Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 21, 2016

A. Cabellos
UPC-BarcelonaTech
S. Barkai
B. Perlman
Hewlett Packard Enterprise
V. Ermagan
F. Maino
Cisco Systems Inc
A. Rodriguez-Natal
UPC-BarcelonaTech

October 19, 2015

Map-Assisted SFC Proxy using LISP
draft-cabellos-sfc-map-assisted-proxy-00.txt

Abstract

This document specifies a map-assisted SFC proxy. The SFC proxy uses the LISP Mapping System to store the NSH header indexed by 5-tuple, before decapsulating and forwarding the packet to the legacy function. After the function has processed the packet, the SFC proxy retrieves the NSH header from the Mapping System to SFC encapsulate it.

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 RFC 2119 [RFC2119].

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 http://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 April 21, 2016.

Copyright Notice

Copyright (c) 2015 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 (http://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

The Locator/ID Separation Protocol (LISP) [RFC6830] is an overlay protocol that creates two namespaces: EIDs (End-point IDentifiers) and RLOCs (Routing LOCators). The LISP Mapping System stores the mappings between both namespaces, LISP provides a standard way for its data-plane elements, called xTRs, to store and retrieve mappings from the Mapping System to make forwarding decisions: Map-Request, Map-Request and Map-Reply. Finally, LISP also offers a flexible syntax for both EIDs and RLOCs by means of LCAFs [I-D.ietf-lisp-lcaf] to define what is an EID and what is an RLOC.

With such architecture in place, the LISP control-plane represents a programmable protocol. The Mapping System is a logically centralized database that stores network state, which is retrieved by data-plane nodes in a standard way to make decisions. Any external control plane can program the LISP Mapping System while any data-plane node can be map-assisted.

This document specifies a map-assisted SFC proxy [I-D.ietf-sfc-architecture]. An SFC acts on behalf the SFC unaware functions on the SFC domain. Basically the SFC Proxy removes the SFC encapsulation, forwards the packet to the SFC unaware function, receives back the packets and reapplies an SFC encapsulation. Specifically this document specifies how to map-assist the encapsulation operation by means of the LISP control-plane.

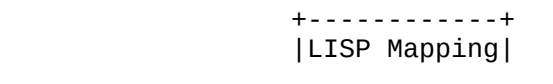
In short, the SFC Proxy before decapsulating the packet stores (Map-Registers) the NSH header (including Context Headers) [I-D.ietf-sfc-nsh] in the LISP Mapping System indexed by the 5-tuple of the packet {5-tuple->NSH}. After the SFC unaware function has processed the packet, the proxy retrieves (Map-Requests based on the 5-tuple of the packet) the NSH+Context headers to SFC encapsulate the packet.

This has two main benefits; first the SFC proxy is stateless and connectionless. Second, in some cases the legacy function may change the headers of the original packet, the SFC control plane can change the stored mapping {5-tuple->NSH} in the Mapping System accordingly and allow for fast reclassification by the proxy.

2. Overview

2.1. Flow example

This section shows a flow example of map-assisted SFC Proxy processing:



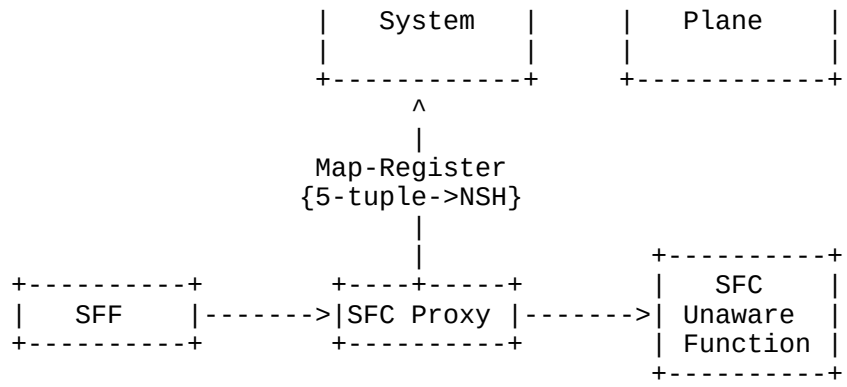


Figure 1.- SFC Proxy Decapsulation

1. An SFC proxy receives an SFC encapsulated packet as defined in the SFC architecture [I-D.ietf-sfc-architecture].

2. The SFC proxy Map-Registers the SFC encapsulation in the LISP Mapping System (figure 1), this includes the entire NSH header: Base Header, Service Header and Context Headers. The NSH header is indexed by the 5-tuple of the payload. Both the 5-tuple and the NSH header are encoded using two different LISP LCAFs, further details can be found in Section 3.

3. The SFC proxy forwards the packet to the SFC unaware function as specified in the SFC architecture [I-D.ietf-sfc-architecture].

4. The SFC unaware function processes the packet and sends it back to the SFC proxy.

5. Upon reception of the processed packet, the SFC proxy must SFC encapsulate the packet. For this it retrieves the NSH header from the LISP Mapping System using a Map-Request indexed by the 5-tuple of the received packet (figure 2). Once the packet is SFC encapsulated, the SFC proxy forwards it as defined in the SFC architecture [I-D.ietf-sfc-architecture].

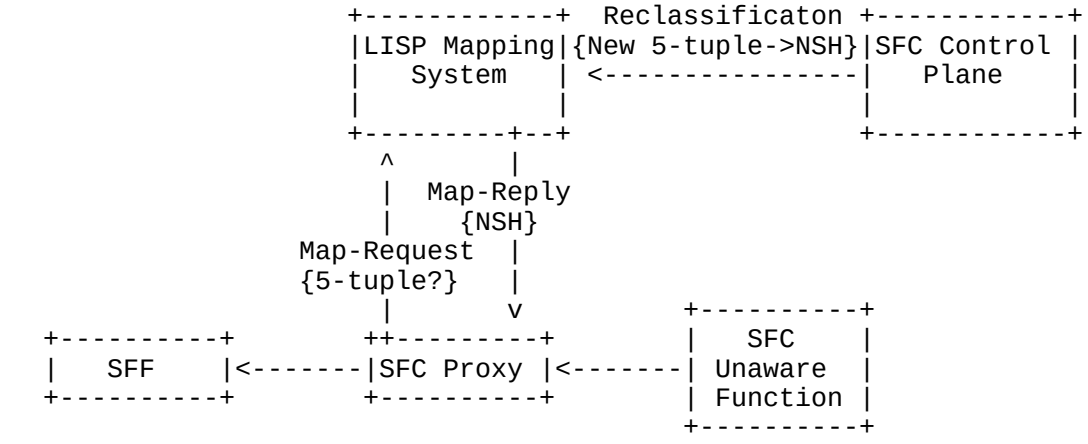


Figure 2.- SFC Proxy Encapsulation

2.2. Benefits of Map-Assisted SFC Proxies

The Map-Assisted encapsulation described in step 5 of the previous section brings the following benefits to the SFC architecture:

- The map-assisted SFC proxy is connectionless and stateless, as such it does not need to store state to forward packets from/to SFC unaware functions. Since the required state is stored in the

Mapping System, any other SFC proxy can receive the processed packets and SFC encapsulate them.

- In some scenarios the legacy functions may change the packet header and hence, the SFC proxy must re-classify it. With map-assisted SFC proxies, the SFC control-plane can change the stored state on the Mapping System to accordingly and allow map-assisted stateless reclassification by the SFC-Proxy. This is illustrated in the figure 2 by the "Reclassification" arrow. How the SFC control plane updates information on the LISP Mapping system is out of the scope of this document. In any case, please note that the SFC proxy still operates as described in this document and remains unaware of the reclassification.

3. Encoding of 5-tuple and NSH in LISP messages

This section describes the LCAFs used to encode both the 5-tuple and NSH header (Base, Service Path and Context Headers). The 5-tuple index is encoded in a LISP record as an EID while the NSH header as an RLOC.

3.1. Encoding of 5-tuple Index

The Multiple-tuple EID [I-D.rodrigueznatal-lisp-multi-tuple-eid] is used to encode the 5-tuple EID that indexes the NSH header, specifically using the "Exact Match" mode and EID mask-ken set to 0.

3.2. Encoding of NSH Header

The NSH header (Base Header, Service Path Header and Context Headers) [I-D.ietf-sfc-nsh] is encoded using the JSON Data Model Type LCAF as defined in [I-D.ietf-lisp-lcaf]. The header is encoded in binary format using BSON [BSON] as a single binary field (subtype "Generic binary subtype"):

document ::= int32 binary "\x00"

A LISP record only transports a single NSH header and all the "Loc" fields are ignored except "Loc-AFI" and "Locator".

4. SFC Proxy Processing

This section specifies the behavior of a map-assisted SFC Proxy, the proxy acts as specified in [I-D.ietf-sfc-architecture] with the following exceptions.

Inbound: For traffic received from the SFF and before removing the SFC encapsulation, the proxy Map-Registers the NSH header (Base, Service and Context) using the 5-tuple and JSON LCAFs defined in Section 3, the 5-tuple is applied to the original payload. After this the SFC Proxy acts as specified in [I-D.ietf-sfc-architecture].

Outbound: For returning traffic from the legacy SF, the SFC Proxy Map-Requests using a 5-tuple lookup LCAF and receives back the entire NSH header encoded using the JSON LCAF. The proxy applies the NSH encapsulation, decrements the Service Index and forwards the traffic as specified in [I-D.ietf-sfc-architecture].

In addition to this please note the following:

- In some scenarios the SFC Control Plane may have changed the {5-tuple->NSH} mapping to account for changes made by the legacy SF to the payload.

- The LISP Mapping System can identify the registering and requesting SFC Proxy using the RLOC of the Map-Register and Map-Request message respectively. This is useful when the inbound and

outbound SFC Proxies are different.

- This document assumes that the payload is IP (IPv4 or IPv6) and a transport header (TCP or UDP). Further revisions of this document will consider other payloads.

5. Security Considerations

The map-assisted SFC Proxy does not introduce additional security considerations beyond the ones described in [I-D.ietf-sfc-architecture] and [I-D.ietf-lisp-threats].

6. IANA Considerations

This memo includes no request to IANA.

7. References

7.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

7.2. Informative References

[BSON] MongoDB, , ""BSON - Binary JSON Specification "http://bsonspec.org/"", June 2015.

[I-D.ietf-lisp-lcaf] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-11 (work in progress), September 2015.

[I-D.ietf-lisp-threats] Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-13 (work in progress), August 2015.

[I-D.ietf-sfc-architecture] Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

[I-D.ietf-sfc-nsh] Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-01 (work in progress), July 2015.

[I-D.rodrigueznatal-lisp-multi-tuple-eid] Rodriguez-Natal, A., Cabellos-Aparicio, A., Barkai, S., Ermagan, V., Maino, F., Lewis, D., and D. Farinacci, ""LISP support for Multi-Tuple EIDs" draft-rodrigueznatal-lisp-multi-tuple-eids-00 (work in progress)", June 2015.

Authors' Addresses

Albert Cabellos
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia 08034
Spain

Email: acabello@ac.upc.edu

Sharon Barkai
Hewlett Packard Enterprise
3000 Hanover Street
Palo Alto, CA
USA

Email: sharon.barkai@hpe.com

Barak Perlman
Hewlett Packard Enterprise
3000 Hanover Street
Palo Alto, CA
USA

Email: barak.perlman@hpe.com

Vina Ermagan
Cisco Systems Inc
170 W Tasman Drive
San Jose, CA 95134
USA

Email: vermagan@cisco.com

Fabio Maino
Cisco Systems Inc
170 W Tasman Drive
San Jose, CA 95134
USA

Email: fmaino@cisco.com

Alberto Rodriguez-Natal
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia 08034
Spain

Email: arnatal@ac.upc.edu