

SPRING
Internet-Draft
Intended status: Informational
Expires: October 9, 2020

J. Guichard, Ed.
Futurewei Technologies Ltd.
C. Filsfils
Cisco Systems, Inc.
D. Bernier
Bell Canada
Z. Li
Huawei Technologies
F. Clad, Ed.
P. Camarillo
A. AbdelSalam
Cisco Systems, Inc.
April 07, 2020

Simplifying Firewall Rules with Network Programming and SRH Metadata
draft-guichard-spring-srv6-simplified-firewall-02

Abstract

A clear application of the SRv6 Network Programming model consists in steering, in a stateless manner, packets through a Service Function Chain (SFC). Each Service Function (SF) is identified by a segment. Each SF can enrich its operation thanks to metadata present in the SRH.

This document describes a practical use-case where the SF is a firewall and the metadata helps to drastically decrease the number of rules that need to be maintained by the operation team.

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 October 9, 2020.

Copyright Notice

Copyright (c) 2020 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.	Use-case overview	3
3.	Demo availability	5
4.	IANA Considerations	5
5.	Security Considerations	5
6.	Acknowledgements	5
7.	References	5
7.1.	Normative References	6
7.2.	Informative References	6
	Authors' Addresses	6

[1.](#) Introduction

The Segment Routing architecture is defined in [[RFC8402](#)].

The IPv6 instantiation of Segment Routing, also known as SRv6, leverages the Segment Routing Header (SRH) defined in [[RFC8754](#)] to encode a list of segments, as well as some complementary information in an IPv6 header. [[I-D.ietf-spring-srv6-network-programming](#)] builds upon the base SRv6 definition and introduces the concept of network programming. In a sense, the list of segments in the SRH is the source code of a network program, while the SRH TLVs represent the memory of that program.

Furthermore, [[I-D.ietf-spring-sr-service-programming](#)] describes how segments can be associated with Service Functions and defines SRH TLVs specifically designed for carrying service metadata. Together, these documents define an integrated solution for underlay, overlay and SFC that uses a single header and does not require any per-flow state in the network fabric.

2. Use-case overview

In an SR domain, firewall policies are applied to control how the various endpoints, users or applications are allowed to communicate between each other. These entities are categorized into classes for the purpose of applying policies to pools rather than individual entities. For example, the endpoints in Class1 may be allowed to communicate with those in either Class3 or Class4, but Class2 is can only communicate with Class4, and Class5 cannot communicate with any other class.

A reference diagram is depicted on Figure 1. An SRv6-enabled network interconnects 4 classes (Class1..4) and a firewall appliance is in charge of enforcing the network policies.

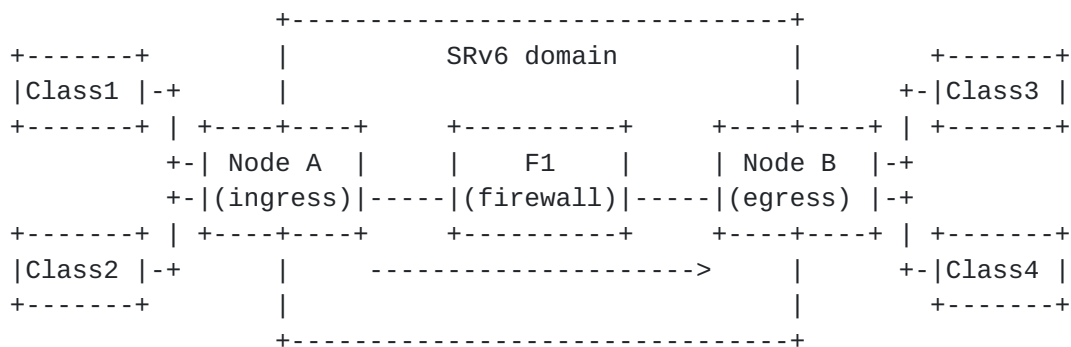


Figure 1: Base diagram

Node A is configured to steer the traffic coming from Class1 or Class2 and headed to Class3 or Class4 into an SRv6 service policy to Node B, via the firewall F1. As part of the steering process, Node A identifies the source and destination classes, encapsulates the traffic and attaches an SRH that contains the SR Policy SID-list, as well as the class information in the SRH TLVs. The procedure to identify the traffic classes is out of the scope of this document.

Node B is similarly configured to handle flows in the reverse direction.

The firewall F1 reads the SRH TLVs and decides to forward or drop the traffic based on the combination of the source and destination classes. The availability of class metadata allows the firewall rule-set size to scale with the number of valid (source class, destination class) pairs. This drastically simplifies the firewall configuration and operation compared to a traditional 5-tuple-based model with tens of thousands of entries.

In Figure 2, a traffic flow from Class1 to Class3 is steered into the SRv6 Policy "<B:F1:A::, B:B:D3::>", where "B:F1:A::" represents a service SID instantiated on the firewall F1 and "B:B:D3::" is an End.DX4 SID on the egress node B that sends the inner packet to Class3. The SRH "S-class" and "D-class" TLVs respectively represent the source and destination class identifiers. This traffic flow is allowed to traverse the firewall and reaches its final destination in Class3.

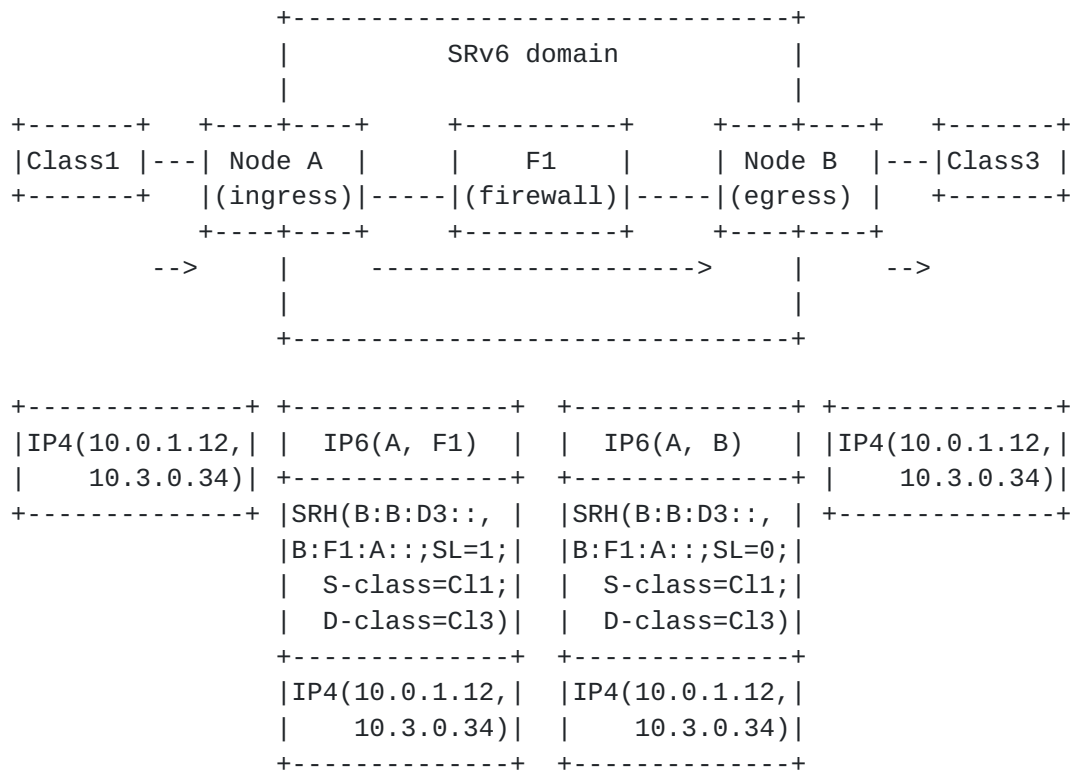


Figure 2: Traffic flow from Class1 to Class3

In Figure 3, a traffic flow from Class2 to Class3 is steered into the exact same SRv6 Policy "<B:F1:A::, B:B:D3::>". The SRH "S-class" and "D-class" TLVs are similarly populated with the source and destination class identifiers. However, "S-class=Cl2" and "D-class=Cl3" does not match an authorized class combination on the firewall. The traffic is considered as invalid and dropped at F1.

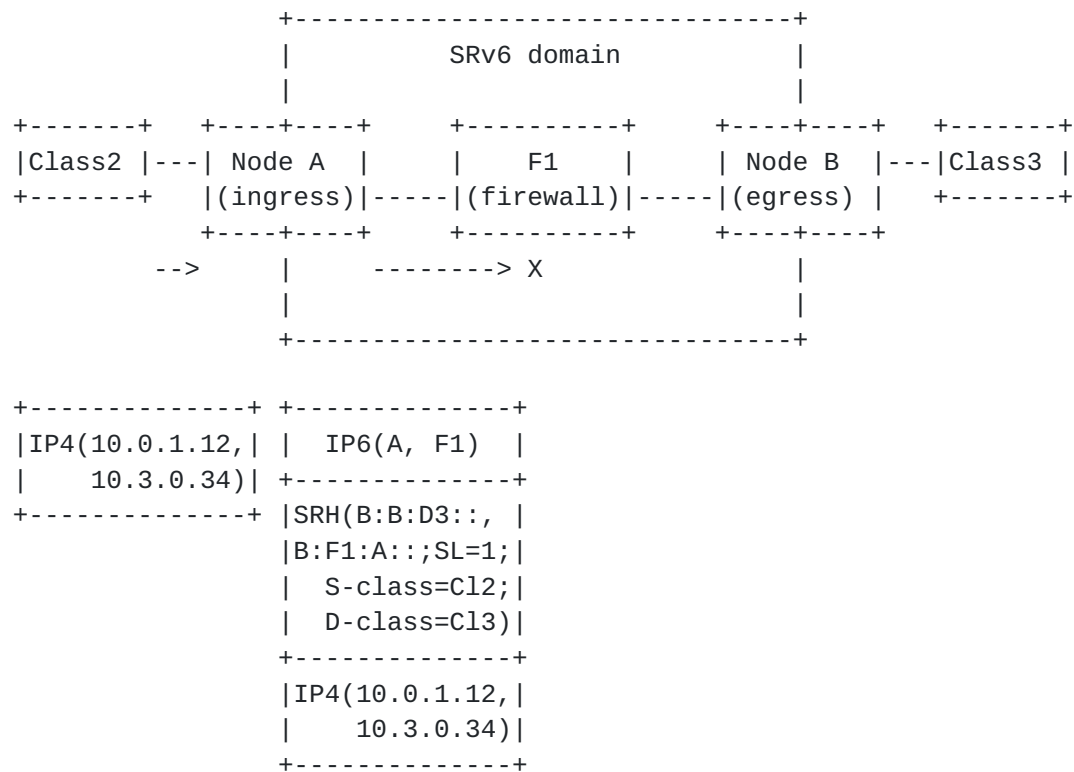


Figure 3: Traffic flow from Class2 to Class3

3. Demo availability

A working demo is available, using FD.io VPP [[FDio](#)] instances as ingress and egress routers and the iptables-based SERA firewall [[SERA](#)].

4. IANA Considerations

To be updated.

5. Security Considerations

To be updated.

6. Acknowledgements

To be updated.

7. References

7.1. Normative References

- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [RFC 8754](#), DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

7.2. Informative References

- [FDio] "The Fast Data Project", The Linux Foundation , 2018, <<https://fd.io>>.
- [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-02](#) (work in progress), March 2020.
- [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.
- [SERA] Abdelsalam, A., Salsano, S., Clad, F., Camarillo, P., and C. Filsfils, "SERA: SEgment Routing Aware Firewall for Service Function Chaining scenarios", IFIP Networking , May 2018.

Authors' Addresses

James N Guichard (editor)
Futurewei Technologies Ltd.

Email: james.n.guichard@futurewei.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cf@cisco.com

Daniel Bernier
Bell Canada

Email: daniel.bernier@bell.ca

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Francois Clad (editor)
Cisco Systems, Inc.

Email: fclad@cisco.com

Pablo Camarillo
Cisco Systems, Inc.

Email: pcamaril@cisco.com

Ahmed AbdelSalam
Cisco Systems, Inc.

Email: ahabdel@cisco.com

