Authors: R. Gandhi, Ed.   C. Filsfils
         Cisco Systems, Inc.   Cisco Systems, Inc.
         N. Vaghamshi   M. Nagarajah   R. Foote   M. Chen
         Reliance   Telstra   Nokia   Huawei
         A. Dhamija
         Rakuten

Enhanced Performance Measurement Using Simple TWAMP in Segment Routing Networks

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 defines procedure for Enhanced
Performance Measurement of end-to-end SR paths including SR Policies
for both SR-MPLS and SRv6 data planes using Simple Two-Way Active
Measurement Protocol (STAMP) defined in RFC 8762. The procedure
allows to improve the scale for number of sessions and failure
detection interval.

Status of This Memo

Copyright Notice

Table of Contents

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 [RFC8402]. SR Policies as defined
   in [RFC9256] are used to steer traffic through a specific, user-
   defined paths using a stack of Segments. A comprehensive SR
   Performance Measurement (PM) for delay and packet loss as well as
   Connectivity Verification (CV) is one of the essential requirements

to measure network performance to provide Service Level Agreements (SLAs).

The Simple Two-Way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762] without the use of a control channel to pre-signal session parameters. As described in [I-D.ietf-spring-stamp-srpm], STAMP can be used for performance measurement of one-way, two-way or round-trip delay and packet loss of end-to-end SR paths.

STAMP requires RFC8762 protocol support on the Session-Reflector to process the received test packets, and hence the received test packets need to be punted from the forwarding fast path and return test packets need to be generated. This limits the scale for number test sessions and the ability to provide faster detection interval. The loopback measurement mode defined in [I-D.ietf-spring-stamp-srpm] does not require STAMP test packet processing on Session-Reflector, however, it does not provide accurate one-way delay.

This document defines procedure for Enhanced Performance Measurement of end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, using Simple Two-Way Active Measurement Protocol (STAMP) defined in [RFC8762]. The procedure allows to improve the scale for number of sessions and failure detection interval.

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.

ECMP: Equal Cost Multi-Path.

EB: Endpoint Behaviour.

HMAC: Hashed Message Authentication Code.

MBZ: Must be Zero.

MPLS: Multiprotocol Label Switching.

PM: Performance Measurement.

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.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

TTL: Time To Live.

2.3. Reference Topology

In the reference topology shown in Figure 1, the STAMP Session-
Sender [RFC8762] S1 initiates a Session-Sender test packet and the
Session-Reflector R1 returns the test packet. The return test packet
may be transmitted back to the Session-Sender S1 on the same path
(same set of links and nodes) or a different path in the reverse
direction from the path taken towards the Session-Reflector R1.

The Session-Sender S1 and Session-Reflector R1 are connected via an
SR path [RFC8402]. The SR path can be an SR Policy [RFC9256] on node
S1 (called head-end) with destination to node R1 (called tail-end).

```
                 T1                      T2
                 /                        \
       +-------+   STAMP Test Packet   +-------+
       |       | - - - - - - - - - - - |       |
       |   S1  |=====================||   R1   |
       |       |<- - - - - - - - - - - |       |
       +-------+   Return Test Packet  +-------+
                 \
                  T4

         Session-Sender              Session-Reflector
                                       (Timestamp,
                                        Pop and Forward)
```
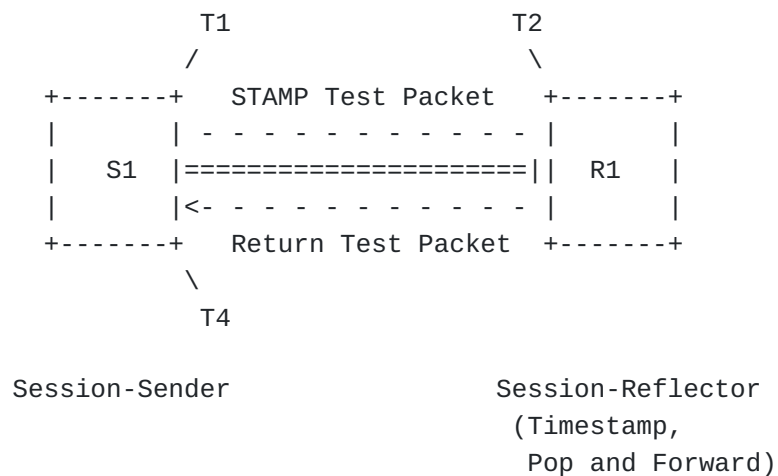
 Figure 1: Loopback Mode Enabled with Network Programming Function

3. Overview

3.1. Enhanced Loopback Mode Enabled with Network Programming Function

As described in [I-D.ietf-spring-stamp-srpm], in loopback mode, the STAMP Session-Sender S1 initiates Session-Sender test packets and the Session-Reflector R1 forwards them back to the Session-Sender S1. The received STAMP test packets are not punted out of the fast path in forwarding at the Session-Reflector. At the Session-Reflector, the loopback function simply makes the necessary changes to the encapsulation including IP and UDP headers to return the STAMP test packet to the Session-Sender S1. No STAMP test session is created on the Session-Reflector R1. As described in [I-D.ietf-spring-stamp-srpm], only round-trip delay can be measured in the loopback mode. In SR networks, there is also a need to measure one-way delay to provide low latency services.

This document defines a new STAMP measurement mode, enhanced loopback mode, that is loopback mode enabled with network programming function. In this mode, both transmit (T1) and receive (T2) timestamps in data plane are collected by the Session-Sender test packets as shown in Figure 1. The network programming function optimizes the "operations of punt test packet and generate return test packet" on the Session-Reflector as timestamping is implemented in forwarding fast path in hardware. This helps to achieve higher STAMP test session scale and faster detection interval.

The Session-Sender adds transmit timestamp (T1) in the payload of the Session-Sender test packet. The Session-Reflector adds the receive timestamp (T2) in the payload of the received test packet in forwarding fast path in hardware without punting the test packet (e.g. to slow path or control-plane). The network programming function enables Session-Reflector to add the receive timestamp (T2) at a specific offset in the payload which is locally provisioned, consistently in the network.

3.2. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 2:
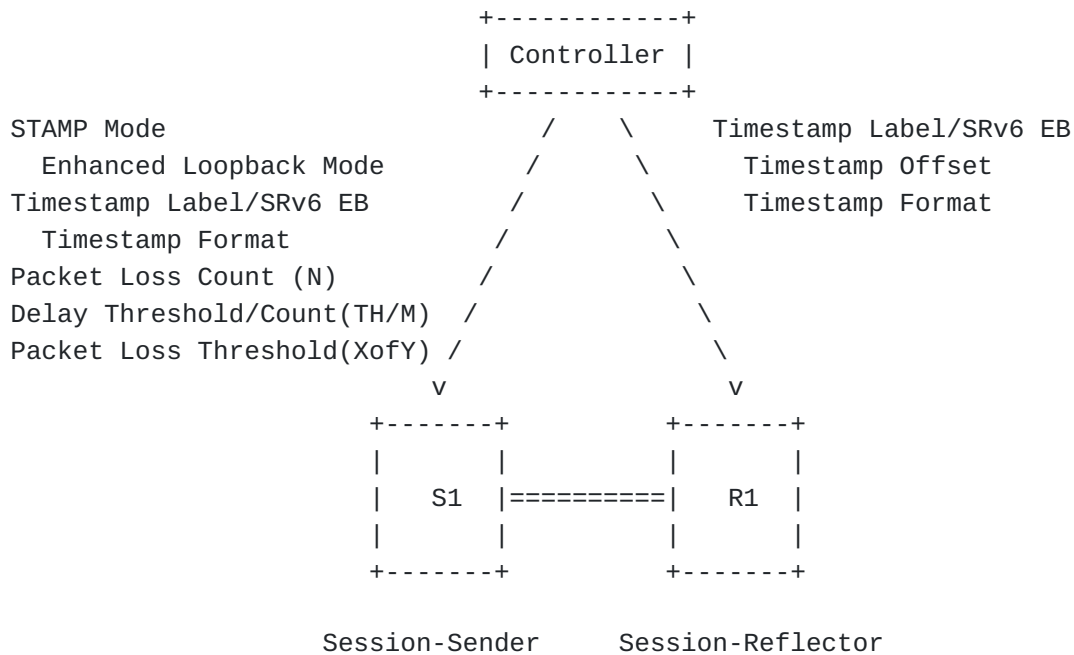
```
                        +------------+
                        | Controller |
                        +------------+
  STAMP Mode                  /    \      Timestamp Label/SRv6 EB
    Enhanced Loopback Mode    /      \       Timestamp Offset
  Timestamp Label/SRv6 EB    /        \      Timestamp Format
    Timestamp Format        /          \
  Packet Loss Count (N)    /            \
  Delay Threshold/Count(TH/M)  /         \
  Packet Loss Threshold(XofY) /           \
                  v                         v
            +-------+            +-------+
            |       |            |       |
            |   S1  |==========|   R1  |
            |       |            |       |
            +-------+            +-------+


           Session-Sender      Session-Reflector

          Figure 2: Example Provisioning Model
```

Example of a STAMP mode is enhanced loopback mode defined in this
document. The values for Timestamp Label and SRv6 Endpoint Behaviour
may be provisioned as described in this document. Example of
Timestamp Format is 64-bit PTPv2 [IEEE1588]. Example of Timestamp
Offset is 16 and 32 bytes for the unauthenticated and authenticated
STAMP Session-Sender test packets, respectively. Example of
threshold values configured for generating notifications are: Packet
Loss Count (N), Delay Exceeded Threshold and Packet Count (TH/M) and
Packet Loss Threshold (XofY), as described in this document.

The mechanisms to provision the Session-Sender and Session-Reflector
are outside the scope of this document.

4. Enhanced Performance Measurement Procedure

For enhanced performance monitoring of an end-to-end SR path
including SR Policy, STAMP Session-Sender test packets are
transmitted in loopback mode enabled with network programming
function to timestamp and forward the packet.

For SR Policy, the Session-Sender test packets are transmitted using
the Segment List (SL) of the Candidate-Path [RFC9256]. When a
Candidate-Path has more than one Segment Lists, multiple Session-
Sender test packets MUST be transmitted, one using each Segment
List.

4.1. Enhanced Performance Measurement Procedure for SR-MPLS Policies

An SR-MPLS Policy may contain a number of Segment Lists (SLs). A Session-Sender test packet MUST be transmitted using each Segment List of the SR-MPLS Policy. The content of an example Session-Sender test packet for an end-to-end SR-MPLS Policy is shown in Figure 3.

The SR-MPLS header can contain the MPLS label stack of the forward path or both the forward and the reverse paths. In the former case, the return test packets are received by the Session-Sender via IP/ UDP [RFC0768] return path and the MPLS header is removed by the Session-Reflector.

In the latter case, the Segment List of the reverse direction SR path is added in the Session-Sender test packet header to receive the return test packet on a specific path, either using the Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. In this case, the MPLS header is not removed by the Session-Reflector.

In both cases, the Session-Sender MUST set the Destination Address equal to the Session-Sender address in the IP header of the test packets.

In this document, two new Timestamp Labels are defined for SR-MPLS data plane to enable network programming function for "timestamp, pop and forward" the received test packet, one for unauthenticated mode and one for authenticated mode.

In the Session-Sender test packets for SR-MPLS Policies, a Timestamp Label is added in the MPLS header as shown in Figure 3, to collect "Receive Timestamp" field in the payload of the test packet. The Label Stack for the reverse direction SR-MPLS path can be added after the Timestamp Label (not shown in the Figure) to receive the return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Label, after timestamping the packet at a specific offset, the Session-Reflector pops the Timestamp Label and forwards the packet using the next label or IP header in the packet (just like the data packets for the normal traffic).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Label(1)                  | TC  |S|     TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Label(n)                  | TC  |S|     TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Extension Label (15)        | TC  |S|     TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Timestamp Label (TBA1 or TBA2)  | TC  |S|     TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IP Header                                                     |
.  Source IP Address = Session-Sender IPv4 or IPv6 Address      .
.  Destination IP Address = Session-Sender IPv4 or IPv6 Address .
.  Protocol = UDP                                               .
.                                                               .
+---------------------------------------------------------------+
| UDP Header                                                    |
.  Source Port = As chosen by Session-Sender                    .
.  Destination Port = As chosen by Session-Sender               .
.                                                               .
+---------------------------------------------------------------+
| Payload = Test Packet as specified in Section 3 of RFC 8972   |
.            in Figure 1 and Figure 3                           .
.                                                               .
+---------------------------------------------------------------+
```

Figure 3: Example STAMP Test Packet with Timestamp Label for SR-MPLS

4.1.1. Timestamp Label Allocation

The timestamp Labels for STAMP test packets in unauthenticated and
authenticated modes can be allocated using one of the following
methods:

  *Labels (values TBA1 and TBA2) assigned by IANA from the "Extended
   Special-Purpose MPLS Values" [RFC9017]. For Label (value TBA1),
   the timestamp offset is fixed at byte-offset 16 from the start of
   the payload for the STAMP test packets in unauthenticated mode,
   and Label (value TBA2) at byte-offset 32 from the start of the
   payload for the STAMP test packets in authenticated mode, both
   using the timestamp format 64-bit PTPv2.

  *Labels allocated by a Controller from the global table of the
   Session-Reflector. The Controller provisions the labels on both

Session-Sender and Session-Reflector, as well as timestamp
offsets and timestamp formats.

*Labels allocated by the Session-Reflector. The signaling and IGP
 flooding extension for the labels (including their timestamp
 offsets and timestamp formats) are outside the scope of this
 document.

4.1.2. Node Capability for Timestamp Label

The STAMP Session-Sender needs to know if the Session-Reflector can
process the Timestamp Label to avoid dropping test packets. The
signaling extension or local configuration for this capability
exchange is outside the scope of this document.

4.2. Enhanced Performance Measurement Procedure for SRv6 Policies

An SRv6 Policy may contain a number of Segment Lists. Each Segment
List may contain a number of SRv6 SIDs as defined in [RFC8986], [I-
D.filsfils-spring-net-pgm-extension-srv6-usid] and [I-D.ietf-spring-
srv6-srh-compression]. A Session-Sender test packet MUST be
transmitted using each Segment List of the SRv6 Policy. An SRv6
Policy may contain an SRv6 Segment Routing Header (SRH) carrying a
Segment List as described in [RFC8754]. The content of an example
Session-Sender test packet for an end-to-end SRv6 Policy using an
SRH is shown in Figure 4.

The SRH can contain the Segment List of the forward path only or
both the forward and the reverse paths. The Destination Address can
contain the SRv6 uSIDs of the forward path only or both the forward
and the reverse paths. In the first case, when the packet contains
only the forward path, an inner IPv6 header (after the SRH and
before the UDP header) is added that contains the Destination
Address equal to the Session-Sender address as shown in Figure 4. In
this case, the SRH if present is removed by the Session-Reflector
and IP/UDP return path is used to forward the packet.

In the second case when the packet contains both the forward and the
reverse paths, the Segment List of the reverse direction SR path is
added in the SRH to receive the return test packet on a specific
path, either using the Binding SID [I-D.ietf-pce-binding-label-sid]
or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-
path]. In this case, the SRH is not removed by the Session-Reflector
and an inner IPv6 header is not required. When the return test
packet contains an SRH at the Session-Sender, the procedure defined
for upper-layer header processing for SRv6 SIDs in [RFC8986] MUST be
used to process the UDP header present after the SRH in the received
test packets.

The [RFC8986] defines SRv6 Endpoint Behaviours (EB) for SRv6 nodes. In this document, two new Timestamp Endpoint Behaviours are defined for Segment Routing Header (SRH) [RFC8754] to enable "Timestamp and Forward (TSF)" function for the received test packets, one for unauthenticated mode and one for authenticated mode.

In the Session-Sender test packets for SRv6 Policies, Timestamp Endpoint Function (End.TSF) is carried with the target Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 4, to collect "Receive Timestamp" field in the payload of the test packet. The Segment List for the reverse direction path can be added after the target SID to receive the return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Endpoint (End.TSF) for the target SID which is local, after timestamping the packet at a specific offset, the Session-Reflector forwards the packet using the next SID in the SRH or inner IPv6 header in the packet (just like the data packets for the normal traffic).

```
+----------------------------------------------------------------+
| IP Header                                                      |
.  Source IP Address = Session-Sender IPv6 Address              .
.  Destination IP Address=Session-Reflector IPv6 Address |      .
.                Segment List[Segments Left]                    .
.  Next-Header = 43, Routing Type = SRH (4)                     .
.                                                                .
+----------------------------------------------------------------+
| SRH as specified in RFC 8754                                   |
.      <Segment List>                                           .
.      SRv6 Endpoint End.TSF (value TBA3 or TBA4)               .
.                                                                .
+----------------------------------------------------------------+
| IP Header                                                      |
.  Source IP Address = Session-Sender IPv6 Address              .
.  Destination IP Address = Session-Sender IPv6 Address         .
.  Next-Header = UDP (17)                                        .
.                                                                .
+----------------------------------------------------------------+
| UDP Header                                                     |
.  Source Port = As chosen by Session-Sender                    .
.  Destination Port = As chosen by Session-Sender               .
.                                                                .
+----------------------------------------------------------------+
| Payload = Test Packet as specified in Section 3 of RFC 8972   |
.            in Figure 1 and Figure 3                           .
.                                                                .
+----------------------------------------------------------------+
```

Figure 4: Example STAMP Test Packet with Endpoint Function for SRv6

4.2.1. Timestamp Endpoint Function Assignment

The Timestamp Endpoint Functions for "Timestamp and Forward" can be
signaled using one of the following methods:

  *Timestamp Endpoint Functions (values TBA3 and TBA4) assigned by
   IANA from the "SRv6 Endpoint Behaviors Registry". For endpoint
   behaviour (value TBA3), the timestamp offset is fixed at byte-
   offset 16 from the start of the payload for the STAMP test
   packets in unauthenticated mode, and endpoint behaviour (value
   TBA4) at byte-offset 32 from the start of the payload for the
   STAMP test packets in authenticated mode, both using the
   timestamp format 64-bit PTPv2.

  *Timestamp Endpoint Functions assigned by a Controller. The
   Controller provisions the values on both Session-Sender and
   Session-Reflector, as well as timestamp offsets and timestamp
   formats.

  *Timestamp Endpoint Functions assigned by the Session-Reflector.
   The signaling and IGP flooding extension for the endpoint
   functions (including timestamp offsets and timestamp formats) are
   outside the scope of this document.

4.2.2. Node Capability for Timestamp Endpoint Function

The STAMP Session-Sender needs to know if the Session-Reflector can
process the Timestamp Endpoint Function to avoid dropping test
packets. The signaling extension for this capability exchange is
outside the scope of this document.

5. Example Notifications

The timestamps T1 and T2 are used to measure the one-way delay. The
delay metrics for an end-to-end SR path are notified, for example,
when consecutive M number of test packets have measured delay values
exceed the user-configured threshold TH, where M (Delay Exceeded
Packet Count) and TH (Absolute and Percentage Delay Exceeded
Thresholds) are also locally provisioned values.

The round-trip packet loss for an end-to-end SR path is calculated
using the Sequence Number in the Session-Sender test packets. The
packet loss metric is notified when X number of Session-Sender test
packets were lost out of last Y number of test packets transmitted
by the Session-Sender, where Threshold XofY is locally provisioned
value.

STAMP session state as UP (i.e. Connectivity Verification success)
for an end-to-end SR path is initially notified as soon as one or
more return test packets are received at the Session-Sender.

STAMP session state as DOWN (i.e. Connectivity Verification failure due to packet loss) for an end-to-end SR path is notified when consecutive N number of return test packets are not received at the Session-Sender after the session state is UP, where N (Packet Loss Count) is a locally provisioned value.

In the loopback mode where the Session-Reflector does not generate reply test packets, a Connectivity Verification failure on the reverse direction path can cause the return test packets to not reach the Session-Sender. This is also true in the case where the return test packets are generated by the stateless Session-Reflector, e.g., in two-way mode. The stateful Session-Reflector can solve this issue by maintaining the forwarding direction state and notifying a Connectivity Verification success and failure to the Session-Sender based on the Packet Loss Count, N.

6. Security Considerations

The STAMP protocol is intended for deployment in limited domains [RFC8799]. As such, it assumes that a node involved in the STAMP protocol operation has previously verified the integrity of the path and the identity of the far-end Session-Reflector.

The security considerations specified in [RFC8762] and [RFC8972] also apply to the procedures defined in this document. Specifically, the message integrity protection using HMAC, as defined in Section 4.4 of [RFC8762] also apply to the procedure described in this document.

7. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>). IANA is requested to allocate Timestamp Label value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	Timestamp Label for offset 16 for STAMP in Unauthenticated Mode	This document
TBA2	Timestamp Label for offset 32 for STAMP in Authenticated Mode	This document

IANA is requested to allocate, within the "SRv6 Endpoint Behaviors
Registry" sub-registry belonging to the top-level "Segment Routing
Parameters" registry [RFC8986], the following allocation:

```
+-------------+--------------------------------+---------------+
| Value       | Endpoint Behavior              | Reference     |
+-------------+--------------------------------+---------------+
| TBA3        | End.TSF (Timestamp and Forward)| This document |
|             | for offset 16 for STAMP        |               |
|             | in Unauthenticated Mode        |               |
+-------------+--------------------------------+---------------+
| TBA4        | End.TSF (Timestamp and Forward)| This document |
|             | for offset 32 for STAMP        |               |
|             | in Authenticated Mode          |               |
+-------------+--------------------------------+---------------+
```

8. References

8.1. 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>.

[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>.

[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>.

[RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A.,
           and E. Ruffini, "Simple Two-Way Active Measurement
           Protocol Optional Extensions", RFC 8972, DOI 10.17487/
           RFC8972, January 2021, <https://www.rfc-editor.org/info/
           rfc8972>.

[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>.

[I-D.ietf-spring-stamp-srpm]
> Gandhi, R., Filsfils, C., Voyer, D., Chen, M., Janssens,
> B., and R. Foote, "Performance Measurement Using Simple
> TWAMP (STAMP) for Segment Routing Networks", Work in
> Progress, Internet-Draft, draft-ietf-spring-stamp-
> srpm-04, 5 July 2022, <https://www.ietf.org/archive/id/
> draft-ietf-spring-stamp-srpm-04.txt>.

8.2. Informative References

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

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,
> Decraene, B., Litkowski, S., and R. Shakir, "Segment
> Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,
> July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy,
> J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing
> Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March
> 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet
> Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020,
> <https://www.rfc-editor.org/info/rfc8799>.

[RFC9017] Andersson, L., Kompella, K., and A. Farrel, "Special-
> Purpose Label Terminology", RFC 9017, DOI 10.17487/
> RFC9017, April 2021, <https://www.rfc-editor.org/info/
> rfc9017>.

[I-D.ietf-spring-srv6-srh-compression]
> Cheng, W., Filsfils, C., Li, Z., Decraene, B., Cai, D.,
> Voyer, D., Clad, F., Zadok, S., Guichard, J. N., Aihua,
> L., Raszuk, R., and C. Li, "Compressed SRv6 Segment List
> Encoding in SRH", Work in Progress, Internet-Draft,
> draft-ietf-spring-srv6-srh-compression-02, 22 July 2022,
> <https://www.ietf.org/archive/id/draft-ietf-spring-srv6-
> srh-compression-02.txt>.

[I-D.filsfils-spring-net-pgm-extension-srv6-usid]
> Filsfils, C., Garvia, P. C., Cai, D., Voyer, D., Meilik,
> I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman,
> D., Liu, Y., and J. Guichard, "Network Programming
> extension: SRv6 uSID instruction", Work in Progress,
> Internet-Draft, draft-filsfils-spring-net-pgm-extension-
> srv6-usid-13, 13 June 2022, <https://www.ietf.org/

archive/id/draft-filsfils-spring-net-pgm-extension-srv6-usid-13.txt>.

[RFC9256] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A.,
           and P. Mattes, "Segment Routing Policy Architecture", RFC
           9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

[I-D.ietf-pce-binding-label-sid]
           Sivabalan, S., Filsfils, C., Tantsura, J., Previdi, S.,
           and C. L. (editor), "Carrying Binding Label/Segment
           Identifier in PCE-based Networks.", Work in Progress,
           Internet-Draft, draft-ietf-pce-binding-label-sid-15, 20
           March 2022, <https://www.ietf.org/archive/id/draft-ietf-pce-binding-label-sid-15.txt>.

[I-D.ietf-pce-sr-bidir-path] Li, C., Chen, M., Cheng, W., Gandhi,
           R., and Q. Xiong, "Path Computation Element Communication
           Protocol (PCEP) Extensions for Associated Bidirectional
           Segment Routing (SR) Paths", Work in Progress, Internet-Draft, draft-ietf-pce-sr-bidir-path-09, 9 March 2022,
           <https://www.ietf.org/archive/id/draft-ietf-pce-sr-bidir-path-09.txt>.

Acknowledgments

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Navin Vaghamshi
Reliance

Email: Navin.Vaghamshi@ril.com

Moses Nagarajah
Telstra

Email: Moses.Nagarajah@team.telstra.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Amit Dhamija
Rakuten

Email: amit.dhamija@rakuten.com