrieve the right SR information when a packet arrives on this interface.

In addition, the inbound policy should check that C(IFACE-IN) has been defined before attempting to restore the MPLS label stack and drop the packet otherwise.

6.2.2. SRv6 pseudocode

The dynamic proxy SRv6 pseudocode is obtained by inserting the following instructions between lines 1 and 2 of the static proxy SRv6 pseudocode.

```

1.  IF NH=SRH & SL > 0 THEN                                    ;; Ref1
2.      Decrement SL and update the IPv6 DA with SRH[SL]
3.      IF C(IFACE-IN) different from IPv6 encaps THEN        ;; Ref2
4.          Copy the IPv6 encaps into C(IFACE-IN)             ;; Ref3
5.  ELSE
6.      Drop the packet

```

Ref1: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

Ref2: "IPv6 encaps" represents the IPv6 header and any attached extension header.

Ref3: C(IFACE-IN) represents the cache entry associated to the dynamic SR proxy segment. It is identified with IFACE-IN in order to efficiently retrieve the right SR information when a packet arrives on this interface.

In addition, the inbound policy should check that C(IFACE-IN) has been defined before attempting to restore the IPv6 encapsulation and drop the packet otherwise.

6.3. Shared memory SR proxy

The shared memory proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy behavior leverages a shared-memory interface with a virtualized service (VNF) in order to hide the SR information from an SR-unaware service while keeping it attached to the packet. We assume in this case that the proxy and the VNF are running on the same compute node. A typical scenario is an SR-capable vrouter running on a container host and forwarding traffic to VNFs isolated within their respective container.

More details will be added in a future revision of this document.

6.4. Masquerading SR proxy

The masquerading proxy is an SR endpoint behavior for processing SRv6 traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an IPv6 header and an SRH on top of an inner payload. The masquerading behavior is independent from the inner payload type. Hence, the inner payload can be of any type but it is usually expected to be a transport layer packet, such as TCP or UDP.

A masquerading SR proxy segment is associated with the following mandatory parameters:

- o S-ADDR: Ethernet or IPv6 address of the service
- o IFACE-OUT: Local interface for sending traffic towards the service
- o IFACE-IN: Local interface receiving the traffic coming back from the service

A masquerading SR proxy segment is thus defined for a specific service and bound to a pair of directed interfaces or sub-interfaces on the proxy. As opposed to the static and dynamic SR proxies, a masquerading segment can be present at the same time in any number of SR SC policies and the same interfaces can be bound to multiple masquerading proxy segments. The only restriction is that a masquerading proxy segment cannot be the last segment in an SR SC policy.

The first part of the masquerading behavior is triggered when the proxy node receives an IPv6 packet whose Destination Address matches a masquerading proxy segment. The proxy inspects the IPv6 extension headers and substitutes the Destination Address with the last segment in the SRH attached to the IPv6 header, which represents the final destination of the IPv6 packet. The packet is then sent out towards the service.

The service receives an IPv6 packet whose source and destination addresses are respectively the original source and final destination. It does not attempt to inspect the SRH, as RFC8200 specifies that routing extension headers are not examined or processed by transit nodes. Instead, the service simply forwards the packet based on its current Destination Address. In this scenario, we assume that the service can only inspect, drop or perform limited changes to the packets. For example, Intrusion Detection Systems, Deep Packet Inspectors and non-NAT Firewalls are among the services that can be supported by a masquerading SR proxy. Variants of the masquerading behavior are defined in Section 6.4.2 and Section 6.4.3 to support a wider range of services.

The second part of the masquerading behavior, also called de-masquerading, is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This policy inspects the incoming traffic and triggers a regular SRv6 endpoint processing (End) on any IPv6 packet that contains an SRH. This processing occurs before any lookup on the packet Destination Address is performed and it is sufficient to restore the right active segment as the Destination Address of the IPv6 packet.

6.4.1. SRv6 masquerading proxy pseudocode

Masquerading: Upon receiving a packet destined for S, where S is an IPv6 masquerading proxy segment, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Update the IPv6 DA with SRH[0]
3. Forward the packet on IFACE-OUT
4. ELSE
5. Drop the packet

De-masquerading: Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Decrement SL
3. Update the IPv6 DA with SRH[SL] ;; Ref1
4. Lookup DA in appropriate table and proceed accordingly

Ref1: This pseudocode can be augmented to support the Penultimate Segment Popping (PSP) endpoint flavor. The exact pseudocode modification are provided in [I-D.filsfils-spring-srv6-network-programming].

6.4.2. Variant 1: Destination NAT

Services modifying the destination address in the packets they process, such as NATs, can be supported by a masquerading proxy with the following modification to the de-masquerading pseudocode.

De-masquerading - NAT: Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Update SRH[0] with the IPv6 DA
3. Decrement SL
4. Update the IPv6 DA with SRH[SL]
5. Lookup DA in appropriate table and proceed accordingly

6.4.3. Variant 2: Caching

Services generating packets or acting as endpoints for transport connections can be supported by adding a dynamic caching mechanism similar to the one described in Section 6.2.

More details will be added in a future revision of this document.

7. Metadata

7.1. MPLS data plane

Metadata can be carried for SR-MPLS traffic in a Segment Routing header inserted between the last MPLS label and the MPLS payload. When used solely as a metadata container, the SRH does not carry any segment but only the mandatory header fields, including the tag and flags, and any TLVs that is required for transporting the metadata.

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata in an MPLS packet is a potential issue to be addressed. One possible solution is to add the indication about the presence of metadata in the semantic of the SIDs. Note that only the SIDs whose behavior involves looking at the metadata or the MPLS payload would need to include such semantic (e.g., service segments). Other segments, such as traffic engineering segments, are not affected by the presence of metadata. Another, more generic, solution is to introduce a protocol identifier

field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

7.2. IPv6 data plane

7.2.1. SRH TLV objects

The IPv6 SRH TLV objects are designed to carry all sorts of metadata. In particular, the NSH carrier TLV is defined as a container for NSH metadata.

TLV objects can be imposed by the ingress edge router that steers the traffic into the SR SC policy.

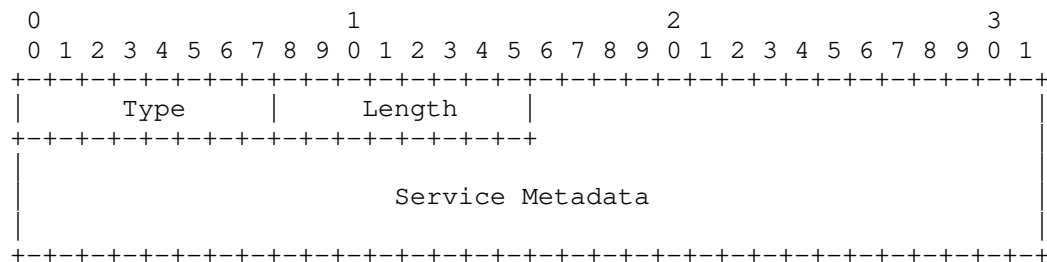
An SR-aware service may impose, modify or remove any TLV object attached to the first SRH, either by directly modifying the packet headers or via a control channel between the service and its forwarding plane.

An SR-aware service that re-classifies the traffic and steers it into a new SR SC policy (e.g. DPI) may attach any TLV object to the new SRH.

Metadata imposition and handling will be further discussed in a future version of this document.

7.2.1.1. Opaque Metadata TLV

This document defines an SRv6 TLV called Opaque Metadata TLV. This is a fixed-length container to carry any type of Service Metadata. No assumption is made by this document on the structure or the content of the carried metadata. The Opaque Metadata TLV has the following format:



where:

- o Type: to be assigned by IANA.

- o Length: 14.
- o Service Metadata: 14 octets of opaque data.

7.2.1.2. NSH Carrier TLV

This document defines an SRv6 TLV called NSH Carrier TLV. It is a container to carry Service Metadata in the form of Variable-Length Metadata as defined in [RFC8300] for NSH MD Type 2. The NSH Carrier TLV has the following format:

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |      Flags      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                Service Metadata                                //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

- o Type: to be assigned by IANA.
- o Length: the total length of the TLV.
- o Flags: 8 bits. No flags are defined in this document. SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- o Service Metadata: a list of Service Metadata TLV as defined in [RFC8300] for NSH MD Type 2.

7.2.2. SRH tag

The SRH tag identifies a packet as part of a group or class of packets [I-D.ietf-6man-segment-routing-header].

In the context of service programming, this field can be used to encode basic metadata in the SRH. An example use case would be to leverage the SRH tag to encode a policy ID which could be leveraged in an SR-aware function to determine which processing policy to apply rather than having doing local classification or leverage alternate encapsulations.

8. Implementation status

This section is to be removed prior to publishing as an RFC.

8.1. SR-aware services

Specific SRv6 support has been implemented for the below open-source services:

- o Iptables (1.6.2 and later)
- o Nftables (0.8.4 and later)
- o Snort

In addition, any service relying on the Linux kernel, version 4.10 and later, or FD.io VPP for packet forwarding can be considered as SR-aware.

8.2. Proxy behaviors

The static SR proxy is available for SR-MPLS and SRv6 on various Cisco hardware and software platforms. Furthermore, the following proxies are available on open-source software.

		VPP	Linux
M P L S	Static proxy	Available	In progress
	Dynamic proxy	In progress	In progress
	Shared memory proxy	In progress	In progress
S R v 6	Static proxy	Available	In progress
	Dynamic proxy	Available	Available
	Shared memory proxy	In progress	In progress
	Masquerading proxy	Available	Available

Figure 6: Open-source implementation status table

9. Related works

The Segment Routing solution addresses a wide problem that covers both topological and service policies. The topological and service instructions can be either deployed in isolation or in combination. SR has thus a wider applicability than the architecture defined in [RFC7665]. Furthermore, the inherent property of SR is a stateless

network fabric. In SR, there is no state within the fabric to recognize a flow and associate it with a policy. State is only present at the ingress edge of the SR domain, where the policy is encoded into the packets. This is completely different from other proposals such as [RFC8300] and the MPLS label swapping mechanism described in [I-D.ietf-mpls-sfc], which rely on state configured at every hop of the service chain.

10. IANA Considerations

10.1. SRv6 Endpoint Behaviors

This I-D requests the IANA to allocate, within the "SRv6 Endpoint Behaviors" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry, the following allocations:

Value	Description	Reference
TBA1	End.AN - SR-aware function (native)	[This.ID]
TBA2	End.AS - Static proxy	[This.ID]
TBA3	End.AD - Dynamic proxy	[This.ID]
TBA4	End.AM - Masquerading proxy	[This.ID]

10.2. Segment Routing Header TLVs

This I-D requests the IANA to allocate, within the "Segment Routing Header TLVs" registry, the following allocations:

Value	Description	Reference
TBA1	Opaque Metadata TLV	[This.ID]
TBA2	NSH Carrier TLV	[This.ID]

11. Security Considerations

The security requirements and mechanisms described in [RFC8402], [I-D.ietf-6man-segment-routing-header] and [I-D.filsfils-spring-srv6-network-programming] also apply to this document.

This document does not introduce any new security vulnerabilities.

12. Acknowledgements

The authors would like to thank Thierry Couture, Ketan Talaulikar, Ioa Andersson, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein

and Joel M. Halpern for their valuable comments and suggestions on the document.

13. Contributors

P. Camarillo (Cisco), B. Peirens (Proximus), D. Steinberg (Steinberg Consulting), A. AbdelSalam (Gran Sasso Science Institute), G. Dawra (LinkedIn), S. Bryant (Huawei), H. Assarpour (Broadcom), H. Shah (Ciena), L. Contreras (Telefonica I+D), J. Tantsura (Individual), M. Vigoureux (Nokia) and J. Bhattacharya (Cisco) substantially contributed to the content of this document.

14. References

14.1. Normative References

- [I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-05 (work in progress), July 2018.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-14 (work in progress), June 2018.
- [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-14 (work in progress), June 2018.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-01 (work in progress), June 2018.
- [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>>.

14.2. Informative References

- [I-D.dawra-idr-bgp-sr-service-chaining]
Dawra, G., Filsfils, C., daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Clad, F., and K. Talaulikar, "BGP Control Plane Extensions for Segment Routing based Service Chaining", draft-dawra-idr-bgp-sr-service-chaining-02 (work in progress), January 2018.
- [I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-02 (work in progress), October 2018.
- [I-D.ietf-mpls-sfc]
Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", draft-ietf-mpls-sfc-03 (work in progress), October 2018.
- [I-D.xu-mpls-payload-protocol-identifier]
Xu, X., Assarpour, H., Ma, S., and F. Clad, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-05 (work in progress), August 2018.
- [IFIP18] Abdelsalam, A., Salsano, S., Clad, F., Camarillo, P., and C. Filsfils, "SEgment Routing Aware Firewall For Service Function Chaining scenarios", IFIP Networking conference , May 2018.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

Authors' Addresses

Francois Clad (editor)
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Xiaohu Xu (editor)
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Cheng Li
Huawei

Email: chengli13@huawei.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Shaowen Ma
Juniper

Email: mashaowen@gmail.com

Chaitanya Yadlapalli
AT&T
USA

Email: cy098d@att.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

SPRING
Internet-Draft
Intended status: Standards Track
Expires: October 25, 2019

F. Clad, Ed.
Cisco Systems, Inc.
X. Xu, Ed.
Alibaba
C. Filsfils
Cisco Systems, Inc.
D. Bernier
Bell Canada
C. Li
Huawei
B. Decraene
Orange
S. Ma
Juniper
C. Yadlapalli
AT&T
W. Henderickx
Nokia
S. Salsano
Universita di Roma "Tor Vergata"
April 23, 2019

Service Programming with Segment Routing
draft-xuclad-spring-sr-service-programming-02

Abstract

This document defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture.

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 October 25, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Classification and steering	4
4. Service segments	5
4.1. SR-aware services	5
4.2. SR-unaware services	6
5. SR service policies	7
5.1. SR-MPLS data plane	8
5.2. SRv6 data plane	10
6. SR proxy behaviors	11
6.1. Static SR proxy	14
6.1.1. SR-MPLS pseudocode	16
6.1.2. SRv6 pseudocode	17
6.2. Dynamic SR proxy	19
6.2.1. SR-MPLS pseudocode	19
6.2.2. SRv6 pseudocode	20
6.3. Shared memory SR proxy	21
6.4. Masquerading SR proxy	21
6.4.1. SRv6 masquerading proxy pseudocode	22
6.4.2. Variant 1: Destination NAT	23
6.4.3. Variant 2: Caching	23
7. Metadata	23
7.1. MPLS data plane	23
7.2. IPv6 data plane	24
7.2.1. SRH TLV objects	24
7.2.2. SRH tag	25
8. Implementation status	25
8.1. SR-aware services	26
8.2. Proxy behaviors	26
9. Related works	26
10. IANA Considerations	27

10.1. SRv6 Endpoint Behaviors	27
10.2. Segment Routing Header TLVs	27
11. Security Considerations	27
12. Acknowledgements	27
13. Contributors	28
14. References	28
14.1. Normative References	28
14.2. Informative References	29
Authors' Addresses	29

1. Introduction

Segment Routing (SR) is an architecture based on the source routing paradigm that seeks the right balance between distributed intelligence and centralized programmability. SR can be used with an MPLS or an IPv6 data plane to steer packets through an ordered list of instructions, called segments. These segments may encode simple routing instructions for forwarding packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of these services, running either on a physical appliance or in a virtual environment, are associated with a segment identifier (SID). These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined together in a SID-list to achieve service programming, but also with other types of segments as defined in [RFC8402]. SR thus provides a fully integrated solution for overlay, underlay and service programming. Furthermore, the IPv6 instantiation of SR (SRv6) supports metadata transportation in the Segment Routing header [I-D.ietf-6man-segment-routing-header], either natively in the tag field or with extensions such as TLVs.

This document describes how a service can be associated with a SID, including legacy services with no SR capabilities, and how these service SIDs are integrated within an SR policy. The definition of an SR Policy and the traffic steering mechanisms are covered in [I-D.ietf-spring-segment-routing-policy] and hence outside the scope of this document.

The definition of control plane components, such as service segment discovery, is outside the scope of this data plane document. For reference, the option of using BGP extensions to support SR service programming is proposed in [I-D.dawra-idr-bgp-sr-service-chaining].

2. Terminology

This document leverages the terminology proposed in [RFC8402] and [I-D.ietf-spring-segment-routing-policy]. It also introduces the following new terms.

Service segment: A segment associated with a service. The service may either run on a physical appliance or in a virtual environment such as a virtual machine or container.

SR-aware service: A service that is fully capable of processing SR traffic. An SR-aware service can be directly associated with a service segment.

SR-unaware service: A service that is unable to process SR traffic or may behave incorrectly due to presence of SR information in the packet headers. An SR-unaware service can be associated with a service segment through an SR proxy function.

3. Classification and steering

Classification and steering mechanisms are defined in section 8 of [I-D.ietf-spring-segment-routing-policy] and are independent from the purpose of the SR policy. From the perspective of a headend node classifying and steering traffic into an SR policy, there is no difference whether this policy contains IGP, BGP, peering, VPN or service segments, or any combination of these.

As documented in the above reference, traffic is classified when entering an SR domain. The SR policy headend may, depending on its capabilities, classify the packets on a per-destination basis, via simple FIB entries, or apply more complex policy routing rules requiring to look deeper into the packet. These rules are expected to support basic policy routing such as 5-tuple matching. In addition, the IPv6 SRH tag field defined in [I-D.ietf-6man-segment-routing-header] can be used to identify and classify packets sharing the same set of properties. Classified traffic is then steered into the appropriate SR policy and forwarded as per the SID-list(s) of the active candidate path.

SR traffic can be re-classified by an SR endpoint along the original SR policy (e.g., DPI service) or a transit node intercepting the traffic. This node is the head-end of a new SR policy that is imposed onto the packet, either as a stack of MPLS labels or as an IPv6 SRH.

4. Service segments

In the context of this document, the term service refers to a physical appliance running on dedicated hardware, a virtualized service inside an isolated environment such as a VM, container or namespace, or any process running on a compute element. A service may also comprise multiple sub-components running in different processes or containers. Unless otherwise stated, this document does not make any assumption on the type or execution environment of a service.

The execution of a service can be integrated as part of an SR policy by assigning a segment identifier, or SID, to the service and including this service SID in the SR policy SID-list. Such a service SID may be of local or global significance. In the former case, other segments, such as prefix or adjacency segments, can be used to steer the traffic up to the node where the service segment is instantiated. In the latter case, the service is directly reachable from anywhere in the routing domain. This is realized with SR-MPLS by assigning a SID from the global label block ([I-D.ietf-spring-segment-routing-mpls]), or with SRv6 by advertising the SID locator in the routing protocol ([I-D.filsfils-spring-srv6-network-programming]). It is up to the network operator to define the scope and reachability of each service SID. This decision can be based on various considerations such as infrastructure dynamicity, available control plane or orchestration system capabilities.

This document categorizes services in two types, depending on whether they are able to behave properly in the presence of SR information or not. These are respectively named SR-aware and SR-unaware services.

4.1. SR-aware services

An SR-aware service can process the SR information in the packets it receives. This means being able to identify the active segment as a local instruction and move forward in the segment list, but also that the service's own behavior is not hindered due to the presence of SR information. For example, an SR-aware firewall filtering SRv6 traffic based on its final destination must retrieve that information from the last entry in the SRH rather than the Destination Address field of the IPv6 header.

An SR-aware service is associated with a locally instantiated service segment, which is used to steer traffic through it.

If the service is configured to intercept all the packets passing through the appliance, the underlying routing system only has to

implement a default SR endpoint behavior (SR-MPLS node segment or SRv6 End function), and the corresponding SID will be used to steer traffic through the service.

If the service requires the packets to be directed to a specific virtual interface, networking queue or process, a dedicated SR behavior may be required to steer the packets to the appropriate location. The definition of such service-specific functions is out of the scope of this document.

SR-aware services also enable advanced network programming functionalities such as conditional branching and jumping to arbitrary SIDs in the segment list. In addition, SRv6 provides several ways of passing and exchanging information between services (e.g., SID arguments, tag field and TLVs). An example scenario involving these features is described in [IFIP18], which discusses the implementation of an SR-aware Intrusion Detection System.

Examples of SR-aware services are provided in section Section 8.1.

4.2. SR-unaware services

Any service that does not meet the above criteria for SR-awareness is considered as SR-unaware.

An SR-unaware service is not able to process the SR information in the traffic that it receives. It may either drop the traffic or take erroneous decisions due to the unrecognized routing information. In order to include such services in an SR policy, it is thus required to remove the SR information as well as any other encapsulation header before the service receives the packet, or to alter it in such a way that the service can correctly process the packet.

In this document, we define the concept of an SR proxy as an entity, separate from the service, that performs these modifications and handle the SR processing on behalf of a service. The SR proxy can run as a separate process on the service appliance, on a virtual switch or router on the compute node or on a different host.

An SR-unaware service is associated with a service segment instantiated on the SR proxy, which is used to steer traffic through the service. Section 6 describes several SR proxy behaviors to handle the encapsulation headers and SR information under various circumstances.

5. SR service policies

An SR service policy is an SR policy, as defined in [I-D.ietf-spring-segment-routing-policy], that includes at least one service. This service is represented in the SID-list by its associated service SID. In case the policy should include several services, the service traversal order is indicated by the relative position of each service SID in the SID-list. Using the mechanisms described in [I-D.ietf-spring-segment-routing-policy], it is possible to load balance the traffic over several services, or instances of the same service, by associating with the SR service policy a weighted set of SID-lists, each containing a possible sequence of service SIDs to be traversed. Similarly, several candidate paths can be specified for the SR service policy, each with its own set of SID-lists, for resiliency purposes.

Furthermore, binding SIDs (BSIDs) can be leveraged in the context of service policies to reduce the number of SIDs imposed by the headend, provide opacity between domains and improve scalability, as described in [I-D.filsfils-spring-sr-policy-considerations]. For example, a network operator may want a policy in its core domain to include services that are running in one of its datacenters. One option is to define an SR policy at ingress edge of the core domain that explicitly includes all the SIDs needed to steer the traffic through the core and in the DC, but that may result in a long SID-list and requires to update the ingress edge configuration every time the DC part of the policy is modified. Alternatively, a separate policy can be defined at the ingress edge of the datacenter with only the SIDs that needs to be executed there and its BSID included in the core domain policy. That BSID remains stable when the DC policy is modified and can even be shared among several core domain policies that would require the same type of processing in the DC.

This section describes how services can be integrated within an SR-MPLS or SRv6 service policy.

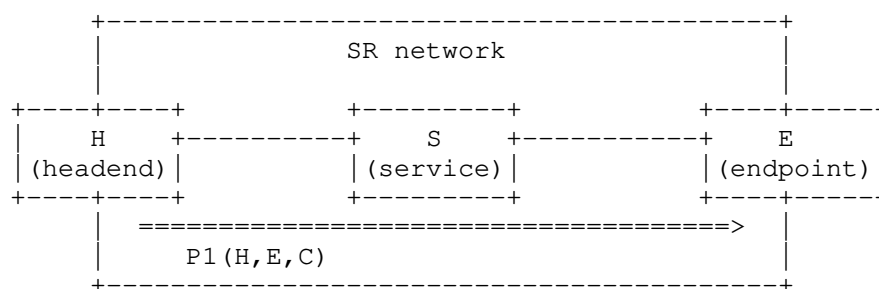


Figure 1: SR service policy

Figure 1 illustrates a basic SR service policy instantiated on a headend node H towards an endpoint E and traversing a service S. The SR policy may also include additional requirements, such as traffic engineering or VPN. On the head-end H, the SR policy P1 is created with a color C and endpoint E and associated with an SR path that can either be explicitly configured, dynamically computed on H or provisioned by a network controller.

In its most basic form, the SR policy P1 would be resolved into the SID-list $\langle \text{SID}(S), \text{SID}(E) \rangle$. This is assuming that $\text{SID}(S)$ and $\text{SID}(E)$ are directly reachable from H and S, respectively, and that the forwarding path meets the policy requirement. However, depending on the dataplane and the segments available in the network, additional SIDs may be required to enforce the SR policy.

This model applies regardless of the SR-awareness of the service. If it is SR-unaware, then S simply represents the proxy that takes care of transmitting the packet to the actual service.

Traffic can then be steered into this policy using any of the mechanisms described in [I-D.ietf-spring-segment-routing-policy].

The following subsections describe the specificities of each SR dataplane.

5.1. SR-MPLS data plane

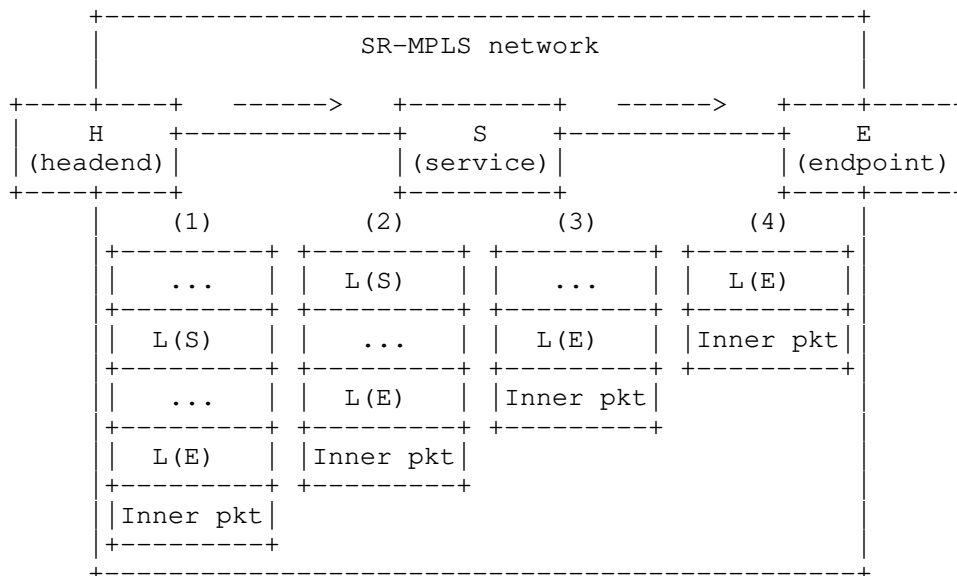


Figure 2: Packet walk in an SR-MPLS network

In an SR-MPLS network, the SR policy SID-list is encoded as a stack of MPLS labels[I-D.ietf-spring-segment-routing-mpls] and pushed on top of the packet.

In the example shown on Figure 2, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an MPLS label stack that includes at least a label L(S) associated to service S and a label L(E) associated to the endpoint E. The label stack may also include additional intermediate segments if these are required for traffic engineering (e.g., to encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID L(S) may be taken from the global or local SID block of node S and, in the latter case, one or more SIDs might be needed before L(S) in order for the packet to reach node S (e.g., a prefix-SID of S), where L(S) can be interpreted. The same applies for the segment L(E) at the SR policy endpoint.

Special consideration must be taken into account when using Local SIDs for service identification due to increased label stack depth and the associated impacts.

When the packet arrives at S, this node determines the MPLS payload type and the appropriate behavior for processing the packet based on the semantic locally associated to the top label L(S). If S is an

SR-aware service, the SID L(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, L(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, L(S) is also popped from the label stack in order to expose the next SID, which may be L(E) or another intermediate segment.

5.2. SRv6 data plane

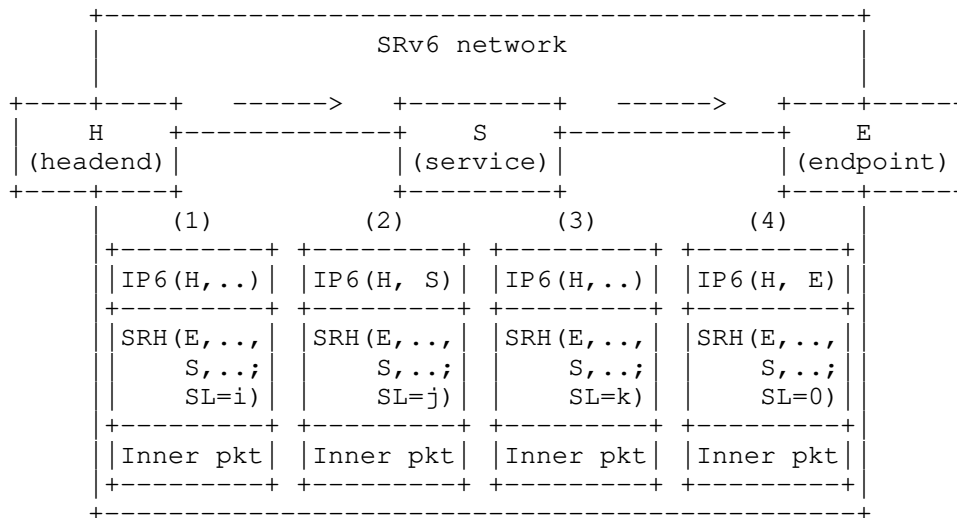


Figure 3: Packet walk in an SRv6 network

In an SRv6 network, the SR Policy is encoded into the packet as an IPv6 header possibly followed by a Segment Routing header (SRH) [I-D.ietf-6man-segment-routing-header].

In the example shown on Figure 3, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an SRH that includes at least a segment SID(S) to the service, or service proxy, S and a segment SID(E) to the endpoint E. The SRH may also include additional intermediate segments if these are required for traffic engineering (e.g., the encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service segment locator may or may not be advertised in the routing protocol and, in the latter case, one or more SIDs might be needed before SID(S) in order to bring the packet up to node S, where SID(S) can be interpreted. The same applies for the segment SID(E) at the SR policy endpoint.

When the packet arrives at S, this node determines how to process the packet based on the semantic locally associated to the active segment SID(S). If S is an SR-aware service, then SID(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, SID(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, the SRv6 End function is also applied in order to make the next SID, which may be SID(E) or another intermediate segment, active.

The "Inner pkt" on Figure 3 represents the SRv6 payload, which may be an encapsulated IP packet, an Ethernet frame or a transport-layer payload, for example.

6. SR proxy behaviors

This section describes several SR proxy behaviors designed to enable SR service programming through SR-unaware services. A system implementing one of these functions may handle the SR processing on behalf of an SR-unaware service and allows the service to properly process the traffic that is steered through it.

A service may be located at any hop in an SR policy, including the last segment. However, the SR proxy behaviors defined in this section are dedicated to supporting SR-unaware services at intermediate hops in the segment list. In case an SR-unaware service is at the last segment, it is sufficient to ensure that the SR information is ignored (IPv6 routing extension header with Segments Left equal to 0) or removed before the packet reaches the service (MPLS PHP, SRv6 End.D or PSP).

As illustrated on Figure 4, the generic behavior of an SR proxy has two parts. The first part is in charge of passing traffic from the network to the service. It intercepts the SR traffic destined for the service via a locally instantiated service segment, modifies it in such a way that it appears as non-SR traffic to the service, then sends it out on a given interface, IFACE-OUT, connected to the service. The second part receives the traffic coming back from the service on IFACE-IN, restores the SR information and forwards it according to the next segment in the list. IFACE-OUT and IFACE-IN are respectively the proxy interface used for sending traffic to the service and the proxy interface that receives the traffic coming back from the service. These can be physical interfaces or sub-interfaces (VLANs) and, unless otherwise stated, IFACE-OUT and IFACE-IN can represent the same interface.

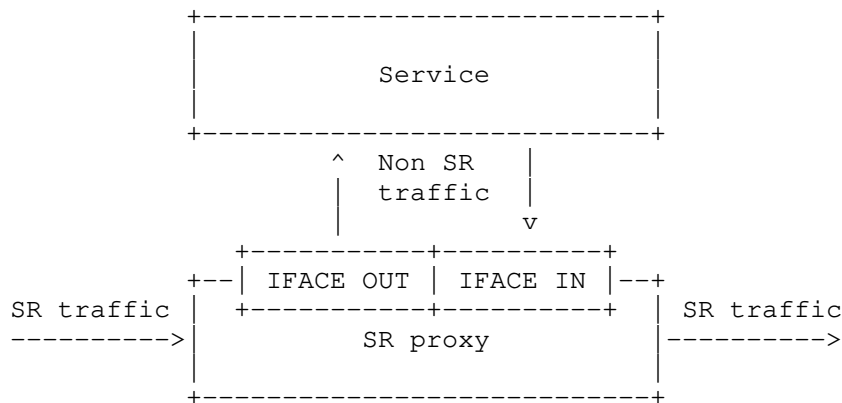


Figure 4: Generic SR proxy

In the next subsections, the following SR proxy mechanisms are defined:

- o Static proxy
- o Dynamic proxy
- o Shared-memory proxy
- o Masquerading proxy

Each mechanism has its own characteristics and constraints, which are summarized in the below table. It is up to the operator to select the best one based on the proxy node capabilities, the service behavior and the traffic type. It is also possible to use different proxy mechanisms within the same service policy.

		S t a t i c	D y n a m i c	S h a r e d m e m .	M a s q u e r a d i n g
SR flavors	SR-MPLS	Y	Y	Y	-
	SRv6 insertion	P	P	P	Y
	SRv6 encapsulation	Y	Y	Y	-
Chain agnostic configuration		N	N	Y	Y
Transparent to chain changes		N	Y	Y	Y
Service support	DA modification	Y	Y	Y	NAT
	Payload modification	Y	Y	Y	Y
	Packet generation	Y	Y	cache	cache
	Packet deletion	Y	Y	Y	Y
	Transport endpoint	Y	Y	cache	cache
Supported traffic	Ethernet	Y	Y	Y	-
	IPv4	Y	Y	Y	-
	IPv6	Y	Y	Y	Y

Figure 5: SR proxy summary

Note: The use of a shared memory proxy requires both the service (VNF) and the proxy to be running on the same node.

6.1. Static SR proxy

The static proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an MPLS label stack or an IPv6 header on top of an inner packet, which can be Ethernet, IPv4 or IPv6.

A static SR proxy segment is associated with the following mandatory parameters

- o INNER-TYPE: Inner packet type
- o NH-ADDR: Next hop Ethernet address (only for inner type IPv4 and IPv6)
- o IFACE-OUT: Local interface for sending traffic towards the service
- o IFACE-IN: Local interface receiving the traffic coming back from the service
- o CACHE: SR information to be attached on the traffic coming back from the service, including at least
 - * CACHE.SA: IPv6 source address (SRv6 only)
 - * CACHE.LIST: Segment list expressed as MPLS labels or IPv6 address

A static SR proxy segment is thus defined for a specific service, inner packet type and cached SR information. It is also bound to a pair of directed interfaces on the proxy. These may be both directions of a single interface, or opposite directions of two different interfaces. The latter is recommended in case the service is to be used as part of a bi-directional SR SC policy. If the proxy and the service both support 802.1Q, IFACE-OUT and IFACE-IN can also represent sub-interfaces.

The first part of this behavior is triggered when the proxy node receives a packet whose active segment matches a segment associated with the static proxy behavior. It removes the SR information from the packet then sends it on a specific interface towards the associated service. This SR information corresponds to the full label stack for SR-MPLS or to the encapsulation IPv6 header with any attached extension header in the case of SRv6.

The second part is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This

policy attaches to the incoming traffic the cached SR information associated with the SR proxy segment. If the proxy segment uses the SR-MPLS data plane, CACHE contains a stack of labels to be pushed on top of the packets. With the SRv6 data plane, CACHE is defined as a source address, an active segment and an optional SRH (tag, segments left, segment list and metadata). The proxy encapsulates the packets with an IPv6 header that has the source address, the active segment as destination address and the SRH as a routing extension header. After the SR information has been attached, the packets are forwarded according to the active segment, which is represented by the top MPLS label or the IPv6 Destination Address. An MPLS TTL or IPv6 Hop Limit value may also be configured in CACHE. If it is not, the proxy should set these values according to the node's default setting for MPLS or IPv6 encapsulation.

In this scenario, there are no restrictions on the operations that can be performed by the service on the stream of packets. It may operate at all protocol layers, terminate transport layer connections, generate new packets and initiate transport layer connections. This behavior may also be used to integrate an IPv4-only service into an SRv6 policy. However, a static SR proxy segment can be used in only one service policy at a time. As opposed to most other segment types, a static SR proxy segment is bound to a unique list of segments, which represents a directed SR SC policy. This is due to the cached SR information being defined in the segment configuration. This limitation only prevents multiple segment lists from using the same static SR proxy segment at the same time, but a single segment list can be shared by any number of traffic flows. Besides, since the returning traffic from the service is re-classified based on the incoming interface, an interface can be used as receiving interface (IFACE-IN) only for a single SR proxy segment at a time. In the case of a bi-directional SR SC policy, a different SR proxy segment and receiving interface are required for the return direction.

The static proxy behavior may also be used for sending traffic through "bump in the wire" services that are transparent to the IP and Ethernet layers. This type of processing is assumed when the inner traffic type is Ethernet, since the original destination address of the Ethernet frame is preserved when the packet is steered into the SR Policy and likely associated with a node downstream of the policy tail-end. In case the inner type is IP (IPv4 or IPv6), the NH-ADDR parameter may be set to a dummy or broadcast Ethernet address, or simply to the address of the proxy receiving interface (IFACE-IN).

6.1.1.1. SR-MPLS pseudocode

6.1.1.1.1. Static proxy for inner type Ethernet

Upon receiving an MPLS packet with top label L, where L is an MPLS L2 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is Ethernet THEN
3. Forward the exposed frame on IFACE-OUT
4. ELSE
5. Drop the packet

Upon receiving on IFACE-IN an Ethernet frame with a destination address different than the interface address, a node N does:

1. Push labels in CACHE on top of the frame Ethernet header
2. Lookup the top label and proceed accordingly

The receiving interface must be configured in promiscuous mode in order to accept those Ethernet frames.

6.1.1.1.2. Static proxy for inner type IPv4

Upon receiving an MPLS packet with top label L, where L is an MPLS IPv4 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is IPv4 THEN
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Upon receiving a non-link-local IPv4 packet on IFACE-IN, a node N does:

1. Decrement TTL and update checksum
2. Push labels in CACHE on top of the packet IPv4 header
3. Lookup the top label and proceed accordingly

6.1.1.1.3. Static proxy for inner type IPv6

Upon receiving an MPLS packet with top label L, where L is an MPLS IPv6 static proxy segment, a node N does:

1. Pop all labels
2. IF payload type is IPv6 THEN
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N does:

1. Decrement Hop Limit
2. Push labels in CACHE on top of the packet IPv6 header
3. Lookup the top label and proceed accordingly

6.1.2. SRv6 pseudocode

6.1.2.1. Static proxy for inner type Ethernet

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for Ethernet traffic, a node N does:

1. IF ENH == 59 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed frame on IFACE-OUT
4. ELSE
5. Drop the packet

Ref1: 59 refers to "no next header" as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving on IFACE-IN an Ethernet frame with a destination address different than the interface address, a node N does:

1. Retrieve CACHE entry matching IFACE-IN and traffic type
2. Push SRH with CACHE.LIST on top of the Ethernet header ;; Ref2
3. Push IPv6 header with

SA = CACHE.SA
 DA = CACHE.LIST[0] ;; Ref3
 Next Header = 43 ;; Ref4
4. Set outer payload length and flow label
5. Lookup outer DA in appropriate table and proceed accordingly

Ref2: Unless otherwise specified, the segments in CACHE.LIST should be encoded in reversed order, Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1, and Next Header should be set to 59.

Ref3: CACHE.LIST[0] represents the first IPv6 SID in CACHE.LIST.

Ref4: If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header value must be set to 59.

The receiving interface must be configured in promiscuous mode in order to accept those Ethernet frames.

6.1.2.2. Static proxy for inner type IPv4

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for IPv4 traffic, a node N does:

1. IF ENH == 4 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Ref1: 4 refers to IPv4 encapsulation as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving a non-link-local IPv4 packet on IFACE-IN, a node N does:

1. Decrement TTL and update checksum
2. IF CACHE.SRH THEN ;; Ref2
3. Push CACHE.SRH on top of the existing IPv4 header
4. Set NH value of the pushed SRH to 4
5. Push outer IPv6 header with SA, DA and traffic class from CACHE
6. Set outer payload length and flow label
7. Set NH value to 43 if an SRH was added, or 4 otherwise
8. Lookup outer DA in appropriate table and proceed accordingly

Ref2: CACHE.SRH represents the SRH defined in CACHE, if any, for the static SR proxy segment associated with IFACE-IN.

6.1.2.3. Static proxy for inner type IPv6

Upon receiving an IPv6 packet destined for S, where S is an IPv6 static proxy segment for IPv6 traffic, a node N does:

1. IF ENH == 41 THEN ;; Ref1
2. Remove the (outer) IPv6 header and its extension headers
3. Forward the exposed packet on IFACE-OUT towards NH-ADDR
4. ELSE
5. Drop the packet

Ref1: 41 refers to IPv6 encapsulation as defined by IANA allocation for Internet Protocol Numbers.

Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N does:

1. Decrement Hop Limit
2. IF CACHE.SRH THEN ;; Ref2
3. Push CACHE.SRH on top of the existing IPv6 header
4. Set NH value of the pushed SRH to 41
5. Push outer IPv6 header with SA, DA and traffic class from CACHE
6. Set outer payload length and flow label
7. Set NH value to 43 if an SRH was added, or 41 otherwise
8. Lookup outer DA in appropriate table and proceed accordingly

Ref2: CACHE.SRH represents the SRH defined in CACHE, if any, for the static SR proxy segment associated with IFACE-IN.

6.2. Dynamic SR proxy

The dynamic proxy is an improvement over the static proxy that dynamically learns the SR information before removing it from the incoming traffic. The same information can then be re-attached to the traffic returning from the service. As opposed to the static SR proxy, no CACHE information needs to be configured. Instead, the dynamic SR proxy relies on a local caching mechanism on the node instantiating this segment.

Upon receiving a packet whose active segment matches a dynamic SR proxy function, the proxy node pops the top MPLS label or applies the SRv6 End behavior, then compares the updated SR information with the cache entry for the current segment. If the cache is empty or different, it is updated with the new SR information. The SR information is then removed and the inner packet is sent towards the service.

The cache entry is not mapped to any particular packet, but instead to an SR SC policy identified by the receiving interface (IFACE-IN). Any non-link-local IP packet or non-local Ethernet frame received on that interface will be re-encapsulated with the cached headers as described in Section 6.1. The service may thus drop, modify or generate new packets without affecting the proxy.

6.2.1. SR-MPLS pseudocode

The dynamic proxy SR-MPLS pseudocode is obtained by inserting the following instructions at the beginning of the static SR-MPLS pseudocode (Section 6.1.1).

```
1.  IF top label S bit is 0 THEN                                ;; Ref1
2.      Pop top label
3.      IF C(IFACE-IN) different from remaining labels THEN    ;; Ref2
4.          Copy all remaining labels into C(IFACE-IN)         ;; Ref3
5.  ELSE
6.      Drop the packet
```

Ref1: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

Ref2: A TTL margin can be configured for the top label stack entry to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a TTL difference smaller than the configured margin should not trigger a cache update (provided that the labels are the same).

Ref3: C(IFACE-IN) represents the cache entry associated to the dynamic SR proxy segment. It is identified with IFACE-IN in order to efficiently retrieve the right SR information when a packet arrives on this interface.

In addition, the inbound policy should check that C(IFACE-IN) has been defined before attempting to restore the MPLS label stack and drop the packet otherwise.

6.2.2. SRv6 pseudocode

The dynamic proxy SRv6 pseudocode is obtained by inserting the following instructions between lines 1 and 2 of the static proxy SRv6 pseudocode.

```
1.  IF NH=SRH & SL > 0 THEN                                    ;; Ref1
2.      Decrement SL and update the IPv6 DA with SRH[SL]
3.      IF C(IFACE-IN) different from IPv6 encaps THEN        ;; Ref2
4.          Copy the IPv6 encaps into C(IFACE-IN)             ;; Ref3
5.  ELSE
6.      Drop the packet
```

Ref1: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

Ref2: "IPv6 encaps" represents the IPv6 header and any attached extension header.

Ref3: C(IFACE-IN) represents the cache entry associated to the dynamic SR proxy segment. It is identified with IFACE-IN in order to efficiently retrieve the right SR information when a packet arrives on this interface.

In addition, the inbound policy should check that C(IFACE-IN) has been defined before attempting to restore the IPv6 encapsulation and drop the packet otherwise.

6.3. Shared memory SR proxy

The shared memory proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy behavior leverages a shared-memory interface with a virtualized service (VNF) in order to hide the SR information from an SR-unaware service while keeping it attached to the packet. We assume in this case that the proxy and the VNF are running on the same compute node. A typical scenario is an SR-capable vrouter running on a container host and forwarding traffic to VNFs isolated within their respective container.

More details will be added in a future revision of this document.

6.4. Masquerading SR proxy

The masquerading proxy is an SR endpoint behavior for processing SRv6 traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an IPv6 header and an SRH on top of an inner payload. The masquerading behavior is independent from the inner payload type. Hence, the inner payload can be of any type but it is usually expected to be a transport layer packet, such as TCP or UDP.

A masquerading SR proxy segment is associated with the following mandatory parameters:

- o S-ADDR: Ethernet or IPv6 address of the service
- o IFACE-OUT: Local interface for sending traffic towards the service
- o IFACE-IN: Local interface receiving the traffic coming back from the service

A masquerading SR proxy segment is thus defined for a specific service and bound to a pair of directed interfaces or sub-interfaces on the proxy. As opposed to the static and dynamic SR proxies, a masquerading segment can be present at the same time in any number of SR SC policies and the same interfaces can be bound to multiple masquerading proxy segments. The only restriction is that a masquerading proxy segment cannot be the last segment in an SR SC policy.

The first part of the masquerading behavior is triggered when the proxy node receives an IPv6 packet whose Destination Address matches a masquerading proxy segment. The proxy inspects the IPv6 extension headers and substitutes the Destination Address with the last segment in the SRH attached to the IPv6 header, which represents the final destination of the IPv6 packet. The packet is then sent out towards the service.

The service receives an IPv6 packet whose source and destination addresses are respectively the original source and final destination. It does not attempt to inspect the SRH, as RFC8200 specifies that routing extension headers are not examined or processed by transit nodes. Instead, the service simply forwards the packet based on its current Destination Address. In this scenario, we assume that the service can only inspect, drop or perform limited changes to the packets. For example, Intrusion Detection Systems, Deep Packet Inspectors and non-NAT Firewalls are among the services that can be supported by a masquerading SR proxy. Variants of the masquerading behavior are defined in Section 6.4.2 and Section 6.4.3 to support a wider range of services.

The second part of the masquerading behavior, also called de-masquerading, is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This policy inspects the incoming traffic and triggers a regular SRv6 endpoint processing (End) on any IPv6 packet that contains an SRH. This processing occurs before any lookup on the packet Destination Address is performed and it is sufficient to restore the right active segment as the Destination Address of the IPv6 packet.

6.4.1. SRv6 masquerading proxy pseudocode

Masquerading: Upon receiving a packet destined for S, where S is an IPv6 masquerading proxy segment, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Update the IPv6 DA with SRH[0]
3. Forward the packet on IFACE-OUT
4. ELSE
5. Drop the packet

De-masquerading: Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Decrement SL
3. Update the IPv6 DA with SRH[SL] ;;; Ref1
4. Lookup DA in appropriate table and proceed accordingly

Ref1: This pseudocode can be augmented to support the Penultimate Segment Popping (PSP) endpoint flavor. The exact pseudocode modification are provided in [I-D.filsfils-spring-srv6-network-programming].

6.4.2. Variant 1: Destination NAT

Services modifying the destination address in the packets they process, such as NATs, can be supported by a masquerading proxy with the following modification to the de-masquerading pseudocode.

De-masquerading - NAT: Upon receiving a non-link-local IPv6 packet on IFACE-IN, a node N processes it as follows.

1. IF NH=SRH & SL > 0 THEN
2. Update SRH[0] with the IPv6 DA
3. Decrement SL
4. Update the IPv6 DA with SRH[SL]
5. Lookup DA in appropriate table and proceed accordingly

6.4.3. Variant 2: Caching

Services generating packets or acting as endpoints for transport connections can be supported by adding a dynamic caching mechanism similar to the one described in Section 6.2.

More details will be added in a future revision of this document.

7. Metadata

7.1. MPLS data plane

Metadata can be carried for SR-MPLS traffic in a Segment Routing header inserted between the last MPLS label and the MPLS payload. When used solely as a metadata container, the SRH does not carry any segment but only the mandatory header fields, including the tag and flags, and any TLVs that is required for transporting the metadata.

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata in an MPLS packet is a potential issue to be addressed. One possible solution is to add the indication about the presence of metadata in the semantic of the SIDs. Note that only the SIDs whose behavior involves looking at the metadata or the MPLS payload would need to include such semantic (e.g., service segments). Other segments, such as traffic engineering segments, are not affected by the presence of metadata. Another, more generic, solution is to introduce a protocol identifier

field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

7.2. IPv6 data plane

7.2.1. SRH TLV objects

The IPv6 SRH TLV objects are designed to carry all sorts of metadata. In particular, the NSH carrier TLV is defined as a container for NSH metadata.

TLV objects can be imposed by the ingress edge router that steers the traffic into the SR SC policy.

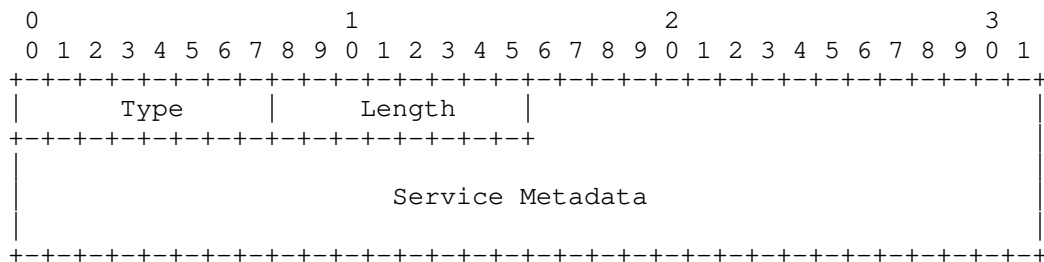
An SR-aware service may impose, modify or remove any TLV object attached to the first SRH, either by directly modifying the packet headers or via a control channel between the service and its forwarding plane.

An SR-aware service that re-classifies the traffic and steers it into a new SR SC policy (e.g. DPI) may attach any TLV object to the new SRH.

Metadata imposition and handling will be further discussed in a future version of this document.

7.2.1.1. Opaque Metadata TLV

This document defines an SRv6 TLV called Opaque Metadata TLV. This is a fixed-length container to carry any type of Service Metadata. No assumption is made by this document on the structure or the content of the carried metadata. The Opaque Metadata TLV has the following format:



where:

- o Type: to be assigned by IANA.

- o Length: 14.
- o Service Metadata: 14 octets of opaque data.

7.2.1.2. NSH Carrier TLV

This document defines an SRv6 TLV called NSH Carrier TLV. It is a container to carry Service Metadata in the form of Variable-Length Metadata as defined in [RFC8300] for NSH MD Type 2. The NSH Carrier TLV has the following format:

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |      Flags      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                Service Metadata                                //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

- o Type: to be assigned by IANA.
- o Length: the total length of the TLV.
- o Flags: 8 bits. No flags are defined in this document. SHOULD be set to 0 on transmission and MUST be ignored on receipt.
- o Service Metadata: a list of Service Metadata TLV as defined in [RFC8300] for NSH MD Type 2.

7.2.2. SRH tag

The SRH tag identifies a packet as part of a group or class of packets [I-D.ietf-6man-segment-routing-header].

In the context of service programming, this field can be used to encode basic metadata in the SRH. An example use case would be to leverage the SRH tag to encode a policy ID which could be leveraged in an SR-aware function to determine which processing policy to apply rather than having doing local classification or leverage alternate encapsulations.

8. Implementation status

This section is to be removed prior to publishing as an RFC.

8.1. SR-aware services

Specific SRv6 support has been implemented for the below open-source services:

- o Iptables (1.6.2 and later)
- o Nftables (0.8.4 and later)
- o Snort

In addition, any service relying on the Linux kernel, version 4.10 and later, or FD.io VPP for packet forwarding can be considered as SR-aware.

8.2. Proxy behaviors

The static SR proxy is available for SR-MPLS and SRv6 on various Cisco hardware and software platforms. Furthermore, the following proxies are available on open-source software.

		VPP	Linux
M P L S	Static proxy	Available	In progress
	Dynamic proxy	In progress	In progress
	Shared memory proxy	In progress	In progress
S R v 6	Static proxy	Available	In progress
	Dynamic proxy	Available	Available
	Shared memory proxy	In progress	In progress
	Masquerading proxy	Available	Available

Figure 6: Open-source implementation status table

9. Related works

The Segment Routing solution addresses a wide problem that covers both topological and service policies. The topological and service instructions can be either deployed in isolation or in combination. SR has thus a wider applicability than the architecture defined in [RFC7665]. Furthermore, the inherent property of SR is a stateless

network fabric. In SR, there is no state within the fabric to recognize a flow and associate it with a policy. State is only present at the ingress edge of the SR domain, where the policy is encoded into the packets. This is completely different from other proposals such as [RFC8300] and the MPLS label swapping mechanism described in [I-D.ietf-mpls-sfc], which rely on state configured at every hop of the service chain.

10. IANA Considerations

10.1. SRv6 Endpoint Behaviors

This I-D requests the IANA to allocate, within the "SRv6 Endpoint Behaviors" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry, the following allocations:

Value	Description	Reference
TBA1	End.AN - SR-aware function (native)	[This.ID]
TBA2	End.AS - Static proxy	[This.ID]
TBA3	End.AD - Dynamic proxy	[This.ID]
TBA4	End.AM - Masquerading proxy	[This.ID]

10.2. Segment Routing Header TLVs

This I-D requests the IANA to allocate, within the "Segment Routing Header TLVs" registry, the following allocations:

Value	Description	Reference
TBA1	Opaque Metadata TLV	[This.ID]
TBA2	NSH Carrier TLV	[This.ID]

11. Security Considerations

The security requirements and mechanisms described in [RFC8402], [I-D.ietf-6man-segment-routing-header] and [I-D.filsfils-spring-srv6-network-programming] also apply to this document.

This document does not introduce any new security vulnerabilities.

12. Acknowledgements

The authors would like to thank Thierry Couture, Ketan Talaulikar, Ioa Andersson, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein

and Joel M. Halpern for their valuable comments and suggestions on the document.

13. Contributors

P. Camarillo (Cisco), B. Peirens (Proximus), D. Steinberg (Steinberg Consulting), A. AbdelSalam (Gran Sasso Science Institute), G. Dawra (LinkedIn), S. Bryant (Huawei), H. Assarpour (Broadcom), H. Shah (Ciena), L. Contreras (Telefonica I+D), J. Tantsura (Individual), M. Vigoureux (Nokia) and J. Bhattacharya (Cisco) substantially contributed to the content of this document.

14. References

14.1. Normative References

- [I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-07 (work in progress), February 2019.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-18 (work in progress), April 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-19 (work in progress), March 2019.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-02 (work in progress), October 2018.
- [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>>.

14.2. Informative References

- [I-D.dawra-idr-bgp-sr-service-chaining]
Dawra, G., Filsfils, C., daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Clad, F., and K. Talaulikar, "BGP Control Plane Extensions for Segment Routing based Service Chaining", draft-dawra-idr-bgp-sr-service-chaining-02 (work in progress), January 2018.
- [I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-03 (work in progress), April 2019.
- [I-D.ietf-mpls-sfc]
Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", draft-ietf-mpls-sfc-07 (work in progress), March 2019.
- [I-D.xu-mpls-payload-protocol-identifier]
Xu, X., Assarpour, H., Ma, S., and F. Clad, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-06 (work in progress), March 2019.
- [IFIP18] Abdelsalam, A., Salsano, S., Clad, F., Camarillo, P., and C. Filsfils, "SEgment Routing Aware Firewall For Service Function Chaining scenarios", IFIP Networking conference , May 2018.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

Authors' Addresses

Francois Clad (editor)
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Xiaohu Xu (editor)
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Cheng Li
Huawei

Email: chengli13@huawei.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Shaowen Ma
Juniper

Email: mashaowen@gmail.com

Chaitanya Yadlapalli
AT&T
USA

Email: cy098d@att.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it