MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2017

X. Xu
S. Bryant
Huawei
H. Assarpour
Broadcom
H. Shah
Ciena
L. Contreras
Telefonica I+D
D. Bernier
Bell Canada
J. Tantsura
Individual
S. Ma
Juniper
M. Vigoureux
Nokia
June 29, 2017

Service Chaining using Unified Source Routing Instructions
draft-xu-mpls-service-chaining-03

Abstract

Source Packet Routing in Networking (SPRING) WG is developing an MPLS source routing mechanism. The MPLS source routing mechanism can be leveraged to realize a unified source routing instruction which works across both IPv4 and IPv6 underlays in addition to the MPLS underlay. This document describes how to leverage the unified source routing instruction to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack.

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].

Table of Contents

1. Introduction

When applying a particular Service Function Chain (SFC) [RFC7665] 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. In order

to steer the selected traffic through the required ordered list of SFs, the service classifier needs to attach information to the packet 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.

The Source Packet Routing in Networking (SPRING) WG is developing an MPLS source routing mechanism which can be used to steer traffic through an ordered set of routers (i.e., an explicit path) and instruct nodes on that path to execute specific operations on the packet. By leveraging the MPLS source routing mechanism, [I-D.xu-mpls-unified-source-routing-instruction] describes a unified source routing instruction which works across both IPv4 and IPv6 underlays in addition to the MPLS underlay. This document describes how to leverage the unified source routing instruction to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack.

2. Terminology

This memo makes use of the terms defined in [I-D.ietf-spring-segment-routing-mpls], [I-D.xu-mpls-unified-source-routing-instruction] and [RFC7665].

3. Solution Description

```
           +----------------------------------------------- ----+
           |               MPLS SPRING Networks                 |
           |           +---------+       +---------+            |
           |           |   SF1   |       |   SF2   |            |
           |           +----+----+       +----+----+            |
           |               ^ | |(3)          ^ | |(6)           |
           |       (1)  (2)| | V     (4)  (5)| | V     (7)      |
      +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
      |Classifier+-------+ SFF1   +--------+ SFF2   +--------+  D   |
      +----------+      +---------+       +---------+       +---+---+
           |                                                    |
           +----------------------------------------------------+
        Figure 1: Service Function Chaining in MPLS-SPRING Networks
```

As shown in Figure 1, SFF1 and SFF2 are two MPLS-SPRING-capable nodes. They are also SFFs, each with one SF attached. In addition, they have allocated and advertised MPLS labels for their locally attached SFs. For example, SFF1 allocates and advertises a label (i.e., L(SF1)) for SF1 while SFF2 allocates and advertises a label (i.e., L(SF2)) for SF2. These labels, which are used to indicate SFs are referred to as SF labels. To encode the SFP information as an

MPLS label stack, local MPLS labels are allocated from SFFs' (e.g., SFF1 in Figure 1) label spaces to identify their locally attached SFs (e.g., SF1 in Figure 1), whilst the SFFs are identified by either nodal SIDs or adjacency SIDs depending on how strictly the network path needs to be specified. In addition, assume node SIDs for SFF1 and SFF2 are L(SFF1) and L(SFF2) respectively. In contrast, to encode the SFC information by an MPLS label stack, those SF labels MUST be domain-wide unique MPLS labels.

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 and 3.2 describe approaches of leveraging the MPLS- based source routing mechanisms to realize the service function chaining by encoding the SFP information within an MPLS label stack and by encoding the SFC information within an MPLS label stack respectively. Since the encoding of the partially specified SFP is just a simple combination of the encoding of the SFP and the encoding of the SFC, this document would not describe how to encode the partially specified SFP anymore.

3.1. Encoding SFP Information by an MPLS Label Stack

```
           +----------------------------------------------- ----+
           |               MPLS SPRING Networks                 |
           |            +---------+       +---------+           |
           |            |   SF1   |       |   SF2   |           |
           |            +----+----+       +----+----+           |
           |  +---------+    |                 |    +---------+ |
           |  | L(SFF2) |    |                 |    |Pkt to D | |
           |  +---------+    |                 |    +---------+ |
           |  | L(SF2)  |    |                 |                |
           |  +---------+    |               ^ | |              |
           |  |Pkt to D | ^  | |             | | |              |
           |  +---------+ |  | |          (5)| | |(6)           |
           |           (2)|  | |(3)          | | V              |
           |       (1)    |  | V     (4)       |      (7)       |
      +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
      |Classifier+------+  SFF1   +-------+  SFF2   +-------+   D   |
      +----------+      +---------+       +---------+       +---+---+
           |    +---------+      +---------+        +---------+ |
           |    | L(SFF1) |      | L(SFF2) |        |Pkt to D | |
           |    +---------+      +---------+        +---------+ |
           |    | L(SF1)  |      | L(SF2)  |                    |
           |    +---------+      +---------+                    |
           |    | L(SFF2) |      |Pkt to D |                    |
           |    +---------+      +---------+                    |
           |    | L(SF2)  |                                     |
           |    +---------+                                     |
           |    |Pkt to D |                                     |
           |    +---------+                                     |
           +----------------------------------------------------+
```

Figure 2: Packet Walk in MPLS underlay

As shown in Figure 2, 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 represented by an 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 [RFC7665]. When the encapsulated packet arrives at SFF1, SFF1 would know which SF should be performed according to the top label (i.e., SID (SF1)) of the received MPLS packet. We first consider the case where SF1 is an encapsulation aware SF, i.e., it understands how to process a packet with a pre-pended MPLS label stack. In this case the packet would be sent to SF1 by SFF1 with the label stack SID(SFF2)-> SID(SF2). SF1 would perform the required service function on the received MPLS packet where the payload is constrained to be an IP packet, and the SF needs to process both IPv4 and IPv6 packets (note that the SF would use the first nibble of the MPLS payload to

identify the payload type). After the MPLS packet is returned from SF1, SFF1 would send it to SFF2 according to the top label (i.e., SID (SFF2)).

If SF1 is a legacy SF, i.e. one that is unable to process the MPLS label stack, the remaining MPLS label stack (i.e., SID(SFF2)->SID(SF2)) MUST be saved and stripped from the packet before sending the packet to SF1. When the packet is returned from SF1, SFF1 would re-impose the MPLS label stack which had been previously stripped and then send the packet to SFF2 according to the current top label (i.e., SID (SFF2)). As for how to associate the corresponding MPLS label stack with the packets returned from legacy SFs, those mechanisms as described in [I-D.song-sfc-legacy-sf-mapping] could be considered.

When the encapsulated packet arrives at SFF2, SFF2 would perform the similar action to that described above.

As shown in Figure 3, 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 for MPLS (e.g., MPLS-in-IP/GRE tunnel [RFC4023], MPLS-in-UDP tunnel [RFC7510] or MPLS-in-L2TPv3 tunnel [RFC4817]) would be used instead, according to the unified source routing instruction as described in [I-D.xu-mpls-unified-source-routing-instruction].

```
           +----------------------------------------------- ----+
           |                      IP Networks                   |
           |            +---------+       +---------+           |
           |            |   SF1   |       |   SF2   |           |
           |            +----+----+       +----+----+           |
           |  +---------+    |                 |    +---------+ |
           |  | L(SFF2) |    |                 |    |Pkt to D | |
           |  +---------+    |                 |    +---------+ |
           |  | L(SF2)  |    |                 |                |
           |  +---------+    |               ^ | |              |
           |  |Pkt to D | ^  | |             | | |              |
           |  +---------+ |  | |          (5)| | |(6)           |
           |           (2)|  | |(3)          | | V              |
           |       (1)    |  | V     (4)       |      (7)       |
      +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
      |Classifier+------+  SFF1   +-------+  SFF2   +-------+   D   |
      +----------+      +---------+       +---------+       +---+---+
           |    +---------+      +---------+                    |
           |    |IP Tunnel|      |IP Tunnel|        +---------+ |
           |    |to SFF1  |      | to SFF2 |        |Pkt to D | |
           |    +---------+      +---------+        +---------+ |
           |    | L(SF1)  |      | L(SF2)  |                    |
           |    +---------+      +---------+                    |
           |    | L(SFF2) |      |Pkt to D |                    |
           |    +---------+      +---------+                    |
           |    | L(SF2)  |                                     |
           |    +---------+                                     |
           |    |Pkt to D |                                     |
           |    +---------+                                     |
           +----------------------------------------------------+
               Figure 3: Packet Walk in IP underlay
```

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 [RFC7665].

3.2. Encoding SFC 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 an MPLS label stack (i.e., L(SF1)->L(SF2)) which indicates that SFC to the packet. Since it's known to the service classifier that SFF1 is attached with an instance of SF1, the service classifier would therefore send the MPLS encapsulated packet through either an MPLS LSP tunnel or an IP-based tunnel towards SFF1 (as shown in Figure 4 and 5 respectively). When the MPLS encapsulated packet arrives at SFF1, SFF1 would know which SF should be performed according to the current top label (i.e.,

L(SF1)). Similarly, SFF1 would send the packet returned from SF1 to SFF2 through either an MPLS LSP tunnel or an IP-based tunnel towards SFF2 since it's known to SFF1 that SFF2 is attached with an instance of SF2. When the encapsulated packet arrives at SFF2, SFF2 would do the similar action as what has been done by SFF1. Since the transport (i.e., the underlay) could be IPv4, IPv6 or even MPLS networks, the above approach of encoding the SFC information by an MPLS label stack is fully transport-independent which is one of the major requirements for the SFC encapsulation [RFC7665].

```
           +----------------------------------------------- ----+
           |                    MPLS Networks                   |
           |            +---------+       +---------+           |
           |            |   SF1   |       |   SF2   |           |
           |            +----+----+       +----+----+           |
           |                 |                 |    +---------+ |
           |                 |                 |    |Pkt to D | |
           |  +---------+    |                 |    +---------+ |
           |  | L(SF2)  |    |                 |                |
           |  +---------+    |               ^ | |              |
           |  |Pkt to D | ^  | |             | | |              |
           |  +---------+ |  | |          (5)| | |(6)           |
           |           (2)|  | |(3)          | | V              |
           |       (1)    |  | V     (4)       |      (7)       |
      +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
      |Classifier+------+  SFF1   +-------+  SFF2   +-------+   D   |
      +----------+      +---------+       +---------+       +---+---+
           |    +---------+      +---------+        +---------+ |
           |    | L(SFF1) |      | L(SFF2) |        |Pkt to D | |
           |    +---------+      +---------+        +---------+ |
           |    | L(SF1)  |      | L(SF2)  |                    |
           |    +---------+      +---------+                    |
           |    | L(SF2)  |      |Pkt to D |                    |
           |    +---------+      +---------+                    |
           |    |Pkt to D |                                     |
           |    +---------+                                     |
           +----------------------------------------------------+
```

Figure 4: Packet Walk in MPLS underlay

```
        +----------------------------------------------- ----+
        |                    IP Networks                   |
        |            +---------+       +---------+           |
        |            |   SF1   |       |   SF2   |           |
        |            +----+----+       +----+----+           |
        |                 |                 |    +---------+ |
        |                 |                 |    |Pkt to D | |
        |  +---------+    |                 |    +---------+ |
        |  | L(SF2)  |    |                 |                |
        |  +---------+    |               ^ | |              |
        |  |Pkt to D | ^  | |             | | |              |
        |  +---------+ |  | |          (5)| | |(6)           |
        |           (2)|  | |(3)          | | V              |
        |       (1)    |  | V     (4)       |      (7)       |
   +----+-----+ ---> +----+----+ ----> +----+----+ --->  +---+---+
   |Classifier+------+  SFF1   +-------+  SFF2   +-------+   D   |
   +----------+      +---------+       +---------+       +---+---+
        |    +---------+      +---------+                    |
        |    |IP Tunnel|      |IP Tunnel|        +---------+ |
        |    |to SFF1  |      | to SFF2 |        |Pkt to D | |
        |    +---------+      +---------+        +---------+ |
        |    | L(SF1)  |      | L(SF2)  |                    |
        |    +---------+      +---------+                    |
        |    | L(SF2)  |      |Pkt to D |                    |
        |    +---------+      +---------+                    |
        |    |Pkt to D |                                     |
        |    +---------+                                     |
        +----------------------------------------------------+
```

Figure 5: Packet Walk in IP underlay

3.3. 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 an MPLS packet is a potential issue to be addressed. One possible way to address the above issue is: 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. Another possible way is to introduce a protocol identifier field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

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, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein and Joel M. Halpern for their valuable comments and suggestions on the document.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

It is fundamental to the SFC design that the classifier is a trusted resource which determines the processing that the packet will be subject to, including for example the firewall. It is also fundamental to the SPRING design that packets are routed through the network using the path specified by the node imposing the SIDs. Where an SF is not encapsulation aware the packet may exist as an IP packet, however this is an intrinsic part of the SFC design which needs to define how a packet is protected in that environment. Where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how it is secured. Thus the secutity vulnerabilities are addressed in the underlying technologies used by this design, which itself does not introduce any new security vulnerabilities.

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>.

7.2. Informative References

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-12 (work in progress), February 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[I-D.song-sfc-legacy-sf-mapping]
Song, H., You, J., Yong, L., Jiang, Y., Dunbar, L., Bouthors, N., and D. Dolson, "SFC Header Mapping for Legacy SF", draft-song-sfc-legacy-sf-mapping-08 (work in progress), September 2016.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X., "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-02 (work in progress), December 2016.

[I-D.xu-mpls-unified-source-routing-instruction]
Xu, X., Bryant, S., Raszuk, R., Chunduri, U., Contreras, L., Jalil, L., Assarpour, H., Velde, G., Tantsura, J., and S. Ma, "Unified Source Routing Instruction using MPLS Label Stack", draft-xu-mpls-unified-source-routing-instruction-02 (work in progress), June 2017.

[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>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Hamid Assarpour
Broadcom

Email: hamid.assarpour@broadcom.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

Jeff Tantsura
Individual

Email: jefftant@gmail.com

Shaowen Ma
Juniper

Email: mashaowen@gmail.com

Martin Vigoureux
Nokia

Email: martin.vigoureux@nokia.com