Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: August 1, 2016

S. Homma
K. Naito
NTT
D. R. Lopez
Telefonica I+D
M. Stiemerling
NEC/H-DA
D. Dolson
Sandvine
A. Gorbunov
Nokia
N. Leymann
Deutsche Telekom AG
P. Bottorff
D. Fedyk
HP Enterprise
January 29, 2016

Analysis on Forwarding Methods for Service Chaining
draft-homma-sfc-forwarding-methods-analysis-05

Abstract

This document presents the results of analyzing packet forwarding methods and path selection patterns for achieving Service Chaining. In Service Chaining, data packets need to be forwarded to the appropriate service functions deployed in networks based on service provided for the packets, and distribution of the service-oriented route information and steering data packets following the route information would be required.

Status of This Memo

This Internet-Draft will expire on August 1, 2016.

Copyright Notice

Table of Contents

1. Introduction

Some IETF working groups of and other Standards Developing Organizations are now discussing use cases of a technology that provides service-oriented traffic forwarding schemes to convey packets to the various service functions, deployed in networks, for providing network services. In this document, we define such technology as Service Chaining. (This draft does not focus only on "Service Function Chaining (SFC)" architecture, and thus, use the term "Service Chaining." SFC is one of approaches to realize Service Chaining.) There are several methods to achieve Service Chaining, and the applicable method will vary depending on the service requirements of individual networks.

This draft assumes that Service Chaining is achieved by the following steps:

a. A traffic classification function identifies the service that is associated to each incoming packets by inspecting the key information such as IP address or 5-tuple.

b. The forwarding path used by packets for reaching the appropriate service functions, is established according to the services provided for the packets. The path might be established in advance.

c. Forwarding functions forward the packets to the next destination along the path established in step b.

d. A service function operates on received packets. Once the invocation of a service function is completed, the packet is forwarded to the next forwarding function.

e. Steps c and d are repeated until each packet has been transferred to all required service functions.

f. After a packet has been transferred to all required Service Functions, it is forwarded to its original destination.

There are several forwarding methods for Service Chaining, and they can be classified into certain categories in terms of distribution of information for setting the paths and decision of the paths. The methods used to distribute the information for path setting and the

patterns used to decide the paths will affect the mechanism of Service Chaining in terms of scalability and service flexibility.

The applicable methods vary depending on network requirements, and thus, classifying and determining forwarding methods will be important in designing the architecture of Service Function Chaining (SFC). This document provides the results of analysis of different forwarding methods for Service Chaining.

OAM, security, and redundancy are outside the scope of this draft.

2. Definition of Terms

Term "Classification", "Classifier" referred to [RFC7665]. Term "Service Function", "Service Node" referred to [I-D.ietf-sfc-dc-use-cases].

Service Chaining: A technology that enables data packets to invoke a set of service functions.

Classification: Locally instantiated matching of traffic flows against policy for subsequent application of the required set of network service functions. The policy may be customer/network/service specific.

Classifier (CF): An element that performs classification.

Service Function (SF): A function that is responsible for specific treatment of received packets. A Service Function can act at various layers of a protocol stack (e.g. at the network layer or other OSI layers). A Service Function can be a virtual element or be embedded in a physical network element. One of multiple Service Functions can be embedded in the same network element. Multiple occurrences of the Service Function can be enabled in the same administrative domain.

One or more Service Functions can be involved in the delivery of added-value services. A non-exhaustive list of Service Functions includes: firewalls. WAN and application acceleration, Deep Packet Inspection (DPI), LI (Lawful Intercept) module, server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], NPTv6 [RFC6296], HOST_ID injection, HTTP Header Enrichment functions, TCP optimizer, etc.

Forwarder (FWD): The entity, responsible for forwarding data packets according to the ordered set of service functions that need to be invoked. A forwarder maintains one or more forwarding tables,

which contain entries that asset the forwarder in its forwarding decision-making process.

Control Entity (CE): One or a set of control entities responsible for managing service topology and indicating forwarding configurations to forwarders.

Service Chain (SC): A service chain defines an ordered list of service functions that must be applied to packets selected as a result of classification. The implied order may not be a linear progression as the architecture allows for nodes that copy to more than one branch.

Service Path (SP): The forwarding path followed by packets that are associated to a given service chain. Packets follow a service path through the requisite service functions that need to be invoked, as per the service chain instructions. Service path shows a specific path that traverses several service function instances. For example, SC is written as SF#1 -> SF#2 -> SF#3 (This shows an ordered list of SFs), and SP is written as SF#1_1(1_1 means instance 1 of SF1) -> SF#2_1 -> SF#3_1.

Segmented Service Path: A Segmented Service Path is an actual path established between FWDs. A service path might be composed of some segmented service paths.

Service Chaining Domain (SC Domain): The domain managed by one or a set of CEs.

Service Path Information (SP Information): The information used to forward packets to the appropriate SFs according to the service that needs to be provided. Examples of SP information include routing configuration for forwarders, headers for forwarding packets to required SFs, and service/flow identifiable tags.

3. Classification of Forwarding Methods and SP Selection Patterns

3.1. Forwarding Methods

In Service Chaining, data packets are transferred to service functions, which might be located outside the regular computed path to the original destination. Therefore, a routing mechanism that is different from general L2/L3 switching/forwarding might be required. The forwarding mechanism can be classified into three methods in terms of distribution of SP information and packet forwarding.

3.1.1. Method 1: Forwarding Based on Flow Identifiable Information

The mechanism of method 1 is shown in Figure 1. In this method, forwarding configuration information is based on flow identifiable information, such as 5-tuple (e.g. dst IP, src IP, dst port, src port, tcp) are indicated to the CF and each FWD. There might be an CE to handle this. The flow identifiable information can be constructed with some fields of L2, L3 or L4 or combination thereof. The information can be configured either before packets arrive, or at the time packets arrive at CF and FWD. Each FWD identifies the packets with flow identifiable information and forwards the packets to the SFs according to the configuration. This method does not require the modification of any field in the original packet header.

```
*Distribution model of SP information*

      +----------------+
      | Control Entity |
      +----------------+
           ^ |     indication of routing configuration
           | |           based on packet identifiable information
           | +---------------+-------------------------------+--------->
           | |               |                               |
           | |               |                               |
           | v               v                               v
       +--------+        +-------+        +------+       +-------+
------>|   CF   |------> |  FWD  |------> | SF#1 |------>|  FWD  |----->
       +--------+        +-------+        +------+       +-------+

////////////////////////////////////////////////////////////////////////
*Forwarding Tables*

Locate:     [CF]             [FWD]                           [FWD]

Table:   192.168.1.1       192.168.1.1                    192.168.1.1
          ->FWD#1           ->SF#1                         ->SF#2
         10.0.1.1          10.0.1.1                       10.0.1.1
          ->FWD#1           ->FWD#2                        ->SF#2
         ...               ...                            ...

////////////////////////////////////////////////////////////////////////
*Condition of Packet*

Locate:     [CF]             [FWD]           [SF#1]          [FWD]

         +--------+        +--------+       +--------+      +--------+
Packet:  |  PDU   |        |  PDU   |       |  PDU   |      |  PDU   |
         +--------+        +--------+       +--------+      +--------+
```

Figure 1: Forwarding Based on Flow Identifiable Information

3.1.2. Method 2: Forwarding with Stacked Headers

The mechanism of method 2 is shown in Figure 2. In this method, the CF classifies packets and stacks headers in which actual network address is included, e.g., MPLS, GRE headers or IPv6 Segment Routes, onto the packets based on the classification. The packet is transferred to the destination according to the outermost header, and a SF or FWD, as the destination, removes the outermost header after receiving the packet. The processes are repeated until all stacked

headers are removed. This method does not require any forwarding entries for forwarding packets based on the service information.

```
*Distribution model of SP information*

        +----------------+
        | Control Entity |
        +----------------+
            ^ |
            | |    indication of
            | |      stacking headers
            | v
         +--------+       +-------+       +------+       +------+
-------->|   CF   |------>| SF#1  |------>| SF#2 |------>| SF#3 |------>
         +--------+       +-------+       +------+       +------+

////////////////////////////////////////////////////////////////////////
*Forwarding Tables*

Locate:       [CF]

Table:    192.168.1.1          __/__/__/__/__/__/__/__/__/__/__/__/__/
           ->Stack #1,2,3      __/ Packets are forwarded to SFs by __/
          10.0.1.1             __/ the outermost header.           __/
           ->Stack #1,3        __/__/__/__/__/__/__/__/__/__/__/__/__/
          ...

////////////////////////////////////////////////////////////////////////
*Condition of Packet*

Locate:       [CF]           [SF#1]          [SF#2]         [SF#3]

           +---------+
Header:    |To SF#1 |
           +---------+      +---------+
           |To SF#2 |       |To SF#2 |
           +---------+      +---------+    +---------+
           |To SF#3 |       |To SF#3 |     |To SF#3 |
           +---------+      +---------+    +---------+
                :                :              :               :
           +---------+      +---------+    +---------+      +---------+
Packet:    |  PDU    |      |  PDU    |    |  PDU    |      |  PDU    |
           +---------+      +---------+    +---------+      +---------+
```

Figure 2: Forwarding with Stacked Multiple Headers

3.1.3. Method 3: Forwarding Based on Service Chain Identifiers

The mechanism of this method is shown in Figure 3. In this method, the corresponding service chain identifier is mapped to each packet by a CF based on the classification. The forwarding configuration based on the identifiers is sent to each FWD. Each FWD identifies the SP assigned to the received packet from the identifier, and forwards the packet to the next hop. After a packet has traversed all SFs, the identifier is removed and the packet is transported to the original destination.

```
  *Distribution model of SP information*

      +----------------+
      | Control Entity |
      +----------------+
           ^ |     indication of attached tag
           | |       and routing configuration based on tags
           | +----------------+------------------------------+--------->
           | |                |                              |
           | |                |                              |
           | v                v                              v
        +--------+        +-------+       +------+       +-------+
  ----->|   CF   |------> |  FWD  |------>| SF#1 |------>|  FWD  |----->
        +--------+        +-------+       +------+       +-------+

  //////////////////////////////////////////////////////////////////////
  *Forwarding Tables*

  Locate:  [CF]             [FWD]                          [FWD]

  Table: 192.168.1.1        IF ID#1,3                   IF ID#1,2,5
          ->Stack ID#1       ->SF#1                       ->SF#2
         10.0.1.1
          ->Stack ID#2
         ...                ...                         ...

  //////////////////////////////////////////////////////////////////////
  *Condition of Packet*

  Locate:  [CF]             [FWD]        [SF#1]           [FWD]

         +-------+        +-------+      +-------+       +-------+
  SC-ID: | ID#1  |        | ID#1  |      | ID#1  |       | ID#1  |
         +-------+        +-------+      +-------+       +-------+
  Packet:|  PDU  |        |  PDU  |      |  PDU  |       |  PDU  |
         +-------+        +-------+      +-------+       +-------+
```

Figure 3: Forwarding Based on Service Chain Identifiers

Then, there are mainly three approaches to map service chain identifiers to packets as follows.

o Tagging an extra header:

In this approach, an extra header which has a service chain identifier is attached on each packet. This document defines such headers as service identifiable tags. Some existing tags, such as

VLAN-tag or MPLS-tag, or dedicated headers, such as NSH, could be used as service identifiable tags. As an example, SFC[RFC7665] is categorized into this approach. An example of packet format in tagging approach with NSH is shown in Figure 4. In this example, a service chain identifier is included in NSH.

```
+----------+-------+--------+-------------~~--+
|   NSH    | Ether | IPv6   |IPv6 Payload     |
| \SC-ID   | Header| Header |                 |
+----------+-------+--------+-------------~~--+

           |---      Ethernet Packet       ---|
```

Figure 4: Packet Format in Tagging Approach

- Inserting into an optional field:

  In this approach, a service chain identifier is inserted into an optional field inside a packet frame, such as IPv6 extension header. An example of an IPv6 packet with a service chain identifier inserted as an extension header is shown in Figure 5.

```
+-------+--------+----------+-------------~~--+
| Ether |IPv6    |IP Opt.Fld|IPv6 Payload     |
| Header|Base Hdr| \SC-ID   |                 |
+-------+--------+----------+-------------~~--+

        |--  IPv6 Header  --|

|---         Ethernet Packet               ---|
```

Figure 5: Packet Format in Inserting Approach

- Overloading on a destination or source address:

  In this approach, service chain identifier is overloaded on a destination or a source address such as MAC or IP address. In other words, the addresses are used for both showing the destination or source in network and identifying service chain which each packet belongs to. An address is required for each hop in a service chain, and FWDs switch the address to new one for the next hop by referring the address of the received packet. An

example of using destination address overloading is shown in Figure 6. In this example, SFs are used as L2 transparent mode, and service chain identifiers are overloaded on destination MAC addresses. FWD2 refers the destination MAC address which shows the address for Port B, and changes it to the address for port D for sending the packet to the next hop in the service chain. When using non-transparent SFs in the overloading approach, the identifier is carried from the FWD to the SF in the source address(SA) and is carried from the SF to the FWD in the destination address(DA). More detailed processes of the overloading approach using MAC addresses is described in Ethernet MAC Chaining[I-D.fedyk-sfc-mac-chain].

```
   *Network Structure*

              Port A        Port B   Port C        Port D
                 |           |          |           |
                 |           |          |           |
        .------. |  +[SF1]+  | .------. |  +[SF2]+  | .------.
       /        \v  |     |  v/        \v  |     |  v/        \
    --#   FWD1   #--+     +--#   FWD2   #--+     +--#   FWD3   #--
       \        /  ^          \        /  ^          \        /
        '------'   |           '------'   |           '------'
                   |                      |
   . . . . . . . . | . . . . . . . . . . .| . . . . . . . . . . . . .
                   |                      |
   *Packet Frame*  |                      |
                   |                      |
        +--------+-------~-+        +--------+-------~-+
        |MAC-DA:B| Payload |        |MAC-DA:D| Payload |
        +--------+-------~-+        +--------+-------~-+

           ==============   Flow Direction   ==============>
```

Figure 6: Overview of DA Overloading Approach

3.2. Service Path Selection Patterns

Since SC contains only logical information (e.g., a set of services that are associated with flows and their sequences), the actual instances, which are called SPs, are needed in order for the forwarding process to work. In this process, an instance of SP is created at certain points during a packet's delivery. Therefore, to forward packets, the SC needs to be turned into an SP, which indicates specific FWDs (or switches, routers) and SFs that the

packets will be forwarded to. From the perspective of points translating SC to SP, the methods that establish SPs from end-to-end are classified into two patterns.

3.2.1. Pattern 1: Static Selection of End-to-End Service Path

The translation point is a CF; that is, the SP is statically pre-established as an end-to-end path and a CF forwards packets along the appropriate path based on the result of the classification. Each FWD on the SP has a forwarding table to uniquely determine the next destination of packets, and each FWD statically forwards the received packets toward the next destination based on the table. FWD requires only a function to receive indications of forwarding configurations from the CE. Pattern 1 can be achieved in the following models.

3.2.1.1. SF Dedicated Model: Network Slicing Model

In this model, an SF instance (or a set of SF instances) is used by only one single SP; in other words, a set of SF instances is prepared for each SP. This model also enables operators to establish SPs without any FWDs by slicing network physically or virtually and deploying a set of SFs required for service providing in each sliced network. A CF assigns packets to the network in which the appropriate SF set is installed inline, and the packets traverse the SFs by being forwarded along the pre-configured route. The overview of network slicing model is shown in Figure 7.

```
  *Path Structure*
               * * * * * * * * * * * * * * * * * * * * * * *
               * network#1                                 *
    +----+     *      +------+     +------+     +------+   *
    |    |SC#1 *      | SF#1 |     | SF#2 |     | SF#3 |   * SP#1
    |    |=======================================================>
    |    |     *      +------+     +------+     +------+   *
    |    |     * * * * * * * * * * * * * * * * * * * * * * *
    |    |
    |    |     * * * * * * * * * * * * * * * * * * * * * * *
    |    |     * network#2                                 *
    |    |     *      +------+                 +------+   *
    |    |SC#2 *      | SF#4 |                 | SF#5 |   * SP#2
    |    |=======================================================>
  ->| CF |     *      +------+                 +------+   *
    |    |     * * * * * * * * * * * * * * * * * * * * * * *

      .           .
      .           .
      .           .
               * * * * * * * * * * * * * * * * * * * * * * *
               * network#n                                 *
               *      +------+     +------+     +------+   *
    |    |SC#n *      | SF#6 |     | SF#7 |     | SF#8 |   * SP#n
    |    |======================================================>
    +----+     *      +------+     +------+     +------+   *
               * * * * * * * * * * * * * * * * * * * * * * *
                                       SC:Service Chain  SP:Service Path
  ///////////////////////////////////////////////////////////////////////
  *How packets traverse*

  Service Chain#1:
  SP#1
    [ CF ]----------->[ SF#1 ]---->[ SF#2 ]----->[ SF#3 ]------->

  Service Chain#2:
  SP#2
    [ CF ]----------->[ SF#4 ]------------------>[ SF#5 ]------->
      :
  Service Chain#n:
  SP#n
    [ CF ]----------->[ SF#6 ]---->[ SF#7 ]----->[ SF#8 ]------->
```

Figure 7: SF Dedicated Model

3.2.1.2. SF Shared Model

In this model, an SF is shared by multiple SPs. Several SPs are mixed at shared SFs, and thus this method requires FWDs for forwarding each packet to the corresponding next hop by identifying the SP which each packet belongs to. The overview of the SF shared model is shown Figure 8.

```
 *Path Structure*

   +----+     +---+   +----+   +---+   +------+   +---+   +----+
   |    |SC#1 |FWD|   |SF#1|   |FWD|   |SF#2_1|   |FWD|   |SF#3| SP#1
   |    |==============================================================>
   |    |SC#2 |   |   |    |   |   |   +------+   |   |   |    | SP#2
   |    |============================# +------+ #======================>
   |    |     |   |   +----+   |   | # |SF#2_2| # |   |   +----+
   |    |     |   |            |   | #==========# |   |
 ->| CF |     +---+            +---+   +------+   +---+
   |    |
     .         .
     .         .
     .         .
              +---+   +----+                      +---+   +----+
   |    |SC#n |FWD|   |SF#4|                      |FWD|   |SF#5| SP#n
   |    |==============================================================>
   +----+     +---+   +----+                      +---+   +----+

                                       SC:Service Chain  SP:Service Path
 ///////////////////////////////////////////////////////////////////////
 *Packet Flow*

 Service Chain#1:
 SP#1
   [ CF ]---->[FWD]-->[SF#1]-->[FWD]-->[SF#2_1]-->[FWD]-->[SF#3]--->

 Service Chain#2:
 SP#2
   [ CF ]---->[FWD]-->[SF#1]-->[FWD]-->[SF#2_2]-->[FWD]-->[SF#3]--->
     :
 Service Chain#n:
 SP#n
   [ CF ]---->[FWD]-->[SF#4]--------------------->[FWD]-->[SF#5]--->
```

Figure 8: SF Shared Model

3.2.2. Pattern 2: Dynamic Selection of Segmented Service Path

The mechanism of this pattern is shown in Figure 9. The translation points are CFs and some FWDs. The SP is established by a series of multiple paths, which are sectioned by CFs and FWDs. The resulting path is referred to as a segmented path in this draft. CFs or FWDs that select the next segmented path might require notification of forwarding configuration information from the CE. Moreover, some FWDs require functions to select the destination of packets from various alternatives and to retrieve the information for selecting the next path. For example, each FWD obtains metric information or load conditions of servers and selects an optimal segmented path based on the information. The CE might support the selection mechanism and may notify CFs or FWDs of it.

```
 *Path Structure*

   +----+     +---+   +----+   +---+   +------+   +---+   +----+
   |    |SC#1 |FWD|   |SF#1|   |FWD|   |SF#2_1|   |FWD|   |SF#3| SP#1
   |    |========================*=====================================>
   |    |     |   |   |    |   | # |   +------+   |   |   |    | SP#2
   |    |     |   |   |    |   | # |   +------+ #======================>
   |    |     |   |   +----+   | # |   |SF#2_2| # |   |   +----+
   |    |     |   |            | #==============# |   |
 ->| CF |     +---+            +---+   +------+   +---+
   |    |
     .         .
     .         .
     .         .
              +---+   +----+                      +---+   +----+
   |    |SC#n |FWD|   |SF#4|                      |FWD|   |SF#5| SP#m
   |    |==============================================================>
   +----+     +---+   +----+                      +---+   +----+

                                       SC:Service Chain  SP:Service Path
 ///////////////////////////////////////////////////////////////////////

 *How packets traverse*

 Service Chain#1:
 SP#1
   [ CF ]---->[FWD]-->[SF#1]-->[FWD]-->[SF#2_1]-->[FWD]-->[SF#3]--->

 SP#2
   [ CF ]---->[FWD]-->[SF#1]-->[FWD]-->[SF#2_2]-->[FWD]-->[SF#3]--->
     :
 Service Chain#n:
 SP#m
   [ CF ]---->[FWD]-->[SF#4]--------------------->[FWD]-->[SF#5]--->
```

Figure 9: Dynamic Selection of Segmented Service Path

In addition, this pattern supports the establishment of hierarchical domains discussed below:

3.2.2.1. Hierarchical Service Path Domains

Complex problems often become manageable with a hierarchical approach. This pattern allows network-wide orchestration of Service Chaining to be relatively simple, while hiding the complexities of fine-grained policy-based path selection within sub-domains. Each

sub-domain can be independently administered and orchestrated. This architecture is described in [I-D.dolson-sfc-hierarchical].

Figure 10 shows two levels of hierarchy in a service provider's network. At the top level in the hierarchy, Service Chaining components are:

1. Edge-classifiers (Edge CF) that reside near the edge of a service provider's domain.

2. SF sub-domains that reside in data centers.

3. Internal Boundary Nodes (IBNs) that reside in data centers, linking together the levels of the hierarchy. To the higher level, sub-domains are viewed as a SF. To the lower level, this is a classifier and FWD.

How packets traverse

```
     +----+    +-----+  +----------------------+   +-----+
     |    |SC#1| FWD |  | IBN#1                |   | FWD |
   ->|    |================*                *=====================>
     |    |    +-----+  |  # (in DC#1)      #  |   +-----+
     |    |             |  V                #  |
     |Edge|             |+---+            +---+|   Top domain
     | CF |    * * * * *||CF | * * * * * *|FWD|| * * * * *
     |    |    *        |+---+            +-+-+|         *
     |    |    *        | | |              | | |    Sub  *
     |    |    *        +-o-o--------------o-o-+   domain*
     |    |    *   SC#1.2 | |SC#1.1        ^ ^       #1  *
     |    |    *    +-----+ |              | |           *
     |    |    *    |       V              | |           *
     |    |    *    |     +---+  +------+  | |           *
     |    |    *    |     |FWD|->|SF#1_1|--+ |           *
     |    |    *    |     +---+  +------+    |           *
     |    |    *    V                        |           *
     |    |    *  +---+  +------+  +---+  +------+       *
     |    |    *  |FWD|->|SF#1_2|->|FWD|->|SF#2_1|       *
     |    |    *  +---+  +------+  +---+  +------+       *
       .       * * * * * * * * * * * * * * * * * * * * * *
       .
     |    |    +-----+   +---------------------+     +-----+
     |    |SC#n| FWD |   | IBN#q                |     | FWD |
     |    |=======================================================>
     |    |    +-----+   |     (in DC#m)       |     +-----+
     +----+              +---------------------+
                     (Details of sub-domain #q not shown)
```

Figure 10: Service Chain Hierarchy in Service Provider Networks

The components within an SF sub-domain are opaque at the top level; each IBN acts as a single SF node in the top-level domain. A service path in the top-level domain may visit multiple sub-domains.

At the lower level in the hierarchy, each sub-domain contains an independently administrated Service Chaining network, generally comprised of multiple instances of multiple types of hosts, most likely (but not necessarily) within the same data center. There is no need for knowledge of the "big picture" at the level of the SF-sub-domain except as required to forward packets to the other SFs that are the next hop of each chain.

Note that different encapsulation methods can be used at each layer in the hierarchy, provided the SF domain-Proxy can translate between them. For example, MPLS could be used to deliver packets from

network edge to the SF clusters within data centers, and NSH [I-D.ietf-sfc-nsh] could be used within the data center.

Details of Top Level of Hierarchy

In this pattern, referring to Figure 11, network-wide Service Chaining orchestration is only concerned with creating service paths from network edge points to sub-domains within data centers and configuring classifiers at a coarse level to get the correct hosts' traffic onto paths that will arrive at appropriate sub-domains. The figure shows one possible service chain passing from edge, through two sub-domains, to network egress.

This top level of orchestration may attach metadata to provide context from the network edge into the data center.

```
                          +------------+
                          |Sub-domain#1|
                          |  in DC1    |
                          +----+-------+
                               |
                        .------+---------.   +--+
                +--+   /     /  |         \--|CF|
            --->|CF|--/---->'   |          \ +--+
                +--+ /  SC#1    |           \
                     |          |            |
                     |          |    .------>|--->
                     |         /    /        |
                     \         |   /        /
                +--+  \        |  /        /  +--+
                |CF|---\       V /        /---|CF|
                +--+    '------+---------'    +--+
                               |
                          +----+-------+
                          |Sub-domain#2|
                          |   in DC2   |
                          +------------+
```

Figure 11: Network-wide view of Top Level of Hierarchy

The orchestration at this top level must ensure bidirectional path symmetry so that inbound packets traverse sub-domains in the reverse order as outbound packets.

Because classifiers must have rules to handle any traffic passing through the network, we believe that a useful approach to classification will be to assign traffic to service function paths on the basis of coarse classification like subscriber tier, tenant or

VRF identifier. These classification rules could be relatively static, changing in response to provisioning but not in response to traffic.

In some networks, it might be possible to create a rule per residential subscriber, resulting in rule updates when subscribers are assigned IP addresses. However, with judicious allocation of IP blocks, entire classes of subscribers could be classified with IP-prefix rules. Similarly, in a mobile network path selection could be based on the APN (Access Point Name) identifier.

Hence, there are methods of globally managing very large networks by choosing a suitable classification granularity.

Details of Lower Levels of Hierarchy

Within each SF sub-domain, there are:

1. An IBN to receive incoming data packets on any of the configured service chains and load-balance (if necessary) traffic to classifiers,

2. Classifier(s) to select internal service chain to use, potentially based on stateful flow analysis, DPI, etc.

3. Service components comprised of FWD and SF.

Local Service Chaining orchestration is concerned with providing viable paths to various functions, providing failure recovery, NFV elasticity, etc.

Classification within each sub-domain can be concerned with determining the local service paths for individual transport-layer flows based on ports, DPI and meta-data provided by the higher-level chain.

For any classifier that is transport-layer-stateful, it is most efficient for the same classifier instance to handle traffic in both directions of a bidirectional connection. State tracking may require that service function paths begin and terminates at the same node with the flow state, where the same classifier instance can be used for both directions of traffic.

4. Consideration on Forwarding Methods and Paths Selection Patterns

This chapter presents the results of analyzing the forwarding methods and architecture patterns in chapter 3.

4.1. Analysis of Forwarding Methods

4.1.1. Analysis of Method 1: Using Flow Identifiable Information

Data Plane Aspects

This method can achieve Service Chaining without changing packet format, such as attaching any header on packets, so it may not imply any overhead or be subject to MTU restrictions. Furthermore, this method does not require additional functions for SFs to apply or handle any header because data packets are transported unaltered. Therefore, it will be easier to use legacy SFs for network operators.

On the other hand, it is difficult to forward a packet to same FWDs several times because flow identifiable information is not basically changed in the forwarding processes. For example, distinction of incoming ports will be required for FWD to resolve the next hop appropriately when a packet traverses it several times.

Control Plane Aspects

This method might be achieved by using existing control mechanisms. For example, openflow, which is able to provide flexible forwarding control, would be available for creating SPs.

However, this methoed might require FWDs to configure forwarding entries for each flow to each FWD. For example, if there are 10,000 flows to be handled at a CF/FWD, the forwarding table for each CF/FWD uses 10,000 flow entries at most. Therefore, it might not be feasible for large-scale networks such as carrier networks that handle a SC per user (which means that individual users will be associated with different policies), because some large carriers have over a million users and even more flows. In addition, control signaling would increase because forwarding configuration for each flow to each FWD is required. Moreover, it may be hard to use this method if some SFs modify header fields of a packet or frame, for example, NAT/NAPT, in a chain. For example, if a NAT changes the IP address of packets dynamically, the FWDs that follow need to renew their forwarding tables. This method also have restriction about forwarding based on high-layer information, such as application information in packet payload. The process of detecting high-layer information is usually heavy compared with L2 or L3 forwarding process, and most existing forwarding functions have capability to refer only under L4 headers. Therefore, it will be difficult to use this method to forward packets along SPs decided by detecting high-layer

information since individual L2-4 packet headers may not retain enough information. An example of this type of problem is a video streaming imbedded within a web page. The identifiable information at the L2-4 level does not allow differentiation between the video stream and the rest of the frame, and thus the all traffic on the web page is forwarded following the same SC.

The results of the above analysis suggest that, although this method is beneficial in terms of impact to existing network, it would not be scalable. Therefore, this method might be suitable for networks with a limited number of flows.

Measurements taken in multiple residential service providers' networks indicate that for each 1Gbps of traffic the sustained rate of new flows can range from 1,000 flows/s to 30,000 flows/s. From this, for example, there would be between 10,000 and 300,000 new flows/s on a 10 Gbps link. Therefore, in some networks at some times of day, this method using 5-tuple as flow identifiable information would require sustaining up to 300,000 table updates per second for each FWD. This incurs a significant amount of control traffic and computational effort.

4.1.2. Analysis of Method 2: Stacking Headers

Data Plane Aspects

In this method, SP information is attached on each packet as headers for forwarding, and the number of the headers increases depending on the number of SFs which the packet will traverse. This means that the size of each packet increases. Packet sizes may be restricted by the minimum available MTU of any link in the network and exceeding the MTU will require to fragment the original packets. Fragmentation adds a new source of errors and may require forwarding processes to be more complex. For example, the whole original packet will be discarded even if one of fragments of the packet gets lost, or in terms of SF equipment, it would be very wasteful of CPU if fragmented packets need to be reassembled at every SF resources, and some equipment has restricted resources and memory for reassembly. Fragmentation will also cause an increase in traffic as more packets have to be processed by the network.

Moreover, this method requires SF to be applied to the headers because they receive packets with optional headers. Therefore SFs will be required to be able to recognize the headers, or proxy functions, which remove the headers before inserting packets into SFs and re-attach the appropriate headers on the returned packet, will be required. In addition, when a SF is used by multiple SCs,

it will be challenging for SFs to process packets because header length attached on each packet may vary and SFs are required to have a mechanism to recognize the header length for each packet.

Control Plane Aspects

In this method, none of the FWDs require any specific forwarding tables for Service Chaining or interface to receive forwarding configuration information. In short, FWDs are stateless or eliminated at hops, and this method has advantages of high scale in SPs managements and lower latency. In addition, no CEs will be required to manage the forwarding configuration of FWDs for Service Chaining, and so the control mechanism might become simple compared with other methods.

On the other hand, some relay nodes such as switches or SFs are required to have a function to remove the outermost header from the received packets. FWDs also don't have to identify flows or services, so cannot change the following SPs. Moreover, CF must grasp all of addresses of relay nodes which packets will traverse, and it will require any CE to manage addresses of relay nodes and a link between CF and the CE. There are already several existing technologies that can be used to achieve this method, such as segment routing.

The results of the above analysis indicate that this method would be appropriate when the number of SFs in a SC is small, and most SFs are deployed in a single domain. On the other hand, it may be unsuitable in cases where there are many SFs in a chain, or packets have to traverse multiple domains.

4.1.3. Analysis of Method 3: Using Service Chain Identifier

Data Plane Aspects

The common features of this method and the individual features of each approach to map service chain identifiers in terms of data plane aspects are described below.

- Common features of method3

  In this method, a service chain identifier, defined for each SC, is mapped into each packet. FWDs recognize the next hops of received packets from the identifiers independent of any information of original packets. Therefore, SFs which modify original packet format can also be used. In addition, it is

easy to change the following SPs on a route by renewing the identifier.

On the other hand, attachment of an identifier might expand packet size, and it would cause an increase of traffic amount or problems which happens as a result of exceeding MTU (The problems are stated in Section 4.1.2.). However, by adopting a single fixed-length identifier, the problems might be prevented. Or, when overloading the identifier on an existing field, such as MAC address, packet size is not changed and such issues would not occur.

Moreover, forwarding along SPs is provided based on service chain identifiers, and so if there are network nodes which are unaware of the identifiers, such as routers without functions to forward packets based on the identifiers, in a SC domain, some tunnel would be required for passing packets over them.

- Tagging an extra header:

  In this approach, the identifiers are prepended to packets, and so a single mapping mechanism could be used independently of the formats of the target packets.

  Conversely, this approach requires SFs to parse the extra headers (Problems which happens as result of inserting packet with optional headers into SFs are stated in Section 4.1.2). In case that an existing header, which SFs can recognize, is used as a service identifiable tag, this problem might be restricted. For example, some SFs can recognize VLAN- tags, and they would not need any improvement for the SFs if they are used as service identifiable tags. However, using an existing header might have some effects on the original uses.

- Inserting into an optional field:

  In this approach, service chain identifiers are inserted in some field of the original packets, and the packets seem normal formats from SFs. Therefore, any improvement for enabling SFs to handle the identifiers would not be required.

  Meanwhile, identifier insertion or packet forwarding mechanisms would vary depending on the formats of the original packets, because positions where identifiers are inserted are different for each packet format. For example, optional field positions of IPv4 and IPv6 headers are different. Furthermore, especially, the inserting approach, using IPv4 optional fields, might have some problems. For example, some server OS and

applications strip the IPv4 optional field due to security concerns. Therefore, it appears this is a difficult solution for IPv4 networks.

Also, in case that existing field is used for storing the identifier, amount of identifier information might be small compared with tagging an extra header approaches.

- Overloading on a destination or source address:

  In this approach, a destination or source address is used for identifying service chain which the packet belongs to in addition to original usage, and so packets size increase caused by attaching additional headers does not occur. Also, any improvement for enabling SFs or any other network equipment to handle the identifiers would not be required, because the packets seem normal formats from them. In other words, this approach can coexist with legacy equipment.

  Meanwhile, the addresses for Service Chaining are overwritten on the original address in this approach, and so an additional encapsulation would be required during the Service Chaining process when retaining the original address information. Therefore, for cases when L2 or L3 addresses are used for identifying subscribers, the overloading approach might require the MTU expansion for additional encapsulation. Moreover, when using L2 addresses as service chain identifier and sending packets to another L2 domain across a L3 domain, an extra means such as L3 tunnel is required.

Control Plane Aspects

The common features of this method and the individual features of the overloading approach in terms of control plane aspects are described below.

- Common features of method3

  This method enables FWDs to save resources for managing forwarding tables and allows all SPs to be established in advance in most of cases. This prevents an increase of control signals such as openflow or Gx/Sd, and also enables changing the following SPs without changing forwarding configuration of FWDs.

On the other hand, this method requires a new control mechanism based on service chain identifiers, therefore, FWDs, CE and interface between them have to be updated to apply forwarding configuration based on the identifiers.

- Overloading on a destination or source address:

  Overloading approach might be achieved without new control mechanisms or drastic remodeling of existing control entities. For example, MAC chaining can be established by using programmed standard openflow switches.

  On the other hand, in the overloading approach, each SP is composed of a set of unique addresses, and thus FWDs are required to have addresses as many as service chains which pass through them.

The results of the above analysis indicate that this method has many advantages in terms of scalability, and it might be appropriate for use in large-scaled networks in which there are so many SFs and various types of flows. On the other hand, when the identifier handling mechanism is an entirely new architecture such as SFC[RFC7665], renewal or introduction of several equipment such as FWDs and CE will be required.

4.2. Analysis of Service Path Selection Patterns

4.2.1. Analysis of Pattern 1: Static SP Selection

In this pattern, the mechanism of FWDs would be simpler than the one in pattern 2 because FWDs do not require any functions to select paths or retrieve any information for next hop resolution purposes. Moreover, it is not necessary to maintain the state of each flow. Therefore, existing network virtuarizing techniques, such as VxLAN or MPLS, can be used to achieve Service Chaining in this pattern. Especially, network slicing model does not require any special forwarding mechanisms.

On the other hand, this pattern has restriction in the management of SPs. When adding new SFs to a SC, removing SFs from a SP, or migrating SFs to other locations, re-establishment of SP would be required. This restriction in network slicing model would be more strict because this model need to establish a new network for adding a SP. For relaxing the restriction, it is desirable to use this pattern together with a means, such as load balancer, which enable to add the same kind SFs into a SP without changing the configuration of the SP. Or the restriction would be relaxed when network

virtuarizing technique progresses significantly and network operaters can install SFs more freely.

In addition, this pattern would also have restriction for use in wide area networks which include multiple domains. This pattern requires unified management of FWDs and SFs, in an SC domain, for setting end-to-end paths. Therefore, the management system of SPs, for example, a CE, for wide-area networks that include several segments might be massive and complex. Figure 12 shows the case in which SPs are established across multiple datacenters in pattern 1. In this case, a CE (or a set of CEs) manages multiple datacenters as a single SC domain for establishing SPs across the datacenters.

```
                            +--------------+
        . . . . . . . . . . |Control Entity| . . . . .
        .         .         +--------------+         .
        .         .                                  .
   * *  . * * * * . * * * * * * * * * * * * * * * *  . * * * * * * * * *
   *    .         .                                  .                 *
   *    .         .                                  .                 *
   *    .      .-----.        .-----------.       .-----.              *
   * +----+   /  DC#1 \      /     WAN     \     /  DC#2 \             *
   * |    |=====================================================> SP#1 *
   * | CF |=====================================================> SP#2 *
   *   :                            :                              :   *
   * |    |=====================================================> SP#n *
   * +----+   \       /      \             /     \       /             *
   *           '-----'        '-----------'       '-----'              *
   *                                                                   *
   *                           SC Domain                               *
   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 12: Establishment of SPs across Multiples DCs in Pattern 1

4.2.2. Analysis of Pattern 2: Dynamic SP Selection

In this pattern, SPs are established with a combination of segmented paths, so it enables SPs to be established flexibly (which means, CEs do not need to constantly manage the entire end-to-end SP) based on additional information such as the SF load conditions.

Furthermore, as described in the previous section, in cases where some SPs traverse multiple datacenters across a WAN, SPs could be established with a combination of segmented paths that each datacenter determines independently based on the Service Chain information. Therefore, it might be possible to separate SC domains

into several small areas for WANs, which would enable a simpler configuration of each CE. Figure 13 shows the case in which SPs are established across multiple datacenters in pattern 2. In Figure 13, each CE manages a single datacenter independently, and the CEs synchronize the Service Chain information for establishing and determining the appropriate segmented SPs in each domain.

However, the (fault) monitoring of the whole SC can become more difficult, as multiple domains are part of the SC. On the other hand, each domain can perform its management as required (and this is probably better as it is more specific). This will require an overarching (fault) monitoring where information from multiple SC domains is collected and aggregated to get a full view of the end-to-end service of the SC.

Moreover, in this pattern, some FWDs may require additional mechanisms to select the next segmented path, and the FWDs must maintain the states of each flow because some SFs require a stateful process, and the FWDs need to insert packets into the same SF instances in the same session.

In case that SC information is conveyed to some components via data plane as any encapsulation, a new protocol such as SFC [RFC7665] will be required.

```
                          Synchronization of
                           Service Chain info.
                 +--------------------------------------+
                 |                                      |
                 v                                      v
            +--------+                              +--------+
            |  CE#1  |                              |  CE#2  |
            +--------+                              +--------+
                 .                                      .
    * * * * * *  .  * * * * * *            * * * * * *  .  * * * * * *
    *            .            *            *            .            *
    *     .-------------.     *            *     .------------.      *
    *    /     DC#1      \    *  .------.  *    /     DC#2     \     *
    * +----+          +-----+ * /  WAN   \ * +-----+            |    *
    * |    |=========>|     | * |        | * | CF/ |==========> SP#1 *
    * | CF |=========>| FWD |===============>| FWD |==========> SP#2 *
    *   :       :        :    * |        | *    :         :       :  *
    * |    |=========>|     | * \        / * |     |==========> SP#n *
    * +----+          +-----+ *  '------'  * +-----+            |    *
    *    \               /    *            *     \             /     *
    *     '-------------'     *            *      '-----------'      *
    *       SC Domain#1       *            *      SC Domain#2        *
    * * * * * * * * * * * * * *            * * * * * * * * * * * * * *
```

Figure 13: Establishment of SPs Across Multiples DCs in pattern 2

Also, the detailed analysis of the establishment of "Hierarchical Service Path domains" is shown in the following section.

4.2.2.1. **Analysis of Hierarchical Service Path domains**

The dynamic selection of SPs pattern allows multiple independent domains of administration. (In the example, two levels were shown, but the pattern could be extended to multiple levels.)

This pattern allows even the largest networks to implement SC from the edges of the network by using coarse-grained classification. Classification choices can be made that are feasible within the constraints of the edge classifiers and FWDs. There is no need to maintain flow state or react to traffic at the top level.

This pattern allows control of sub-domains to be delegated to different owners. Each domain is simpler to comprehend than would be the case by dealing with a single flat network. Furthermore, failures and errors are localized (See Figure 14.).

```
   +----------+
   |Top-level | . . . . . . . . . . . . . . . . . . . . .
   |Control   |                                         .
   |Entity    |          +------------+     +--------+  .
   +----------+          |sub-domain#1|. . .|  CE#1  |  .
        .                +-----+------+     +--------+  .
        .                      |                        .
        .               .------+---------.   +---+      .
        .      +---+   /                  \--|CF |. . . .
        . . . .|CF |--/                    \ |FWD|      .
        .      |FWD| /                      \+---+      .
        .      +---+ |                       |          .
        .            |                       |          .
        .            |                       |          .
        .      +---+ \                      /           .
        .      |CF |  \                    /  +---+     .
        . . . .|FWD|---\                  /---|CF | . . .
               +---+    '------+---------'    |FWD|
                               |              +---+
          +--------+     +------------+
          |  CE#2  |. . .|sub-domain#2|
          +--------+     +-----+------+
```

Figure 14: Multiple Control Entities in Hierarchical Service Chaining

This hierarchical model supports the management of large networks by adhering to these principles:

1. At higher levels of hierarchy, packet classification is coarse, to minimize state and control-plane chatter.

2. At lower levels of hierarchy, packet classification can be more granular because classifiers in the lower levels deal with a subset of the entire network: fewer flows, lower bit-rate and a subset of network policy.

However, in this model, a new component that can proxy between the different domains, termed "Internal Boundary Node (IBN)," will be required. It has some commonality with the legacy SF proxy discussed in [I-D.song-sfc-legacy-sf-mapping].

This model also requires some coordination of path information within the IBN, since the IBN must map packets back and forth between domains. Solving this probably requires sharing metadata dictionaries among controllers and inventing a scheme that provides a level of indirection by naming path identifiers and metadata values.

4.3. Example of selecting Methods and Patterns

In this section, clarifications about the most suitable method and pattern are made for the following example networks based on the results of the above analysis.

4.3.1. Example#1: Enterprise Datacenter Network

The conditions of the target network are as follows:

Network type: Network with a single DC.

Intended service: For providing several network service to traffic of one or several business offices.

Variation of service: A group of adopting network service varies per office.

The number of SFs included in a service chain: Less than 5 (ref. section 3.2.1. Sample north-south service function chains in [I-D.ietf-sfc-dc-use-cases]).

Features of SFs: SFs are set statically, and SFs are exclusively used for each service.

On the basis of the conditions "network type" and "features of SFs", pattern 1 with SF dedicated model would be selected.

As the condition "variation of service" describes, such network requires few flow entries for each FWD, so method 1 would be applicable. Method 1 also does not require SFs to have any additional mechanism to apply any header, thus the impact of implementing this method would be less than other methods.

4.3.2. Example#2: Current Mobile Service Providers Network

The conditions of the target network are as follows:

Network type: Network with a single DC (e.g., (S)Gi-LAN (3GPP, [TS.23.203])).

Intended service: For providing network access service and several network service to traffic of millions customers.

Variation of service: Service varies per user or applications.

The number of SFs included in a service chain: Around 5(ref. examples of service in [I-D.ietf-sfc-use-case-mobility].).

Features of SFs: Many SFs are hardware equipment and they are deployed statically. Also, many SFs are used for several service. A function to inspect user traffic in detail, such as TDF (3GPP, [TS.23.203]), is located at the ingress of the network, and it might behave as a CF.

On the basis of the conditions "network type" and "features of SFs," pattern 1 with SF shared model would be selected. In such network, classification based on deep packet inspection such as application type inspections is done, and paths branching will not be happen.

As the other conditions describe, the operator must handle millions of flows and the flows traverse multiple SFs, so method 3 would be applicable. Configuring such amounts of flows among large scale network might be too much work for operators.

The examples of concrete service of such network are described as follows:

1. HTTP Modification

   Packet Gateway(P-GW), which is defined in 3GPP (ref. [tS.23.203]), detects traffic to the specific website and that traffic must be sent through a special element to insert additional data to the HTTP header or advertisement to the HTTP traffic, so the destination site can apply specific deals with the operator's customer (simplify DRM, premium service, etc.) That would require flow entries with mobile source IP, destination IP and port.

2. VoLTE Calls

   VoLTE calls are sent via a special SP. The VoLTE control plane selects all application network elements. But to reach application network elements it fully relies on standard routing and switching mechanisms. With Service Chaining it is possible to select the SP which can provide required QoS. That would require to set flow entries with mobile source IP, destination IP and port.

3. Secure Internet Access

   Some customers' HTTP traffic is forwarded to one or more security functions to inspect for malware. This case would require flow entries with source IP, destination IP and port.

4. Content Optimizer

Based on the policy rules, a SC/SP with the Content Optimization might be provided. Content optimization primarily affects video and HTTP traffic, and saves valuable radio resources in the specific radio cells during times of congestion. A controller might monitor Key Performance Indicators (KPIs) of the radio network to detect congestion. When congestion is detected, the controller might enforce a content optimization policy for the users on the congested radio cell. Most resource-expensive traffic can be transcoded by a content optimizer to save bandwidth. Selecting traffic for optimization would require to set flow entries with mobile source IP, destination IP and port. Also, content optimization might require changing SCs/SPs assigned to users flows based on the result of KPI monitoring or the time of day.

On the other hand, method 1 might be also selected with pattern 1 with SF dedicated model. For example, the series of the above service might be achieved by static configured flow entries, for example, with incoming port. However, it will require many incoming ports for FWDs when the operator would like to share a SF with multiple SCs, and it will not be scalable.

4.3.3. Example#3: Fixed and Mobile Converged Service Providers Network

The conditions of the target network are as follows:

Network type: Network with multiple DCs (e.g., SFs are deployed at multiple DCs based on their applications).

Intended service: For providing network access service or several network service to traffic of millions customers.

Variation of service: Service varies per user. Also, the service assigned to each flow might vary based on using applications.

The number of SFs included in a service chain: More than 5. (Various services such as enriched security service and value added services would be provided)

Features of SFs: Many SFs are deployed as VNFs (Virtualized Network Functions), and some SFs are shared with multiple SCs. Also, some SFs changes the following SPs dynamically based on the result of the process.

On the basis of the conditions "network type" and "features of SFs," pattern 2 would be selected. Pattern 2 allows hierarchical approach which enables operators to deploy SFs in multiple domains easily based on service requirements. For example, operators can deploy SFs

into several domains based on application types. This concept is introduced in [I-D.ietf-sfc-dc-use-cases].

From the above conditions describe, the operator must handle enormous flows and paths branching, thus method 3 will be appreciable for such network. Especially, security scenario sometimes requires paths branching based on the result of packet inspection such as processes of DPI or traffic analyzer. Some security functions such as web application firewall (WAF) are specialized for each application, and it might be inefficient to insert all traffic into such SFs. Therefore, for inserting only target packets to appropriate security functions, classifying and paths branching based on packet inspection would be required.

5. Acknowledgements

The authors would like to thank Konomi Mochizuki and Lily Guo for their reviews and comments.

6. Contributors

The following people are active contributors to this document and have provided review, content and concepts (listed alphabetically by surname):

Poul Bottorff
Hewlett Packard Networking

Mohamed Boucadair
France Telecom

Nicolas Bouthors
Qosmos

Hiroshi Dempo
NEC

Christian Jacquenet
France Telecom

Ron Parker
Affirmed Networks

Chuong D. Pham
Telstra

Paul Quinn
Cisco Systems

7. IANA Considerations

This memo includes no request to IANA.

8. References

[I-D.dolson-sfc-hierarchical]
Dolson, D., Homma, S., Lopez, D., Boucadair, M., and D. Liu, "Hierarchical Service Function Chaining", draft-dolson-sfc-hierarchical-04 (work in progress), December 2015.

[I-D.fedyk-sfc-mac-chain]
Bottorff, P., don.fedyk@hpe.com, d., and H. Assarpour, "Ethernet MAC Chaining", draft-fedyk-sfc-mac-chain-01 (work in progress), January 2016.

[I-D.ietf-sfc-dc-use-cases]
Surendra, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-04 (work in progress), January 2016.

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-02 (work in progress), January 2016.

[I-D.ietf-sfc-use-case-mobility]
Haeffner, W., Napper, J., Stiemerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", draft-ietf-sfc-use-case-mobility-05 (work in progress), October 2015.

[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-06 (work in progress), August 2015.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <http://www.rfc-editor.org/info/rfc3022>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <http://www.rfc-editor.org/info/rfc6146>.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <http://www.rfc-editor.org/info/rfc6296>.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <http://www.rfc-editor.org/info/rfc7498>.

[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

Shunsuke Homma
NTT, Corp.
3-9-11, Midori-cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3486
Email: homma.shunsuke@lab.ntt.co.jp

Kengo Naito
NTT, Corp.
3-9-11, Midori-cho
Musashino-shi, Tokyo 180-8585
Japan

Email: k.naito@nttv6.jp

Diego R. Lopez
Telefonica I+D.
Don Ramon de la Cruz, Street
Madrid 28006
Spain

Phone: +34 913 129 041
Email: diego.r.lopez@telefonica.com

Martin Stiemerling
NEC Laboratories Europe / Hochschule Darmstadt
Kurfuerstenanlage 36
Heidelberg 69115
Germany

URI: ietf.stiemerling.org

David Dolson
Sandvine
408 Albert Street
Waterloo, Ontario N2L 3V3
Canada

Email: ddolson@sandvine.com

Alexey Gorbunov
Nokia
6000 Connection Drive
Irving, Texas 75039
USA

Phone: +1 214 516 11 41
Email: Alexey.gorbunov82@gmail.com

Nicolai Leymann
DT
Winterfeldtstrasse 21-27
Berlin 10781
Germany

Phone: +49 (0)30 835392761
Email: n.leymann@telekom.de

Paul Bottorff
Hewlett Packard Enterprise
8000 Foothills Blvd.
Roseville, CA
USA

Email: paul.bottorff@hpe.com

Don Fedyk
Hewlett Packard Enterprise
153 Taylor Street
Littleton, MA
USA

Email: don.fedyk@hpe.com