SFC working group
Internet-Draft
Intended status: Standards Track
Expires: October 12, 2014

H. Song
L. Yong
Y. Jiang
Huawei

April 10, 2014

SFC Header Mapping for Legacy SF
draft-song-sfc-legacy-sf-mapping-00

Abstract

SFC (Service Function Chaining) is used to manipulate service functions with easy creation, updating and deletion. A service function chain goes through a list of ordered service function instances. One assumption of this document is that legacy service function instances can participate in the service chain. They are not aware of the SFC header, nor interpret it. This document provides a mechanism between a Service Forwarding Entity (SFE) and a Service Function Instance (SFI), to identify the SFC header associated with a packet that is returned from an SFI, without SFC header being explicitly carried in the wired protocol between SFE and SFI.

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 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

SFC is used to manipulate service functions with easy creation, updating and deletion. A service function chain goes through a list of ordered service functions. One assumption of this document is that certain service functions can be kept as legacy. They do not have to be aware of the SFC header, nor interprets it. This document provides a mechanism between a Service Forwarding Entity and a Service Function Instance, to identify the SFC header associated with a packet that is returned from an SFI, without anything in the SFC header being explicitly carried in the wired protocol between a SFE and SFI.

```
            +----------------+
            |Service Function|
            |Instance        |
            +----+----+------+
                 ^    |
                 |    |
                 |    |
              (2)|    |(3)
                 |    |
                 |    |
            +----+----V--------+
      (1)   |Service Forwarding| (4)
   -------->|Entity            +------->
            +------------------+
```

Figure 1: Procedure of a packet processed by a legacy service function

The legacy service function (i.e. SFI in the Figure 1) only handles packets without SFC header, because it does not understand the SFC header. One advantage is that the existing service functions don't need to be upgraded to support SFC. Otherwise it may be a hindrance for the widely adoption of SFC.

We assume that for most SFIs, the packet header is transparent to a legacy SFI. SFI will not modify the layer 2 or layer 3 packet headers. If the payload in the SFC encapsulation is layer 3 traffic, it will be kept as it is, and a new layer 2 header will be added before sending to the SFI. However if the payload in the SFC encapsulation is layer 2 traffic, we may modify the original source MAC address and use it for mapping to the stored SFC header, but it will not impact the SFI processing. The SFI will send the traffic back after processing. For the current stage, we leave the legacy SFIs which modify the original packet headers as an open issue for further study.

As shown in Figure 1, there are four steps. The SFE receives a packet, and removes its SFC header, which may optionally contain metadata, and stores the SFC header locally, and then sends the original packet to the SFI. After SFI processing the packet, the traffic will be sent back to the SFE. The SFE retrieves the pre-stored SFC header accordingly, and encapsulates the packet with the SFC header, and then sends the packet to next-hop service function. The key problem here is how to map the packet to its original SFC header.

If the SFC header is not changed per flow at a certain point, e.g., a specific SFE, (i.e. each flow has a specific SFC header in a SFE, but in another SFE, the SFC header is different), then the SFE needs to find the original SFC header per flow. If the SFC header is changed

per packet for a specific flow at a certain point, then the SFE needs to find the original SFC header per packet. The second case may be happened if different packets in a flow carry different metadata (e.g. the metadata can be injected to the packet by a DPI appliance). It's also the reason why we cannot use five-tuple for the mapping to retrieve the original SFC header.

We use an expiration time for each mapping entry in the SFE. If the SFC header in that entry has not been retrieved after the expiration time, the entry will be deleted from the entry table.

2. Terminology

The terminology used in this document is defined below:

Legacy SF: A conventional service function that does not support SFC header.

Transparent SF: A service function that does not change any bit of the original service packet header (Layer 2, layer 3, and layer 4) sent to it.

Non-transparent SF: A service function that changes some part of the original service packet header sent to it.

Original Service Packet: The payload in a SFC encapsulation packet or a packet constructed based on the original payload.

3. Mechanisms

The mechanisms used in this document require that each forwarding entity and its connected service functions in a same layer 2 network. The following are considerations mainly for transparent SFIs. If the original payload packet is a layer 2 packet, and the mapping method used is layer 2 MAC address, then the assumption is that the SFI does not need to look into the layer 2 header. If it does, other mechanisms should be used.

3.1. For Transparent Service Functions

If the service function is transparent to packet headers, we can use the following methods for SFC header mapping.

3.1.1. Layer 2 MAC Address

We can use layer 2 MAC address to associate a SFC header between SFE and SFI, i.e. each SFC header will be assigned a source MAC address on the SFE. If SFC header can be changed per packet, then SFE

assigns a new source MAC address for each packet it received, otherwise, it assigns a new MAC address for each flow it received.

When SFE received the returned packet from the SFI, it retrieves the packet's original SFC header by using the MAC address as a key. And then it encapsulates the packet with that SFC header and sends to the next hop.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

 Outer Ethernet Header:

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             SFI Destination MAC Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |SFI Destination MAC Address    | SFE Source MAC Address        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             SFE Source MAC Address                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Ethertype = 0x0800         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 Original IP Payload:

    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Original Payload                                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.2. VLAN

If the network between the SFE and SFI is a layer 2 network, and in the case that a SFI need to look into the MAC address of the packet, then VLAN can be used for the mapping between them. The SFE removes the SFC header and sends the packet to the SFI, with encapsulating a certain VLAN ID. It locally maintains the mapping between VLAN ID and the SFC header. When it gets the returned packet from the SFI, it removes the VLAN part from the packet and retrieves the corresponding SFC header according to the VLAN ID, and then encapsulates SFC header into that packet before sending to the next service function.

The VLAN ID can be used for mapping per flow, i.e. each flow will be assigned a new VLAN ID. If SFC header could be changed per packet, the length of VLAN ID is not enough for mapping.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

    Outer Ethernet Header:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             SFI Destination MAC Address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |SFI Destination MAC Address    | SFE Source MAC Address        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             SFE Source MAC Address                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |OptnlEthtype = C-Tag 802.1Q    |Outer.VLAN Tag Information     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |    Ethertype = 0x0800         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Original IP Payload:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           Original Payload                                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.3. QinQ

If the network between the SFE and SFI is already a VLAN network, and the SFI needs to look into the MAC address, then QinQ is used for the communication between SFE and SFI. The SFE remove the SFC header and send the original traffic to SFI with a certain outer VLAN ID. It locally maintains the mapping between outer VLAN ID and the SFC header.

If the network between SFE and SFI is not a VLAN network, then QinQ can be used for either per flow mapping or per packet mapping, using two layer VLAN fields. If the network between SFE and SFI is a VLAN network, then QinQ can only be used for per flow mapping, using one VLAN field.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

    Outer Ethernet Header:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             SFI Destination MAC Address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |SFI Destination MAC Address    | SFE Source MAC Address        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |             SFE Source MAC Address                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |OptnlEthtype = S-Tag 802.1Q    |Outer.VLAN Tag Information     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Ethertype = C-Tag 802.1Q       |Inner.VLAN Tag Information     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |    Ethertype = 0x0800         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Original IP Payload:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           Original Payload                                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.1.4. VXLAN

If the SFE and SFI are already deployed in a QinQ network, then we can use VXLAN [I-D.mahalingam-dutt-dcops-vxlan] for the mapping. This tunneling technology is only used when the original packet type is at layer 2 and the SFI has to look into the layer 2 MAC header.

The drawback of this mechanism is that it requires both SFE and SFI to support VXLAN.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

   Outer Ethernet Header:

      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             SFI Destination MAC Address                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |SFI Destination MAC Address    | SFE Source MAC Address        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             SFE Source MAC Address                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |OptnlEthtype = C-Tag 802.1Q    |Outer.VLAN Tag Information     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Ethertype = 0x0800         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Outer IP Header:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Version|  IHL  |Type of Service|          Total Length         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Identification        |Flags|      Fragment Offset    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Time to Live |Protocol=17(UDP) |   Header Checksum             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |               Outer Source IPv4 Address                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |               Outer Destination IPv4 Address                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Outer UDP Header:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |       Source Port = xxxx      |       Dest Port = VXLAN Port  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    VXLAN Header:

       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |R|R|R|R|I|R|R|R|            Reserved                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |              VXLAN Network Identifier (VNI) |   Reserved      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.2. For Non-transparent Service Functions

Non transparent service functions including NAT (Network Address Translation), WOC (WAN Optimization Controller) and etc, are more complicated, as they may change any part of the original packet sent to them. It is better to analyze case by case, to utilize a specific filed that the SFI does not change for the mapping and retrieving the SFC header. We would like to leave it for open discussion.

The use case below is just one example that SFE can learn the behavior of the SFI changing the packet. In this example, we can use the following method for SFC header mapping. The SFI needs to report its mapping rules (e.g. 5-tuple mapping rules) to the control plane (step 1), and then the control plane can notify the SFE the mapping information (step 2). According to the mapping information, the SFE can establish a mapping table for the SFC header, the original header, and the processed header of the packet. After receiving the packet from the SFI (step 5), the SFE retrieves the SFC header from the mapping table by using the processed header as a key.

```
        +-------------+
        |Control Plane|
        +--+-----+----+
           ^     ^
           |     |
           |     |(1)    +----------------+
           |     +------->Service Function|
        (2)|             |Instance        |
           |             +-----+---+------+
           |                (4)^   |(5)
           +---------------+   |   |
                           |   |   |
                        +--V---+---V-------+
                 (3)    |Service Forwarding| (6)
              --------->+Entity            +------->
                        +------------------+
```

4. Security considerations

When we modify the layer 2 header of the original packet and send it to SFI, if the SFI needs to look into the layer 2 header, it may cause security threats.). It also provides diagrams of the main entities that the information model is comprised of.

5. Acknowledgement

6. Informative References

[I-D.jiang-sfc-arch]
Jiang, Y. and L. Hongyu, "An Architecture of Service Function Chaining", draft-jiang-sfc-arch-01 (work in progress), February 2014.

[I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-08 (work in progress), February 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Haibin Song
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: haibin.song@huawei.com

Lucy Yong
Huawei
5340 Legacy Drive
Plano, TX 75025
U.S.A.

Email: lucy.yong@huawei.com

Yuanlong Jiang
Huawei
Bantian, Longgang district
Shenzhen 518129
China

Email: jiangyuanlong@huawei.com