

BESS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2017

A. Farrel  
J. Drake  
E. Rosen  
Juniper Networks  
J. Uttaro  
AT&T  
L. Jalil  
Verizon  
October 30, 2016

**BGP Control Plane for NSH SFC**  
**draft-mackie-bess-nsh-bgp-control-plane-01**

Abstract

This document describes the use of BGP as a control plane for networks that support Service Function Chaining (SFC). The document introduces a new BGP address family called the SFC AFI/SAFI with two route types. One route type is originated by a node to advertise that it hosts a particular instance of a specified service function. This route type also provides "instructions" on how to send a packet to the hosting node in a way that indicates that the service function has to be applied to the packet. The other route type is used by a Controller to advertise the paths of "chains" of service functions, and to give a unique designator to each such path so that they can be used in conjunction with the Network Service Header.

This document adopts the SFC architecture described in [RFC 7665](#).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

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 May 3, 2017.

## Copyright Notice

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

This document is subject to [BCP 78](#) 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="#">1.1.</a>	Terminology . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Overview . . . . .	<a href="#">5</a>
<a href="#">2.1.</a>	Functional Overview . . . . .	<a href="#">5</a>
<a href="#">2.2.</a>	Control Plane Overview . . . . .	<a href="#">7</a>
<a href="#">3.</a>	BGP SFC Routes . . . . .	<a href="#">9</a>
<a href="#">3.1.</a>	Service Function Instance Route (SFIR) . . . . .	<a href="#">10</a>
<a href="#">3.2.</a>	Service Function Path Route (SFPR) . . . . .	<a href="#">10</a>
<a href="#">3.2.1.</a>	The SFP Attribute . . . . .	<a href="#">11</a>
<a href="#">3.2.2.</a>	General Rules For The SFP Attribute . . . . .	<a href="#">15</a>
<a href="#">4.</a>	Mode of Operation . . . . .	<a href="#">16</a>
<a href="#">4.1.</a>	Route Targets . . . . .	<a href="#">16</a>
<a href="#">4.2.</a>	Service Function Instance Routes . . . . .	<a href="#">17</a>
<a href="#">4.3.</a>	Service Function Path Routes . . . . .	<a href="#">17</a>
<a href="#">4.4.</a>	Classifier Operation . . . . .	<a href="#">19</a>
<a href="#">4.5.</a>	Service Function Forwarder Operation . . . . .	<a href="#">19</a>
<a href="#">5.</a>	Selection in Service Function Paths . . . . .	<a href="#">20</a>
<a href="#">6.</a>	Looping, Jumping, and Branching . . . . .	<a href="#">21</a>
<a href="#">6.1.</a>	Protocol Control of Looping, Jumping, and Branching . . . . .	<a href="#">22</a>
<a href="#">6.2.</a>	Implications for Forwarding State . . . . .	<a href="#">23</a>
<a href="#">7.</a>	Advanced Topics . . . . .	<a href="#">23</a>
<a href="#">7.1.</a>	Preserving Entropy . . . . .	<a href="#">23</a>
<a href="#">7.2.</a>	Correlating Service Function Path Instances . . . . .	<a href="#">24</a>
<a href="#">7.3.</a>	VPN Considerations and Private Service Functions . . . . .	<a href="#">24</a>
<a href="#">8.</a>	Examples . . . . .	<a href="#">25</a>
<a href="#">8.1.</a>	Example Explicit SFP With No Choices . . . . .	<a href="#">26</a>



<a href="#">8.2.</a>	<a href="#">Example SFP With Choice of SFIs . . . . .</a>	<a href="#">27</a>
<a href="#">8.3.</a>	<a href="#">Example SFP With Open Choice of SFIs . . . . .</a>	<a href="#">28</a>
<a href="#">8.4.</a>	<a href="#">Example SFP With Choice of SFTs . . . . .</a>	<a href="#">28</a>
<a href="#">8.5.</a>	<a href="#">Example Correlated Bidirectional SFPs . . . . .</a>	<a href="#">29</a>
<a href="#">8.6.</a>	<a href="#">Example Correlated Asymmetrical Bidirectional SFPs . . . . .</a>	<a href="#">29</a>
<a href="#">8.7.</a>	<a href="#">Example Looping in an SFP . . . . .</a>	<a href="#">30</a>
<a href="#">8.8.</a>	<a href="#">Example Branching in an SFP . . . . .</a>	<a href="#">31</a>
<a href="#">9.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">31</a>
<a href="#">10.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">32</a>
<a href="#">10.1.</a>	<a href="#">New BGP AF/SAFI . . . . .</a>	<a href="#">32</a>
<a href="#">10.2.</a>	<a href="#">New BGP Path Attribute . . . . .</a>	<a href="#">32</a>
<a href="#">10.3.</a>	<a href="#">New SFP Attribute TLVs Type Registry . . . . .</a>	<a href="#">32</a>
<a href="#">10.4.</a>	<a href="#">New SFP Association Type Registry . . . . .</a>	<a href="#">33</a>
<a href="#">10.5.</a>	<a href="#">New Service Function Type Registry . . . . .</a>	<a href="#">34</a>
<a href="#">11.</a>	<a href="#">Contributors . . . . .</a>	<a href="#">34</a>
<a href="#">12.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">35</a>
<a href="#">13.</a>	<a href="#">References . . . . .</a>	<a href="#">35</a>
<a href="#">13.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">35</a>
<a href="#">13.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">36</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">36</a>

## **[1.](#) Introduction**

As described in [[RFC7498](#)], the delivery of end-to-end services can require a packet to pass through a series of Service Functions (SFs) (e.g., classifiers, firewalls, TCP accelerators, and server load balancers) in a specified order: this is termed "Service Function Chaining" (SFC). There are a number of issues associated with deploying and maintaining service function chaining in production networks, which are described below.

Conventionally, if a packet needs to travel through a particular service chain, the nodes hosting the service functions of that chain are placed in the network topology in such a way that the packet cannot reach its ultimate destination without first passing through all the service functions in the proper order. This need to place the service functions at particular topological locations limits the ability to adapt a service function chain to changes in network topology (e.g., link or node failures), network utilization, or offered service load. These topological restrictions on where the service functions can be placed raise the following issues:

1. The process of configuring or modifying a service function chain is operationally complex and may require changes to the network topology.
2. Alternate or redundant service functions may need to be co-located with the primary service functions.



3. When there is more than one path between source and destination, forwarding may be asymmetric and it may be difficult to support bidirectional service function chains using simple routing methodologies and protocols without adding mechanisms for traffic steering or traffic engineering.

In order to address these issues, the SFC architecture includes Service Function Chains that are built in their own overlay network (the service function overlay network), coexisting with other overlay networks, over a common underlay network [[RFC7665](#)]. A Service Function Chain is a sequence of Service Functions through which packet flows satisfying specified criteria will pass.

This document describes the use of BGP as a control plane for networks that support Service Function Chaining (SFC). The document introduces a new BGP address family called the SFC AFI/SAFI with two route types. One route type is originated by a node to advertise that it hosts a particular instance of a specified service function. This route type also provides "instructions" on how to send a packet to the hosting node in a way that indicates that the service function has to be applied to the packet. The other route type is used by a Controller to advertise the paths of "chains" of service functions, and to give a unique designator to each such path so that they can be used in conjunction with the Network Service Header.

This document adopts the SFC architecture described in [[RFC7665](#)].

### **1.1. Terminology**

This document uses the following terms from [[RFC7665](#)]:

- o Bidirectional Service Function Chain
- o Classifier
- o Service Function (SF)
- o Service Function Chain (SFC)
- o Service Function Forwarder (SFF)
- o Service Function Instance (SFI)
- o Service Function Path (SFP)
- o SFC branching



Additionally, this document uses the following terms from [\[I-D.ietf-sfc-nsh\]](#):

- o Network Service Header (NSH)
- o Service Index (SI)
- o Service Path Identifier (SPI)

This document introduces the following terms:

- o Service Function Instance Route (SFIR)
- o Service Function Overlay Network
- o Service Function Path Route (SFPR)
- o Service Function Type (SFT)

## **[2.](#) Overview**

### **[2.1.](#) Functional Overview**

In [\[I-D.ietf-sfc-nsh\]](#) a Service Function Chain (SFC) is an ordered list of Service Functions (SFs). A Service Function Path (SFP) is an indication of which instances of SFs are acceptable to be traversed in an instantiation of an SFC in a service function overlay network. The Service Path Identifier (SPI) is a 24-bit number that identifies a specific SFP, and a Service Index (SI) is an 8-bit number that identifies a specific point in that path. In the context of a particular SFP (identified by an SPI), an SI represents a particular Service Function, and indicates the order of that SF in the SFP.

In fact, each SI is mapped to one or more SFs that are implemented by one or more Service Function Instances (SFIs) that support those specified SFs. Thus an SI may represent a choice of SFIs of one or more Service Function Types. By deploying multiple SFIs for a single SF, one can provide load balancing and redundancy.

A special Service Function, called a Classifier, is located at each ingress point to a service function overlay network. It assigns the packets of a given packet flow to a specific Service Function Path. This may be done by comparing specific fields in a packet's header with local policy, which may be customer/network/service specific. The classifier picks an SFP and sets the SPI accordingly it then sets the SI to the value of the SI for the first hop in the SFP and then prepending a Network Services Header (NSH) [\[I-D.ietf-sfc-nsh\]](#), to that packet containing the assigned SPI/SI. Note that the Classifier





and the node that hosts the first Service Function in a Service Function Path need not be located at the same point in the service function overlay network.

Note that the presence of the NSH can make it difficult for nodes in the underlay network to locate the fields in the original packet that would normally be used to constrain equal cost multipath (ECMP) forwarding. Therefore, it is recommended, as described in [Section 7.1](#), that the node prepending the NSH also provide some form of entropy indicator that can be used in the underlay network.

The Service Function Forwarder (SFF) receives a packet from the previous node in a Service Function Path, removes the packet's link layer or tunnel encapsulation and hands the packet and the NSH to the Service Function Instance for processing.

When the SFF receives the packet and the NSH back from the SFI it must select the next SFI along the path using the SPI and SI in the NSH and potentially choosing between multiple SFIs (possibly of different Service Function Types) as described in [Section 5](#). In the normal case the SPI remains unchanged and the SI will have been decremented to indicate the next SF along the path. But other possibilities exist if the SF makes other changes to the NSH through a process of re-classification:

- o The SI in the NSH may indicate:
  - \* A previous SF in the path: known as "looping" (see [Section 6](#)).
  - \* An SF further down the path: known as "jumping" (see also [Section 6](#)).
- o The SPI and the SI may point to an SF on a different SFP: known as "branching" (see also [Section 6](#)).

Such modifications are limited to within the same service function overlay network. That is, an SPI is known within the scope of service function overlay network. Furthermore, the new SI value is interpreted in the context of the SFP identified by the SPI, and SI values that do not form part of the definition of the path are invalid.

An unknown or invalid SPI/SI combination SHALL be treated as an error and the SFF MUST drop the packet. Such errors SHOULD be logged, and such logs MUST be subject to rate limits. See [[I-D.ietf-sfc-nsh](#)] for more details of handling this situation in received NSH packets.



The SFF then selects an SFI that provides the SF denoted by the SPI/SI, and forwards the packet to the SFF that supports that SFI.

## **2.2. Control Plane Overview**

To accomplish the function described in [Section 2.1](#), this document introduces a new BGP AFI/SAFI [values to be assigned by IANA] for "SFC Routes". Two SFC Route Types are defined by this document: the Service Function Instance Route (SFIR), and the Service Function Path Route (SFPR). As detailed in [Section 3](#), the route type is indicated by a sub-field in the NLRI.

- o The SFIR is advertised by the node hosting the service function instance. The SFIR describes a particular instance of a particular Service Function and the way to forward a packet to it through the underlay network, i.e., IP address and encapsulation information.
- o The SFPRs are originated by Controllers. One SFPR is originated for each Service Function Path. The SFPR specifies:
  - A. the SPI of the path
  - B. the sequence of SFTs and/or SFIs of which the path consists
  - C. for each such SFT or SFI, the SI that represents it in the identified path.

This approach assumes that there is an underlay network that provides connectivity between SFFs and Controllers, and that the SFFs are grouped to form one or more service function overlay networks through which SFPs are built. We assume BGP connectivity between the Controllers and all SFFs within each service function overlay network.

In addition, we also introduce the Service Function Type (SFT) that is the category of SF that is supported by an SFF (such as "firewall"). An IANA registry of Service Function Types is introduced in [Section 10](#). An SFF may support SFs of multiple different SFTs, and may support multiple SFIs of each SF.

When choosing the next SFI in a path, the SFF uses the SPI and SI as well as the SFT to choose among the SFIs, applying, for example, a load balancing algorithm or direct knowledge of the underlay network topology as described in [Section 4](#).



The SFF then encapsulates the packet using the encapsulation specified by the SFIR of the selected SFI and forwards the packet. See Figure 1.

Thus the SFF can be seen as a portal in the underlay network through which a particular SFI is reached.

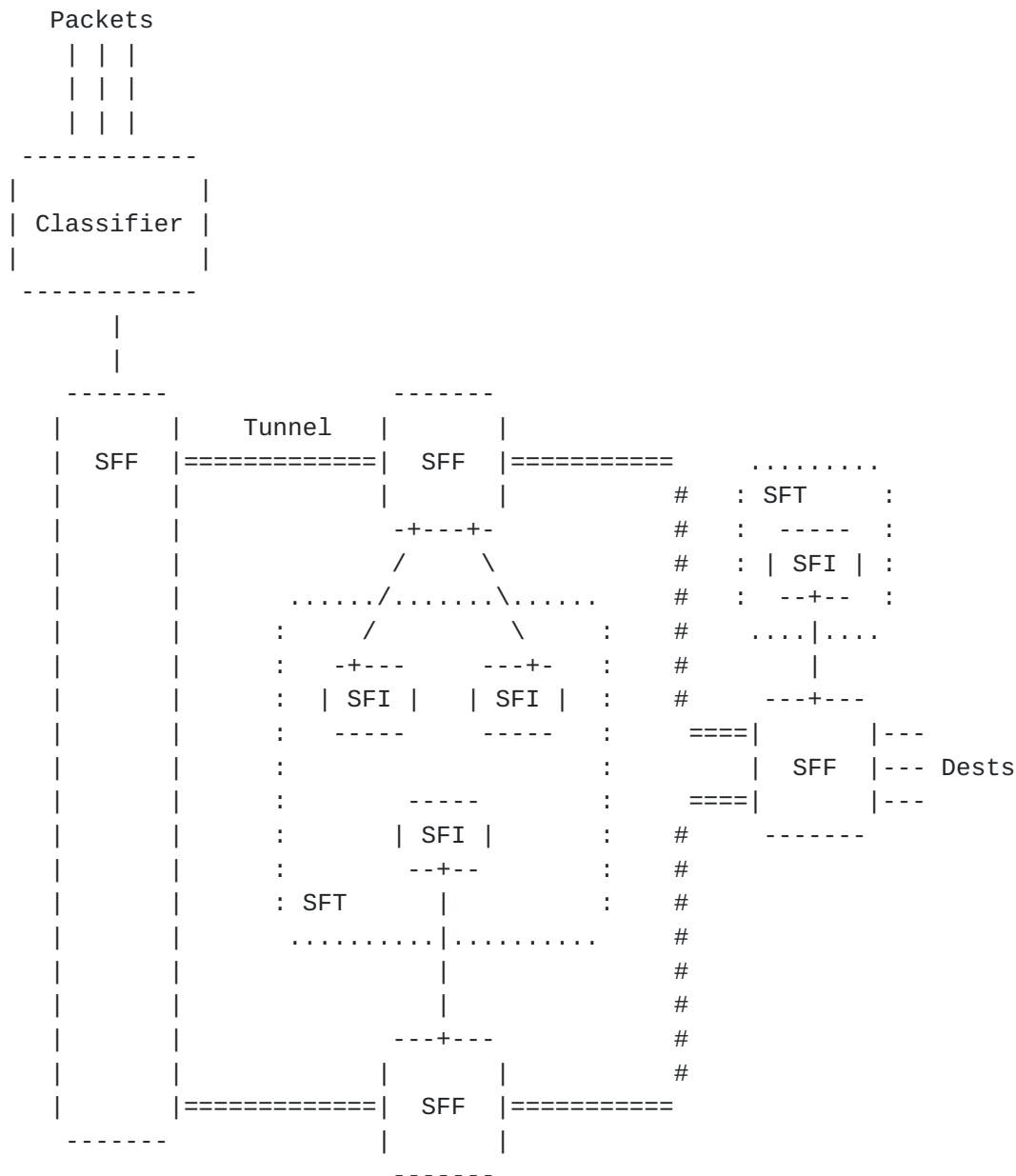


Figure 1: The SFC Architecture Reference Model



### 3. BGP SFC Routes

This document defines a new AFI/SAFI for BGP, known as "SFC", with an NLRI that is described in this section.

The format of the SFC NLRI is shown in Figure 2.

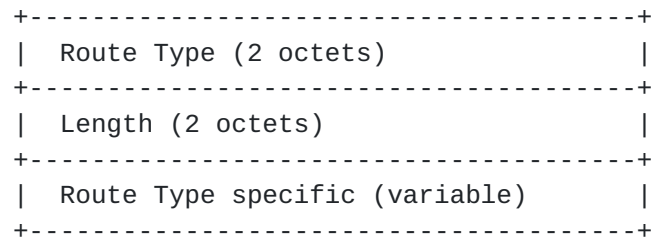


Figure 2: The Format of the SFC NLRI

The Route Type field determines the encoding of the rest of the route type specific SFC NLRI.

The Length field indicates the length in octets of the route type specific field of the SFC NLRI.

This document defines the following Route Types:

1. Service Function Instance Route (SFIR)
2. Service Function Path Route (SFPR)

A Service Function Instance Route (SFIR) is used to identify an SFI. A Service Function Path Route (SFPR) defines a sequence of Service Functions (each of which has at least one instance advertised in an SFIR) that form an SFP.

The detailed encoding and procedures for these Route Types are described in subsequent sections.

The SFC NLRI is carried in BGP [[RFC4271](#)] using BGP Multiprotocol Extensions [[RFC4760](#)] with an Address Family Identifier (AFI) of TBD1 and a Subsequent Address Family Identifier (SAFI) of TBD2. The NLRI field in the MP\_REACH\_NLRI/MP\_UNREACH\_NLRI attribute contains the SFC NLRI, encoded as specified above.

In order for two BGP speakers to exchange SFC NLRIs, they must use BGP Capabilities Advertisements to ensure that they both are capable of properly processing such NLRIs. This is done as specified in





[RFC4760], by using capability code 1 (Multiprotocol BGP) with an AFI of TBD1 and a SAFI of TBD2.

### 3.1. Service Function Instance Route (SFIR)

Figure 3 shows the Route Type specific NLRI of the SFIR.

```

+-----+
| Route Distinguisher (RD) (8 octets) |
+-----+
| Service Function Type (2 octets) |
+-----+

```

Figure 3: SFIR Route Type specific NLRI

Per [RFC4364] the RD field comprises a two byte Type field and a six byte Value field. Two SFIs of the same SFT must be associated with different RDs, where the association of an SFI with an RD is determined by provisioning. If two SFIRs are originated from different administrative domains, they must have different RDs. In particular, SFIRs from different VPNs (for different service function overlay networks) must have different RDs, and those RDs must be different from any non-VPN SFIRs.

The Service Function Type identifies a service function, e.g., classifier, firewall, load balancer, etc. There may be several SFIs that can perform a given Service Function. Each node hosting an SFI must originate an SFIR for each SFI that it hosts. The SFIR representing a given SFI will contain an NLRI with RD field set to an RD as specified above, and with SFT field set to identify that SFI's Service Function Type. The values for the SFT field are taken from a registry administered by IANA (see [Section 10](#)). A BGP Update containing one or more SFIRs will also include a Tunnel Encapsulation attribute [[I-D.ietf-idr-tunnel-encaps](#)]. If a data packet needs to be sent to an SFI identified in one of the SFIRs, it will be encapsulated as specified by the Tunnel Encapsulation attribute, and then transmitted through the underlay network.

### 3.2. Service Function Path Route (SFPR)

Figure 4 shows the Route Type specific NLRI of the SFPR.



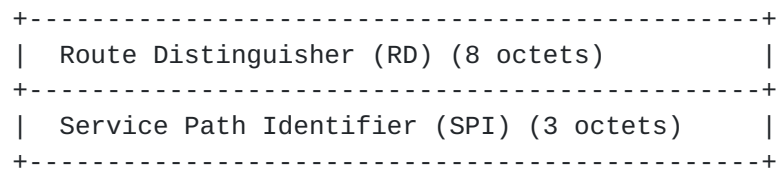


Figure 4: SFP Route Type Specific NLRI

Per [RFC4364] the RD field comprises a two byte Type field and a six byte Value field. All SFPs must be associated with different RDs. The association of an SFP with an RD is determined by provisioning. If two SFPs are originated from different Controllers they must have different RDs. Additionally, SFPs from different VPNs (i.e., in different service function overlay networks) must have different RDs, and those RDs must be different from any non-VPN SFPs.

The Service Path Identifier is defined in [I-D.ietf-sfc-nsh] and is the value to be placed in the Service Path Identifier field of the NSH header of any packet sent on this Service Function Path. It is expected that one or more Controllers will originate these routes in order to configure a service function overlay network.

The SFP is described in a new BGP Path attribute, the SFP attribute. [Section 3.2.1](#) shows the format of that attribute.

### **3.2.1. The SFP Attribute**

[RFC4271] defines the BGP Path attribute. This document introduces a new Path attribute called the SFP attribute with value TBD3 to be assigned by IANA. The first SFP attribute MUST be processed and subsequent instances MUST be ignored.

The common fields of the SFP attribute are set as follows:

- o Optional bit is set to 1 to indicate that this is an optional attribute.
- o The Transitive bit is set to 1 to indicate that this is a transitive attribute.
- o The Extended Length bit is set according to the length of the SFP attribute as defined in [RFC4271].
- o The Attribute Type Code is set to TBD3.



The content of the SFP attribute is a series of Type-Length-Variable (TLV) constructs. Each TLV may include sub-TLVs. All TLVs and sub-TLVs have a common format that is:

- o Type: A single octet indicating the type of the SFP attribute TLV. Values are taken from the registry described in [Section 10.3](#).
- o Length: A two octet field indicating the length of the data following the Length field counted in octets.
- o Value: The contents of the TLV.

The formats of the TLVs defined in this document are shown in the following sections. The presence rules and meanings are as follows.

- o The SFP attribute contains a sequence of zero or more Association TLVs. That is, the Association TLV is optional. Each Association TLV provides an association between this SFPR and another SFPR. Each associated SFPR is indicated using the RD with which it is advertised (we say the SFPR-RD to avoid ambiguity).
- o The SFP attribute contains a sequence of one or more Hop TLVs. Each Hop TLV contains all of the information about a single hop in the SFP.
- o Each Hop TLV contains an SI value and a sequence of one or more SFT TLVs. Each SFT TLV contains an SFI reference for each instance of an SF that is allowed at this hop of the SFP for the specific SFT. Each SFI is indicated using the RD with which it is advertised (we say the SFIR-RD to avoid ambiguity).

#### **[3.2.1.1](#). The Association TLV**

The Association TLV is an optional TLV in the SFP attribute. It may be present multiple times. Each occurrence provides an association with another SFP as advertised in another SFPR. The format of the Association TLV is shown in Figure 5



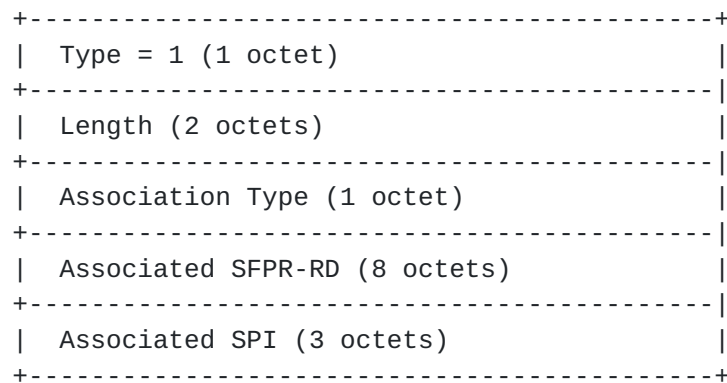


Figure 5: The Format of the Association TLV

The fields are as follows:

Type is set to 1 to indicate an Association TLV.

Length indicates the length in octets of the Association Type and Associated SFPR-RD fields. The value of the Length field is 12.

The Association Type field indicate the type of association. The values are tracked in an IANA registry (see [Section 10.4](#)). Only one value is defined in this document: type 1 indicates association of two unidirectional SFPs to form a bidirectional SFP. An SFP attribute SHOULD NOT contain more than one Association TLV with Association Type 1: if more than one is present, the first one MUST be processed and subsequent instances MUST be ignored. Note that documents that define new Association Types must also define the presence rules for Association TLVs of the new type.

The Associated SFPR-RD contains the RD of some other SFPR advertisement that contains the SFP with which this SFP is associated.

The Associated SPI contains the SPI of the associated SFP as advertised in the SFPR indicated by the Associated SFPR-RD field.

Association TLVs with unknown Association Type values SHOULD be ignored. Association TLVs that contain an Associated SFPR-RD value equal to the RD of the SFPR in which they are contained SHOULD be ignored. If the Associated SPI is not equal to the SPI advertised in the SFPR indicated by the Associated SFPR-RD then the Association TLV SHOULD be ignored.





Note that when two SFPRs reference each other using the Association TLV, one SFPR advertisement will be received before the other. Therefore, processing of an association MUST NOT be rejected simply because the Associated SFPR-RD is unknown.

Further discussion of correlation of SFPRs is provided in [Section 7.2](#).

#### **[3.2.1.2](#). The Hop TLV**

There is one Hop TLV in the SFP attribute for each hop in the SFP. The format of the Hop TLV is shown in Figure 6. At least one Hop TLV must be present in an SFP attribute.

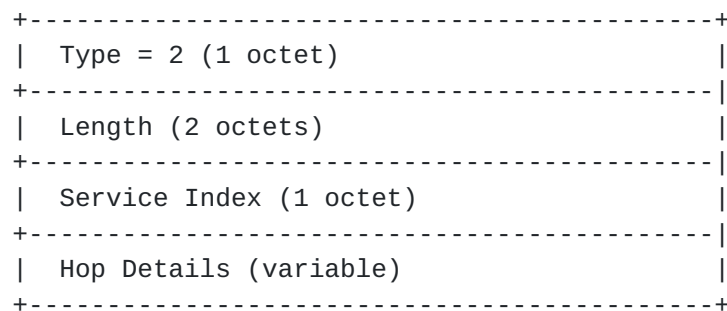


Figure 6: The Format of the Hop TLV

The fields are as follows:

Type is set to 2 to indicate a Hop TLV.

Length indicates the length in octets of the Service Index and Hop Details fields.

The Service Index is defined in [[I-D.ietf-sfc-nsh](#)] and is the value found in the Service Index field of the NSH header that an SFF will use to lookup to which next SFI a packet should be sent.

The Hop Details consist of a sequence of one or more SFT TLVs.

#### **[3.2.1.3](#). The SFT TLV**

There is one or more SFT TLV in each Hop TLV. There is one SFT TLV for each SFT supported in the specific hop of the SFP. The format of the SFT TLV is shown in Figure 7.



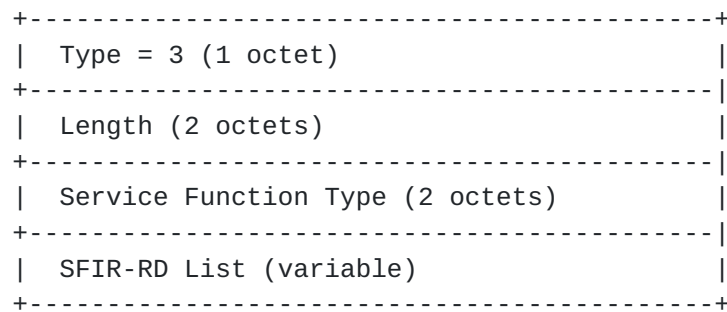


Figure 7: The Format of the SFT TLV

The fields are as follows:

Type is set to 3 to indicate an SFT TLV.

Length indicates the length in octets of the Service Function Type and SFIR-RD List fields.

The Service Function Type is used to identify a Service Function Instance Route in the service function overlay network which, in turn, will allow lookup of routes to SFIs implementing the SF. SFT values in the range 1-31 are Special Purpose SFT values and have meanings defined by the documents that describe them - the value 'Change Sequence' is defined in [Section 6.1](#) of this document.

The SFIR-RD List is made up of one or more SFIR-RD values from the advertisements of SFIs in SFIRs. An SFIR-RD of value zero has special meaning as described in [Section 5](#). Each entry in the list is 8 octets long, and the number of entries in the list can be deduced from the value of the Length field.

### **3.2.2. General Rules For The SFP Attribute**

It is possible for the same SFI, as described by an SFIR, to be used in multiple SFPRs.

When two SFPRs have the same SPI but different SFPR-RDs there can be three cases:

- o Two or more Controllers are originating SFPRs for the same SFP. In this case the content of the SFPRs is identical and the duplication is to ensure receipt and to provide Controller redundancy.



- o There is a transition in content of the advertised SFP and the advertisements may originate from one or more Controllers. In this case the content of the SFPRs will be different.
- o The reuse of an SPI may result from a configuration error.

In all cases, there is no way for the receiving SFF to know which SFPR to process, and the SFPRs could be received in any order. At any point in time, when multiple SFPRs have the same SPI but different SFPR-RDs, the SFF MUST use the SFPR with the numerically lowest SFPR-RD. The SFF SHOULD log this occurrence to assist with debugging.

Furthermore, a Controller that wants to change the content of an SFP is RECOMMENDED to use a new SPI and so create a new SFP onto which the Classifiers can transition packet flows before the SFPR for the old SFP is withdrawn. This avoids any race conditions with SFPR advertisements.

Additionally, a Controller SHOULD NOT re-use an SPI after it has withdrawn the SFPR that used it until at least a configurable amount of time has passed. This timer SHOULD have a default of one hour.

#### **4. Mode of Operation**

This document describes the use of BGP as a control plane to create and manage a service function overlay network.

##### **4.1. Route Targets**

The main feature introduced by this document is the ability to create multiple service function overlay networks through the use of Route Targets (RTs) [[RFC4364](#)].

Every BGP UPDATE containing an SFIR or SFPR carries one or more RTs. The RT carried by a particular SFIR or SFPR is determined by the provisioning of the route's originator.

Every node in a service function overlay network is configured with one or more import RTs. Thus, each SFF will import only the SFPRs with matching RTs allowing the construction of multiple service function overlay networks or the instantiation of Service Function Chains within an L3VPN or EVPN instance (see [Section 7.3](#)). An SFF that has a presence in multiple service function overlay networks (i.e., imports more than one RT) may find it helpful to maintain separate forwarding state for each overlay network.



#### **4.2. Service Function Instance Routes**

The SFIR (see [Section 3.1](#)) is used to advertise the existence and location of a specific Service Function Instance and consists of:

- o The RT as just described.
- o A Service Function Type (SFT) that is the category of Service Function that is provided (such as "firewall").
- o A Route Distinguisher (RD) that is unique to a specific instance of a service function.

#### **4.3. Service Function Path Routes**

The SFPR (see [Section 3.2](#)) describes a specific path of a Service Function Chain. The SFPR contains the Service Path Identifier (SPI) used to identify the SFP in the NSH in the data plane. It also contains a sequence of Service Indexes (SIs). Each SI identifies a hop in the SFP, and each hop is a choice between one of more SFIs.

As described in this document, each Service Function Path Route is identified in the service function overlay network by an RD and an SPI. The SPI is unique across all service function overlay networks supported by the underlay network.

The SFPR advertisement comprises:

- o An RT as described in [Section 4.1](#).
- o A tuple that identifies the SFPR
  - \* An RD that identifies an advertisement of an SFPR.
  - \* The SPI that uniquely identifies this path within all service function overlay networks supported by the underlay network. This SPI also appears in the NSH.
- o A series of Service Indexes. Each SI is used in the context of a particular SPI and identifies one or more SFs (distinguished by their SFTs) and for each SF a set of SFIs that instantiate the SF. The values of the SI indicate the order in which the SFs are to be executed in the SFP that is represented by the SPI.
- o The SI is used in the NSH to identify the entries in the SFP. Note that the SI values have meaning only relative to a specific path. They have no semantic other than to indicate the order of Service Functions within the path and are assumed to be





monotonically decreasing from the start to the end of the path [[I-D.ietf-sfc-nsh](#)].

- o Each Service Index is associated with a set of one or more Service Function Instances that can be used to provide the indexed Service Function within the path. Each member of the set comprises:
  - \* The RD used in an SFIR advertisement of the SFI.
  - \* The SFT that indicates the type of function as used in the same SFIR advertisement of the SFI.

This may be summarized as follows where the notations "SFPR-RD" and "SFIR-RD" are used to distinguish the two different RDs:

$$RT, \{SFPR-RD, SPI\}, m * \{SI, \{n * \{SFT, p * SFIR-RD\} \} \}$$

Where:

RT: Route Target

SFPR-RD: The Route Descriptor of the Service Function Path Route advertisement

SPI: Service Path Identifier used in the NSH

m: The number of hops in the Service Function Path

n: The number of choices of Service Function Type for a specific hop

p: The number of choices of Service Function Instance for given Service Function Type in a specific hop

SI: Service Index used in the NSH to indicate a specific hop

SFT: The Service Function Type used in the same advertisement of the Service Function Instance Route

SFIR-RD: The Route Descriptor used in an advertisement of the Service Function Instance Route

Note that the values of SI are from the set {255, ..., 1} and are monotonically decreasing within the SFP. SIs MUST appear in order within the SFPR (i.e., monotonically decreasing) and MUST NOT appear more than once. Malformed SFPRs MUST be discarded and MUST cause any previous instance of the SFPR (same SFPR-RD and SPI) to be discarded.



The choice of SFI is explained further in [Section 5](#). Note that an SFIR-RD value of zero has special meaning as described in that Section.

#### **4.4. Classifier Operation**

As shown in Figure 1, the Classifier is a special Service Function that is used to assign packets to an SFP.

The Classifier is responsible for determining to which packet flow a packet belongs (usually by inspecting the packet header), imposing an NSH, and initializing the NSH to include the SPI of the selected SFPR and to include the SI from first hop of the selected SFP.

The Classifier may also provide an entropy indicator as described in [Section 7.1](#).

#### **4.5. Service Function Forwarder Operation**

Each packet sent to an SFF is transmitted encapsulated in an NSH. The NSH includes an SPI and SI: the SPI indicates the SFPR advertisement that announced the Service Function Path; the tuple SPI/SI indicates a specific hop in a specific path and maps to the RD/SFT of a particular SFIR advertisement.

When an SFF gets an SFPR advertisement it will first determine whether to import the route by examining the RT. If the SFPR is imported the SFF then determines whether it is on the SFP by looking for its own SFIR-RDs in the SFPR. For each occurrence in the SFP, the SFF creates forwarding state for incoming packets and forwarding state for outgoing packets that have been processed by the specified SFI.

The SFF creates local forwarding state for packets that it receives from other SFFs. This state makes the association between the SPI/SI in the NSH of the received packet and one or more specific local SFIs as identified by the SFIR-RD/SFT. If there are multiple local SFIs that match this is because a single advertisement was made for a set of equivalent SFIs and the SFF may use local policy (such as load balancing) to determine to which SFI to forward a received packet.

The SFF also creates next hop forwarding state for packets received back from the local SFI that need to be forwarded to the next hop in the SFP. There may be a choice of next hops as described in [Section 4.3](#). The SFF could install forwarding state for all potential next hops, or it could choose to only install forwarding state to a subset of the potential next hops. If a choice is made then it will be as described in [Section 5](#).



The installed forwarding state may change over time reacting to changes in the underlay network and the availability of particular SFIs.

Note that SFFs only create and store forwarding state for the SFPs on which they are included. They do not retain state for all SFPs advertised.

An SFF may also install forwarding state to support looping, jumping, and branching. The protocol mechanism for explicit control of looping, jumping, and branching is described in [Section 6.1](#) using a special value of the SFT within an entry in an SFPR.

## 5. Selection in Service Function Paths

As described in [Section 2](#) the SPI/SI in the NSH passed back from an SFI to the SFF may leave the SFF with a choice of next hop SFTs, and a choice of SFIs for each SFT. That is, the SPI indicates an SFPR, and the SI indicates an entry in that SFPR. Each entry in an SFPR is a set of one or more SFT/SFIR-RD pairs. The SFF must choose one of these, identify the SFF that supports the chosen SFI, and send the packet to that next hop SFF.

In the typical case, the SFF chooses a next hop SFF by looking at the set of all SFFs that support the SFs identified by the SI (that set having been advertised in individual SFIR advertisements), finding the one or more that are "nearest" in the underlay network, and choosing between next hop SFFs using its own load-balancing algorithm.

An SFI may influence this choice process by passing additional information back along with the packet and NSH. This information may influence local policy at the SFF to cause it to favor a next hop SFF (perhaps selecting one that is not nearest in the underlay), or to influence the load-balancing algorithm.

This selection applies to the normal case, but also applies in the case of looping, jumping, and branching (see [Section 6](#)).

Suppose an SFF in a particular service overlay network (identified by a particular import RT, RT-z) needs to forward an NSH-encapsulated packet whose SPI is SPI-x and whose SI is SI-y. It does the following:

1. It looks for an installed SFPR that carries RT-z and that has SPI-x in its NLRI. If there is none, then such packets cannot be forwarded.



2. From the SFP attribute of that SFPR, it finds the Hop TLV with SI value set to SI-y. If there is no such Hop TLV, then such packets cannot be forwarded.
3. It then finds the "relevant" set of SFIRs by going through the list of of SFT TLVs contained in the Hop TLV as follows:
  - A. An SFIR is relevant if it carries RT-z, the SFT in its NLRI matches the SFT value in one of the SFT TLVs, and the RD value in its NLRI matches an entry in the list of SFIR-RDs in that SFT TLV.
  - B. If an entry in the SFIR-RD list of an SFT TLV contains the value zero, then an SFIR is relevant if it carries RT-z and the SFT in its NLRI matches the SFT value in that SFT TLV. I.e., any SFIR in the service function overlay network defined by RT-z and with the correct SFT is relevant.

Each of the relevant SFIRs identifies a single SFI, and contains a Tunnel Encapsulation attribute that specifies how to send a packet to that SFI. For a particular packet, the SFF chooses a particular SFI from the set of relevant SFIRs. This choice is made according to local policy.

A typical policy might be to figure out the set of SFIs that are closest, and to load balance among them. But this is not the only possible policy.

## **6. Looping, Jumping, and Branching**

As described in [Section 2](#) an SFI or an SFF may cause a packets to "loop back" to a previous SF on a path in order that a sequence of functions may be re-executed. This is simply achieved by replacing the SI in the NSH with a higher value instead of decreasing it as would normally be the case to determine the next hop in the path.

[Section 2](#) also describes how an SFI or an SFF may cause a packets to "jump forward" to an SF on a path that is not the immediate next SF in the SFP. This is simply achieved by replacing the SI in the NSH with a lower value than would be achieved by decreasing it by the normal amount.

A more complex option to move packets from one SFP to another is described in [[I-D.ietf-sfc-nsh](#)] and [Section 2](#) where it is termed "branching". This mechanism allows an SFI or SFF to make a choice of downstream treatments for packets based on local policy and output of the local SF. Branching is achieved by changing the SPI in the NSH





to indicate the new path and setting the SI to indicate the point in the path at which the packets should enter.

Note that the NSH does not include a marker to indicate whether a specific packet has been around a loop before. Therefore, the use of NSH metadata may be required in order to prevent infinite loops.

### **6.1. Protocol Control of Looping, Jumping, and Branching**

If the SFT value in an SFT TLV in an SFPR has the Special Purpose SFT value "Change Sequence" (see [Section 10](#)) then this is an indication that the SFF may make a loop, jump, or branch according to local policy and information returned by the local SFI.

In this case, the SPI and SI of the next hop is encoded in the eight bytes of an entry in the SFIR-RD list as follows:

3 bytes SPI

2 bytes SI

3 bytes Reserved (SHOULD be set to zero and ignored)

If the SI in this encoding is not part of the SFPR indicated by the SPI in this encoding, then this is an explicit error that SHOULD be detected by the SFF when it parses the SFPR. The SFPR SHOULD NOT cause any forwarding state to be installed in the SFF and packets received with the SPI that indicates this SFPR SHOULD be silently discarded.

If the SPI in this encoding is unknown, the SFF SHOULD NOT install any forwarding state for this SFPR, but MAY hold the SFPR pending receipt of another SFPR that does use the encoded SPI.

If the SPI matches the current SPI for the path, this is a loop or jump. In this case, if the SI is greater than the current SI it is a loop. If the SPI matches and the SI is less than the next SI, it is a jump.

If the SPI indicates another path, this is a branch and the SI indicates the point at which to enter that path.

The Change Sequence SFT is just another SFT that may appear in a set of SFI/SFT tuples within an SI and is selected as described in [Section 5](#).

Note that Special Purpose SFTs MUST NOT be advertised in SFIRs.



## **6.2. Implications for Forwarding State**

Support for looping and jumping requires that the SFF has forwarding state established to an SFF that provides access to an instance of the appropriate SF. This means that the SFF must have seen the relevant SFIR advertisements and known that it needed to create the forwarding state. This is a matter of local configuration and implementation: for example, an implementation could be configured to install forwarding state for specific looping/jumping.

Support for branching requires that the SFF has forwarding state established to an SFF that provides access to an instance of the appropriate entry SF on the other SFP. This means that the SFF must have seen the relevant SFIR and SFPR advertisements and known that it needed to create the forwarding state. This is a matter of local configuration and implementation: for example, an implementation could be configured to install forwarding state for specific branching (identified by SPI and SI).

## **7. Advanced Topics**

This section highlights several advanced topics introduced elsewhere in this document.

### **7.1. Preserving Entropy**

Forwarding decisions in the underlay network in the presence of equal cost multipath (ECMP) are usually made by inspecting key invariant fields in a packet header so that all packets from the same packet flow receive the same forwarding treatment. However, when an NSH is included in a packet, those key fields may be inaccessible. For example, the fields may be too far inside the packet for a forwarding engine to quickly find them and extract their values, or the node performing the examination may be unaware of the format and meaning of the NSH and so unable to parse far enough into the packet.

Various mechanisms exist within forwarding technologies to include an "entropy indicator" within a forwarded packet. For example, in MPLS there is the entropy label [[RFC6790](#)], while for encapsulations in UDP the source port field is often used to carry an entropy indicator (such as for MPLS in UDP [[RFC7510](#)]).

Implementations of this specification are RECOMMENDED to include an entropy indicator within the packet's underlay network header, and SHOULD preserve any entropy indicator from a received packet for use on the same packet when it is forwarded along the path but MAY choose to generate a new entropy indicator so long as the method used is constant for all packets. Note that preserving per packet entropy



may require that the entropy indicator is passed to and returned by the SFI to prevent the SFF from having to maintain per-packet state.

## **7.2. Correlating Service Function Path Instances**

It is often useful to create bidirectional SFPs to enable packet flows to traverse the same set of SFs, but in the reverse order. However, packets on SFPs in the data plane (per [[I-D.ietf-sfc-nsh](#)]) do not contain a direction indicator, so each direction must use a different SPI.

As described in [Section 3.2.1.1](#) an SFPR can contain one or more correlators encoded in Association TLVs. If the Association Type indicates "Bidirectional SFP" then the SFP advertised in the SFPR is one direction of a bidirectional pair of SFPs where the other in the pair is advertised in the SFPR with RD as carried in the Associated SFPR-RD field of the Association TLV. The SPI carried in the Associated SPI field of the Association TLV provides a cross-check and should match the SPI advertised in the SFPR with RD as carried in the Associated SFPR-RD field of the Association TLV.

As noted in [Section 3.2.1.1](#) SFPRs reference each other one SFPR advertisement will be received before the other. Therefore processing of an association will require that the first SFPR is not rejected simply because the Associated SFPR-RD it carries is unknown. However, the SFP defined by the first SFPR is valid and SHOULD be available for use as a unidirectional SFP even in the absence of an advertisement of its partner.

Furthermore, in error cases where SFPR-a associates with SFPR-b, but SFPR-b associates with SFPR-c such that a bidirectional pair of SFPs cannot be formed, the individual SFPs are still valid and SHOULD be available for use as unidirectional SFPs. An implementation SHOULD log this situation because it represents a Controller error.

Usage of a bidirectional SFP may be programmed into the Classifiers by the Controller. Alternatively, a Classifier may look at incoming packets on a bidirectional packet flow, extract the SPI from the received NSH, and look up the SFPR to find the reverse direction SFP to use when it sends packets.

See [Section 8](#) for an example of how this works.

## **7.3. VPN Considerations and Private Service Functions**

Likely deployments include reserving specific instances of Service Functions for specific customers or allowing customers to deploy their own Service Functions within the network. Building Service



Functions in such environments requires that suitable identifiers are used to ensure that SFFs distinguish which SFIs can be used and which cannot.

This problem is similar to how VPNs are supported and is solved in a similar way. The RT field is used to indicate a set of Service Functions from which all choices must be made.

## **8. Examples**

Assume we have a service function overlay network with four SFFs (SFF1, SFF2, SFF3, and SFF4). The SFFs have addresses in the underlay network as follows:

```
SFF1 192.0.2.1
SFF2 192.0.2.2
SFF3 192.0.2.3
SFF4 192.0.2.4
```

Each SFF provides access to some SFIs from the four Service Function Types SFT=41, SFT=42, SFT=43, and SFT=44 as follows:

```
SFF1 SFT=41 and SFT=42
SFF2 SFT=41 and SFT=43
SFF3 SFT=42 and SFT=44
SFF4 SFT=43 and SFT=44
```

The service function network also contains a Controller with address 198.51.100.1.

This example service function overlay network is shown in Figure 8.





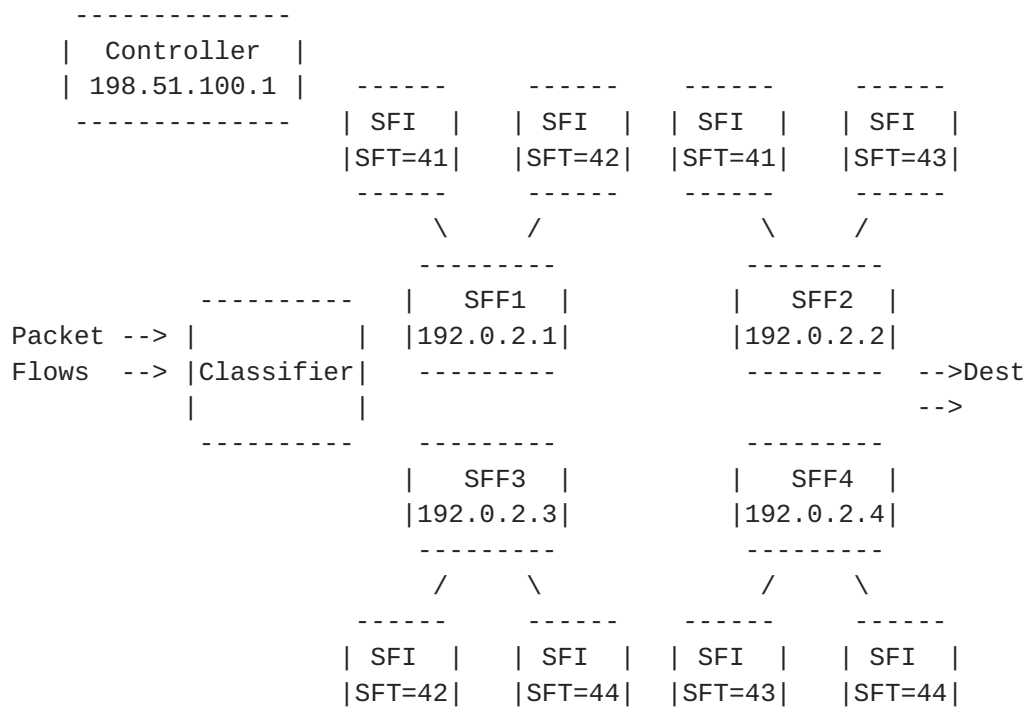


Figure 8: Example Service Function Overlay Network

The SFFs advertise routes to the SFIs they support. So we see the following SFIRs:

```

RD = 192.0.2.1,1, SFT = 41
RD = 192.0.2.1,2, SFT = 42
RD = 192.0.2.2,1, SFT = 41
RD = 192.0.2.2,2, SFT = 43
RD = 192.0.2.3,7, SFT = 42
RD = 192.0.2.3,8, SFT = 44
RD = 192.0.2.4,5, SFT = 43
RD = 192.0.2.4,6, SFT = 44
  
```

Note that the addressing used for communicating between SFFs is taken from the Tunnel Encapsulation attribute of the SFIR and not from the SFIR-RD.

### 8.1. Example Explicit SFP With No Choices

Consider the following SFPR.



```
SFP1:  RD = 198.51.100.1,101, SPI = 15,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 43, RD = 192.0.2.2,2]
```

The Service Function Path consists of an SF of type 41 located at SFF1 followed by an SF of type 43 located at SFF2. This path is fully explicit and each SFF is offered no choice in forwarding packet along the path.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (15). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 has no flexibility in the choice of SFF to support the next hop SFI and will forward the packet to SFF2 which will send the packets to the SFI that supports SFT 43 before forwarding the packets to their destinations.

## **8.2. Example SFP With Choice of SFIs**

```
SFP2:  RD = 198.51.100.1,102, SPI = 16,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,],  
       [SI = 250, SFT = 43, {RD = 192.0.2.2,2,  
                           RD = 192.0.2.4,5 } ]
```

In this example the path also consists of an SF of type 41 located at SFF1 and this is followed by an SF of type 43, but in this case the SI = 250 contains a choice between the SFI located at SFF2 and the SFI located at SFF4.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (16). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a choice of next hop SFF to execute the next hop in the path. It can either forward packets to SFF2 or SFF4 to execute a function of type 43. It uses its local load balancing algorithm to make this choice. The chosen SFF will send the packets to the SFI that supports SFT 43 before forwarding the packets to their destinations.



### **8.3. Example SFP With Open Choice of SFIs**

```
SFP3:  RD = 198.51.100.1,103, SPI = 17,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 44, RD = 0]
```

In this example the path also consists of an SF of type 41 located at SFF1 and this is followed by an SI with an RD of zero and SF of type 44. This means that a choice can be made between any SFF that supports an SFI of type 44.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (17). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a free choice of next hop SFF to execute the next hop in the path selecting between all SFFs that support SFs of type 44. Looking at the SFIRs it has received, SFF1 knows that SF type 44 is supported by SFF3 and SFF4. SFF1 uses its local load balancing algorithm to make this choice. The chosen SFF will send the packets to the SFI that supports SFT 44 before forwarding the packets to their destinations.

### **8.4. Example SFP With Choice of SFTs**

```
SFP4:  RD = 198.51.100.1,104, SPI = 18,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, {SFT = 43, RD = 192.0.2.2,2,  
                  SFT = 44, RD = 192.0.2.3,8 } ]
```

This example provides a choice of SF type in the second hop in the path. The SI of 250 indicates a choice between SF type 43 located through SF2 and SF type 44 located at SF3.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (18). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a free choice of next hop SFF to execute the next hop in the path selecting between all SFF2 that support an SF of type 43 and SFF3 that supports an SF of type 44. These may be completely different functions that are to



be executed dependent on specific conditions, or may be similar functions identified with different type identifiers (such as firewalls from different vendors). SFF1 uses its local policy and load balancing algorithm to make this choice, and may use additional information passed back from the local SFI to help inform its selection. The chosen SFF will send the packets to the SFI that supports the chose SFT before forwarding the packets to their destinations.

### **8.5. Example Correlated Bidirectional SFPs**

SFP5: RD = 198.51.100.1,105, SPI = 19,  
Assoc-Type = 1, Assoc-RD = 198.51.100.1,106, Assoc-SPI = 20,  
[SI = 255, SFT = 41, RD = 192.0.2.1,1],  
[SI = 250, SFT = 43, RD = 192.0.2.2,2]

SFP6: RD = 198.51.100.1,106, SPI = 20,  
Assoc-Type = 1, Assoc-RD = 198.51.100.1,105, Assoc-SPI = 19,  
[SI = 254, SFT = 43, RD = 192.0.2.2,2],  
[SI = 249, SFT = 41, RD = 192.0.2.1,1]

This example demonstrates correlation of two SFPs to form a bidirectional SFP as described in [Section 7.2](#).

Two SFPRs are advertised by the Controller. They have different SPIs (19 and 20) so they are known to be separate SFPs, but they both have Association TLVs with Association Type set to 1 indicating bidirectional SFPs. Each has an Associated SFPR-RD fields containing the value of the other SFPR-RD to correlated the two SFPs as a bidirectional pair.

As can be seen from the SFPRs in this example, the paths are symmetric: the hops in SFP5 appear in the reverse order in SFP6.

### **8.6. Example Correlated Asymmetrical Bidirectional SFPs**





SFP7: RD = 198.51.100.1,107, SPI = 21,  
Assoc-Type = 1, Assoc-RD = 198.51.100.1,108, Assoc-SPI = 22,  
[SI = 255, SFT = 41, RD = 192.0.2.1,1],  
[SI = 250, SFT = 43, RD = 192.0.2.2,2]

SFP8: RD = 198.51.100.1,108, SPI = 22,  
Assoc-Type = 1, Assoc-RD = 198.51.100.1,107, Assoc-SPI = 21,  
[SI = 254, SFT = 44, RD = 192.0.2.4,6],  
[SI = 249, SFT = 41, RD = 192.0.2.1,1]

Asymmetric bidirectional SFPs can also be created. This example shows a pair of SFPs with distinct SPIs (21 and 22) that are correlated in the same way as in the example in [Section 8.5](#).

However, unlike in that example, the SFPs are different in each direction. Both paths include a hop of SF type 41, but SFP7 includes a hop of SF type 43 supported at SFF2 while SFP8 includes a hop of SF type 44 supported at SFF4.

### [8.7](#). Example Looping in an SFP

SFP9: RD = 198.51.100.1,109, SPI = 23,  
[SI = 255, SFT = 41, RD = 192.0.2.1,1],  
[SI = 250, SFT = 44, RD = 192.0.2.4,5],  
[SI = 245, SFT = 1, RD = {SPI=23, SI=255, Rsv=0}],  
[SI = 245, SFT = 42, RD = 192.0.2.3,7]

Looping and jumping are described in [Section 6](#). This example shows an SFP that contains an explicit loop-back instruction that is presented as a choice within an SFP hop.

The first two hops in the path (SI = 255 and SI = 250) are normal. That is, the packets will be delivered to SFF1 and SFF4 in turn for execution of SFs of type 41 and 44 respectively.

The third hop (SI = 245) presents SFF4 with a choice of next hop. It can either forward the packets to SFF3 for an SF of type 42 (the second choice), or it can loop back.

The loop-back entry in the SFP for SI = 245 is indicated by the special purpose SFT value 1 ("Change Sequence"). Within this hop, the RD is interpreted as encoding the SPI and SI of the next hop (see [Section 6.1](#)). In this case the SPI is 23 which indicates that this is loop or branch: i.e., the next hop is on the same SFP. The SI is set



to 255: this is a higher number than the current SI (245) indicating a loop.

SFF4 must make a choice between these two next hops. Either the packets will be forwarded to SFF3 with the NSH SI decreased to 245 or looped back to SFF1 with the NSH SI reset to 255. This choice will be made according to local policy, information passed back by the local SFI, and details in the packets' metadata that are used to prevent infinite looping.

### **8.8. Example Branching in an SFP**

```
SFP10:  RD = 198.51.100.1,110, SPI = 24,  
        [SI = 254, SFT = 42, RD = 192.0.2.3,7],  
        [SI = 249, SFT = 43, RD = 192.0.2.2,2]  
  
SFP11:  RD = 198.51.100.1,111, SPI = 25,  
        [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
        [SI = 250, SFT = 1, RD = {SPI=24, SI=254, Rsv=0}]
```

Branching follows a similar procedure to that for looping (and jumping) as shown in [Section 8.7](#) however there are two SFPs involved.

SFP10 shows a normal path with packets forwarded to SFF3 and SFF2 for execution of service functions of type 42 and 43 respectively.

SFP11 starts as normal (SFF1 for an SF of type 41), but then SFF1 processes the next hop in the path and finds a "Change Sequence" Special Purpose SFT. The SFIR-RD field includes an SPI of 24 which indicates SFP10, not the current SFP. The SI in the SFIR-RD is 254, so SFF1 knows that it must set the SPI/SI in the NSH to 24/254 and send the packets to the appropriate SFF as advertised in the SFPR for SFP10 (that is, SFF3).

## **9. Security Considerations**

This document inherits all the security considerations discussed in the documents that specify BGP, the documents that specify BGP Multiprotocol Extensions, and the documents that define the attributes that are carried by BGP UPDATEs of the SFC AFI/SAFI. For more information look in [\[RFC4271\]](#), [\[RFC4760\]](#), and [\[I-D.ietf-idr-tunnel-encaps\]](#).

Service Function Chaining provides a significant attack opportunity: packets can be diverted from their normal paths through the network, can be made to execute unexpected functions, and the functions that



are instantiated in software can be subverted. However, this specification does not change the existence of Service Function Chaining and security issues specific to Service Function Chaining are covered in [[RFC7665](#)] and [[I-D.ietf-sfc-nsh](#)].

This document defines a control plane for Service Function Chaining. Clearly, this provides an attack vector for a Service Function Chaining system as an attack on this control plane could be used to make the system misbehave. Thus, the security of the BGP system is critically important to the security of the whole Service Function Chaining system.

## **[10.](#) IANA Considerations**

### **[10.1.](#) New BGP AF/SAFI**

IANA maintains a registry of "Address Family Numbers". IANA is requested to assign a new Address Family Number from the "Standards Action" range called "BGP SFC" (TBD1 in this document) with this document as a reference.

IANA maintains a registry of "Subsequent Address Family Identifiers (SAFI) Parameters". IANA is requested to assign a new SAFI value from the "Standards Action" range called "BGP SFC" (TBD2 in this document) with this document as a reference.

### **[10.2.](#) New BGP Path Attribute**

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters" with a subregistry of "BGP Path Attributes". IANA is requested to assign a new Path attribute called "SFP attribute" (TBD3 in this document) with this document as a reference.

### **[10.3.](#) New SFP Attribute TLVs Type Registry**

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters". IANA is request to create a new subregistry called the "SFP Attribute TLVs" registry.

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [[RFC5226](#)].

This document should be given as a reference for this registry.



The new registry should track:

- o Type
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:

Type	Name	Reference	Date
1	Association TLV	[This.I-D]	Date-to-be-set
2	Hop TLV	[This.I-D]	Date-to-be-set
3	SFT TLV	[This.I-D]	Date-to-be-set

#### **10.4. New SFP Association Type Registry**

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters". IANA is request to create a new subregistry called the "SFP Association Type" registry.

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [[RFC5226](#)].

This document should be given as a reference for this registry.

The new registry should track:

- o Association Type
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:





Association Type	Name	Reference	Date
1	Bidirectional SFP	[This.I-D]	Date-to-be-set

### **10.5. New Service Function Type Registry**

IANA is request to create a new top-level registry called "Service Function Chaining Service Function Types".

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 31 are to be assigned by "Standards Action" [[RFC5226](#)] and are referred to as the Special Purpose SFT values.
- o Other values (32 through 65534) are to be assigned according to the "First Come First Served" policy [[RFC5226](#)].

This document should be given as a reference for this registry.

The new registry should track:

- o Value
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:

Value	Name	Reference	Date
1	Change Sequence	[This.I-D]	Date-to-be-set

## **11. Contributors**



Stuart Mackie  
Juniper Networks

Email: [wsmackie@juniper.net](mailto:wsmackie@juniper.net)

Keyur Patel  
Arrcus, Inc.

Email: [keyur@arrcus.com](mailto:keyur@arrcus.com)

## **12. Acknowledgements**

Thanks to Tony Przygienda for helpful comments, and to Joel Halpern for discussions that improved this document.

## **13. References**

### **13.1. Normative References**

- [I-D.ietf-idr-tunnel-encaps]  
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", [draft-ietf-idr-tunnel-encaps-02](#) (work in progress), May 2016.
- [I-D.ietf-sfc-nsh]  
Quinn, P. and U. Elzur, "Network Service Header", [draft-ietf-sfc-nsh-10](#) (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), DOI 10.17487/RFC4271, January 2006, <<http://www.rfc-editor.org/info/rfc4271>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", [RFC 4364](#), DOI 10.17487/RFC4364, February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", [RFC 4760](#), DOI 10.17487/RFC4760, January 2007, <<http://www.rfc-editor.org/info/rfc4760>>.



- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

### **13.2. Informative References**

- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", [RFC 6790](#), DOI 10.17487/RFC6790, November 2012, <<http://www.rfc-editor.org/info/rfc6790>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", [RFC 7498](#), DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", [RFC 7510](#), DOI 10.17487/RFC7510, April 2015, <<http://www.rfc-editor.org/info/rfc7510>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", [RFC 7665](#), DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

#### Authors' Addresses

Adrian Farrel  
Juniper Networks

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

John Drake  
Juniper Networks

Email: [jdrake@juniper.net](mailto:jdrake@juniper.net)

Eric Rosen  
Juniper Networks

Email: [erosen@juniper.net](mailto:erosen@juniper.net)



Jim Uttaro  
AT&T

Email: [ju1738@att.com](mailto:ju1738@att.com)

Luay Jalil  
Verizon

Email: [luay.jalil@verizon.com](mailto:luay.jalil@verizon.com)