Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2017

X. Xu
Huawei
H. Shah
Ciena
L. Contreras
Telefonica I+D
D. Bernier
Bell Canada
July 3, 2016

Service Function Chaining Using MPLS-SPRING
draft-xu-sfc-using-mpls-spring-06

Abstract

Source Packet Routing in Networking (SPRING) WG specifies a special source routing mechanism. Such source routing mechanism can be leveraged to realize the service path layer functionality of the service function chaining (i.e, steering traffic through a particular service function path) by encoding the service function path or the service function chain information as the explicit path information. This document describes how to leverage the MPLS-based source routing mechanism as developed by the SPRING WG to realize the service path layer functionality of the service function chaining.

Table of Contents

1. Introduction

When applying a particular Service Function Chain (SFC) [I-D.ietf-sfc-architecture] to the traffic selected by a service classifier, the traffic need to be steered through an ordered set of Service Functions (SF) in the network. This ordered set of SFs in the network indicates the Service Function Path (SFP) associated with the above SFC. To steer the selected traffic through an ordered list of SFs in the network, the traffic need to be attached by the service classifier with the information about the SFP (i.e., specifying exactly which Service Function Forwarders (SFFs) and which SFs are to be visited by traffic), the SFC, or the partially specified SFP which is in between the former two extremes. Source Packet Routing in Networking (SPRING) WG specifies a special source routing mechanism which can be used to steer traffic through an ordered set of routers (i.e., an explicit path). Such source routing mechanism can be leveraged to realize the service path layer functionality of the SFC (i.e., steering traffic through a particular SFP) by encoding the SFP information as the explicit path information contained in packets. The source routing mechanism specified by the SPRING WG can be applied to the MPLS data plane

[I-D.ietf-spring-segment-routing-mpls]. This document describes how to leverage the MPLS-based source routing mechanisms to realize the service path layer functionality of the service function chaining. Note that this approach is aligned with the Transport Derived SFF mode as described in Section 4.3.1 of [I-D.ietf-sfc-architecture].

1.1. 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].

2. Terminology

This memo makes use of the terms defined in [I-D.ietf-spring-segment-routing] and [I-D.ietf-sfc-architecture].

3. Solution Description

```
           +----------------------------------------------- ----+
           |                  SPRING Networks                   |
           |            +---------+       +---------+           |
           |            |   SF1   |       |   SF2   |           |
           |            +----+----+       +----+----+           |
           |                 |                 |                |
           |       (1)       |      (2)        |      (3)       |
      +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
      |Classifier+------+  SFF1   +-------+  SFF2   +-------+   D   |
      +----------+      +---------+       +---------+       +---+---+
           |                                                    |
           +----------------------------------------------------+
```

Figure 1: Service Function Chaining in SPRING Networks

As shown in Figure 1, assume SFF1 and SFF2 are two MPLS-SPRING-capable nodes. They are also Service Function Forwarders (SFF) to which two SFs (i.e., SF1 and SF2) are attached respectively. In addition, they have allocated and advertised Segment IDs (SID) for their locally attached SFs. In the MPLS-SPRING context, SIDs are intercepted as MPLS labels. For example, SFF1 allocates and advertises an SID (i.e., SID(SF1)) for SF1 while SFF2 allocates and advertises an SID (i.e., SID(SF2)) for SF2. These SIDs which are used to indicate SFs are referred to as SF SIDs. To encode the SFP information by an MPLS label stack, those SF SIDs as mentioned above would be interpreted as local MPLS labels. In addition, assume node SIDs for SFF1 and SFF2 are SID(SFF1) and SID(SFF2) respectively. Now assume a given traffic flow destined for destination D is selected by the service classifier to go through a particular SFC (i.e., SF1->SF2) before reaching its final destination D. Section 3.1 describes

how to leverage the MPLS- based source routing mechanisms to realize the service path functionality of the service function chaining (i.e., by encoding the SFP information within an MPLS label stack). Section 3.2 describes how to carry metadata over MPLS packets.

3.1. Encoding SFP Information by an MPLS Label Stack

Since the selected packet needs to travel through an SFC (i.e., SF1->SF2), the service classifier would attach a segment list of (i.e., SID(SFF1)->SID(SF1)->SID(SFF2)-> SID(SF2)) which indicates the corresponding SFP to the packet. This segment list is actually represented by a MPLS label stack. To some extent, the MPLS label stack here could be looked as a specific implementation of the SFC encapsulation used for containing the SFP information [I-D.ietf-sfc-architecture]. When the encapsulated packet arrives at SFF1, SFF1 would know which SF should be performed according to the current top label (i.e., SID (SF1)) of the received MPLS packet. If SF1 is an SFC encapsulation-aware SF, the MPLS packet would be sent to SF1 after the top label is poped. After receiving the MPLS packet returned from SF1, SFF1 would send it to SFF2 according to the current top label (i.e., SID (SFF2)). If SF1 is a legacy SF which could not process the MPLS label stack, the whole MPLS label stack (i.e., SID(SFF2)->SID(SF2)) MUST be stripped before sending the packet to SF1. After receiving the packet returned from SF1, SFF1 would re-impose the MPLS label stack which had been stripped before to the packet and then send it to SFF2 according to the current top label (i.e., SID (SFF2)). When the encapsulated packet arrives at SFF2, SFF2 would perform the similar action as what has been done by SFF1.

If there is no MPLS LSP towards the next node segment (i.e., the next SFF identified by the current top label), the corresponding IP-based tunnel (e.g., MPLS-in-IP/GRE tunnel [RFC4023], MPLS-in-UDP tunnel [RFC7510] or MPLS-in-L2TPv3 tunnel [RFC4817]) could be used instead (For more details about this special usage, please refer to [I-D.xu-spring-islands-connection-over-ip]). Since the transport (i.e., the underlay) could be IPv4, IPv6 or even MPLS networks, the above approach of encoding the SFP information by an MPLS label stack is fully transport-independent which is one of the major requirements for the SFC encapsulation [I-D.ietf-sfc-architecture].

In addition, the service classifier could further impose metadata on the MPLS packet through the Network Service Header (NSH) [I-D.ietf-sfc-nsh] (As for how to contain the NSH within a MPLS packet, please see Section 3.3). Here the Service Path field wihin the NSH would not be used for the path selection purpose anymore and therefore it MUST be set to a particular value to indicate such particular usage. In addition, the service index value within the

NSH is set to a value indicating the total number of SFs within the service function path. The service index SHOULD be decreased by one on each SF or SFC-proxy on behalf of the corresponding legacy SF. When the service index become zero, the NSH MUST be removed from the packet by the SF or SFC-proxy on behalf of the corresponding legacy SF.

3.2. How to Contain Metadata within an MPLS Packet

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata (i.e., the NSH which is only used as a metadata containner) in MPLS packets is a potential issue. There is a possible way to address the above issue: SFFs allocate two different labels for a given SF, one indicates the presence of NSH while the other indicates the absence of NSH. This approach has no change to the current MPLS architecture but it would require more than one label binding for a given SF. More details about how to contain metadata within an MPLS packet would be considered in the future version of this draft.

4. Acknowledgements

The authors would like to thank Loa Andersson and Andrew G. Malis for their valuable comments and suggestions on the draft. The authors would like to thank Adrian Farrel, Stewart Bryant, Alexander Vainshtein, Joel M. Halpern for their comments on how to indicate the presence of metadata within an MPLS packet.

5. Contributors

Zhenbin Li (Huawei Technologies)

6. IANA Considerations

TBD.

7. Security Considerations

TBD

8. References

8.1. Normative References

[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-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-04 (work in progress), March 2016.

[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>.

8.2. Informative References

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-05 (work in progress), May 2016.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-08 (work in progress), May 2016.

[I-D.xu-spring-islands-connection-over-ip]
Xu, X., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Connecting MPLS-SPRING Islands over IP Networks", draft-xu-spring-islands-connection-over-ip-05 (work in progress), March 2016.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <http://www.rfc-editor.org/info/rfc4023>.

[RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <http://www.rfc-editor.org/info/rfc4817>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid, 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://people.tid.es/LuisM.Contreras/

Daniel Bernier
Bell Canada

Email: daniel.bernier@bell.ca