

Service Function Chaining  
Internet-Draft  
Intended status: Informational  
Expires: January 4, 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  
July 3, 2015

Analysis on Forwarding Methods for Service Chaining  
draft-homma-sfc-forwarding-methods-analysis-03

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 is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2016.

Internet-Draft [draft-homma-sfc-forwarding-methods-analysis](#)

July 2015

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Definition of Terms . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Classification of Forwarding Methods and SP Decision Patterns	<a href="#">5</a>
<a href="#">3.1.</a>	Forwarding Methods . . . . .	<a href="#">5</a>
<a href="#">3.1.1.</a>	Method 1: Forwarding Based on Flow Identifiable Information . . . . .	<a href="#">5</a>
<a href="#">3.1.2.</a>	Method 2: Forwarding with Stacked Headers . . . . .	<a href="#">7</a>
<a href="#">3.1.3.</a>	Method 3: Forwarding Based on Service Chain Identifiable Tags . . . . .	<a href="#">9</a>
<a href="#">3.2.</a>	Service Path Selection Patterns . . . . .	<a href="#">10</a>
<a href="#">3.2.1.</a>	Pattern 1: Static Selection of End-to-End Service Path . . . . .	<a href="#">11</a>
<a href="#">3.2.2.</a>	Pattern 2: Dynamic Selection of Segmented Service Path . . . . .	<a href="#">13</a>
<a href="#">4.</a>	Consideration on Forwarding Methods and Paths Selection Patterns . . . . .	<a href="#">18</a>
<a href="#">4.1.</a>	Analysis of Forwarding Methods . . . . .	<a href="#">19</a>
<a href="#">4.1.1.</a>	Analysis of Method 1 . . . . .	<a href="#">19</a>
<a href="#">4.1.2.</a>	Analysis of Method 2 . . . . .	<a href="#">20</a>
<a href="#">4.1.3.</a>	Analysis of Method 3 . . . . .	<a href="#">21</a>
<a href="#">4.2.</a>	Analysis of Service Path Selection Patterns . . . . .	<a href="#">22</a>
<a href="#">4.2.1.</a>	Analysis of Pattern 1 . . . . .	<a href="#">22</a>
<a href="#">4.2.2.</a>	Analysis of Pattern 2 . . . . .	<a href="#">23</a>
<a href="#">4.3.</a>	Example of selecting Methods and Patterns . . . . .	<a href="#">26</a>
<a href="#">4.3.1.</a>	Example#1: Enterprise Datacenter Network . . . . .	<a href="#">26</a>
<a href="#">4.3.2.</a>	Example#2: Current Mobile Service Providers Network . . . . .	<a href="#">27</a>

4.3.3. Example#3: Fixed and Mobile Converged Service Providers Network . . . . .	<a href="#">28</a>
<a href="#">5.</a> Acknowledgements . . . . .	<a href="#">29</a>
<a href="#">6.</a> Contributors . . . . .	<a href="#">29</a>
<a href="#">7.</a> IANA Considerations . . . . .	<a href="#">30</a>

<a href="#">8.</a> References . . . . .	<a href="#">30</a>
Authors' Addresses . . . . .	<a href="#">31</a>

[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 .

- 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 [[I-D.ietf-sfc-architecture](#)]. 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 assist 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 Decision Patterns

#### 3.1. Forwarding Methods

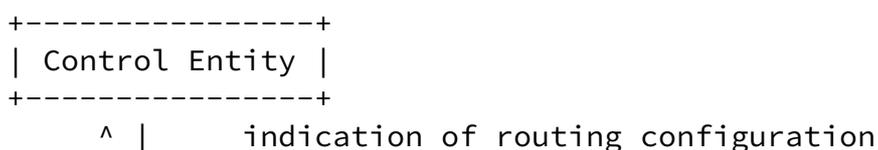
In Service Chaining, data packets are transferred to service functions, which might be located outside the regular computed path to the original destination. Therefore, a routing mechanism that is different from general L2/L3 switching/forwarding might be required. The forwarding mechanism can be classified 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

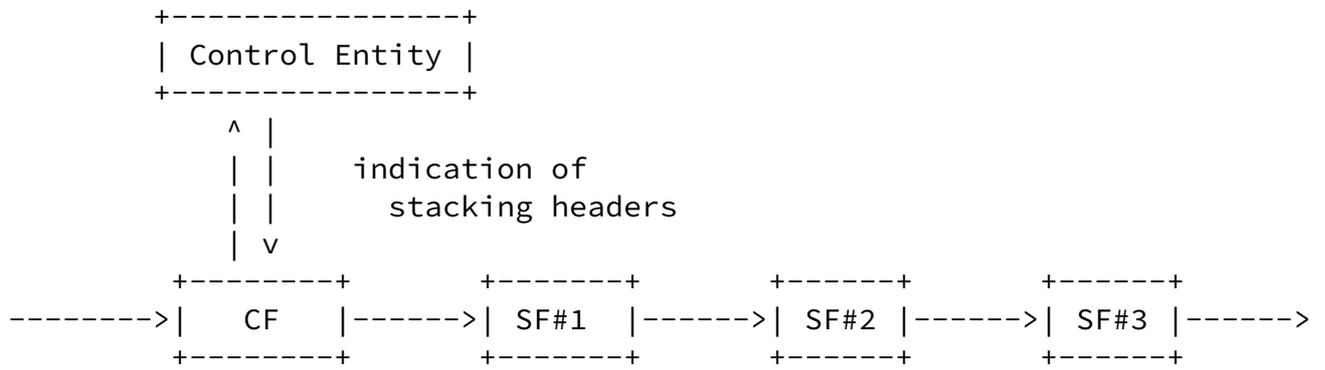
port, tcp) are indicated to the CF and each FWD. There may be an CE to handle this. The flow identifiable information can be constructed with some fields of L2 or L3 or combination thereof. The information can be configured either before packets arrive, or at the time packets arrive at CF and FWD. Each 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\*





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.



////////////////////////////////////  
\*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\*

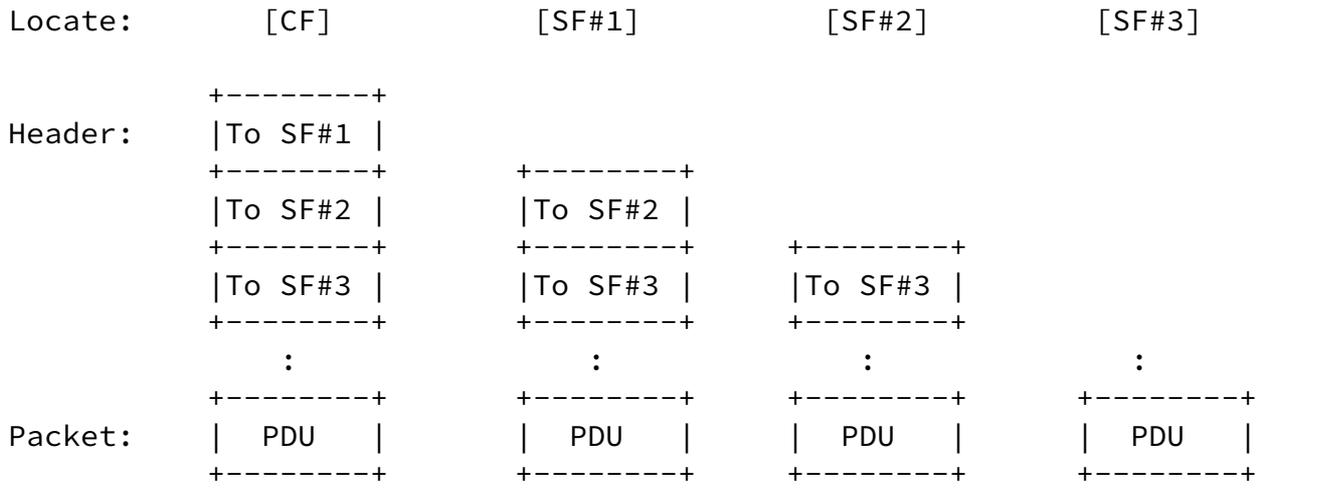


Figure 2: Forwarding with Stacked Multiple Headers

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

The mechanism of this method is shown in Figure 3. In this method, a CF classifies each packet and attaches a tag for identifying the service or flow to which the packets belong, based on the classification. The forwarding configuration based on the tags is sent to each FWD (from some CE) in advance. Each FWD forwards packets to the SFs following the configuration and the tag. After a packet has traversed all SFs, the tag is removed and the packet is transported to the original destination.



## [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 ways.

#### [3.2.1.1.](#) SF Shared Model

Figure 4 shows the mechanism of this model. In this model, an SF is shared by multiple SPs. Therefore, FWDs require a function to identify the SP followed by each packet and forward the packets to the corresponding next hop.



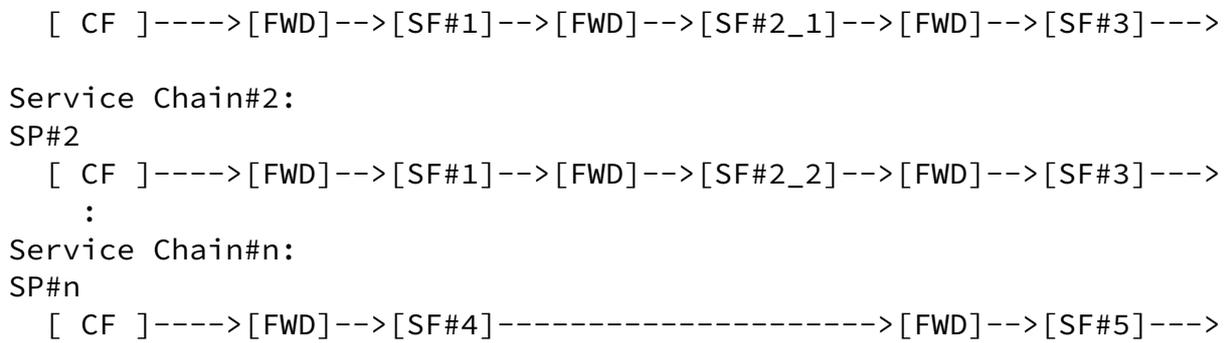
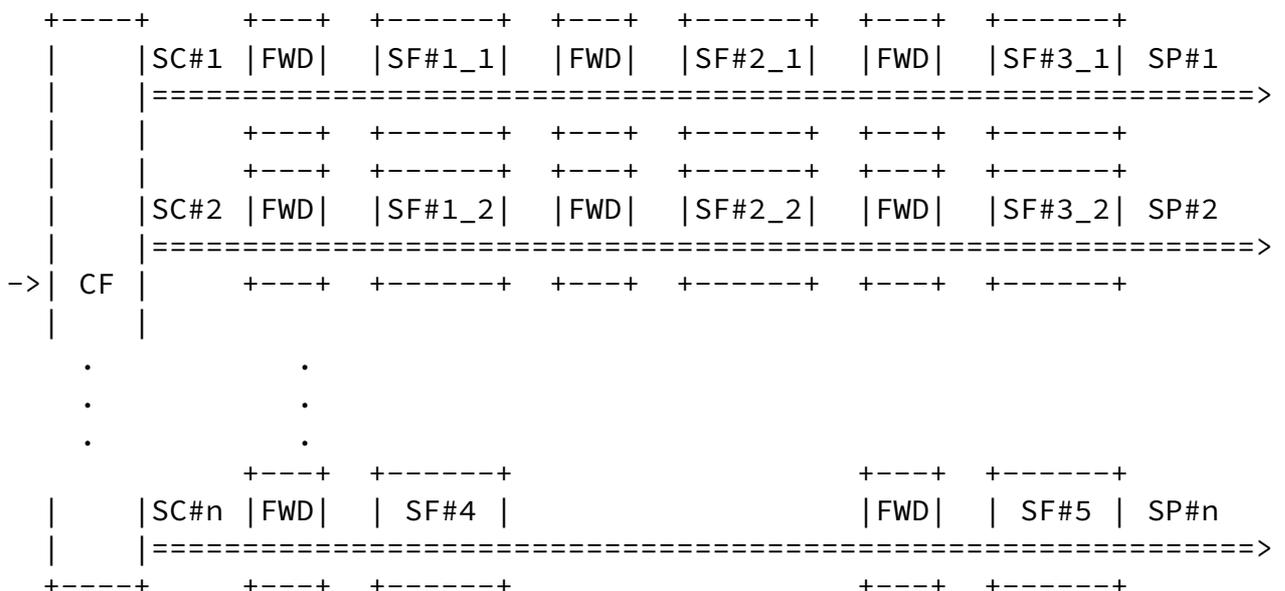


Figure 4: SF Shared Model

### 3.2.1.2. SF Dedicated Model

Figure 5 shows the mechanism of this 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. At each FWD, incoming packets are statically forwarded to the single pre-defined next hop.

\*Path Structure\*



```

                                                                    SC:Service Chain  SP:Service Path
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
*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 ]---->

```

Figure 5: SF Dedicated Model

### 3.2.2. Pattern 2: Dynamic Selection of Segmented Service Path

The mechanism of this pattern is shown in Figure 6. 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

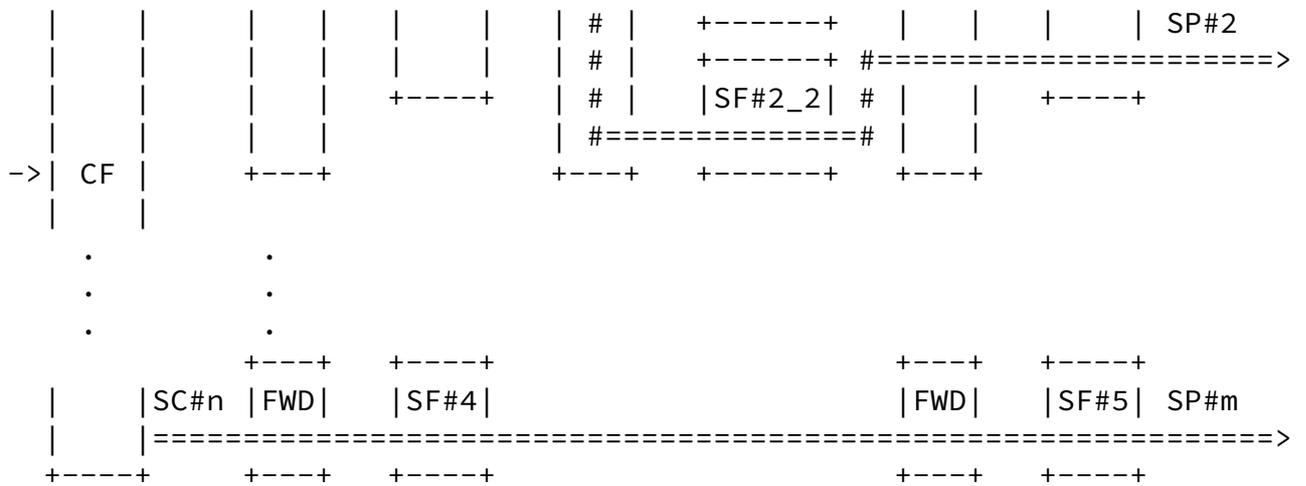
the next path. For example, each FWD obtains metric information or load conditions of servers and selects an optimal segmented path based on the information. The CE might support the selection mechanism and may notify CFs or FWDs of it.

\*Path Structure\*

```

+----+      +----+   +----+   +----+   +-----+   +----+   +----+
|      |SC#1 |FWD|   |SF#1|   |FWD|   |SF#2_1|   |FWD|   |SF#3| SP#1
|      |=====*<

```



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 6: 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 7 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\*

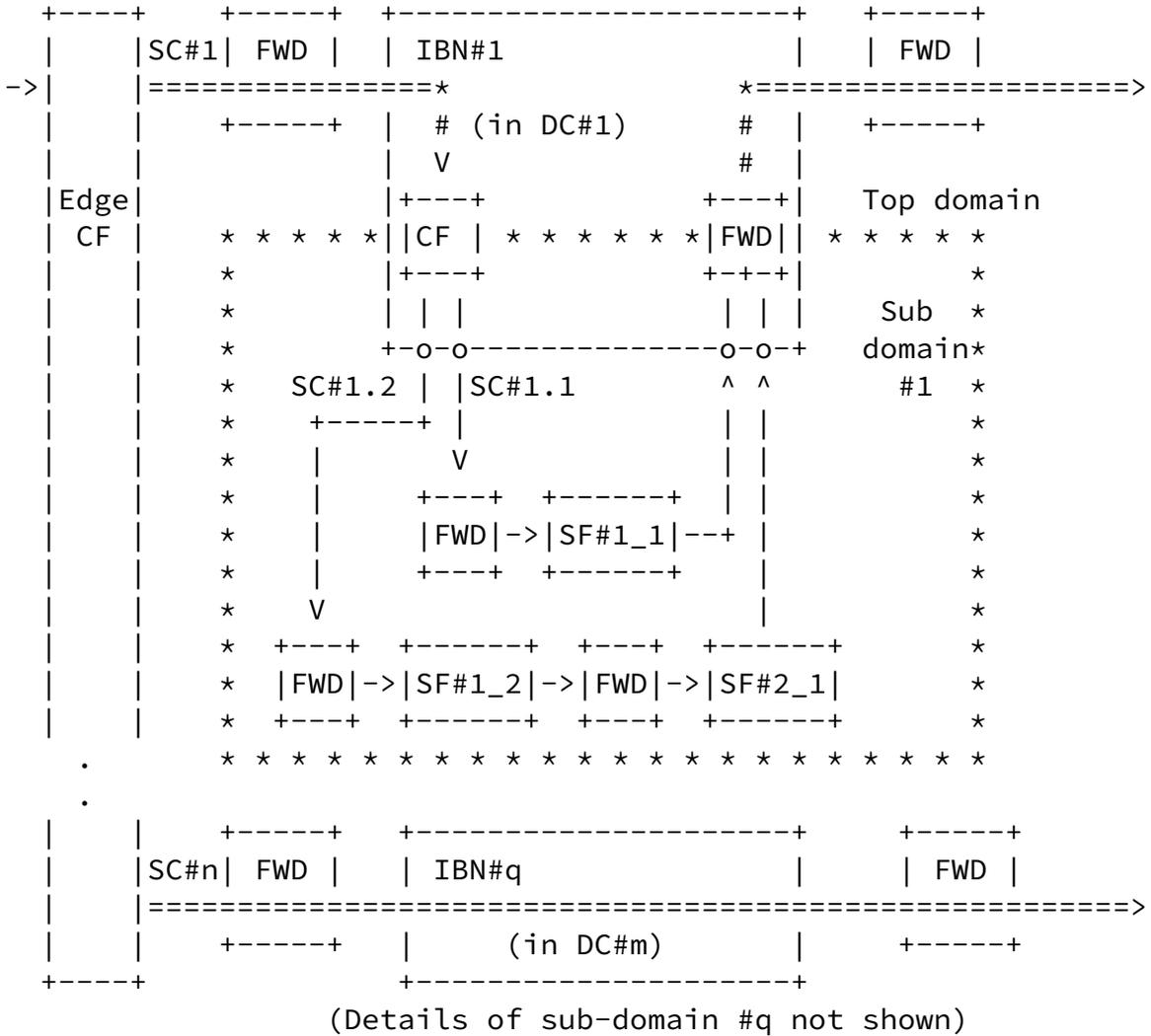


Figure 7: Service Chain Hierarchy in Service Provider Networks

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 8, 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.

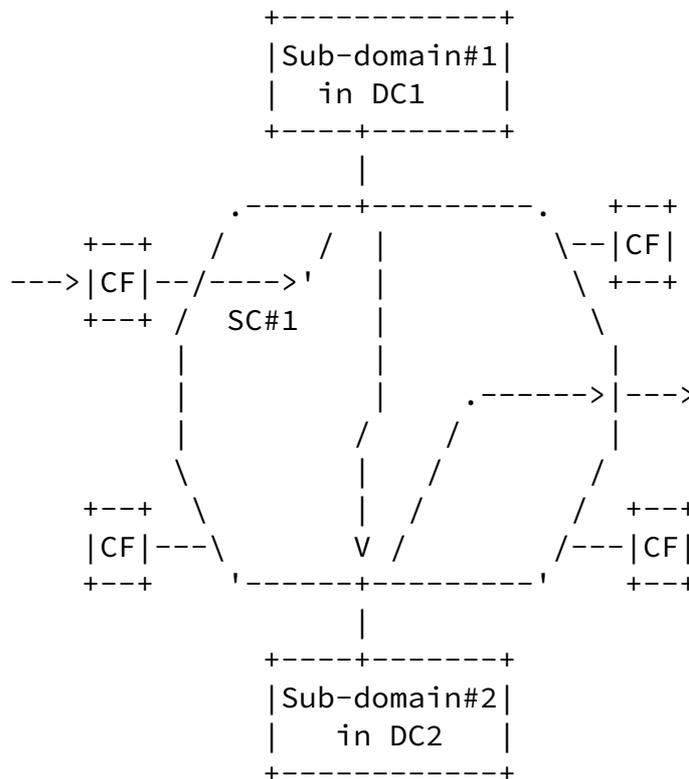


Figure 8: Network-wide view of Top Level of Hierarchy

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.

Homma, et al.

Expires January 4, 2016

[Page 18]

---

Internet-Draft [draft-homma-sfc-forwarding-methods-analysis](#)

July 2015

#### [4.1.](#) Analysis of Forwarding Methods

##### [4.1.1.](#) Analysis of Method 1

###### 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 requires FWDs to set forwarding entries for each flow. 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. Another concern is the increase of control signaling because route setting is required for each flow. 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.

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

##### 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 tags before inserting packets into SFs and re-attach the appropriate tag 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. Also, no CEs will be required to manage the forwarding configuration of FWDs, so the control plane might become simple.

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

## Data Plane Aspects

In this method, a tag is defined for each SC and attached on each packet. By adopting a single fixed-length tag, this method can prevent an increase of the amount of traffic, and can provide an upper bound on packet size (Problems which happen as a result of exceeding MTU are stated in [Section 4.1.2](#)). Also, FWDs recognize the next hops of received packets from the tags 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 tag.

On the other hand, this method requires SFs to be applied to the tags because SFs receive packets with the tags. (Problems which happens as result of inserting packet with optional tags into SFs are stated in [Section 4.1.2](#)) By using existing headers as tags or outer header for forwarding, effect on network nodes such as existing router and switches might be limited.

## Control Plane Aspects

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

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

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 many SFs and flows.

By the way, if the tag handling mechanism is an entirely new architecture such as SFC[I-D.ietf-sfc-architecture], 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

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 protocols for virtualizing networks, such as VxLAN or MPLS, can be used to achieve Service Chaining in this pattern.

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 the creation of a new 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 9 shows the case in which SPs are established across multiple datacenters in pattern 1. In Figure 9, 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 route 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.



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 [[I-D.ietf-sfc-architecture](#)] will be required.

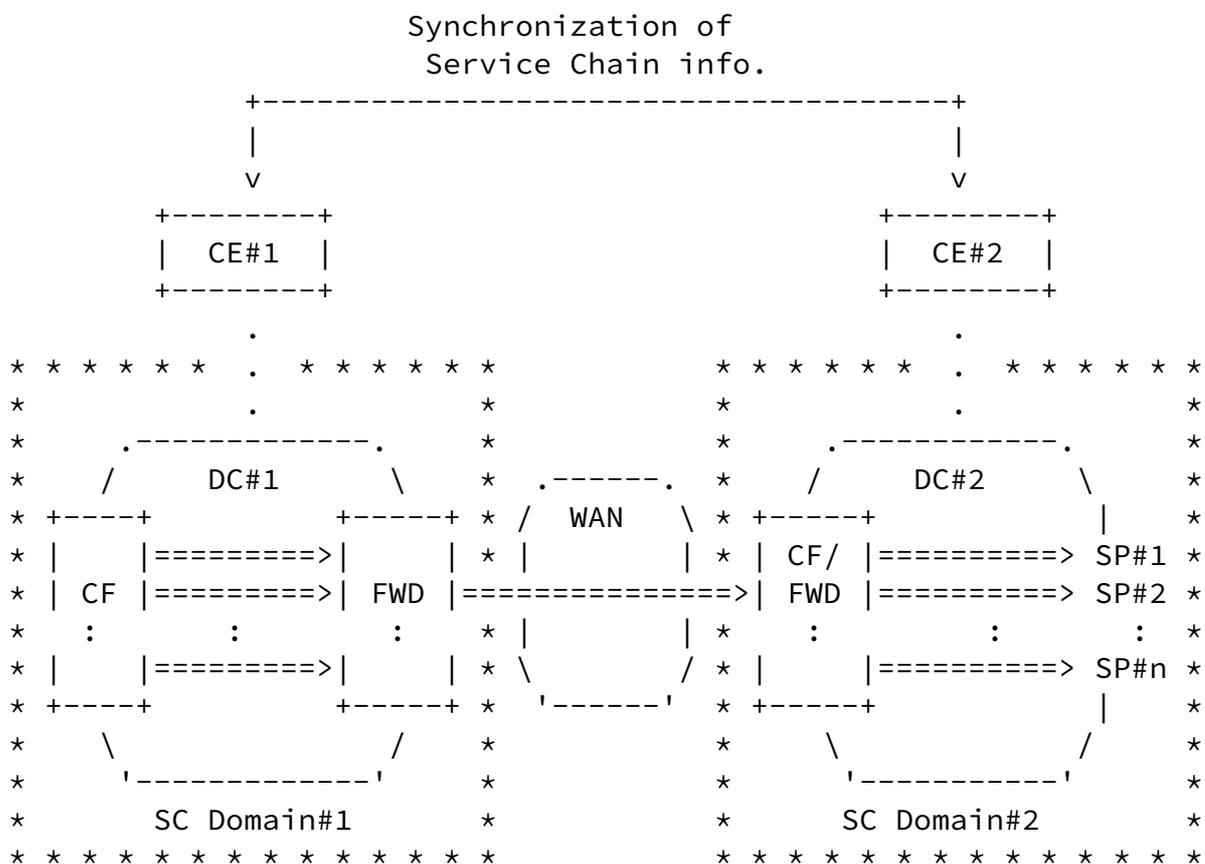


Figure 10: Establishment of SPs Across Multiples DCs in pattern 2

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



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):

Mohamed Boucadair  
France Telecom

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., and D. Lopez, "Hierarchical Service Chaining", [draft-dolson-sfc-hierarchical-01](#) (work in progress), June 2015.

[I-D.ietf-sfc-architecture]

Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", [draft-ietf-sfc-architecture-09](#) (work in progress), June 2015.

[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-02](#) (work in progress), January 2015.

[I-D.ietf-sfc-nsh]

Quinn, P. and U. Elzur, "Network Service Header", [draft-ietf-sfc-nsh-00](#) (work in progress), March 2015.

[I-D.ietf-sfc-use-case-mobility]

Haeffner, W., Napper, J., Stiernerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", [draft-ietf-sfc-use-case-mobility-03](#) (work in progress), January 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-04](#) (work in progress), December 2014.

---

Internet-Draft [draft-homma-sfc-forwarding-methods-analysis](#)

July 2015

- [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.
- [RFC7498] Quinn, P. and T. Nadeau, "Problem Statement for Service Function Chaining", [RFC 7498](#), April 2015.

#### 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](mailto: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](mailto: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](mailto:diego.r.lopez@telefonica.com)

Internet-Draft [draft-homma-sfc-forwarding-methods-analysis](#)

July 2015

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

URI: [ietf.stiemerling.org](http://ietf.stiemerling.org)

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

Email: [ddolson@sandvine.com](mailto:ddolson@sandvine.com)

Alexey Gorbunov  
Nokia  
6000 Connection Drive  
Irving, Texas 75039  
USA

Phone: +1 214 516 11 41  
Email: [Alexey.gorbunov82@gmail.com](mailto:Alexey.gorbunov82@gmail.com)

Nicolai Leymann  
DT  
Winterfeldtstrasse 21-27  
Berlin 10781  
Germany

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

Homma, et al.

Expires January 4, 2016

[Page 32]