SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 30, 2018

Z. Ali
C. Filsfils
N. Kumar
C. Pignataro
F. Iqbal
R. Gandhi
Cisco Systems, Inc.
J. Leddy
Comcast
S. Matsushima
SoftBank
R. Raszuk
Bloomberg LP
B. Peirens
Proximus
G. Naik
Drexel University
February 26, 2018

Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)
draft-ali-spring-srv6-oam-00.txt

Abstract

This document describes mechanisms for Operations, Administration, and Maintenance (OAM) in Segment Routing with IPv6 data plane (SRv6) network.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes mechanisms for Operations, Administrations, and Maintenance (OAM) in Segment Routing using IPv6 data plane (SRv6) networks.

Additional mechanisms will be added in a future revision of the document.

2. Conventions Used in This Document

2.1. Abbreviations

ECMP: Equal Cost Multi-Path.

SID: Segment ID.

SL: Segment Left.

SR: Segment Routing.

SRH: Segment Routing Header.

SRv6: Segment Routing with IPv6 Data plane.

TC: Traffic Class.

UCMP: Unequal Cost Multi-Path.

2.2. Terminology and Reference Topology

In this document, the simple topology shown in Figure 1 is used for illustration.

```
                             --------
    +------------------------| N100 |------------------------+
    |                        --------                        |
    |                                                        |
      ====== link1====== link3------ link5====== link9------
      ||N1||======||N2||======| N3 |======||N4||======| N5 |
      ||  ||------||  ||------|    |------||  ||------|    |
      ====== link2====== link4------ link6======link10------
                     |                      |
                     |        ------        |
                     +--------| N6 |--------+
                       link7  |    | link8
                              ------
```

Figure 1: Reference Topology

In the reference topology:

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

Nodes N3, N5 and N6 are classic IPv6 nodes.

Node 100 is a controller.

Node Nk has a classic IPv6 loopback address Bk::/128

Node Nk has Ak::/48 for its local SID space from which Local SIDs are explicitly allocated.

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::.

Ak::0 is explicitly allocated as the END function at Node k.

Ak::Cij is explicitly allocated as the END.X function at node k towards neighbor node i via jth Link between node i and node j. e.g., A2::C31 represents END.X at N2 towards N3 via link3 (the 1st link between N2 and N3). Similarly, A4::C52 represents the END.X at N4 towards N5 via link10.

<S1, S2, S3> represents a SID list where S1 is the first SID and S3 is the last SID. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID (S1) in the SRH is the first SID and the leftmost SID (S3) in the SRH is the last SID.

(SA, DA) (S3, S2, S1; SL) represents an IPv6 packet, SA is the IPv6 Source Address, DA the IPv6 Destination Address, (S3, S2, S1; SL) is the SRH header that includes the SID list <S1, S2, S3>.

SR policy is defined in Section 3 of [I-D.spring-segment-routing-policy].

3. OAM Mechanisms

This section describes how ping and traceroute mechanisms can be used in an SRv6 network. Additional OAM mechanisms will be added in a future revision of the document.

3.1. Ping

[RFC4443] describes Internet Control Message Protocol for IPv6 (ICMPv6) that is used by IPv6 devices for network diagnostic and error reporting purposes. As Segment Routing with IPv6 data plane (SRv6) simply adds a new type of Routing Extension Header, existing ICMPv6 mechanisms can be used in an SRv6 network. This section describes the applicability of ICMPv6 in the SRv6 network and how the existing ICMPv6 mechanisms can be used for providing OAM functionality.

Throughout this document, unless otherwise specified, the acronym ICMPv6 refers to multi-part ICMPv6 messages [RFC4884]. The document does not propose any changes to the standard ICMPv6 [RFC4443], [RFC4884] or standard ICMPv4 [RFC792].

There is no hardware or software change required for ping operation at the classic IPv6 nodes in an SRv6 network. This includes the classic IPv6 node with ingress, egress or transit roles. Furthermore, no protocol changes are required to the standard ICMPv6 [RFC4443], [RFC4884] or standard ICMPv4 [RFC792]. In other words, existing ICMP ping mechanisms work seamlessly in SRv6 networks.

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

3.1.1. Classic Ping

The existing mechanism to ping a remote IP prefix, 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 prefix 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 node N2 and node N3, which have IGP metric set to 100. User issues a ping from node N1 to a loopback of node N5, via via segment list <A2::C31, A4::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 <A2::C31, A4::C52>.

```
> ping B5:: via segment-list A2::C31, A4::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:

- o Node N1 initiates an ICMPv6 ping packet with SRH as follows (B1::,A2::C31)(B1::, A4::C52, A2::C31, SL=2, NH: ICMPv6)(ICMPv6 Echo Request).

- o Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (A2::C31) on the echo request packet.

- o Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA A4::C52 in the IPv6 header.

- o Node N4, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it observes the END.X function (A4::C52) with PSP (Penultimate Segment Popping) on the echo request packet and removes the SRH and forwards the packet across link10 to N5.

- o The echo request packet at N5 arrives as an IPv6 packet without a SRH. Node N5, which is a classic IPv6 node, performs the standard IPv6/ICMPv6 processing on the echo request and responds, accordingly.

3.1.2. Pinging SID Function

The classic ping described in the previous section cannot be used to ping a remote SID function, as explained using an example in the following.

Consider the case where the user wants to ping the remote SID function A4::C52, via A2::C31, from node N1. Node N1 constructs the ping packet (B1::0, A2::C31)(A4::C52, A2::C31, SL=1;NH=ICMPv6)(ICMPv6 Echo Request). When the node N4 receives the ICMPv6 echo request with DA set to A4::C52 and next header set to ICMPv6, it silently drops it (as per [I-D.filsfils-spring-srv6-network-programming]). To solve this

problem, the initiator needs to mark the ICMPv6 echo request as an OAM packet.

The OAM packets are identified either by setting the O-bit in SRH [I-D.6man-segment-routing-header] or by inserting the SID Function END.OTP at an appropriate place in the SRH [I-D.filsfils-spring-srv6-network-programming].

In an SRv6 network, the user can exercise two flavors of the ping: end-to-end ping or segment-by-segment ping, as outlined in the following.

3.1.2.1. End-to-end Ping Using END.OTP

Consider the same example where the user wants to ping a remote SID function A4::C52 , via A2::C31, from node N1. To force a punt of the ICMPv6 echo request at the node N4, node N1 inserts the SID function END.OTP just before the target SID A4::C52 in the SRH. The ICMPv6 echo request is processed at the individual nodes along the path as follows:

- o Node N1 initiates an ICMPv6 ping packet with SRH as follows (B1::0, A2::C31)(A4::C52, A4::OTP, A2::C31; SL=2; NH=ICMPv6)(ICMPv6 Echo Request).

- o Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (A2::C31) on the echo request packet.

- o Node N3 receives the packet as follows (B1::0, A4::OTP)(A4::C52, A4::OTP, A2::C31 ; SL=1; NH=ICMPv6)(ICMPv6 Echo Request). Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA A4::OTP in the IPv6 header.

- o When node N4 receives the packet (B1::0, A4::OTP)(A4::C52,A4::OTP, A2::C31 ; SL=1; NH=ICMPv6)(ICMPv6 Echo Request), it processes the SID Function END.OTP, as described in the pseudocode in [I-D.filsfils-spring-srv6-network-programming]. The packet gets punted to the ICMPv6 process for processing. The ICMPv6 process checks if the next SID in SRH (the target SID A4::C52) is locally programmed.

- o If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA1 by IANA)", Code: "SID not locally implemented (TBA2 by IANA)"); otherwise a success is returned.

3.1.2.2. Segment-by-segment Ping Using O-bit (Proof of Transit)

Consider the same example where the user wants to ping a remote SID function A4::C52 , via A2::C31, from node N1. However, in this ping, the node N1 wants to get a response from each segment node in the SRH. In other words, in the segment-by-segment ping case, the node N1 expects a response from node N2 and node N4 for their respective local SID function.

To force a punt of the ICMPv6 echo request at node N2 and node N4, node N1 sets the O-bit in SRH [I-D.6man-segment-routing-header]. 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 (B1::0, A2::C31)(A4::C52, A2::C31; SL=1, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request).

- When node N2 receives the packet (B1::0, A2::C31)(A4::C52, A2::C31; SL=1, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request) packet, it processes the O-bit in SRH, as described in the pseudo code in [I-D.filsfils-spring-srv6-network-programming]. A time-stamped copy of the packet is punted to the ICMPv6 process in control plane for processing. Node N2 continues to apply the A2::C31 SID function on the original packet and forwards it, accordingly. Due to SRH.Flags.O=1, Node N2 also disables the PSP behaviour, i.e., does not remove the SRH. The ICMPv6 process at node N2 checks if its local SID (A2::C31) is locally programmed or not and responds to the ICMPv6 Echo Request.

- If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA1 by IANA)", Code: "SID not locally implemented (TBA2 by IANA)"); otherwise a success is returned. Note that, as mentioned in [I-D.filsfils-spring-srv6-network-programming], if node N2 does not support the O-bit, it simply ignores it and process the local SID, A2::C31.

- Node N3, which is a classic IPv6 node, performs standard IPv6 processing. Specifically, it forwards the echo request based on DA A4::C52 in the IPv6 header.

- When node N4 receives the packet (B1::0, A4::C52)(A4::C52, A2::C31; SL=0, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request), it processes the O-bit in SRH, as described in the pseudo code in [I-D.filsfils-spring-srv6-network-programming]. A time-stamped copy of the packet is punted to the ICMPv6 process in control plane for processing. The ICMPv6 process at node N4 checks if its

local SID (A2::C31) is locally programmed or not and responds to the ICMPv6 Echo Request.

If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA1 by IANA)", Code: "SID not locally implemented (TBA2 by IANA)"); otherwise a success is returned.

Support for O-bit is part of node capability advertisement. This enables node N1 to know which segment nodes are capable of responding to the ICMPv6 echo request. Node N1 processes the echo responses and presents the data to the user, accordingly.

Please note that segment-by-segment ping described in this Section can be used to address proof of transit use-case.

3.2. Error Reporting

Any IPv6 node can use ICMPv6 control messages to report packet processing errors to the source that originated the datagram packet. To name a few such scenarios:

- If the router receives an undeliverable IP datagram, or

- If the router receives a packet with a Hop Limit of zero, or

- If the router receives a packet such that if the router decrements the packet's Hop Limit it becomes zero, or

- If the router receives a packet with problem with a field in the IPv6 header or the extension headers such that it cannot complete processing the packet, or

- If the router cannot forward a packet because the packet is larger than the MTU of the outgoing link.

In the scenarios listed above, the ICMPv6 response also contains the IP header, IP extension headers and leading payload octets of the "original datagram" to which the ICMPv6 message is a response. Specifically, the "Destination Unreachable Message", "Time Exceeded Message", "Packet Too Big Message" and "Parameter Problem Message" ICMPV6 messages can contain as much of the invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU [RFC4443], [RFC4884]. In an SRv6 network, the copy of the invoking packet contains the SR header. The packet originator can use this information for diagnostic purposes. For example, traceroute can use this information as detailed in the following.

3.3. 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.

3.3.1. Classic Traceroute

The existing mechanism to traceroute a remote IP prefix, 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 node may be an SRv6 node or a classic IPv6 node.

If an SRv6 capable ingress node wants to traceroute to IPv6 prefix 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>. This is illustrated using the topology in Figure 1. Assume all the links have IGP metric 10 except both links between node N2 and node N3, which have IGP metric set to 100. User issues a traceroute from node N1 to a loopback of node N5, via segment list <A2::C31, A4::C52>. Figure 3 contains sample output for the traceroute request.

```
> traceroute B5:: via segment-list A2::C31, A4::C52

Tracing the route to B5::

 1  2001:DB8:1:2:21:: 0.512 msec 0.425 msec 0.374 msec
    SRH: (B5::, A4::C52, A2::C31, SL=2)

 2  2001:DB8:2:3:31:: 0.721 msec 0.810 msec 0.795 msec
    SRH: (B5::, A4::C52, A2::C31, SL=1)

 3  2001:DB8:3:4:41:: 0.921 msec 0.816 msec 0.759 msec
    SRH: (B5::, A4::C52, A2::C31, SL=1)

 4  2001:DB8:4:5:52:: 0.879 msec 0.916 msec 1.024 msec
```

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 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 A2::C31 and A4::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 A2::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 A4::C52 (link10).

3.3.2. Traceroute to a SID Function

The classic traceroute described in the previous Section cannot be used to traceroute a remote SID function, as explained using an example as follows.

Consider the case where the user wants to traceroute the remote SID function A4::C52, via A2::C31, from node N1. Node N1 constructs the traceroute packet (B1::0, A2::C31, HC=1) (A4::C52, A2::C31, SL=1; NH=UDP) (traceroute probe). Even though Hop Count of the packet is set to 1, when the node N4 receives the traceroute probe with DA set to A4::C52 and next header set to UDP, it silently drops it (as per

[I-D.filsfils-spring-srv6-network-programming]). To solve this problem, the initiator node needs to mark the traceroute probe as an OAM packet.

The OAM packets are identified either by setting the O-bit in SRH [I-D.6man-segment-routing-header] or by inserting the SID Function END.OTP at an appropriate place in the SRH [I-D.filsfils-spring-srv6-network-programming].

In SRv6 networks, the user can exercise two flavors of the traceroute: hop-by-hop traceroute or overlay traceroute.

- o In hop-by-hop traceroute, user gets responses from all nodes including classic IPv6 transit nodes, SRv6 capable transit nodes as well as SRv6 capable segment endpoints. E.g., consider the example where the user wants to traceroute to a remote SID function A4::C52, via A2::C31, from node N1. The traceroute output will also display information about node N3, which is a transit (underlay) node.

- o The overlay traceroute, on the other hand, does not trace the underlay nodes. In other words, the overlay traceroute only displays the nodes that acts as SRv6 segments along the route. I.e., in the example where the user wants to traceroute to a remote SID function A4::C52, via A2::C31, from node N1, the overlay traceroute would only display the traceroute information from node N2 and node N4 and will not display information from node N3.

3.3.2.1. Hop-by-hop Traceroute Using END.OTP

In this Section, hop-by-hop traceroute to a SID function is exemplified using UDP probes. However, the procedure is equally applicable to other implementation of traceroute mechanism.

Consider the same example where the user wants to traceroute to a remote SID function A4::C52 , via A2::C31, from node N1. To force a punt of the traceroute probe only at the node N4, node N1 inserts the SID Function END.OTP just before the target SID A4::C52 in the SRH. The traceroute probe is processed at the individual nodes along the path as follows:

- o Node N1 initiates a traceroute probe packet with a monotonically increasing value of hop count and SRH as follows (B1::0,A2::C31)(A4::C52, A4::OTP, A2::C31; SL=2; NH=UDP)(Traceroute probe).

- o 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").

- o When Node N2 receives the packet with hop-count > 1, it performs the standard SRH processing. Specifically, it executes the END.X function (A2::C31) on the traceroute probe.

- o When node N3, which is a classic IPv6 node, receives the packet (B1::0, A4::OTP)(A4::C52, A4::OTP, A2::C31 ; HC=1, SL=1; NH=UDP)(Traceroute probe) 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").

- o 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 A4::OTP in the IPv6 header.

- o When node N4 receives the packet (B1::0, A4::OTP)(A4::C52, A4::OTP, A2::C31 ; SL=1; HC=1, NH=UDP)(Traceroute probe), it processes the SID Function END.OTP, as described in the pseudocode in [I-D.filsfils-spring-srv6-network-programming]. The packet gets punted to the traceroute process for processing. The traceroute process checks if the next SID in SRH (the target SID A4::C52) is locally programmed. If the target SID A4::C52 is locally programmed, node N4 responses with the ICMPv6 message (Type: Destination unreachable, Code: Port Unreachable). If the target SID A4::C52 is not a local SID, node N4 silently drops the traceroute probe.

Figure 4 displays a sample traceroute output for this example.

```
> traceroute srv6 A4::C52 via segment-list A2::C31

Tracing the route to SID function A4::C52

 1  2001:DB8:1:2::21 0.512 msec 0.425 msec 0.374 msec   SRH:
 (A4::C52, A4::OTP, A2::C31; SL=2)

 2  2001:DB8:2:3::31 0.721 msec 0.810 msec 0.795 msec   SRH:
 (A4::C52, A4::OTP, A2::C31; SL=1)

 3  2001:DB8:3:4::41 0.921 msec 0.816 msec 0.759 msec   SRH:
 (A4::C52, A4::OTP, A2::C31; SL=1)
```

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

3.3.2.2. Tracing SRv6 Overlay

The overlay traceroute does not trace the underlay nodes, i.e., only displays the nodes that acts as SRv6 segments along the path. This is achieved by setting the SRH.Flags.O bit.

In this section, overlay traceroute to a SID function is exemplified using UDP probes. However, the procedure is equally applicable to other implementation of traceroute mechanism.

Consider the same example where the user wants to traceroute to a remote SID function A4::C52 , via A2::C31, from node N1.

o Node N1 initiates a traceroute probe with SRH as follows (B1::0,A2::C31)(A4::C52, A2::C31; HC=64, SL=1, Flags.O=1; NH=UDP)(Traceroute Probe). Please note that the hop-count is set to 64 to skip the underlay nodes from tracing. The O-bit in SRH is set to make the overlay nodes (nodes processing the SRH) respond.

o When node N2 receives the packet (B1::0, A2::C31)(A4::C52,A2::C31; SL=1, HC=64, Flags.O=1; NH=UDP)(Traceroute Probe), it processes the O-bit in SRH, as described in the pseudocode in [I-D.filsfils-spring-srv6-network-programming]. A time-stamped copy of the packet gets punted to the traceroute process for processing. Node N2 continues to apply the A2::C31 SID function on the original packet and forwards it, accordingly. As SRH.Flags.O=1, Node N2 also disables the PSP flavor, i.e., does not remove the SRH. The traceroute process at node N2 checks if its local SID (A2::C31) is locally programmed. If the SID is not locally programmed, it silently drops the packet. Otherwise, it performs the egress check by looking at the SL value in SRH.

o As SL is not equal to zero (i.e., it's not egress node), node N2 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA1 by IANA)", Code: "O-bit punt at Transit (TBA3 by IANA)"). Note that, as mentioned in [I-D.filsfils-spring-srv6-network-programming], if node N2 does not support the O-bit, it simply ignores it and processes the local SID, A2::C31.

o When node N3 receives the packet (B1::0, A4::C52)(A4::C52, A2::C31; SL=0, HC=63, Flags.O=1; NH=UDP)(Traceroute Probe), performs the standard IPv6 processing. Specifically, it forwards the traceroute probe based on DA A4::C52 in the IPv6 header. Please note that there is no hop-count expiration at the transit nodes.

o When node N4 receives the packet (B1::0, A4::C52)(A4::C52,A2::C31;

SL=0, HC=62, Flags.O=1; NH=UDP)(Traceroute Probe), it processes the O-bit in SRH, as described in the pseudocode in [I-D.filsfils-spring-srv6-network-programming]. A time-stamped copy of the packet gets punted to the traceroute process for processing. The traceroute process at node N4 checks if its local SID (A2::C31) is locally programmed. If the SID is not locally programmed, it silently drops the packet. Otherwise, it performs the egress check by looking at the SL value in SRH. As SL is equal to zero (i.e., N4 is the egress node), node N4 tries to consume the UDP probe. As UDP probe is set to access an invalid port, the node N4 responses with the ICMPv6 message (Type: Destination unreachable, Code: Port Unreachable).

Figure 5 displays a sample overlay traceroute output for this example. Please note that the underlay node N3 does not appear in the output.

```
> traceroute srv6 A4::C52 via segment-list A2::C31

Tracing the route to SID function A4::C52

 1  2001:DB8:1:2::21 0.512 msec 0.425 msec 0.374 msec
    SRH: (A4::C52, A4::OTP, A2::C31; SL=2)

 2  2001:DB8:3:4::41 0.921 msec 0.816 msec 0.759 msec
    SRH: (A4::C52, A4::OTP, A2::C31; SL=1)
```

Figure 5: A sample output for overlay traceroute to a SID function

4. In-situ OAM Applicability

[I-D.brockners-inband-oam-requirements] describes motivation and requirements for In-situ OAM (iOAM). iOAM records operational and telemetry information in the data packet while the packet traverses the network of telemetry domain. iOAM complements out-of-band probe based OAM mechanisms such ICMP ping and traceroute by directly encoding tracing and the other kind of telemetry information to the regular data traffic.

[I-D.brockners-inband-oam-transport] describes transport mechanisms for iOAM data including IPv6 and Segment Routing traffic. Furthermore, [I-D.brockners-inband-oam-data] defines information encoding for iOAM data.

One of the application of iOAM is to perform inband traceroute. In SRv6 network, iOAM traceroute feature can be used to trace the order set of segment ID executed by SRv6 nodes for packet forwarding along

the packet path. This is achieved by recording the node details that the packet traversed in the packet header itself.

Another important application of iOAM is to perform delay measurement in anycast server scenarios. Anycast server deployment is commonly seen for redundancy and load balancing purpose. In SRv6 network, iOAM can be used to collect the timestamp from different anycats servers to measure the delay induced by each server within the anycast cluster that helps to provide SLA constrained services.

One of the other applications of iOAM is to provide the Proof of Transit (POT). Among other features of iOAM, SRv6 networks can use the POT feature of iOAM to verify that all the function SIDs in SRH have been executed before the packet is delivered to the destination. It can also ensure that the order of execution of the SID function has been consistent with the SRH contents.

More details on various applications of iOAM in SRv6 networks will be included in future versions of this document.

5. Seamless BFD Applicability

[RFC7880] defines Seamless BFD (S-BFD) architecture that simplifies BFD mechanism and enables it to perform path monitoring in a controlled and scalable manner. [RFC7881] describes the procedure to perform continuity check using S-BFD in different environments including IPv6 networks. Section 5.1 of [RFC7881] explains the SBFDInitiator specification and procedure to initiate S-BFD control packet in IP and MPLS network. The specification described for IP-routed S-BFD control packet is also directly applicable to the SRv6 network.

S-BFD has a fast bootstrapping capability. Furthermore, in S-BFD, only the ingress is required to keep BFD states; the egress and transit node does not have any knowledge of the BFD session. These attributes of S-BFD make it an excellent candidate for rapid failure detection in the SRv6 network. More details on various S-BFD usage on the SRv6 network will be included in a future version.

6. Monitoring of SRv6 Paths

In the recent past, network operators are interested in performing network OAM functions in a centralized manner. Various data models like YANG are available to collect data from the network and manage it from a centralized entity.

The SR technology enables a centralized OAM entity to perform path monitoring without control plane intervention on monitored nodes.

[I-D.ietf-spring-oam-usecase] describes such centralized OAM mechanism. Specifically, it 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 intervention on the nodes. However, the document focuses on SR networks with MPLS data plane. The same concept is also applicable to the SRv6 networks. This document describes how the concept can be used to perform path monitoring in an SRv6 network as follows.

In the reference topology in Figure 1, N100 is the controller implementing an END function A100::. In order to verify a segment list <A2::C31, A4::C52>, N100 generates a probe packet with SRH set to (A100::, A4::C52, A2::C31, SL=2). The controller routes the probe packet towards the first segment, which is A2::C31. N2 performs the standard SRH processing and forwards it over link3 with the DA of IPv6 packet set to A4::C52. N4 also performs the normal SRH processing and forwards it over link10 with the DA of IPv6 packet set to A100::. This makes the probe packet loop back to the controller.

In our 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. In other words, 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.

7. 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], [RFC792] and RFCs that updates these RFCs.

8. IANA Considerations

This document requests IANA to allocate a new Type for ICMPv6 message for "SRv6 OAM".

9. References

9.1. Normative References

[RFC792] J. Postel, "Internet Control Message Protocol", RFC 792, September 1981.

[RFC4443] A. Conta, S. Deering, M. Gupta, Ed., "Internet Control

Message Protocol (ICMPv6) for the Internet Protocol
Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4884] R. Bonica, D. Gan, D. Tappan, C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.

[RFC5837] A. Atlas, Ed., R. Bonica, Ed., C. Pignataro, Ed., N. Shen, JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.

[RFC7880] C.Pignataro, D.Ward, N.Akiya, M.Bhatia, S.Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, July 2016.

[RFC7881] C.Pignataro, D.Ward, N.Akiya, "Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS", RFC 7881 July 2016.

[I-D.filsfils-spring-srv6-network-programming] C. Filsfils, et al., "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming, work in progress.

[I-D.6man-segment-routing-header] Previdi, S., Filsfils, et al, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.

9.2. Informative References

[I-D.ietf-spring-oam-usecase] A Scalable and Topology-Aware MPLS Dataplane Monitoring System. R. Geib, C. Filsfils, C. Pignataro, N. Kumar, draft-ietf-spring-oam-usecase, work in progress.

[I-D.brockners-inband-oam-data] F. Brockners, et al., "Data Formats for In-situ OAM", draft-brockners-inband-oam-data, work in progress.

[I-D.brockners-inband-oam-transport] F.Brockners, at al., "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport, work in progress.

[I-D.brockners-inband-oam-requirements] F.Brockners, et al., "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements, work in progress.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy for Traffic Engineering",

draft-filsfils-spring-segment-routing-policy, work in progress.

10. Acknowledgments

To be added.

Authors' Addresses

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

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

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

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

Faisal Iqbal
Cisco Systems, Inc.
Email: faiqbal@cisco.com

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

John Leddy
Comcast
Email: John_Leddy@cable.comcast.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY10022, USA
Email: robert@raszuk.net

Satoru Matsushima
SoftBank
Japan
Email: satoru.matsushima@g.softbank.co.jp

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

Gaurav Naik
Drexel University
United States of America
Email: gn@drexel.edu