

Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: January 2017

P. Bottorff  
D. Fedyk  
HP Enterprise  
H. Assarpour  
Broadcom  
July 21, 2016

**Ethernet MAC Chaining**  
**draft-fedyk-sfc-mac-chain-02.txt**

**Abstract**

This document introduces and describes a simple and highly scalable service function chaining mechanism called MAC chaining which is built largely on existing Ethernet frame and forwarding capabilities. MAC chaining uses IEEE 802 Media Access Control (MAC) addresses to provide flexible and complete service function chains. It is largely transparent to layers above Ethernet and designed to augment and coexist with existing virtual and physical network forwarding. MAC chaining is achievable in some devices and virtual switches today using existing protocols.

**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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 2017.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Conventions used in this document.....</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Terminology.....</a>	<a href="#">5</a>
<a href="#">4.</a>	<a href="#">MAC Chaining.....</a>	<a href="#">6</a>
<a href="#">4.1.</a>	<a href="#">MAC Chaining Packet and Address Formats.....</a>	<a href="#">7</a>
<a href="#">4.2.</a>	<a href="#">Meta-Data Encoding Consideration for NSH.....</a>	<a href="#">10</a>
<a href="#">4.3.</a>	<a href="#">Forwarding.....</a>	<a href="#">11</a>
<a href="#">4.3.1.</a>	<a href="#">Destination Address MAC Chaining Operation.....</a>	<a href="#">13</a>
<a href="#">4.3.2.</a>	<a href="#">Destination and Source Address MAC Chaining.....</a>	<a href="#">14</a>
<a href="#">4.3.3.</a>	<a href="#">Forwarding by Service Functions.....</a>	<a href="#">14</a>
<a href="#">4.3.4.</a>	<a href="#">Reverse Path Forwarding by Service Functions.....</a>	<a href="#">15</a>
<a href="#">4.3.5.</a>	<a href="#">Proxy Forwarders.....</a>	<a href="#">16</a>
<a href="#">4.3.6.</a>	<a href="#">Example MAC Chaining Walk Through using DA/SA Chaining .....</a>	<a href="#">17</a>
<a href="#">4.3.7.</a>	<a href="#">Forwarding by Chain Termination Functions.....</a>	<a href="#">19</a>
<a href="#">5.</a>	<a href="#">Programming a Service Chain.....</a>	<a href="#">19</a>
<a href="#">6.</a>	<a href="#">Considerations for Operation over NV03 Tunnel Transports.....</a>	<a href="#">19</a>
<a href="#">7.</a>	<a href="#">Domain of operation.....</a>	<a href="#">20</a>
<a href="#">8.</a>	<a href="#">Security Considerations.....</a>	<a href="#">20</a>
<a href="#">9.</a>	<a href="#">IANA Considerations.....</a>	<a href="#">20</a>
<a href="#">10.</a>	<a href="#">References.....</a>	<a href="#">21</a>
<a href="#">10.1.</a>	<a href="#">Normative References.....</a>	<a href="#">21</a>
<a href="#">10.2.</a>	<a href="#">Informative References.....</a>	<a href="#">21</a>
<a href="#">11.</a>	<a href="#">Acknowledgments.....</a>	<a href="#">21</a>

## 1. Introduction

Service Function Chaining (SFC) enables the creation of composite (network) services that consist of a directed graph of Service Functions (SF) which must be applied to packets selected as a result of classification. SFC is described in detail in the SFC architecture document [RFC7665], and is not repeated here.

This document describes a new highly scalable, low resource, service function chain (SFC) mechanism called MAC chaining that is based on the current IEEE 802 [802-2001] Ethernet header for physical and virtualized environments. Service function chaining is an active area in the standards and various proposals for how to do SFCs are being put forward. The basic mechanism used for MAC chaining is the use of MAC addresses in the Ethernet header as a mechanism both for identifying chains and for forwarding packets along a MAC chain. The forwarding mechanism used in MAC chaining is independent from virtual or overlay networks used to form subnets. MAC chaining addresses are terminated at each Service Function Forwarder (SFF) and replaced by a new set of MAC chaining addresses used to forward through the next Service Function in the chain.

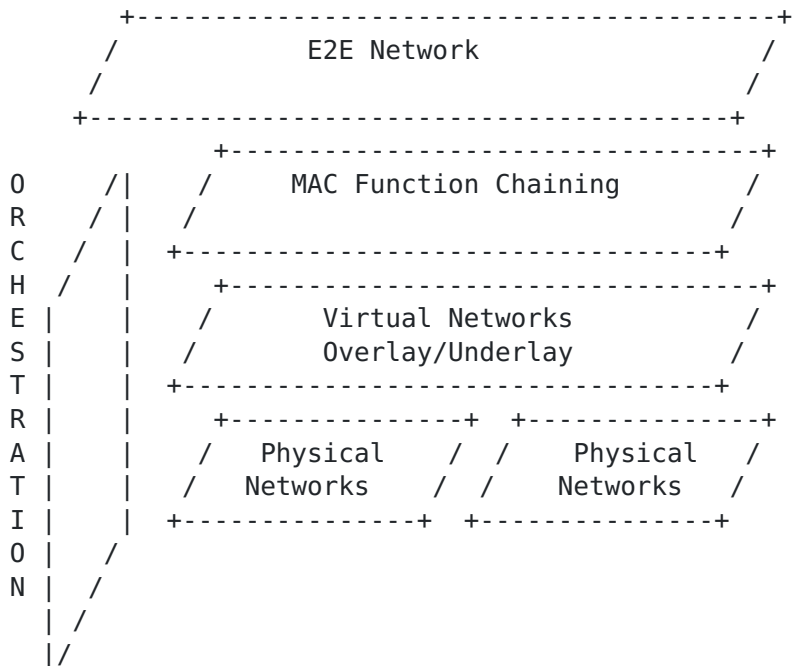


Figure 1 Service Forwarding Plane

MAC chaining can be viewed as a network service plane as shown in Figure 1. The SFC architecture document [[RFC7665](#)] describes chain forwarding in terms of 3 main architecture components which are the Service Classification Function (SCF), Service Function Forwarder (SFF) and the Service Function (SF). When managed with MAC chaining, Service Functions (SF) are simple links in the service chain and require little context of the overall chain. MAC chaining Service Function Forwarders (SFF) enable the chain and control the path to and from the SFs. Logically the SFF forms a switching layer above the existing virtual networking layers. In MAC chaining, a Chain Termination Function (CTF) is added to the architecture to separate the operation of de-encapsulating the packet and sending it toward the final destination from the operations of service function classification and service function forwarding described in the I-I.ietf-sfc-architecture.

MAC Chain forwarding is performed by a MAC Chaining Service Function Forwarder (SFF) using DA and SA address swapping. The operation of a MAC Chaining SFF has characteristics of a router in that it uses information in the packet to determine a new link destination, however unlike a router the new link decision is based on the previous MAC address rather than the IP address. This arrangement has the advantage that the IP addresses retains the end-to-end address eliminating the need for NAT addresses on entry and exit of the chain. A MAC Chain Service Function Forwarder also has characteristics of a Bridge in that it uses a promiscuous receiver with exact matching of every frame presented on its links to a MAC DA and VLAN entry in the filtering database. This matching prevents forwarding frames which don't contain allocated Chain MAC addresses. The exact matching performed by a MAC Chaining SFF provides orders of magnitude higher table scaling than longest match forwarding characteristic of routers.

MAC chaining can interwork with other chaining mechanisms when used in hybrid chains. In Hybrid chains the SFFs or proxies are responsible for converting the packet formats for the appropriate elements.

## **2. Conventions used in this document**

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](#)].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

### 3. Terminology

Chain Termination Function (CTF): See [[RFC7665](#)]. The chain termination function terminates a Service Function Path performing any de-encapsulation and operations required to continue forwarding to the final destination. The CTF may also be the final destination of the chain.

CS-MAC: A MAC address which identifies a MAC Chain Segment

CS-MAC Authority: CS-MAC Authority refers to the purely administrative mechanism to ensure CS-MACs are unique but allows the optional reuse of MACs in different VNs. Each VN port has a single CS-MAC Authority. Multiple ports may share the same Authority. A MAC Chain may be under a single CS-MAC Authority or it may be split among multiple CS-MAC Authorities.

DA: MAC destination address

I/G The Individual/Group (I/G) address bit (LSB of octet 0).

MAC Address: IEEE 802 Media Access Control Address a 48 bit address.

MAC Chain Segment (CS): A hop between either Service Forwarding Functions, a Service Classification Function and a Service Forwarding Function, or a Service Forwarding Function and a Chain Termination Function

MTU: Maximum Transmission Unit. Layer 2 has a maximum frame size and Layer 3 has a Maximum Packet Unit. This documents uses the term Layer 2 MTU to identify that MAC chaining does not affect L3 or IP MTU.

VN Port: In this document a port is the logical interface context for a MAC address in a virtual network (VN). A VN port may be implemented on any type of physical port or logical supporting Ethernet.

SA: MAC source address

Service Function (SF): See [[RFC7665](#)].

Service Classification Function (SCF):  
See [[RFC7665](#)].

Service Function Chain (SFC): See [[RFC7665](#)].

Service Function Forwarder (SFF): See [[RFC7665](#)].

Service Function Path (SFP): See [[RFC7665](#)].

U/L: The Universally or Locally administered (U/L) address bit is the bit of octet 0 adjacent to the I/G bit.

Virtual Network (VN): A Virtual network is used to identify a network segment controlled by a CS-MAC Authority.

#### 4. MAC Chaining

MAC chaining uses controlled assignment of Ethernet 48 bit MAC addresses to form the chain. Ethernet MAC addresses are selected to uniquely identify both the chain and the particular chain segment (or hop) within the identified chain. These assigned Ethernet addresses are called Chain Segment MAC (CS-MAC) addresses in this document. These CS-MACs allow MAC chaining to be implemented on existing Ethernet infrastructure making it broadly interoperable with the majority of installed base including existing Ethernet, Carrier Ethernet and IP equipment.

Each MAC chain is composed of a series of Chain Segments (CS) which are hops between Classifiers, Service Function Forwarders and Chain Terminating Functions (see figure 2). Some of the chain segments include Service Functions while others perform forwarding between the SCF, SFF and CTF. For each chain segment, a Destination MAC address (DA), and optionally a Source MAC address (SA) are selected, from a locally administered MAC address space, to uniquely identify the chain segment within the SFC domain. MAC chaining uses these CS-MACs, in the Ethernet header, as an identifier to enable forwarding packets in a MAC chain.

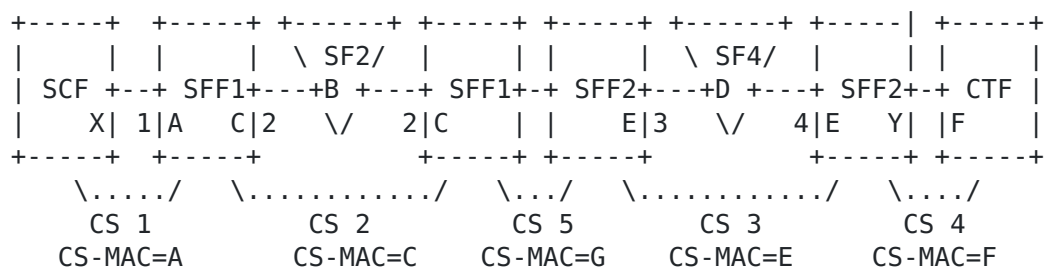


Figure 2 MAC Chain Segments Addressing

In Figure 2 five chain segments are illustrated. The first chain segment is between the classifier and service function forwarder identified as SFF1. This chain segment, designated CS1, has been assigned CS-MAC A. (For brevity the 48 bit MAC addresses are identified by letters). The next chain segment is from SFF1 VN port 2 through service function 2 and back to SFF1 VN port 2. This chain segment designated CS2 has been assigned CS-MAC C. SF2 on CS2 is a single armed SF with MAC address B attached to SFF VN port 2. (See [section 4.3.3](#) for a description of the types of armed SF). Chain segment CS5 is between SFF1 and SFF2. It has an assigned CS-MAC G. Chain segment CS3 from SFF2 VN port 3 to SFF2 VN port 4 is identified by CS-MAC E. SF4, lying on CS3, is a dual armed SF with MAC address D on the side connecting to SFF2 VN port 3. The final chain segment, CS4, of the path is between SFF2 and the CTF and is identified by CS-MAC F.

As described here, MAC chaining operates in the context of Virtual Networks (VN). To fully describe each MAC chaining address the tuple (CS-MAC Authority, CS-MAC) is used which uniquely identifies each chain segment as well as the entire chain. Each chain segment and each VN MUST belong to a single CS-MAC Authority which is a management construct that assigns unique CS-MACs for that segment and VN. If a chain segment crosses between two independent VNs, then both the VNs must have the same CS-MAC Authority.

#### **4.1. MAC Chaining Packet and Address Formats**

The IEEE 802.3 [[802-2001](#)] frame header consists of a Destination MAC Address (DA) (6 bytes) followed by a Source MAC Address (SA) (6 Bytes) followed by a number of possible fields which are identified by an Ethertype (2 Bytes) following the SA. A VLAN tag (4 bytes) is a common TAG that also carries Priority Code Points and Discard Eligible Information for traffic classification. For the purpose of this document the DA and SA are the primary fields used for MAC chaining however the frame may optionally have VLAN Tags. The MAC chaining frame can also be carried inside other encapsulations (i.e. within an overlay) like VxLAN, Geneve, GUE, L2VPN or Provider Backbone Bridging (PBB).

Figure 3 illustrates the formats MAC chaining uses to carry the original IPv4 or L2 packets when entering the classifier. Since MAC chaining encodes a SFC path solely in the MAC addresses of the Ethernet header the SPI and SI fields of the Network Service Header (NSH) [[I-D.ietf-quinn-sfc-nsh](#)] are not necessary and therefore NSH is an optional addition when using a MAC segment chain.

Format 1: Original IPv4, MAC Chaining without NSH:

```
+-----+
| Outer Ethernet, ET=0x0800          | original IP Packet |
+-----+
```

Format 2: Original IPv4, MAC Chaining with NSH

```
+-----+-----+-----+
| Outer Ethernet, ET=0x894F | NSH, NP = 0x1 | original IP Packet |
+-----+-----+-----+
```

Format 3: Original L2, MAC Chaining without NSH

```
+-----+-----+-----+
| Outer Ethernet, ET=0x****          | original L2 frame |
+-----+-----+-----+
```

Format 4: Original L2, MAC Chaining with NSH

```
+-----+-----+-----+
| Outer Ethernet, ET=0x894F | NSH, NP = 0x3 | original L2 frame |
+-----+-----+-----+
```

Figure 3 MAC Chaining Formats for IPv4 and L2 Service Packets

For original L3 packets MAC Chaining can forward a standard L3 frame without any further encapsulation. In addition if the SFs or Proxy functions are NSH aware, the format 2 for original L3 packets allows adding the NSH header to pass meta-data between SFs.

Figure 3 provides two alternate encapsulations for original L2 packets. The simple format 3 encoding without NSH uses an Ethertype to designate an L2 frame follows. This encoding provides an L2 encapsulated in an L2 frame. The format 4 encoding uses the NSH header with the Next Protocol set to 0x3 designating an L2 frame encapsulation. The NSH encapsulation provides all the functions of the simple L2 encapsulation and therefore can be used whenever the SFs or Proxy functions are NSH aware.

In addition to meta-data carried in the NSH it is also possible to encode a small amount of meta-data in the Chain Segment MAC addresses. The branch taken bit (figure 4) is a small piece of meta-data that can be used in the MAC chaining header. For more elaborate meta-data, the Network Service Header draft [\[I-D.ietf-quinn-sfc-nsh\]](#) header is compatible with MAC chaining. [Section 11.3](#) of [\[I-D.ietf-quinn-sfc-nsh\]](#) illustrates that the Ether type following a MAC chaining outer header has a registered type of 0x894F (TBC) with an NSH that subsequently defines the payload. SFs can use the NSH within the chain. Any SCF, SF or CTF can remove or modify the NSH as





BT        Branch Taken Chain indicator. Required. This bit may be set or reset in context of a chain.

CS-ID     Chain Segment ID 23 bits. (Example only).

CS-MAC addresses, which identify chain segments, SHOULD be allocated by the CS-MAC Authority from the local space using a Company ID assigned to the CS-MAC Authority. MAC addresses also have an Individual (unicast) or Group (Multicast) bit I/G. MAC chaining MAY use individual or group addresses for the CS-MACs though restriction on the use of group CS-MACs may apply depending on the type of forwarding performed by the SFF for the particular segment.

MAC chaining may also use global or local MAC addresses. The MAC address assigned to a Service Function MAY be Global or Local and can be assigned by any authority, not necessarily the CS-MAC Authority.

As with other types of Service chaining, a packet or a frame travels through a network until it encounters an initial classifier. Forwarding before the classifier is out of the scope of this specification. The native packet format (L2 or L3 or tunneled, etc.) arriving at the classifier does not matter but the classifier (or set of classifiers) need to inspect the packet and determine that the packet is part of a service chain.

In all cases of MAC chaining after a frame (L2, L3, etc.) has been classified the MAC chain begins by prepending the packet with an Ethernet L2 Frame header. The frame will also have a valid 4 byte CRC checksum.

One advantage of MAC chaining is the MAC frame has an overhead of bytes that can leave the L2 MTU unaffected. As with all Ethernet II frames payload must be a minimum of 64 bytes or must be padded to 64 bytes.

#### **4.2. Meta-Data Encoding Consideration for NSH**

A Network Service Header (NSH) is required on NSH aware chain segments where meta-data is being carried. A NSH may be inserted and deleted from the chain depending on the requirements of the specific SFs by the MAC Chaining SFFs as discussed in 4.1.

The current NSH draft [[I-D.ietf-quinn-sfc-nsh](#)] has a mandatory 32 byte service path header. MAC Chaining doesn't require the Service path and Service index, of the NSH for forwarding, and when used for MAC Chaining and utilizing NSH these fields are simply transparent metadata. When used in hybrid chains of MAC chaining and other chain

types the service path header SHOULD carry Service path and Service index values that are relevant to the particular chain segments. MAC chaining can also be used with service functions that understand the NSH header and in this case a proxy operation MUST ensure the service path header has the appropriate values for the service function. A proxy operation may be a distinct function or part of an SFF that Maps or sets the service header appropriately for an SF.

There are several drafts that are proposing context based headers [[I-D.meng-sfc-nsh-broadband-allocation](#)], [[I-D.guichard-sfc-nsh-dc-allocation](#)], and [[I-D.napper-sfc-nsh-mobility-allocation](#)]. In a hybrid SFC chain environment that supports different SFC forwarding on different chain segments, SFFs SHOULD map MAC chainIDs to Service paths or vice versa where required.

### **[4.3. Forwarding](#)**

Forwarding of a packet proceeds from a classifier (SCF) to the Service Function Forwarder (SFF) to the service function (SF) to the next SFF to the next SF and so on until the chain is finished. MAC chaining makes the distinction that the forwarding operations performed by a SFF and a SF are distinct and independent. However implementations may place SFF and SF functions as combined or separate entities. This makes MAC chaining particularly useful for deployment in virtualization environments where a virtual machine may implement one or more SFs and SFFs. Forwarding is a table driven operation. Note that all active chains are normally preprogrammed.

Figure 5 illustrates the table driven forwarding operation of a MAC chaining SFF. Every frame arriving on the ingress VN port is matched to the MAC chaining filtering database. On arrival at the SFF the DA always contains a CS-MAC for the chain segment just crossed. The DA is looked up in the context of its ingress Port (a VN port). A subsequent DA prime (DA' a CS-MAC used as the new frame DA if this segment is DA forwarding) and SA' (a CS-MAC used as the new frame SA if this segment is DA/SA forwarding) and egress Port prime (l') are determined by the lookup.

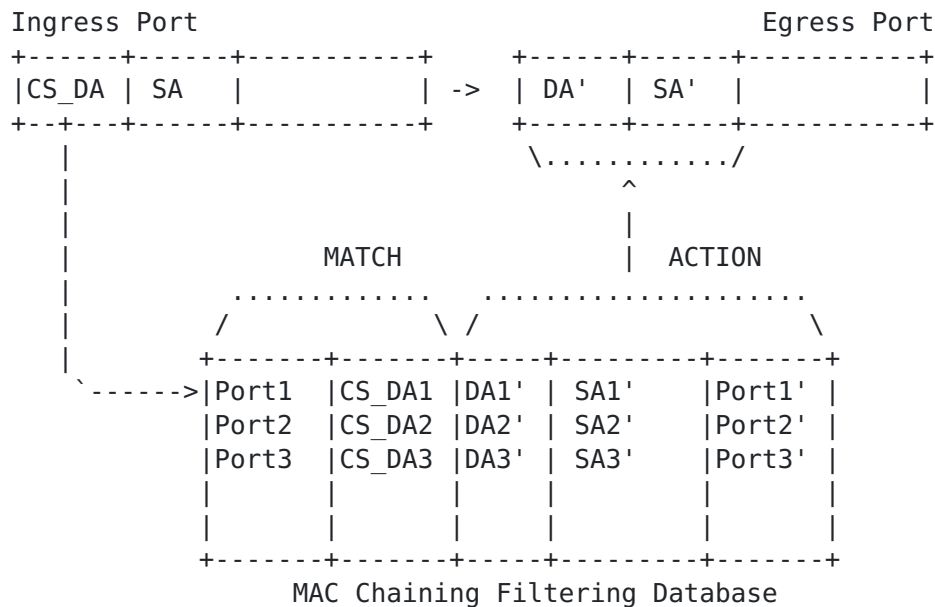


Figure 5 MAC Chain Destination Match Forwarding

Any frame that arrives at an SFF and doesn't exactly match an entry in the MAC chaining filtering database **MUST** be discarded. The filtering database itself is configured by network controller (see [section 6](#)) which creates the chain by programming the MAC chaining filtering database. The MAC chaining filtering database is an exact match database which may use existing Bridge match logic. The exact match filtering with a hash implementation allows the filtering database to easily scale to a large number of chains. Devices in the forwarding path that are not MAC service chaining aware are free to bridge the frame normally or to route using any underlay Layer 3 or 2.5 VN encapsulation.

The Branch Taken (BT) Operation bit (Figure 4) allows an SF to branch the chain by changing the BT bit. Not all service chains are branch capable. If a simple branch chain is desired it must be programmed in the SFF filtering database. A branch operation is indicated by setting a reserved bit in the CS-MAC address. This bit is not read as a bit but as a paired address to the forward direction. The context of the BT bit may be maintained in both the SA and the DA once the bit has been set. An SFF receiving a frame with the BT bit set will

look up the exact match address and forward the frame in the context of the received VN.

More complicated branching requires SF chain awareness. The next hop addresses may be overridden by chain aware SFs to perform more advance branching. A SF must be provided with the allocated addresses for larger branches.

MAC Chaining supports two different types of forwarding methods for SFs which are called DA forwarding and DA/SA forwarding. These two types of forwarding are used for coupling different types of SFs into a chain. The DA forwarding method is suited to operating on existing SFs (such as Firewalls) which provide transparent or Bridge forwarding modes. The DA/SA forwarding method is suited for use with Virtualized SFs that are operating in Virtual Machines or Containers. The main difference between the two methods is when using DA forwarding the SFF encodes the Chain Segment MAC in the DA field. The DA therefore contains an address for the next SFF hop rather than an explicit address for the SF. In DA/SA forwarding the SFF encodes the Chain Segment MAC in the SA rather than the DA. For DA/SA forwarding the SFF uses the DA to directly address the SF. This allows any SF to receive on a single unique address which may be shared by all chains passing through the SF. The choice of DA or DA/SA forwarding is made for each SF of the chain depending on the requirements of the specific SF.

A MAC chain aware SF can determine if the packet is using DA or DA/SA forwarding by determining if the received DA addresses the SF itself and if the SA contains an address allocated by the CS-MAC authority.

#### **4.3.1. Destination Address MAC Chaining Operation**

Destination MAC address chaining uses only the Destination MAC address to key on and implement a chain. Destination Address MAC chaining is used to operate with a MAC chaining un-aware SF which operates in a transparent Bridge mode. A Classifier/Service Function Forwarder (SFF), composing a DA MAC chaining hop, encodes the Chain Segment MAC in the frame DA and an address for the SFF in the frame SA. This encoding will address the next SFF or CTF in the chain. DA MAC chaining may only be used with dual-arm or multi-arm SFs since an unmodified frame can't be returned to the same network where it was received. Any SF along a DA MAC chaining segment must be operating in

Transparent or in Bridging mode so it behaves as a "bump in the wire".

For a Service Function to participate in a DA MAC chaining it must operate in promiscuous receiver, like an Ethernet Bridge, rather than explicit receiver used by Ethernet stations. In promiscuous receiver the SF receives and inspects every frame presented to it independent of the addressing on the frame.

DA MAC chaining is determined by the configuration of the forwarding table in the SFF. After the initial classification the packet is passed to the first SFF (this may be a virtual operation completely within a single chaining switch). The SFF formats the Ethernet frame with a DA of the next hop in the Chain. At each hop the MAC address is looked up in a table similar to figure 5.

#### **4.3.2. Destination and Source Address MAC Chaining**

DA and SA MAC chaining is a variation of MAC chaining that allows MAC chaining aware SFs to use an explicit receiver mode and to support single armed as well as dual and multi-armed SFs. When using DA/SA MAC chaining the SF is individually addressed by a MAC DA and therefore does not need to operate as a promiscuous receiver. This type of SF does not need a MAC lookup table and may be provisioned with a single global or local address under any administration authority (not necessarily the MAC chaining address authority). Service Functions using DA/SA MAC chaining require only a single MAC address regardless of the number of chains passing through them. DA/SA MAC chaining is particularly advantageous for virtual service functions (VNFs) since it reduces the need to flood frames into the virtual NIC supporting the SFs virtual machines and server I/O accelerators.

DA/SA MAC chaining uses both addresses in the Ethernet L2 Header. The DA is used for the next hop device and the SA is used for the subsequent next hop device of the chain. A SF receives a frame; processes the frame; replaces the DA with the received SA and uses resulting DA (received SA) to forward the frame. By specifying 2 hops in a chain the SF can be a very generic operation. The original SA of the received frame does not have to be the address at the SFF that created the header, allowing forwarding flexibility.

#### **4.3.3. Forwarding by Service Functions**

MAC chaining Service Functions (SFs) must be able to pass Ethernet DA/SA addresses through the SF unless the SF is supported by a Proxy

Forwarder (see Proxy Forwarders section below). SFs are not required to pass VLAN Tags. Service Functions supported by MAC chaining can be classified by how they attach to the network as single armed, dual armed or multi-armed. Single arm SFs receive and send all packets on the same VN port. Single armed SFs are typically used when the direction of travel is unimportant to the SF. Dual arm SFs have two VN ports and pass packets between the two VN ports. Dual arm SFs are typically used when the SF needs to know the direction of travel. Multi-arm SFs have more than two VN ports. In a multi-arm (two or more) the SF selects the egress VN port based on its' re-classification of the packet. Each VN port of a multi-arm SF must attach to a different VN or the SF must be MAC Chaining aware. These SFs allow the SFs to branch the chain based on re-classification or to replicate in the chain.

#### **4.3.4. Reverse Path Forwarding by Service Functions**

The BT bit SHOULD be used for the special case where a SF reverses the chain direction. An example of this type of SF is one which proxies TCP. Another example is a loopback function.

An SF uses the BT bit for reverse chain forwarding by setting the BT bit in the destination MAC for frames directed in the reverse direction and leaves the BT bit clear for frames following normal forwarding.

To enable reverse path forwarding, for a particular SF by using the BT bit, the SFF must be configured with a separate match rules for the reverse paths. To support an SF that is reverse path capable the SFF can have up to 4 table match entries (see Figure 5), one for the right direction (e.g. proceeding from left to right), one for the left direction, one for the reverse (BT set to 1) of the right direction, and one for the reverse (BT set to 1) of the left direction. The right and right reverse, or the left and left reverse paths can be independently programmed to allow the reverse paths to be symmetric or asymmetric with the right or left paths.

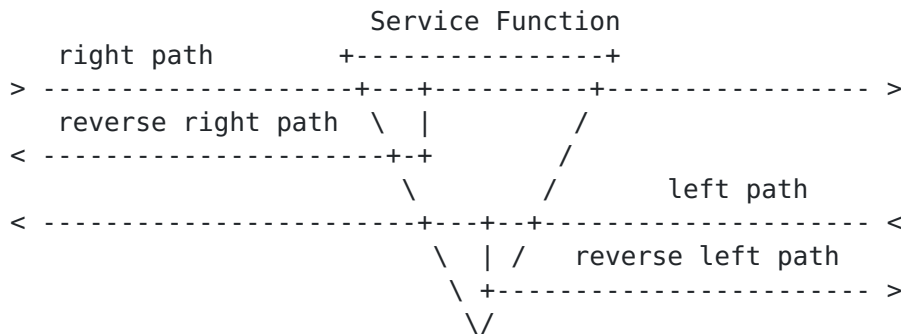


Figure 6: Reversing Paths at a Service Function

In addition to reversing the Service Function Path some chains will need to have re-arranged or re-constructed meta-data for the reverse path. In general this may require re-classification of the packet. If re-construction of meta-data is required this can be handled using exception processing at the SFF initiated on the reverse path match.

#### 4.3.5. Proxy Forwarders

Proxy forwarding is typically for legacy devices or other devices that do not have an ability to support MAC chaining by passing through L2 headers.

Some service functions may reside on devices that do not understand MAC chaining. Legacy functions on middle boxes are one example and service functions that use the NSH header are another example. In these cases a proxy forwarding function is used. Proxies may be integrated with the SFF or located in the switches attaching to the SF. The proxy removes the MAC chaining header and forwards the packet in an appropriate format to the SF. The SF then returns the packet to the proxy upon completion of its operation. The Specific formats of frames between the proxy and the SF, when using a proxy, is out of the scope of this document.

The most basic proxy is a transparent proxy, which must be located between the SF and any underlay entity. A transparent proxy provides a provisioned Ethernet destination which is used for forwarding all frames egressed by the SF at a specific VN port. The use of a transparent proxy constrains the service chain in which it is



inserted since no explicit chain state is passed through the SF by the proxy, however can be highly useful for supporting existing SFs. The combination of a transparent proxy with extended match SFFs can allow simple support for existing high layer SF.

A NSH proxy function converting a packet to a NSH aware SF sets the service path header of the NSH header while leaving other meta data in the NSH untouched.

A more specific proxy technique for chain unaware SFs is to store the CS-MAC by reading the SA on ingress to the SF and then inserting the stored CS-MAC in the DA on egress from the SF.

#### **4.3.6. Example MAC Chaining Walk Through using DA/SA Chaining**

Figure 7 outlines the general path and operations of a MAC chaining.

The Service Classification Function (SCF) determines if a packet matches a predetermined policy for the chain by inspecting the packet then selecting the chain by encoding the frame with next destination equal to the chain segment 1 MAC Address A and itself as the Source Address (SA) designated as H in figure 7.

SFF1 receives a frame from the SCF with Destination Address (DA) equal (exact match) to A and finds the next chain segment by looking up A to find the next DA equal to SF2 MAC Address B and sets the SA equal to chain segment 2 MAC Address C.

SF2 is a single armed Service Function which receives and sends all data on a single network interface. The single SF2 network interface normally connects to a single virtual or physical network. SF2 receives a frame from SFF1 performs its function and then returns the frame to C. This process requires the SF to forward back to the frame's SA, by swapping DA and SA, on the same VN port.

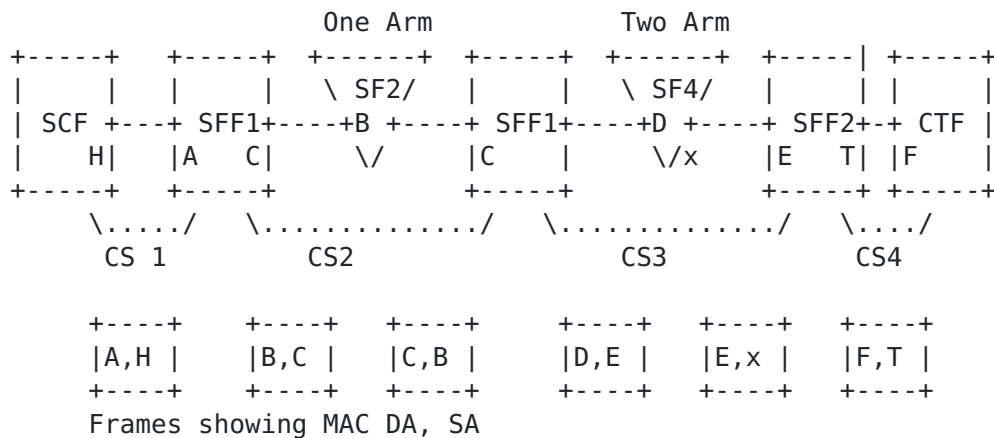


Figure 7 MAC Service Chain Example

SCF: Service Classification Function  
 CTF: Chain Termination Function  
 SF: Service Function  
 SFF: Service Function Forwarder  
 CSx: Chain Segment x.

SFF1 receives a frame from SF2 with DA equal to chain segment 2 MAC Address C, finds the next chain segment by looking up C to find the next destination equal to SF4 MAC Address D and the SA equals chain segment 3 MAC Address E.

SF4 is a dual armed Service Function which receives and sends data on two network interfaces. SF4 always forwards frames between its two interfaces. The two interfaces of SF4 are normally connected to separate virtual or physical networks. SF4 receives the frame from SFF1 with DA equals D and SA equals E performs its function then forwards to E by swapping DA with SA and sends out the packet to the other VN port (A VN port that supports address E as a destination).

SFF2 receives a frame from SF4 with DA equals chain segment 3 MAC Address E, finds the next chain segment by looking up E to find the next destination equals chain segment 4 MAC Address F and SA equals SFF2 MAC Address T.

The CTF receives a frame from SFF2 with destination equals F. The CTF must perform any required packet header adjustment and egress VN port

determination based on the destination equals F and the frame payload (i.e. uses the IP address to route the packet).

#### **4.3.7. Forwarding by Chain Termination Functions**

The forwarding to the final destination by the CTF typically does not use MAC chaining. The CTF is responsible for receiving frames addresses to the termination CS-MAC for each chain, de-encapsulating the packets, and forwarding the packets toward their final destination. One common method which may be used by the CTF for forwarding to the final destination is to route the packets using the IP address of the service packet.

If the service packet (data payload) is an L2 packet then the CTF may use either the IP network addresses or the L2 addresses to forward the packet. The choice between these two CTF forwarding models will depend on the application. Other CTF forwarding models are possible using by using the CS-MAC or meta-data for forwarding.

### **5. Programming a Service Chain**

The capability exists today with open flow enabled switches to specify MAC match criteria and actions that match MAC forwarding all operations. However not all switches are Openflow enabled.

A Yang model could be specified to enable the MAC Chaining operations using an I2RS agent.

Chains must be preprogrammed. Care must be taken to ensure that service chain loops are not programmed (this can easily be verified before a chain is active) however MAC chains that are programmed correctly are inherently loop free in the data plane. The policy is to drop a frame that is not an exact match on any MAC chaining aware SFF.

MAC chaining may be programmed be allowed to pass through bridges that are not MAC chaining aware. It is recommended that this operation be explicitly controlled by setting up port based VLANs designed for this purpose. Ports can add a VLAN tag as part of their forwarding operation. This can be usually be achieved with existing Ethernet controls that allow ports to have service tags added. The VLAN tagging is independent of MAC chaining in this regard.

### **6. Considerations for Operation over NV03 Tunnel Transports**

MAC Chaining can be used with the L3 encapsulation transport tunnels being specified in NV03 ([[I-D.ietf-nvo3-vxlan-gpe](#)], etc.). When using

an NV03 encapsulation it is preferable to use an encapsulation which supports encapsulation of an L2 packet such as [I-D.ietf-nv03-vxlan-gpe], since this allows encoding the MAC addresses used for chain forwarding at the natural layer boundary used to address Virtual Machines or containers.

It is desirable to encode any meta-data header, such as the NSH within an NV03 transport tunnel following an encapsulated L2 MAC header as an Ethernet tag since this will allow a natural protocol layering for delivery of the meta-data to an addressed Virtual Machine or container. Since Virtual Machines and containers are addressed by MAC addresses at the hypervisor vSwitch or system level, the meta-data will be carried as part of the frame layer into the guest OS or container environments along with the MAC addresses used for chaining.

## **7. Domain of operation**

MAC chaining requires connectivity of L2 virtual networks over the service chain path. (This may include multiple VNs that are interconnected.) In many networks this is readily available. Data centers for example can use MAC chain within a physical site that has L2 connectivity.

If Virtualization of the L2 domain is enabled MAC chaining could operate over L2 networks such as NV03 or Ethernet EVPN and an existing L2 Overlay.

## **8. Security Considerations**

MAC chaining is an Ethernet based forwarding operation that follows standard Ethernet rules. VN ports should be qualified with VLANs that limit the scope of MAC chaining frames. This prevents MAC chaining messages from being flooded to external parts of the network or injected into a network from external sources. Programming the VLAN that support MAC chaining is controlled and access to those VLANs is allowed only by trusted devices.

MAC chaining is IP agnostic but like any tunneling protocol it will deliver IP frames to other parts of a network.

## **9. IANA Considerations**

There are no IANA considerations for this document.

## **10. References**

### **10.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [802-2001] "Standard for Local and Metropolitan Area Networks: Overview and Architecture", IEEE 802, Standard 2014.

### **10.2. Informative References**

- [RFC7665] Halpern, J., Pignataro, C. Editors, "Service Function Chaining (SFC) Architecture", [RFC 7665](#), June 2015.
- [I-D.ietf-quinn-sfc-nsh]  
P.Quinn et al., "Network Service Header", [draft-ietf-sfc-nsh-05](#) work in progress), May 26, 2016.
- [I-D.meng-sfc-nsh-broadband-allocation]  
W. Meng et al., "NSH Context Header - Broadband", [draft-meng-sfc-nsh-broadband-allocation-01](#), May 10, 2016
- [I-D.guichard-sfc-nsh-dc-allocation]  
J. Guichard et al., "Network Service Header (NSH) Context Header Allocation (Data Center)", [draft-guichard-sfc-nsh-dc-allocation-04](#), February 15, 2016
- [I-D.napper-sfc-nsh-mobility-allocation]  
J. Napper et al., "NSH Context Header Allocation - Broadband", [draft-napper-sfc-nsh-broadband-allocation-00](#), March 21, 2016
- [I-D.ietf-nvo3-vxlan-gpe]  
P. Quinn et al., "Generic Protocol Extension for VxLAN", [draft-ietf-nvo3-vxlan-gpe-02](#), April 21, 2016

## **11. Acknowledgments**

This document was prepared using 2-Word-v2.0.template.dot.

Copyright (c) 2016 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Authors' Addresses

Don Fedyk  
Hewlett Packard Enterprise  
153 Taylor Street  
Littleton, MA  
Email: don.fedyk@hpe.com

Paul Bottorff  
Hewlett Packard Enterprise  
8000 Foothills Blvd.  
Roseville, CA  
Email: paul.bottorff@hpe.com

Hamid Assarpour  
Broadcom Corporation  
600 Federal Street  
Andover, MA  
Email: hamid@broadcom.com