Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: December 10, 2015

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
June 8, 2015

Analysis on Forwarding Methods for Service Chaining
draft-homma-sfc-forwarding-methods-analysis-02

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 located remotely 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 of approaches 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 requirements of individual networks.

This document presents the results of analyzing packet forwarding methods and path selection patterns for achieving Service Chaining. For forwarding data packets to the appropriate service functions, distribution of route information and steering data packets following the route information, 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.

Status of This Memo

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 December 10, 2015.

Copyright Notice

Table of Contents

1. Introduction

Service Chaining is a technology to provide service oriented forwarding which enables data packets to traverse the appropriate service functions deployed in networks. This draft assumes that Service Chaining is achieved by 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. The paths may be established in advance.

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

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 path through the requisite service functions. Service path shows a specific path of traversing 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 (SP Information): The information used to forward packets to the appropriate SFs based on the selected

service. Examples of SP information 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 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, 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 identifiable 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 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 Transport Headers

The mechanism of method 2 is shown in Figure 2. In this method, the CF classifies packets and stacks transport headers in which actual network address is included, e.g., MPLS or GRE headers, onto the packets based on the classification. This method does not require any forwarding function for forwarding packets based on the service

information. Forwarding functions of underlay networks forward the packets to SFs following the outermost header. The outermost header is removed after service process of the SF. The actions are repeated until all headers are removed.

```
*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 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   |
            +---------+      +---------+    +---------+     +---------+
```

Figure 2: Forwarding with Stacked Multiple Transport 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 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 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.

```
  *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]

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

Figure 3: Forwarding Based on Service Chain Identifiable Tags

3.2. Service Path Selection Patterns

Since SC contains only logical information (e.g. series of services that are applied to 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 only a CF; that is, the SP is statically pre-established as an end-to-end path and a CF inserts packets into the appropriate path based on the result of the classification. Each FWD on the route has a forwarding table to uniquely determine the next destination of packets, and each FWD statically forwards the received packets to 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 SP for each packet 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  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 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 instance is prepared for each SP. At each FWD, incoming packets are statically forwarded to the 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  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 path, which is sectioned by CFs and FWDs, is referred to as a segmented path in this draft. CFs or FWDs that select the next segmented path may require notification of forwarding 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 the information. The CE may have 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 6: Dynamic Selection of Segmented Service Path

In addition, this pattern accepts establishment of hierarchical domains as following:

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 and

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

3. SF Domain Gateways that reside in data centers, linking together the levels of the hierarchy. To the higher level, this is an SF. To the lower level, this is a classifier and FWD.

How packets traverse

```
  +-----+    +------+  +----------------------+   +------+
  |    |SC#1| FWD |  |SF Domain Gateway#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 |   | SF Domain Gateway#q |     | FWD |
     |    |=======================================================>
     |    |    +-----+   |     (in DC#m)       |     +-----+
     +----+              +---------------------+
                     (Details of sub-domain #q not shown)
```

Figure 7: Service Chain Hierarchy in Service Provider Network

The components within an SF sub-domain are opaque at the top level; each SF domain gateway 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 meta-data 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 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 APN.

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

Details of Lower Level of Hierarchy

Within each SF sub-domain, there are:

1. An SF domain-gateway 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 end at the same node with the flow state, where the same classifier instance can be used for both directions of traffic.

4. Consideration of 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 3.1. 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 cause any increase in packet size 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 in original format. 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 chainged in the forwarding processes. For example, distinction of incoming ports will be required for FWD to decide the next hop appropriately when a packet traverse 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 have their own policies), because some large carriers have over a million users and even more flows. Another concern is 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 transport headers, and the number of the headers increases depending on the number of SFs which the packet will traverse. This means that size of each packet increases. Packet sizes may be restricted by the minimal 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 get 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 reattaches 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 indications of forwarding configuration. 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 identify flow or service so can not 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 technologies proposed 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 single fixed-length tag, this method can prevent an increase in 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 be also 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 SF are stated in Section 4.1.2) By using existing transport headers as the tags or outer header for forwarding, effect on network nodes such as existing router and switches might be restrained.

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, and also enables to change the following SPs without changing forwarding configurations 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 3.2. Service Paths 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 determination of the next hop. 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 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 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 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| . . . . .
     .         .           +--------------+          .
     .         .                                     .
 * *  . * * * * . * * * * * * * * * * * * * * * *  . * * * * * * * * *
```

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

Figure 9: Establishment of SPs Across 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, as it is 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 10 shows the case in which SPs are established across multiple datacenters in pattern 2. In Figure 10, 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 perform 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.

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.

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

Also, the detail analysis of 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 11.)

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

Figure 11: Multiple Control Entities in Hierarchical Service Chaining

This hierarchical model supports 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 "SF Domain Gateway," 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 SF Domain Gateway component, since the gateway 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 smaller 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 set statically. Also, many SFs are used for several service. A function to inspect the user traffic in detail, such as TDF (3GPP, [TS.23.203]), is located around the edge 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 protocols. 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 are 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 apply 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 service such as enriched security service and value added service would be provided)

Features of SFs: Many SFs are deployed as vNF, 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 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):

Chuong D. Pham
Telstra

Hiroshi Dempo
NEC

Ron Parker
Affirmed Networks

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-00 (work in progress), May 2015.

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-08 (work in progress), May 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., Stiemerling, 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.

[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

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

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