SPRING
Internet-Draft
Intended status: Informational
Expires: May 2, 2020

R. Nakamura, Ed.
The University of Tokyo
Y. Ueno
NTT Communications Corporation
T. Kamata
Cisco Systems, Inc.
October 30, 2019

An Experiment of SRv6 Service Chaining at Interop Tokyo 2019 ShowNet
draft-upa-srv6-service-chaining-exp-00

Abstract

This document reports lessons learned from an experimental deployment of service chaining with Segment Routing over the IPv6 data plane (SRv6) at an event network. The service chaining part of the network was comprised of four SRv6-capable nodes (three products from different vendors), five SRv6 proxy nodes (two products from different vendors and three open source software), and six services. This network was deployed at Interop Tokyo 2019, and it successfully provided network connectivity and services to all the exhibitors and visitors on the event.

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 2, 2020.

Copyright Notice

Table of Contents

1. Introduction

Standardizing functionalities for SRv6 service programming is still an ongoing agenda. Meanwhile, some fundamental parts have begun to be implemented: basic transit behaviors, endpoint functions at ingress and egress nodes [I-D.filsfils-spring-srv6-network-programming], and SR proxies for SR-unaware services [I-D.ietf-spring-sr-service-programming]. Trying out such running codes and devices would clarify statuses of recent implementations and provide feedback to current and future standardization processes.

To clarify the current status, we conducted an experiment of SRv6 service chaining at Interop Tokyo. Interop Tokyo is a large exhibition of networking technologies, and ShowNet is the event network built at Interop Tokyo while demonstrating new technologies and conducting interoperability tests. In 2019, we deployed SRv6 service chaining with the latest implementations at ShowNet and provided Internet connectivity for over 200 exhibitors and over 155,000 visitors. Through the experiment, we made several

observations from both implementation and specification perspectives: transparency of SRv6 header (SRH), services that cannot co-exist with the original masquerading proxy, how to integrate service liveness into advertisement of service segments, behaviors of TTL decrement on the masquerading proxy, necessity of control planes, and behavior of the longest prefix match and SRv6 functions. This document reports and discusses these lessons learned from the experiment of SRv6 service chaining.

2. Terminology

This document leverages the terminology proposed in [RFC8402], [I-D.ietf-spring-segment-routing-policy], and [I-D.ietf-spring-sr-service-programming].

3. SRv6 service chaining at Interop Tokyo 2019 ShowNet

The experiment was conducted on Interop Tokyo, from June 12 to June 14, 2019. This section describes the overview of SRv6 service chaining at ShowNet 2019.

The devices contributed to the experiment are listed below:

FUNCTION	NAME	CONTRIBUTOR
T.Insert	FX201	Furukawa Electric
T.Encaps	FX201	Furukawa Electric
End.DT4	NCS55A1	Cisco Systems
	NE40E-F1A	Huawei
End	FX201	Furukawa Electric
End.AM	FX201	Furukawa Electric
	Kamuee	NTT Communications
	VPP	FD.io
End.AN	TM VNFS	Trend Micro
Variant of End.AD	Two open source software on Linux	

The variant of End.AD was the SRv6 tagging proxy designed for IPv4 traffic encapsulated in SRv6 and implemented for ShowNet. The detail is described in [I-D.eden-srv6-tagging-proxy]. In addition to the SRv6-capable devices, we deployed six SR-unaware services into the service chaining. These services were applied to user traffic in accordance with service menus.

SERVICE	NAME	CONTRIBUTOR
Security	FortiGate 3601E	Fortinet
	Lastline Defender	Lastline
	PA-5280	Palo Alto Networks
	Thunder 3230S CFW	A10 Networks
	TBA	
CGN	Thunder 7440-11 CFW	A10 Networks

Note that the detailed security services were different depending on each device (e.g., DPI, URL filtering, etc).

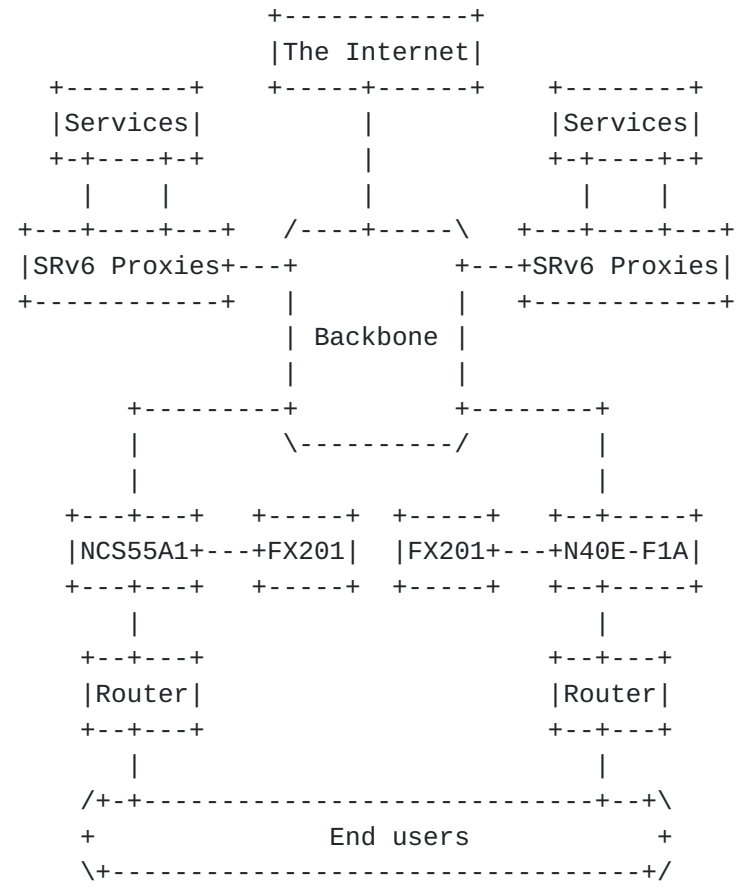

Figure 1: Overview of SRv6 Service Chaining at ShowNet

Figure 1 illustrates an overview of the experimental topology. End users, i.e., exhibitors and visitors, were accommodated by two non-SR-capable routers. The two FX201 had default routes for users, and the FX201 applied T.Insert and T.Encaps to received IPv6 and IPv4 packets from the users, respectively. The packets with the SRH were delivered to the SRv6 proxies by the active service segments. The SRv6 proxies connected to the backbone received the packets and

performed the proxy behaviors to take the packets through the services. Lastly, Segment List[0] representing endpoint functions took the packets to FX201 again for End and popping the SRH, or NCS55A1 or NE40E-F1A for End.DT4.

From the Internet to users, packets followed a similar path: FX201 inserted SRH to the packets, services were applied to the packets in the reverse order, and FX201 removed the SRH from IPv6 packets, or NCS55A1 or NE40E-F1A decapsulated IPv4 packets in the SRH and outer IPv6 headers.

The reason why we chose T.Insert instead of T.Encaps for IPv6 packets is to use the masquerading proxy (End.AM). SR proxies excluding End.AM require a proxy instance or an interface for receiving traffic returning from the service (IFACE-IN) for each SR service policy. In the ShowNet, there were over 200 individual users (exhibitors); therefore, there was the possibility that we needed to prepare over 200 proxy instances or interfaces for each service. It was possible, however, not reasonable for the temporal network for the three-days event. Therefore, we used T.Insert and End.AM for IPv6 packets to multiplex service chains on a proxy instance for a service.

End.AM is applicable to only SRv6 insertion; thus, IPv4 traffic still has the same issue. To address this issue, we designed the SRv6 tagging proxy, which is a new variant of End.AD, for IPv4 packets encapsulated in SRv6 [I-D.eden-srv6-tagging-proxy]. An XDP-based implementation was used for delivering all user traffic, and a Linux kernel-based implementation was also tested at ShowNet.

Although we faced some challenges described in the next section, this SRv6 service chaining finally fulfilled the role. It delivered all user traffic and applied the services during the period of Interop Tokyo 2019.

4. Lessons Learned

This section shares lessons learned from the experimental deployment.

4.1. Transparency of SRv6 header

First, we report that all the services contributed to ShowNet transparently delivered IPv6 packets under the End.AM proxies, although SRH is a new IPv6 extension header. There were no devices that dropped packets due to the unrecognized extension header.

4.2. Services that cannot co-exist with End.AM

When we designed the ShowNet, we intended to deploy a captive portal at service chaining. However, we could not accomplish that. The masquerading proxy assumes that the service under the proxy can only inspect, drop, or perform limited changes to the packets as noted in [I-D.ietf-spring-sr-service-programming]. Besides, it assumes that the packets returning from IFACE-IN have masqueraded SRH. This means that packets originated by SR-unaware services do not have SRH; therefore, the packets cannot be de-masqueraded. A captive portal is one of the examples that originate packets. These SR-unaware services cannot be integrated with the original masquerading proxy.

Instead, the variant 2 of masquerading proxy (Caching) defined in Section 6.4.3 of [I-D.ietf-spring-sr-service-programming] would be capable of handling such services.

4.3. Service liveness detection and conditional advertisement of service segments

Advertising service segments should be stopped when corresponding services are down to achieve redundancy of services in SRv6 service chaining. It is also expected that SR proxies are capable of stopping service segment advertisements. Meanwhile, the proxies contributed to ShowNet had not implemented such functionality yet because they were focusing on data plane functionalities at that time.

Link downs do not always represent service downs; services might be physical appliances behind switches or virtual machines on hypervisors. To detect failures on the services, we suggest that SR proxies should have some probing mechanisms for determining the statuses of services under the proxies and integrate the statuses with the advertisement process of the service segments. For example, Bidirectional Forwarding Detection (BFD) [RFC8562] between IFACE-IN and IFACE-OUT would determine the liveness of the services, and the liveness property could be integrated into triggers for advertising corresponding service segments.

4.4. TTL Decrement on SRv6 Proxies

We confirmed that there were variations on TTL decrement behaviors of the End.AM proxies. An implementation decreases TTL of outer IPv6 headers when sending packets to IFACE-OUT, and another implementation does not decrease TTL on IFACE-OUT. TTL decrement also occurs for forwarding packets from IFACE-IN to the next segments. As a result, TTL values of packets varied depending on the proxy implementations that the packets passed through. The variation of TTL decrement is

not a significant matter for just forwarding traffic; however, it would complicate Operation, Administration, and Maintenance (OAM) because traceroute results would also vary depending on implementations. Forwarding packets to IFACE-OUT is also layer-3 processing based on IPv6 headers and SRH; therefore, we suggest that TTL should be decreased on IFACE-OUT.

4.5. Control Plane Capabilities

The SR node implementations in ShowNet were not capable of control planes; therefore, we configured all the SR policies in ShowNet manually. Despite manually configuring SR nodes for service chaining is possible, it is hard to operate in realistic environments. Advertising SR policies via BGP [I-D.ietf-idr-segment-routing-te-policy] is also an essential capability for service chaining that is a work in progress [I-D.dawra-idr-bgp-ls-sr-service-segments].

4.6. Match Condition for Applying SRv6 Functions

The SRv6 node implementations contributed to ShowNet applied SRv6 functions by the longest prefix match: SRv6 functions are represented by pairs of destination prefixes for segments and functions. Packets from users to the Internet would have arbitrary destinations; therefore, the transit behaviors must be associated with default routes. On the other hand, SR service policies inserted to packets vary depending on users and their service menus. To apply different SR service policies by the transit behaviors with default routes, we used VRFs on FX201 that performed T.Insert and T.Encaps. An SR service policy was associated with a transit behavior for a default route on a VRF. The user networks were attached to VRFs corresponding to their service menus. By this technique, we accomplished per-user service chains at the transit nodes.

Per-VRF transit behavior is a practical candidate for SRv6 service chaining. On the other hand, users and their traffic can be distinguished by source addresses of packets. Thus, applying SRv6 functions in accordance with source addresses or other fields in packets can also be a candidate implementation. For example, leveraging SRv6 functions as actions of BGP Flowspec [RFC5575] is a possible way for inserting SR policies into packets flexibly. This is a different adaptation of BGP Flowspec to SRv6 in addition to [I-D.ietf0-idr-srv6-flowspec-path-redirect].

5. IANA Considerations

This document has no IANA implications.

6. Security Considerations

The security requirements and mechanisms described in [RFC8402], [I-D.ietf-6man-segment-routing-header] and [I-D.filsfils-spring-srv6-network-programming] also apply to this document.

This document does not introduce any new security vulnerabilities.

7. Contributors

J. Cao (Huawei), T. Fujiwara (Furukawa Network Solution), M. Udaka (Furukawa Network Solution), Y. Oga (Furukawa Network Solution), Y. Ohara (NTT Communications), H. Shirokura (NTT Communications), Y. Yamada (Keysight Technologies), K. Noda (Keysight Technologies), L. Zhou (Spirent), T. Kawada (TOYO Corporation), T. Matsuba (TOYO Corporation), Y. Matsubayashi (Trend Micro), and H. Nishikawa (Trend Micro) substantially contributed to the content of this document.

8. Acknowledgements

The authors would like to thank all the members and contributors of Interop Tokyo 2019 ShowNet. The authors are thankful to Francois Clad for his comments.

9. Normative References

[I-D.dawra-idr-bgp-ls-sr-service-segments]
Dawra, G., Filsfils, C., daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Clad, F., and K. Talaulikar, "BGP-LS Advertisement of Segment Routing Service Segments", draft-dawra-idr-bgp-ls-sr-service-segments-02 (work in progress), July 2019.

[I-D.eden-srv6-tagging-proxy]
Ueno, Y., Nakamura, R., and T. Kamata, "SRv6 Tagging proxy", draft-eden-srv6-tagging-proxy-00 (work in progress), October 2019.

[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. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-07 (work in progress), July 2019.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-03 (work in progress), May 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.

[I-D.ietf0-idr-srv6-flowspec-path-redirect]
Velde, G., Patel, K., Li, Z., and H. Chen, "Flowspec Indirection-id Redirect for SRv6", draft-ietf0-idr-srv6-flowspec-path-redirect-02 (work in progress), July 2019.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[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>.

[RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <https://www.rfc-editor.org/info/rfc8562>.

Authors' Addresses

Ryo Nakamura (editor)
The University of Tokyo
Tokyo
JP

Phone: +81-3-5841-2710
Email: upa@haeena.net

Yukito Ueno
NTT Communications Corporation
Tokyo
JP

Phone: +80 90 3085 5274
Email: yukito.ueno@ntt.com

Teppei Kamata
Cisco Systems, Inc.
Tokyo
JP

Email: tkamata@cisco.com