Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

S. Homma
K. Naito
NTT
D. R. Lopez
Telefonica I+D
October 27, 2014

Analysis on Forwarding Methods for Service Chaining
draft-homma-sfc-forwarding-methods-analysis-00

Abstract

Some working groups of the IETF and other Standards Developing Organizations are now discussing use cases of a technology that enables data packets to traverse appropriate service functions through networks. This is called Service Chaining in this document. (Also, in Network Functions Virtualisation (NFV), a subject that forwarding packets to required service functions in appropriate order is called VNF Forwarding Graph.) This draft does not focus only on SFC method, and thus, use the term "Service Chaining". SFC may be one method to realize Service Chaining. There are several Service Chaining methods to forward data packets to service functions, and the applicable methods will vary depending on the service/network requirements of individual networks.

This document presents the results of analyzing packet forwarding methods and path decision patterns for achieving Service Chaining. For forwarding data packets to the appropriate service functions, distribution of route infromation and steering data packets following the route infromation, are required. Examples of route information are packet identifier and the routing configurations based on the identifier. Also, forwarding functions are required to decide the path according to the route information.

Table of Contents

1. Introduction

Service Chaining is a technology that enables data packets to traverse the appropriate service functions deployed in a network. This draft assumes that Service Chaining is achieved in the following steps:

a. A classification function identifies data packets and determines the set of services that will be provided for the packets and in which order.

b. The path, that the packets will traverse for reaching the required service functions, is established based on the result of step a.

c. Forwarding functions determine the appropriate destination and forward each packet to the next hop according to the path.

d. A service function provides services to received packets and return each packet to the 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 and the patterns used to decide the paths will affect the mechanism of Service Chaining as well as 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 analyzing 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 draft-merged-sfc-architecture-01. Term "Service Function", "Service Node" referred to draft-ietf-sfc-dc-use-cases-01.

Service Chaining: A technology that lets data packets traverse a series of service functions.

Classification: Locally instantiated policy and customer/network/service profile matching of traffic flows for identification of appropriate outbound forwarding actions.

Classifier (CF): The entity 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.

Service Node (SN): A virtual or physical device that hosts one or more service functions, which can be accessed via the network location associated with it.

Forwarder (FWD): The entity, responsible for forwarding data packets along the service path, which includes delivery of traffic to the connected service functions. FWD handles Forwarding Tables, which is used for forwarding packets.

Control Entity (CE): The entity 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 user 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 instantiation of a service chain in the network. Packets follow a service function path through the requisite service functions. SP shows a specific path of taraversing SF instance. 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.

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

Service Path Information (SPI): The information used to forward packets to The appropriate SFs based on the selected service. Examples of SPI include routing configurations for Forwarders, transport headers for forwarding packets to required SFs, and service/flow identifiable tags.

3. Classification of Forwarding Methods and SP Decision Patterns

3.1. Forwarding Methods

In Service Chaining, data packets are transferred to service functions, which can be located outside the regular computed path to the original destination. Therefore, a routing mechanism that is different from general L2/L3 switching/routing may be required. The routing mechanism can be classified into three methods in terms of distribution of SPI 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, routing configurations 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 may be an CE to handle this. The flow identifiable information can be constructed with some fields of L2 or L3 or combination of those. 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 identifable information and forwards the packets to the SFs according to the configuration. This method does not require changing any fields of the original packet frame.

```
*Distribution model of SPI*

        +----------------+
        | 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   |
         +--------+        +--------+        +--------+      +--------+

         Fig.1 Forwarding Based on Flow Identifiable Information
```

3.1.2. Method 2: Forwarding with Stacked Transport Headers

The mechanism of method 2 is shown in Figure 2. In this method, the CF classifies packets and stacks transport headers, e.g., MPLS or GRE headers, onto the packets based on the classification. The configuration about how FWDs handle the headers is pre-configured. Each FWD forwards the packets to SFs following the outermost header. The outermost header is removed after each forwarding or service process. The actions are repeated until all headers are removed.

```
*Distribution model of SPI*

           +----------------+
           | 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 FWD1              * the outermost transport 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    |
           +---------+      +---------+     +---------+     +---------+
```

Fig.2 Forwarding with Stacked Multiple Transport Headers

3.1.3. Method 3: Forwarding Based on Service Chain Identifiable Tags

This method is shown in Figure 3. In this method, a CF classifies each packet and attaches a tag for identifying the service or flows on the packets based on the classification. The routing

configuration based on the tags is sent to each FWD (from some CE) in advance. Each FWD fowards packets to the SFs following the configuration and the tag. After a packet has traversed all SFs, the tag is removed.

```
*Distribution model of SPI*

    +----------------+
    | 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 FWD1
        ->Stack ID#2
       ...                ...                         ...

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

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

       +--------+        +--------+     +--------+       +--------+
Tag:   | ID#1   |        | ID#1   |     | ID#1   |       | ID#1   |
       +--------+        +--------+     +--------+       +--------+
Packet:|  PDU   |        |  PDU   |     |  PDU   |       |  PDU   |
       +--------+        +--------+     +--------+       +--------+

    Fig.3  Forwarding Based on Service Chain Identifiable Tags
```

3.2. Service Path Decision Patterns

Since Service Chain contains only logical information (e.g. series of services that are applied to flows and their sequences), the actual instances, which are called Service Paths, are needed in order for the forwarding process to work. In this process, an instance of Service Path is created at certain points during a packet's delivery. Therefore, to forward packets, the Service Chain needs to be turned into an SP, which indicates specific FWDs (or switches, routers) and SFs that the packets will be forwarded to. In the Service Chain to SP change points, the paths that determine the Service Chaining are classified into two patterns.

3.2.1. Pattern 1: End to End Static Service Path

The translation point is only a CF; that is, the SP is statically pre-established as an end-to-end path. A CF inserts packets into the appropriate pre-established path based on their classification. Each FWD on the route has a routing table to uniquely determine the next destination of packets, and each FWD statically forwards the received packets to the next destination. FWDs require only a function to receive indications of routing configurations from the CE. Pattern 1 can be achieved in the following ways.

3.2.1.1. SF Shared Model

Figure 4 shows the mechanism of this way. An SF is shared by multiple SPs. In this way, the FWDs require a function to identify packets and insert the packets into the next appropriate hop.

```
*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
////////////////////////////////////////////////////////////////////////////
*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 ]-->

                         Fig.4 SF Shared Model
```

3.2.1.2. SF Dedicated Model

Figure 5 shows the mechanism of this style. An SF (instance) is used by only one single SP; in other words, there is an SF instance per SP. At each FWD, incoming packets are statically routed to a single predefined next hop.

```
*Path Structure*

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

                                                            SC:Service Chain
/////////////////////////////////////////////////////////////////////////////
*How packets traverse*

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

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

Fig.5 SF Dedicated Model

3.2.2. Pattern 2: Dynamically Determined Service Path

The mechanism of this style is shown in Figure 6. The translation points are a CF and FWDs. The SP is established by a series of multiple paths, which are sectioned by CFs and FWDs. Each path determined by CFs and FWDs is referred to as a segmented path in this draft. CFs or FWDs that determine the next segmented path may require notification of routing configurations 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 that information. The CE may have the selection mechanism and may notify CSs/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
////////////////////////////////////////////////////////////////////////////

*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 ]-->
```

Fig.6 Dynamically Determined Service Path

4. Consideration of Service Chaining Methods and Architecture Patterns

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

4.1. **Analysis of 3.1.** Forwarding Methods

4.1.1. **Analysis of Method 1**

This method can achieve Service Chaining without adding any headers to packets, so it may not cause any increase in packet size or be subject to MTU restrictions. Furthermore, this method does not require additional functions within SFs to be applied to any headers because data packets are transported in original format. Therefore, it will be easier to use legacy SFs for network operators.

However, forwarding entries or static configuration for a flow at each FWD is required. For example, if there are 10,000 flows to be handled at a CF/FWD, the routing 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 Service Chain per user (which means that individual users have their own policies), because some large carriers have over a million users and even more flows. Another concern is the traffic increase in the control plane because route setting is required for each flow. Moreover, it may be hard to use this method if some service functions 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 routing tables. The results of the above analysis suggest that this method may be suitable for networks with a limited number of flows.

4.1.2. **Analysis of Method 2**

In this method, none of the FWDs require any specific routing tables for Service Chaining, but they require a function to forward packets based on header information, and to remove the outermost header from the received packets. Therefore, the control plane would be simple because the SC controller would not be required to manage the routing configuration of FWDs. Also, there are already several technologies proposed that can be used to achieve this method, such as MPLS.

However, the more the SFs packets traverse, the more headers have to be added to the packet and this in turn means that the packet size increases. But packet sizes are restricted by the minimal available MTU of any link in the network path and exceeding the MTU will require to fragement the original packet before starting to add more headers required to the service chaining. This requires more complexity in processing due to the fragementation, adds a new source of errors, as fragments of packets can get lost and so the whole original packet will get discared, and also will cause an increase in traffic as more packets have to be processed by the network. Moreover, from a hardware point of view, it might be challenging for

FWDs or SFs to process packets with variable length headers. In terms of SF equipments, if fragmented packets need to be reassembled at every SF, this would be very wasteful of CPU resources, and some equipment has restricted resources and memory for reassembly. The results of the above analysis indicate that this method would be appropriate when the number of SFs in an SC is small, or most packets are forwarded to a static SP. On the other hand, it may be unsuitable in cases where there are many SFs in a chain.

4.1.3. Analysis of Method 3

In this method, a tag is defined for each Service Chain. By adopting single fixed-length tags, this method can prevent an increase in the amount of traffic in the data plane, and can provide an upper bound on packet size. (Problems which happen as a result of exceeding MTU are stated in 4.1.2.) This method also enables FWDs to save resources when handling flow entries. Therefore, this method has many advantages in terms of scalability, and it might be appropriate for use in large-scale networks.

However, this method might require renewal of equipments, or Operating Systems (OSes) installed in hardware, or softwares, or any other components to realize the method in network which includes SFs, if this tag handling is an entirely new mechanism. Furthermore discussion might be required to deploy such standardized technologies.

4.2. Analysis of 3.2. Determination of Service Paths

4.2.1. Analysis of Pattern 1

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 determination of the next hop. Moreover, it is not necessary to maintain the state of each flow.

However, this pattern will impact the flexibility of the SCs, as adding new SFs to a SC, removing SFs from a SC, or migrating SFs to other locations requires an update or new creation of a path in the Service Path. Furthermore, unified management of FWDs and SFs in an SC domain would be required in setting end-to-end paths. Therefore, the management system of SPs, for example, a CE, for wide-area networks that include several segments may be massive and complex. Figure 7 shows the case in which SPs are established across multiple datacenters in pattern 1. In Figure 7, a CE manages multiple datacenters as a single SC domain for establishing SPs across multiple datacenters.

In pattern 4.2.1.2 (SF Dedicated Model), the number of flow entries that FWDs hold can be extremely small, as FWDs hold only static next-hop information. Also, the CF function would be simple, as the CF only determines the gateway of each SP. However, because the SF (instance) is settled for each SP, resource usage would be high if there were many SPs.

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

Fig.7 Establishment of SPs Accross Multiples DCs in Pattern 1

4.2.2. Analysis of Pattern 2

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 load condition of SFs.

Furthermore, 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. Figures 8 shows the case in which SPs are established across multiple datacenters in pattern 2. In Figure 8, 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 get harder as multiple domains are part of the SC. On the other hand each domain can perfom its fault management as required (and 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.

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

Fig.8 Establishment of SPs Accross Multiples DCs in pattern 2

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 A: Datacenter Network

The conditions of network A are as follows:

1. The network is used for several business offices as a single DC.

2. Service Chain varies per office (not per user).

3. The number of SF included in for each Service Chain is few. (e.g. within 5.)

4. SF (instance) cost is not so high.

5. MTU should not be restricted.

6. Service Chains do not fork paths through end-to-end. (As monitoring, or controlling will be harder, some operators may not want to change paths after packets got into a service chain.)

On the basis of conditions 4 and 6, Pattern 1 (SF Dedicated Model) would be selected. In this case, any method would be applicable. (Even if method 2 is selected, only one header that shows the gateway to the specific SC is stacked on packets. This does not restrict the MTU.)

4.3.2. Example B: Current Mobile Carrier Network

The conditions of network B are as follows:

1. The network handles millions of users.

2. Service Chain (SF set and order) is predefined and limited.

3. The number of SF, included in for each Service Chain, is few. (e.g. within 5.)

4. The user chooses or the provider can choose for the user a predefined Service Chain to adopt to their traffic.

5. SFs are located in (S)Gi-LAN.(Term referred to draft-ietf-sfc-use-case-mobility-01)

6. Service Chains do not require to fork paths through end-to-end.

On the basis of conditions 1, 2, and 5, Pattern 1 (SF Shared Model) would be selected because the architecture would be simple.

On the basis of conditions 3 and 4, method 1 (unless the configuration or forwarding table does not increase explosively) or 3 would be applicable.

4.3.3. Example C: Fixed Mobile Convergence Network

Conditions of the network A is as follows:

1. The network handles millions of users.

2. The user chooses or the provider can choose for the user multiple SFs to adopt to their traffic.

3. Many SFs (e.g. 5 or more,) are included in for each Service Chain.

4. SFs are located in multiple DCs.(e.g. Some delay sensitive SFs, or SFs which should be placed near users' locations are installed in DCs located locally, and added-value SFs are installed in DCs located centrally.)

5. There are some expansive SFs (instance) that should be shared by several SPs.

6. Service Chains may be forked according to the process of SF.

On the basis of conditions 1, 2, 3, 4, and 5, Method 3 would be applicable in terms of scalability. Pattern 2 should be selected based on conditions 1 and 6. Although the operation would be complex, there may be a case in which some carriers set multiple DCs and separate SC domains according to their network or service policy. The use case and architecture pattern is introduced in draft-ietf-sfc-dc-use-cases-01.

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):

Hiroshi Dempo
NEC

David Dolson

Sandvine

Ron Parker
Affirmed Networks

Paul Quinn
Cisco Systems

Martin Stiemerling
NEC

7. IANA Considerations

This memo includes no request to IANA.

8. References

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-02 (work in progress), September 2014.

[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-01 (work in progress), July 2014.

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-10 (work in progress), August 2014.

[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-01 (work in progress), July 2014.

[I-D.quinn-sfc-nsh]
Quinn, P., Guichard, J., Fernando, R., Surendra, S., Smith, M., Yadav, N., Agarwal, P., Manur, R., Chauhan, A., Elzur, U., Garg, P., McConnell, B., and C. Wright, "Network Service Header", draft-quinn-sfc-nsh-03 (work in progress), July 2014.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

Authors' Addresses

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

Email: homma.shunsuke@lab.ntt.co.jp

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

Email: naito.kengo@lab.ntt.co.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