

spring
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2020

Z. Ali
C. Filsfils
N. Nainar
C. Pignataro
F. Clad
F. Iqbal
Cisco Systems, Inc.
X. Xu
Alibaba
November 1, 2019

**OAM for Service Programming with Segment Routing
draft-ali-spring-sr-service-programming-oam-02**

Abstract

This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP 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 May 4, 2020.

Copyright Notice

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

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

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

Table of Contents

1.	Introduction	2
2.	Requirements notation	2
3.	Terminology	3
4.	Document Scope	3
5.	Programmable OAM	3
5.1.	Service Programming OAM Packet Marker	3
5.2.	OAM with SR-aware services	3
5.3.	OAM with SR-unaware services	4
5.4.	Controlling OAM packet processing in Services	4
6.	OAM for Service Programming with SRv6	4
6.1.	OAM Tool Kit	5
6.1.1.	OAM Flag Processing	5
6.1.2.	OAM Segment ID	5
6.1.3.	ICMP	5
6.1.4.	UDP Probes	6
6.1.5.	In-band OAM	6
7.	OAM for Service Programming with SR-MPLS	6
8.	IANA Considerations	7
9.	Security Considerations	7
10.	Acknowledgement	7
11.	Normative References	7
	Authors' Addresses	8

[1.](#) Introduction

[I-D.ietf-spring-sr-service-programming] defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture. This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

[2.](#) Requirements notation

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](#)].

3. Terminology

This document uses the terminologies defined in [[RFC8402](#)], [[I-D.filsfils-spring-srv6-network-programming](#)] [[I-D.ietf-spring-sr-service-programming](#)] and so the readers are expected to be familiar with the same.

4. Document Scope

The initial focus of this document to define and document the machinery required to apply OAM mechanisms on SRv6 based service programming.

Future version of this document will include the required details to apply OAM mechanism on other data planes.

5. Programmable OAM

Section 4 of [[I-D.ietf-spring-sr-service-programming](#)] introduces Service segments and the procedure of service programming when the services are SR-aware and SR-unaware. By integrating the OAM functionality in the services, versatile OAM tool kits can be used to execute programmable OAM for service programming with Segment Routing.

This section describes the procedure to perform basic OAM mechanisms such as path validation and path tracing of Service Programming environment in Segment Routing network.

5.1. Service Programming OAM Packet Marker

Any services upon receiving OAM packet may apply the service treatment if it cannot differentiate the OAM packet from normal data packet. Depending on the service type, service treatment on OAM packet may result in dropping the OAM probe packet that may cause uncertainty in OAM mechanism.

To avoid such uncertainty, any node that is originating the OAM probe for Service Programming OAM MUST mark the packet as OAM packet so that the services can differentiate the OAM packet from data traffic.

5.2. OAM with SR-aware services

As defined in section 4.1 of [[I-D.ietf-spring-sr-service-programming](#)], an SR-aware service can process the SR information in the packet header such as performing lookup or executing the next segment etc. An SR-aware service may need to identify the packet payload and/or interpret SR information

to apply the right policy to the received packet. While processing SR information in the packet header, it can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet.

An SR-aware service SHOULD skip applying the service on the OAM packet while forwarding the packet to the next segment or IP address. As defined in [section 9](#), a local policy may be used to control any malicious use of OAM marker.

5.3. OAM with SR-unaware services

As defined in section 4.2 of [\[I-D.ietf-spring-sr-service-programming\]](#), an SR-unaware service may be a legacy service that is not able to process the SR information in the packet header. SR Proxy, an entity that is external to the service is used to handle the SR information processing on behalf of the service. SR Proxy will remove the SR header before forwarding the packet to SR-unaware services to avoid any erroneous decision due to the presence of SR header that the service cannot recognize.

While processing SR information in the packet header, SR proxy can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet. SR Proxy MUST skip forwarding the packets with OAM marker to the service while forwarding the packet to the next segment or IP address. As defined in [section 9](#), a local policy may be used to control any malicious use of OAM marker.

5.4. Controlling OAM packet processing in Services

As mentioned in the above sections, SR-aware service or the SR proxy can use the OAM marker to differentiate the OAM packet from data packet to skip the service treatment. Any intentional or unintentional use of OAM marker in data traffic may result in skipping service treatment for data traffic. To avoid such condition, a local policy will be used in the SR-aware service or SR Proxy that SHOULD rate limit or MAY drop the packets received with OAM marker.

6. OAM for Service Programming with SRv6

[\[I-D.ietf-6man-segment-routing-header\]](#) defines SRH.Flags.O-bit in SRH header. When service programming is implemented with SRv6 dataplane, SRH.Flags.O-bit is used as OAM marker. An IPv6 packet received with a local END.OP or END.OTP SID is also considered as an OAM packet.

Any node that is originating OAM probe to a service in SRv6 dataplane MUST set SRH.Flags.O-bit in the SRH.

6.1. OAM Tool Kit

This section describes the availability of different tool kits that can be used to perform OAM functionality for Service Programming with SRv6 dataplane.

6.1.1. OAM Flag Processing

An SR-aware service or SR proxy MUST implement the SRH.Flags.0 bit. An SR-aware service SHOULD skip applying the service to the packet when SRH.Flags.0-bit is set and SHOULD forward the packet based on the next header. SR Service Proxy MUST skip applying the service to the packet when SRH.Flags.0-bit is set and SHOULD forward the packet based on the next header.

An SR-aware service and SR proxy may choose to time-stamp and punt the packet with SRH.Flags.0-bit set for further processing and this is a local implementation matter.

6.1.2. OAM Segment ID

[Section 3.2](#) of [\[\[I-D.ali-spring-srv6-oam\]\]](#) defines OAM segment ID and the associated forwarding semantics to implement the punt behavior for OAM packets. Specifically, the draft defines END.OP and END.OTP SIDs. An IPv6 packet received with DA set to a local END.OP or END.OTP SID is considered as an OAM packet.

Any service policy head end MAY include OAM segment ID in the desired segment list position of SRH. The inclusion of OAM SID in SRH can be used to control the services that are required to punt the OAM packet for processing.

6.1.3. ICMP

There is no hardware or software changes required to use ICMP for Ping operation. It can be triggered from the service policy head end or from any classical IPv6 nodes by sending ICMPv6 Echo Request. The existing ICMP Ping mechanism works seamlessly in SRv6 dataplane with no protocol changes required to the standard ICMPv6 [\[\[RFC4443\]\]](#), [\[\[RFC4884\]\]](#) or the standard ICMPv4 [\[\[RFC0792\]\]](#).

An SR-aware service SHOULD skip the service and forward to next segment based on the SR information in the packet header. An SR Service Proxy MUST skip the service and forward to next segment based on the SR information in the packet header.

6.1.3.1. Pinging Service SID Function

When a remote node pings a service segment, it MUST set SRH.Flags.0 = 1. If the target service segment is implemented with USP behavior, the ICMP packet can be constructed without adding END.OP or END.OTP SIDs defined in [[I-D.ali-spring-srv6-oam](#)]. However, if the target service SID observes a PSP behavior, the sender needs to insert END.OP/ END.OTP SIDs before the target service SID in the segment-list. In either case, the target SR-aware service or SR proxy receives the ICMP echo request with either SRH.Flags.0-bit set or with the local END.OP or END.OTP SID. In both cases, the packet is punted for slow-path processing and service is skipped.

The Egress node process the packet as per procedure defined in [[I-D.ali-spring-srv6-oam](#)]. The Egress checks if the target SID is locally programmed or not.

If the target SID is not locally programmed, the Egress responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned [[I-D.ali-spring-srv6-oam](#)].

6.1.4. UDP Probes

A classic traceroute mechanism relies on UDP probes by sending packets with sequentially incrementing the TTL. More details are available in section 4.3.1 of [[I-D.ali-spring-srv6-oam](#)].

An SR-aware service or SR proxy upon receiving the probe with TTL=1, may follow the traditional behavior of replying with ICMPv6 Time Exceeded Message (Type 3) as defined in [[RFC4443](#)] without applying the service.

Use of SRH.Flags.0 bit and END.OP/ END.OTP SIDs as OAM marker in the UDP probe for trace route is same as discussed for ICMPv6 ping discussed in the last section.

6.1.5. In-band OAM

To be Updated.

7. OAM for Service Programming with SR-MPLS

To be updated.

8. IANA Considerations

None.

9. Security Considerations

A local policy may be used to control any malicious use of OAM marker. More details are to be added in a future revision of the document.

10. Acknowledgement

Authors would like to thank Bruno Decraene for his review and useful comments.

11. Normative References

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

Ali, Z., Filsfils, C., Kumar, N., Pignataro, C., faiqbal@cisco.com, f., Gandhi, R., Leddy, J., Matsushima, S., Raszuk, R., daniel.voyer@bell.ca, d., Dawra, G., Peirens, B., Chen, M., and G. Naik, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", [draft-ali-spring-srv6-oam-02](#) (work in progress), October 2018.

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", [draft-filsfils-spring-srv6-network-programming-07](#) (work in progress), February 2019.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-26](#) (work in progress), October 2019.

[I-D.ietf-spring-sr-service-programming]

Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", [draft-ietf-spring-sr-service-programming-00](#) (work in progress), October 2019.

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

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, [RFC 4443](#), DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [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>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", [RFC 7665](#), DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cfilsfils@cisco.com

Nagendra Kumar Nainar
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Francois Clad
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Faisal Iqbal
Cisco Systems, Inc.
2000 Innovation Dr
Ottawa, ON 3E8
Canada

Email: faiqbal@cisco.com

Xiaohu Xu
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

