6man
Internet-Draft
Intended status: Standards Track
Expires: December 14, 2020

Z. Ali
C. Filsfils
Cisco Systems
S. Matsushima
Softbank
D. Voyer
Bell Canada
M. Chen
Huawei

June 12, 2020

# Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6) draft-ietf-6man-spring-srv6-oam-05

## Abstract

This document describes how the existing IPv6 OAM mechanisms can be used in an SRv6 network. The document also introduces enhancements for controller-based OAM mechanisms for SRv6 networks.

## 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 December 14, 2020.

## Copyright Notice

Table of Contents



## 1. Introduction

As Segment Routing with IPv6 data plane (SRv6) simply adds a new type of Routing Extension Header, existing IPv6 OAM mechanisms can be used in an SRv6 network. This document describes how the existing IPv6 mechanisms for ping and trace route can be used in an SRv6 network.

The document also introduces enhancements for controller-based OAM mechanism for SRv6 networks. Specifically, the document describes an OAM mechanism for performing controllable and predictable flow sampling from segment endpoints using, e.g., IP Flow Information Export (IPFIX) protocol [RFC7011]. The document also outlines how centralized OAM technique in [RFC8403] can be extended for SRv6 to

perform a path continuity check between any nodes within an SRv6 domain from a centralized monitoring system, with minimal or no control plane intervene on the nodes.

## 1.1. 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].

## 1.2. Abbreviations

The following abbreviations are used in this document:

SID: Segment ID.

SL: Segments Left.

SR: Segment Routing.

SRH: Segment Routing Header.

SRv6: Segment Routing with IPv6 Data plane.

TC: Traffic Class.

ICMPv6: ICMPv6 Specification [RFC4443].

## 1.3. Terminology and Reference Topology

This document uses the terminology defined in [I-D.ietf- spring-srv6-network-programming]. The readers are expected to be familiar with the same.

Throughout the document, the following simple topology is used for illustration.

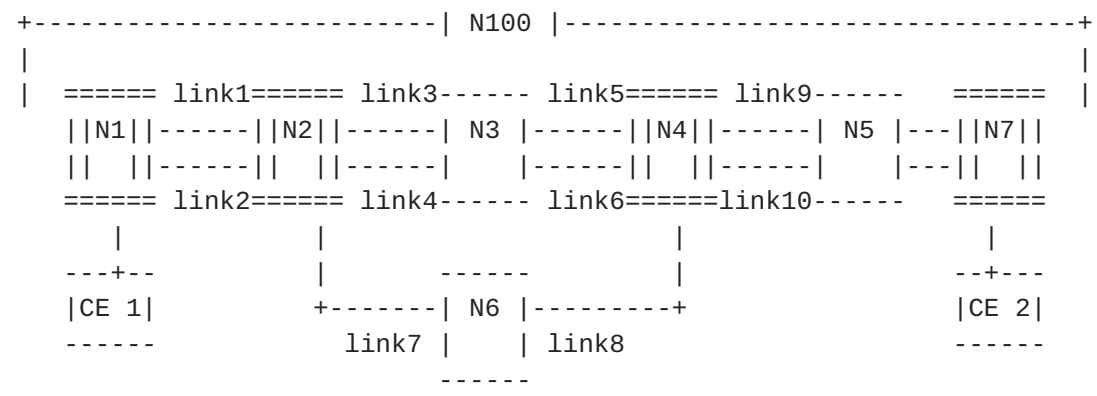



Figure 1 Reference Topology

In the reference topology:

Node k has a classic IPv6 loopback address 2001:DB8:A:k::/128.

Nodes N1, N2, and N4 are SRv6 capable nodes.

Nodes N3, N5 and N6 are IPv6 nodes that are not SRv6 capable. Such nodes are referred as classic IPv6 nodes.

A SID at node k with locator block 2001:DB8:B::/48 and function F is represented by 2001:DB8:B:k:F::.

Node N100 is a controller.

The IPv6 address of the nth Link between node X and Y at the X side is represented as 2001:DB8:X:Y:Xn::, e.g., the IPv6 address of link6 (the 2nd link) between N3 and N4 at N3 in Figure 1 is 2001:DB8:3:4:32::. Similarly, the IPv6 address of link5 (the 1st link between N3 and N4) at node 3 is 2001:DB8:3:4:31::.

2001:DB8:B:k:Cij:: is explicitly allocated as the END.X function at node k towards neighbor node i via jth Link between node i and node k. e.g., 2001:DB8:B:2:C31:: represents END.X at N2 towards N3 via link3 (the 1st link between N2 and N3). Similarly, 2001:DB8:B:4:C52:: represents the END.X at N4 towards N5 via link10.

A SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SR path.

(SA,DA) (S3, S2, S1; SL)(payload) represents an IPv6 packet with:

* IPv6 header with source address SA, destination addresses DA and SRH as next-header

* SRH with SID list <S1, S2, S3> with SegmentsLeft = SL

* Note the difference between the < > and () symbols: <S1, S2, S3> represents a SID list where S1 is the first SID and S3 is the last SID to traverse. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID. When referring to an SR policy in a high-level use-case, it is simpler to use the <S1, S2, S3> notation. When referring to an illustration of the detailed packet behavior, the (S3, S2, S1; SL) notation is more convenient.

* (payload) represents the the payload of the packet.

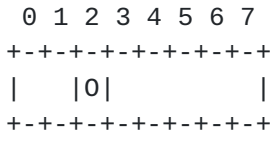
SRH[SL] represents the SID pointed by the SL field in the first SRH. In our example SID list (S3, S2, S1; SL), SRH[2] represents S1, SRH[1] represents S2 and SRH[0] represents S3.

## 2. OAM Mechanisms

This section defines OAM enhancement for the SRv6 networks.

### 2.1. O-flag in Segment Routing Header

[RFC8754] describes the Segment Routing Header (SRH) and how SR capable nodes use it. The SRH contains an 8-bit "Flags" field. This document defines the following bit in the SRH.Flags to carry the O-flag:




Where:

O-flag: OAM flag.

The document does not define any other flag in the SRH.Flags and meaning and processing of any other bit in SRH.Flags is outside of the scope of this document.

### 2.1.1. O-flag Processing

The O-flag in SRH is used as a marking-bit in the user packets to trigger the telemetry data collection and export at the segment endpoints.

Without the loss of generality, this document assumes IP Flow Information Export (IPFIX) protocol [RFC7011] is used for exporting the traffic flow information from the network devices to a controller for monitoring and analytics. The requested information elements are configured by the management plane through data set templates (e.g., as in IPFIX [RFC7012]).

Implementation of the O-flag is OPTIONAL. If a node does not support the O-flag, then upon reception it simply ignores it. If a node supports the O-flag, it can optionally advertise its potential via control plan protocol(s).

When N receives a packet whose IPv6 DA is S and S is a local SID, the line S01 of the pseudo-code associated with the SID S, as defined in section 4.3.1.1 of [RFC8754], is modified as follows for the O-flag processing.

```
S01.1. IF SRH.Flags.O-flag is set and local configuration permits
       O-flag processing THEN
          a. Make a copy of the packet.
          b. Send the copied packet, along with a timestamp
          to the OAM process for telemetry data collection
          and export.      ;; Ref1
Ref1: An implementation SHOULD copy and record the timestamp as
soon as possible during packet processing. Timestamp or any other
metadata is not
carried in the packet forwarded to the next hop.
```

Please note that the O-flag processing happens before execution of regular processing of the local SID S.

Based on the requested information elements configured by the management plane through data set templates [RFC7012], the OAM process exports the requested information elements. The information elements include parts of the packet header and/or parts of the packet payload for flow identification. The OAM process uses information elements defined in IPFIX [RFC7011] and PSAMP [RFC5476] for exporting the requested sections of the mirrored packets.

If the telemetry data from the last node in the segment-list (egress node) is desired, the ingress uses an Ultimate Segment Pop (USP) SID advertised by the egress node.

The processing node SHOULD rate-limit the number of packets punted to the OAM process to avoid hitting any performance impact.

The OAM process MUST NOT process the copy of the packet or respond to any upper-layer header (like ICMP, UDP, etc.) payload to prevent multiple evaluations of the datagram.

Specification of the OAM process or the external controller operations are beyond the scope of this document. section 3 illustrates use of the SRH.Flags.O-flag for implementing a controller-based hybrid OAM mechanism, where the "hybrid" classification is based on RFC7799 [RFC7799]. The illustration is different than the In-situ OAM defined in [I.D-draft-ietf-ippm-ioam-data]. This is because In-situ OAM records operational and telemetry information in the packet as the packet traverses a path between two points in the network [I.D-draft-ietf- ippm-ioam-data]. The controller-based OAM mechanism using O-flag illustration in section 3 does not require the recording of OAM data in the packet.

## 3. Illustrations

This section shows how some of the existing IPv6 OAM mechanisms can be used in an SRv6 network. It also illustrates an OAM mechanism for performing controllable and predictable flow sampling from segment endpoints. How centralized OAM technique in [RFC8403] can be extended for SRv6 is also described in this Section.

### 3.1. Ping in SRv6 Networks

The following subsections outline some use cases of the ICMP ping in the SRv6 networks.

#### 3.1.1. Classic Ping

The existing mechanism to query liveliness of a remote IP address, along the shortest path, continues to work without any modification. The initiator may be an SRv6 node or a classic IPv6 node. Similarly, the egress or transit may be an SRv6 capable node or a classic IPv6 node.

If an SRv6 capable ingress node wants to ping an IPv6 address via an arbitrary segment list <S1, S2, S3>, it needs to initiate ICMPv6 ping with an SR header containing the SID list <S1, S2, S3>. This is illustrated using the topology in Figure 1. Assume all the links

have IGP metric 10 except both links between node2 and node3, which have IGP metric set to 100. User issues a ping from node N1 to a loopback of node 5, via segment list <2001:DB8:B:2:C31::, 2001:DB8:B:4:C52::>.

Figure 2 contains sample output for a ping request initiated at node N1 to the loopback address of node N5 via a segment list <2001:DB8:B:2:C31::, 2001:DB8:B:4:C52::>.

```
> ping 2001:DB8:A:5:: via segment-list 2001:DB8:B:2:C31::,
       2001:DB8:B:4:C52::

Sending 5, 100-byte ICMP Echos to B5::, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0.625
/0.749/0.931 ms
```

Figure 2 A sample ping output at an SRv6 capable node

All transit nodes process the echo request message like any other data packet carrying SR header and hence do not require any change. Similarly, the egress node (IPv6 classic or SRv6 capable) does not require any change to process the ICMPv6 echo request. For example, in the ping example of Figure 2:

- Node N1 initiates an ICMPv6 ping packet with SRH as follows (2001:DB8:A:1::, 2001:DB8:B:2:C31::) (2001:DB8:A:5::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=2, NH = ICMPv6)(ICMPv6 Echo Request). If 2001:DB8:B:4:C52:: is a PSP SID, the OAM probes encodes the PSP SID in the packet (just mimicking data packets). No special consideration is needed to handle PSP SIDs.

- Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (2001:DB8:B:2:C31::) and forwards the packet on link3 to N3.

- Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on the DA 2001:DB8:B:4:C52:: in the IPv6 header.

- Node N4, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it observes the END.X function (2001:DB8:B:4:C52::) and forwards the packet on link10 towards N5. If 2001:DB8:B:4:C52:: is a PSP SID, The penultimate node (Node N4) does not, should not and cannot differentiate between the data packets and OAM probes. Specifically, if 2001:DB8:B:4:C52:: is a

PSP SID, node N4 executes the SID like any other data packet with DA = 2001:DB8:B:4:C52:: and removes the SRH.

- The echo request packet at N5 arrives as an IPv6 packet with or without an SRH. If N5 receives the packet with SRH, it skips SRH processing (SL=0). In either case, Node N5 performs the standard IPv6/ ICMPv6 processing on the echo request.

### 3.1.2. Pinging a SID

The classic ping described in the previous section applies equally to ping a remote SID function, as explained using an example in the following.

Consider the example where the user wants to ping a remote SID function 2001:DB8:B:4::, via 2001:DB8:B:2:C31::, from node N1. The ICMPv6 echo request is processed at the individual nodes along the path as follows:

- Node N1 initiates an ICMPv6 ping packet with SRH as follows (2001:DB8:A:1::, 2001:DB8:B:2:C31::) (2001:DB8:B:4::, 2001:DB8:B:2:C31::; SL=1; NH=ICMPv6)(ICMPv6 Echo Request). If 2001:DB8:B:2:C31:: is a PSP SID, the OAM probes encodes the PSP SID in the packet (just mimicking data packets). No special consideration is needed to handle PSP SIDs.

- Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (2001:DB8:B:2:C31::) on the echo request packet. If 2001:DB8:B:2:C31:: is a PSP SID, node N4 executes the SID like any other data packet with DA = 2001:DB8:B:2:C31:: and removes the SRH.

- Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA = 2001:DB8:B:4:: in the IPv6 header.

- When node N4 receives the packet, it processes the 2001:DB8:B:4:: SID, as described in the pseudocode in [I-D.ietf-spring-srv6-network-programming].

- If the 2001:DB8:B:4:: SID is not locally programmed, the packet is discarded

- If the target SID (2001:DB8:B:4::) is locally programmed, the node processes the upper layer header. As part of the upper layer header processing node N4 respond to the ICMPv6 echo request message.

### 3.2. Traceroute

There is no hardware or software change required for traceroute operation at the classic IPv6 nodes in an SRv6 network. That includes the classic IPv6 node with ingress, egress or transit roles. Furthermore, no protocol changes are required to the standard traceroute operations. In other words, existing traceroute mechanisms work seamlessly in the SRv6 networks.

The following subsections outline some use cases of the traceroute in the SRv6 networks.

#### 3.2.1. Classic Traceroute

The existing mechanism to traceroute a remote IP address, along the shortest path, continues to work without any modification. The initiator may be an SRv6 node or a classic IPv6 node. Similarly, the egress or transit may be an SRv6 node or a classic IPv6 node.

If an SRv6 capable ingress node wants to traceroute to IPv6 address via an arbitrary segment list <S1, S2, S3>, it needs to initiate traceroute probe with an SR header containing the SID list <S1, S2, S3>. That is illustrated using the topology in Figure 1. Assume all the links have IGP metric 10 except both links between node2 and node3, which have IGP metric set to 100. User issues a traceroute from node N1 to a loopback of node 5, via segment list <2001:DB8:B:2:C31::, 2001:DB8:B:4:C52::>. Figure 3 contains sample output for the traceroute request.

```
> traceroute 2001:DB8:A:5:: via segment-list 2001:DB8:B:2:C31::,
             2001:DB8:B:4:C52::

Tracing the route to 2001:DB8:A:5::
1  2001:DB8:1:2:21:: 0.512 msec 0.425 msec 0.374 msec
   DA: 2001:DB8:B:2:C31::,
   SRH:(2001:DB8:A:5::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=2)
2  2001:DB8:2:3:31:: 0.721 msec 0.810 msec 0.795 msec
   DA: 2001:DB8:B:4:C52::,
   SRH:(2001:DB8:A:5::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=1)
3  2001:DB8:3:4::41:: 0.921 msec 0.816 msec 0.759 msec
   DA: 2001:DB8:B:4:C52::,
   SRH:(2001:DB8:A:5::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=1)
4  2001:DB8:4:5::52:: 0.879 msec 0.916 msec 1.024 msec
   DA: 2001:DB8:A:5::
```

Figure 3 A sample traceroute output at an SRv6 capable node

Please note that information for hop2 is returned by N3, which is a classic IPv6 node. Nonetheless, the ingress node is able to display SR header contents as the packet travels through the IPv6 classic node. This is because the "Time Exceeded Message" ICMPv6 message can contain as much of the invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU [RFC4443]. The SR header is also included in these ICMPv6 messages initiated by the classic IPv6 transit nodes that are not running SRv6 software. Specifically, a node generating ICMPv6 message containing a copy of the invoking packet does not need to understand the extension header(s) in the invoking packet.

The segment list information returned for hop1 is returned by N2, which is an SRv6 capable node. Just like for hop2, the ingress node is able to display SR header contents for hop1.

There is no difference in processing of the traceroute probe at an IPv6 classic node and an SRv6 capable node. Similarly, both IPv6 classic and SRv6 capable nodes may use the address of the interface on which probe was received as the source address in the ICMPv6 response. ICMP extensions defined in [RFC5837] can be used to also display information about the IP interface through which the datagram would have been forwarded had it been forwardable, and the IP next hop to which the datagram would have been forwarded, the IP interface upon which a datagram arrived, the sub-IP component of an IP interface upon which a datagram arrived.

The information about the IP address of the incoming interface on which the traceroute probe was received by the reporting node is very useful. This information can also be used to verify if SID functions 2001:DB8:B:2:C31:: and 2001:DB8:B:4:C52:: are executed correctly by N2 and N4, respectively. Specifically, the information displayed for hop2 contains the incoming interface address 2001:DB8:2:3:31:: at N3. This matches with the expected interface bound to END.X function 2001:DB8:B:2:C31:: (link3). Similarly, the information displayed for hop5 contains the incoming interface address 2001:DB8:4:5::52:: at N5. This matches with the expected interface bound to the END.X function 2001:DB8:B:4:C52:: (link10).

### 3.2.2. Traceroute to a SID

The classic traceroute described in the previous section applies equally to traceroute a remote SID function, as explained using an example in the following.

Please note that traceroute to a SID function is exemplified using UDP probes. However, the procedure is equally applicable to other implementations of traceroute mechanism.

Consider the example where the user wants to traceroute a remote SID function 2001:DB8:B:4::, via 2001:DB8:B:2:C31::, from node N1. The traceroute probe is processed at the individual nodes along the path as follows:

- Node N1 initiates a traceroute probe packet with a monotonically increasing value of hop count and SRH as follows (2001:DB8:A:1::, 2001:DB8:B:2:C31::) (2001:DB8:B:4::, 2001:DB8:B:2:C31::; SL=1; NH=UDP)(Traceroute probe). If 2001:DB8:B:2:C31:: is a PSP SID, the OAM probes encodes the PSP SID in the packet (just mimicking data packets). No special consideration is needed to handle PSP SIDs.

- When node N2 receives the packet with hop-count = 1, it processes the hop count expiry. Specifically, the node N2 responses with the ICMPv6 message (Type: "Time Exceeded", Code: "Time to Live exceeded in Transit").

- When Node N2 receives the packet with hop-count > 1, it performs the standard SRH processing. Specifically, it executes the END.X function (2001:DB8:B:2:C31::) on the traceroute probe. If 2001:DB8:B:2:C31:: is a PSP SID, node N4 executes the SID like any other data packet with DA = 2001:DB8:B:2:C31:: and removes the SRH.

- When node N3, which is a classic IPv6 node, receives the packet with hop-count = 1, it processes the hop count expiry. Specifically, the node N3 responses with the ICMPv6 message (Type: "Time Exceeded", Code: "Time to Live exceeded in Transit").

- When node N3, which is a classic IPv6 node, receives the packet with hop-count > 1, it performs the standard IPv6 processing. Specifically, it forwards the traceroute probe based on DA 2001:DB8:B:4:: in the IPv6 header.

- When node N4 receives the packet with DA set to the local SID 2001:DB8:B:4::, it processes the END SID, as described in the pseudocode in [I-D.ietf-spring-srv6-network-programming].

- If the 2001:DB8:B:4:: SID is not locally programmed, the packet is discarded.

- If the target SID (2001:DB8:B:4::) is locally programmed, the node processes the upper layer header. As part of the upper layer header processing node N4 responses with the ICMPv6 message (Type: Destination unreachable, Code: Port Unreachable).

Figure 4 displays a sample traceroute output for this example.

```
> traceroute 2001:DB8:B:4:C52:: via segment-list 2001:DB8:B:2:C31::

Tracing the route to SID function 2001:DB8:B:4:C52::
1  2001:DB8:1:2:21:: 0.512 msec 0.425 msec 0.374 msec
   DA: 2001:DB8:B:2:C31::,
   SRH:(2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=1)
2  2001:DB8:2:3:31:: 0.721 msec 0.810 msec 0.795 msec
   DA: 2001:DB8:B:4:C52::,
   SRH:(2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=0)
3  2001:DB8:3:4:41:: 0.921 msec 0.816 msec 0.759 msec
   DA: 2001:DB8:B:4:C52::,
   SRH:(2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=0)
```

Figure 4 A sample output for hop-by-hop traceroute to a SID

## 3.3. A Controller-Based Hybrid OAM Using O-flag

Consider the example where the user wants to monitor sampled IPv4 VPN 100 traffic going from CE1 to CE2 via a low latency SR policy P installed at Node N1. To exercise a low latency path, the SR Policy P forces the packet via segments 2001:DB8:B:2:C31:: and 2001:DB8:B:4:C52::. The VPN SID at N7 associated with VPN100 is 2001:DB8:B:7:DT100::. 2001:DB8:B:7:DT100:: is a USP SID. N1, N4, and N7 are capable of processing SRH.Flags.O-flag but N2 is not capable of processing SRH.Flags.O-flag. N100 is the centralized controller capable of processing and correlating the copy of the packets sent from nodes N1, N4, and N7. N100 is aware of SRH.Flags.O-flag processing capabilities. Controller N100 with the help from nodes N1, N4, N7 and implements a hybrid OAM mechanism using the SRH.Flags.O-flag as follows:

- A packet P1:(IPv4 header)(payload) is sent from CE1 to Node N1.

- Node N1 steers the packet P1 through the Policy P. Based on a local configuration, Node N1 also implements logic to sample traffic steered through policy P for hybrid OAM purposes. Specification for the sampling logic is beyond the scope of this document. Consider the case where packet P1 is classified as a packet to be monitored via the hybrid OAM. Node N1 sets SRH.Flags.O-flag during encapsulation required by policy P. As part of setting the SRH.Flags.O-flag, node N1 also send a timestamped copy of the packet P1: (2001:DB8:A:1::, 2001:DB8:B:2:C31::) (2001:DB8:B:7:DT100::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=2; O-flag=1; NH=IPv4)(IPv4 header)(payload) to a local OAM process. The local OAM process sends a full or partial copy of the packet P1 to the controller N100. The OAM

process includes the recorded timestamp, additional OAM information like incoming and outgoing interface, etc. along with any applicable metadata. Node N1 forwards the original packet towards the next segment 2001:DB8:B:2:C31::.

- When node N2 receives the packet with SRH.Flags.O-flag set, it ignores the SRH.Flags.O-flag. This is because node N2 is not capable of processing the O-flag. Node N2 performs the standard SRv6 SID and SRH processing. Specifically, it executes the END.X function (2001:DB8:B:2:C31::) and forwards the packet P1 (2001:DB8:A:1::, 2001:DB8:B:4:C52::) (2001:DB8:B:7:DT100::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=1; O-flag=1; NH=IPv4)(IPv4 header)(payload) over link 3 towards Node N3.

- When node N3, which is a classic IPv6 node, receives the packet P1 , it performs the standard IPv6 processing. Specifically, it forwards the packet P1 based on DA 2001:DB8:B:4:C52:: in the IPv6 header.

- When node N4 receives the packet P1 (2001:DB8:A:1::, 2001:DB8:B:4:C52::) (2001:DB8:B:7:DT100::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=1; O-flag=1; NH=IPv4)(IPv4 header)(payload), it processes the SRH.Flags.O-flag. As part of processing the O-flag, it sends a timestamped copy of the packet to a local OAM process. The local OAM process sends a full or partial copy of the packet P1 to the controller N100. The OAM process includes the recorded timestamp, additional OAM information like incoming and outgoing interface, etc. along with any applicable metadata. Node N4 performs the standard SRv6 SID and SRH processing on the original packet P1. Specifically, it executes the END.X function (2001:DB8:B:4:C52::) and forwards the packet P1 (2001:DB8:A:1::, 2001:DB8:B:7:DT100::) (2001:DB8:B:7:DT100::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=0; O-flag=1; NH=IPv4)(IPv4 header)(payload) over link 10 towards Node N5.

- When node N5, which is a classic IPv6 node, receives the packet P1, it performs the standard IPv6 processing. Specifically, it forwards the packet based on DA 2001:DB8:B:7:DT100:: in the IPv6 header.

- When node N7 receives the packet P1 (2001:DB8:A:1::, 2001:DB8:B:7:DT100::) (2001:DB8:B:7:DT100::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::; SL=0; O-flag=1; NH=IPv4)(IPv4 header)(payload), it processes the SRH.Flags.O-flag. As part of processing the O-flag, it sends a timestamped copy of the packet to a local OAM process. The local OAM process sends a full or partial copy of the packet P1 to the controller N100. The OAM

process includes the recorded timestamp, additional OAM information like incoming and outgoing interface, etc. along with any applicable metadata. Node N4 performs the standard SRv6 SID and SRH processing on the original packet P1. Specifically, it executes the VPN SID (2001:DB8:B:7:DT100::) and based on lookup in table 100 forwards the packet P1 (IPv4 header)(payload) towards CE 2.

- The controller N100 processes and correlates the copy of the packets sent from nodes N1, N4 and N7 to find segment-by-segment delays and provide other hybrid OAM information related to packet P1.

- The process continues for any other sampled packets.

### 3.4. Monitoring of SRv6 Paths

In the recent past, network operators are interested in performing network OAM functions in a centralized manner. [RFC8403] describes such a centralized OAM mechanism. Specifically, the document describes a procedure that can be used to perform path continuity check between any nodes within an SR domain from a centralized monitoring system, with minimal or no control plane intervene on the nodes. However, the document focuses on SR networks with MPLS data plane. This document describes how the concept can be used to perform path monitoring in an SRv6 network from the centralized controller.

In the reference topology in Figure 1, N100 uses an IGP protocol like OSPF or ISIS to get the topology view within the IGP domain. N100 can also use BGP-LS to get the complete view of an inter-domain topology. The controller leverages the visibility of the topology to monitor the paths between the various endpoints without control plane intervention required at the monitored nodes.

The controller N100 advertises an END SID 2001:DB8:B:100:1::. To monitor any arbitrary SRv6 paths, the controller can create a loopback probe that originates and terminates on Node N100. For example, in order to verify a segment list <2001:DB8:B:2:C31::, 2001:DB8:B:4:C52::>:

- N100 generates an OAM packet (2001:DB8:A:100::, 2001:DB8:B:2:C31::)(2001:DB8:B:100:1::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=2)(OAM Payload). The controller routes the probe packet towards the first segment, which is 2001:DB8:B:2:C31::.

- Node N2 executes the END.X function (2001:DB8:B:2:C31::) and forwards the packet (2001:DB8:A:100::, 2001:DB8:B:4:C52::)(2001:DB8:B:100:1::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=1)(OAM Payload) on link3 to N3.

- Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the packet based on the DA 2001:DB8:B:4:C52:: in the IPv6 header.

- Node N4 executes the END.X function (2001:DB8:B:4:C52::) and forwards the packet (2001:DB8:A:100::, 2001:DB8:B:100:1::)(2001:DB8:B:100:1::, 2001:DB8:B:4:C52::, 2001:DB8:B:2:C31::, SL=0)(OAM Payload) on link10 to N5.

- Node N5, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the packet based on the DA 2001:DB8:B:100:1:: in the IPv6 header.

- Node N100 executes the standard SRv6 END function. It decapsulates the header and consume the probe for OAM processing. The information in the OAM payload is used to detect any missing probes, round trip delay, etc.

The OAM payload type or the information carried in the OAM probe is a local implementation decision at the controller and is outside the scope of this document.

## 4. Implementation Status

This section is to be removed prior to publishing as an RFC.

See [I-D.matsushima-spring-srv6-deployment-status] for updated deployment and interoperability reports.

## 5. Security Considerations

This document does not define any new protocol extensions and relies on existing procedures defined for ICMP. This document does not impose any additional security challenges to be considered beyond security considerations described in [RFC4884], [RFC4443], [RFC0792], and [RFC8754].

## 6. IANA Considerations

### 6.1. Segment Routing Header Flags

This I-D requests to IANA to allocate bit position 2, within the "Segment Routing Header Flags" registry defined in [RFC8754].

## 7. Acknowledgements

The authors would like to thank Joel M. Halpern, Greg Mirsky, Bob Hinden, Loa Andersson and Gaurav Naik for their review comments.

## 8. Contributors

The following people have contributed to this document:

Robert Raszuk
Bloomberg LP
Email: robert@raszuk.net

John Leddy
Individual
Email: john@leddy.net

Gaurav Dawra
LinkedIn
Email: gdawra.ietf@gmail.com

Bart Peirens
Proximus
Email: bart.peirens@proximus.com

Nagendra Kumar
Cisco Systems, Inc.
Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
Email: cpignata@cisco.com

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

Frank Brockners
Cisco Systems, Inc.
Germany
Email: fbrockne@cisco.com

Darren Dukes
Cisco Systems, Inc.
Email: ddukes@cisco.com

Cheng Li
Huawei
Email: chengli13@huawei.com

Faisal Iqbal
Individual
Email: faisal.ietf@gmail.com

## 9. References

### 9.1. 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>.

[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>.

### 9.2. Informative References

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-15 (work in progress), March 2020.

[I-D.matsushima-spring-srv6-deployment-status]
Matsushima, S., Filsfils, C., Ali, Z., Li, Z., and K. Rajaraman, "SRv6 Implementation and Deployment Status", draft-matsushima-spring-srv6-deployment-status-07 (work in progress), April 2020.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[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>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC5476] Claise, B., Ed., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, DOI 10.17487/RFC5476, March 2009, <https://www.rfc-editor.org/info/rfc5476>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <https://www.rfc-editor.org/info/rfc5837>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <https://www.rfc-editor.org/info/rfc7012>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[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>.

[RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <https://www.rfc-editor.org/info/rfc8403>.

## Authors' Addresses

Zafar Ali
Cisco Systems

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems

Email: cfilsfil@cisco.com

Satoru Matsushima
Softbank

Email: satoru.matsushima@g.softbank.co.jp

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach Chen
Huawei

Email: mach.chen@huawei.com