BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 21, 2020

A. Farrel
Old Dog Consulting
J. Drake
E. Rosen
Juniper Networks
J. Uttaro
AT&T
L. Jalil
Verizon

August 20, 2019

BGP Control Plane for NSH SFC
draft-ietf-bess-nsh-bgp-control-plane-12

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 defined in RFC 8300.

This document adopts the SFC architecture described in RFC 7665.

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 https://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 February 21, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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

1. Introduction 3
1.1. Requirements Language 4
1.2. Terminology 5
2. Overview 6
2.1. Overview of Service Function Chaining 6
2.2. Control Plane Overview 7
3. BGP SFC Routes 11
3.1. Service Function Instance Route (SFIR) 12
3.1.1. SFI Pool Identifier Extended Community 13
3.1.2. MPLS Mixed Swapping/Stacking Extended Community . . . 14
3.2. Service Function Path Route (SFPR) 14
3.2.1. The SFP Attribute 15
3.2.2. General Rules For The SFP Attribute 21
4. Mode of Operation 22
4.1. Route Targets 22
4.2. Service Function Instance Routes 22
4.3. Service Function Path Routes 22
4.4. Classifier Operation 24
4.5. Service Function Forwarder Operation 25
4.5.1. Processing With 'Gaps' in the SI Sequence 26
5. Selection within Service Function Paths 27
6. Looping, Jumping, and Branching 29
6.1. Protocol Control of Looping, Jumping, and Branching . . . 29
6.2. Implications for Forwarding State 30
7. Advanced Topics 31
7.1. Correlating Service Function Path Instances 31
7.2. Considerations for Stateful Service Functions 32
7.3. VPN Considerations and Private Service Functions 33
7.4. Flow Spec for SFC Classifiers 33
7.5. Choice of Data Plane SPI/SI Representation 34
7.5.1. MPLS Representation of the SPI/SI 36

7.6. MPLS Label Swapping/Stacking Operation 36
7.7. Support for MPLS-Encapsulated NSH Packets 36
8. Examples 37
8.1. Example Explicit SFP With No Choices 38
8.2. Example SFP With Choice of SFIs 39
8.3. Example SFP With Open Choice of SFIs 40
8.4. Example SFP With Choice of SFTs 40
8.5. Example Correlated Bidirectional SFPs 41
8.6. Example Correlated Asymmetrical Bidirectional SFPs . . . 41
8.7. Example Looping in an SFP 42
8.8. Example Branching in an SFP 43
8.9. Examples of SFPs with Stateful Service Functions 43
8.9.1. Forward and Reverse Choice Made at the SFF 44
8.9.2. Parallel End-to-End SFPs with Shared SFF 45
8.9.3. Parallel End-to-End SFPs with Separate SFFs 47
8.9.4. Parallel SFPs Downstream of the Choice 49
9. Security Considerations 52
10. IANA Considerations 53
10.1. New BGP AF/SAFI 53
10.2. New BGP Path Attribute 53
10.3. New SFP Attribute TLVs Type Registry 53
10.4. New SFP Association Type Registry 54
10.5. New Service Function Type Registry 55
10.6. New Generic Transitive Experimental Use Extended Community Sub-Types 56
10.7. New BGP Transitive Extended Community Types 56
10.8. SPI/SI Representation 56
11. Contributors 56
12. Acknowledgements 57
13. References 57
13.1. Normative References 57
13.2. Informative References 58
Authors' Addresses 59

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., WAN and application accelerators, Deep Packet Inspection (DPI) engines, firewalls, TCP optimizers, 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.

Historically, if a packet needed to travel through a particular service chain, the nodes hosting the service functions of that chain were placed in the network topology in such a way that the packet

could not 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 limited 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 raised 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 describes 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 that satisfy 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 [RFC8300].

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

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

This document uses the following terms from [RFC7665]:

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

Additionally, this document uses the following terms from [RFC8300]:

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

This document introduces the following terms:

- Service Function Instance Route (SFIR). A new BGP Route Type advertised by the node that hosts an SFI to describe the SFI and to announce the way to forward a packet to the node through the underlay network.
- Service Function Overlay Network. The logical network comprised of Classifiers, SFFs, and SFIs that are connected by paths or tunnels through underlay transport networks.
- Service Function Path Route (SFPR). A new BGP Route Type originated by Controllers to advertise the details of each SFP.
- Service Function Type (SFT). An indication of the function and features of an SFI.

2. Overview

2.1. Overview of Service Function Chaining

In [RFC8300] 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 functional element, 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 prepends a Network Services Header (NSH) [RFC8300] containing the assigned SPI/SI to that packet. 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 that the node prepending the NSH also provide some form of entropy indicator that can be used in the underlay network. How this indicator is generated and supplied, and how an SFF generates a new entropy indicator when it forwards a packet to the next SFF are out of scope of this document.

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. The SFI has no knowledge of the SFP.

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:

- 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).
- 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.

As described in [RFC8300], an unknown or invalid SPI is treated as an error and the SFF drops the packet. Such errors should be logged, and such logs are subject to rate limits.

An SFF receiving an SI that is unknown in the context of the SPI can reduce the value to the next meaningful SI value in the SFP indicated by the SPI. If no such value exists or if the SFF does not support this function, the SFF drops the packet and should log the event: such logs are also subject to rate limits.

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.

[RFC8300] makes it clear that the intended scope is for use within a single provider's operational domain.

2.2. Control Plane Overview

To accomplish the function described in Section 2.1, this document introduces 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.

This document also 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.

- The SFIR is advertised by the node hosting the service function instance. The SFIR describes a particular instance of a particular Service Function (i.e., an SFI) and the way to forward a packet to it through the underlay network, i.e., IP address and encapsulation information.

- 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.

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 shows a reference model for the SFC architecture. There are four SFFs (SFF-1 through SFF-4) connected by tunnels across the underlay network. Packets arrive at a Classifier and are channelled along SFPs to destinations reachable through SFF-4.

SFF-1 and SFF-4 each have one instance of one SF attached (SFa and SFe). SFF-2 has two types of SF attached: there is one instance of

one (SFc), and three instances of the other (SFb). SFF-3 has just one instance of an SF (SFd), but it in this case the type of SFd is the same type as SFb (SFTx).

This figure demonstrates how load balancing can be achieved by creating several SFPs that satisfy the same SFC. Suppose an SFC needs to include SFa, an SF of type SFTx, and SFc. A number of SFPs can be constructed using any instance of SFb or using SFd. Load balancing may be applied at two places:

- The Classifier may distribute different flows onto different SFPs to share the load in the network and across SFIs.

- SFF-2 may distribute different flows (on the same SFP) to different instances of SFb to share the processing load.

Note that, for convenience and clarity, Figure 1 shows only a few tunnels between SFFs. There could be a full mesh of such tunnels, or more likely, a selection of tunnels connecting key SFFs to enable the construction of SFPs and to balance load and traffic in the network.

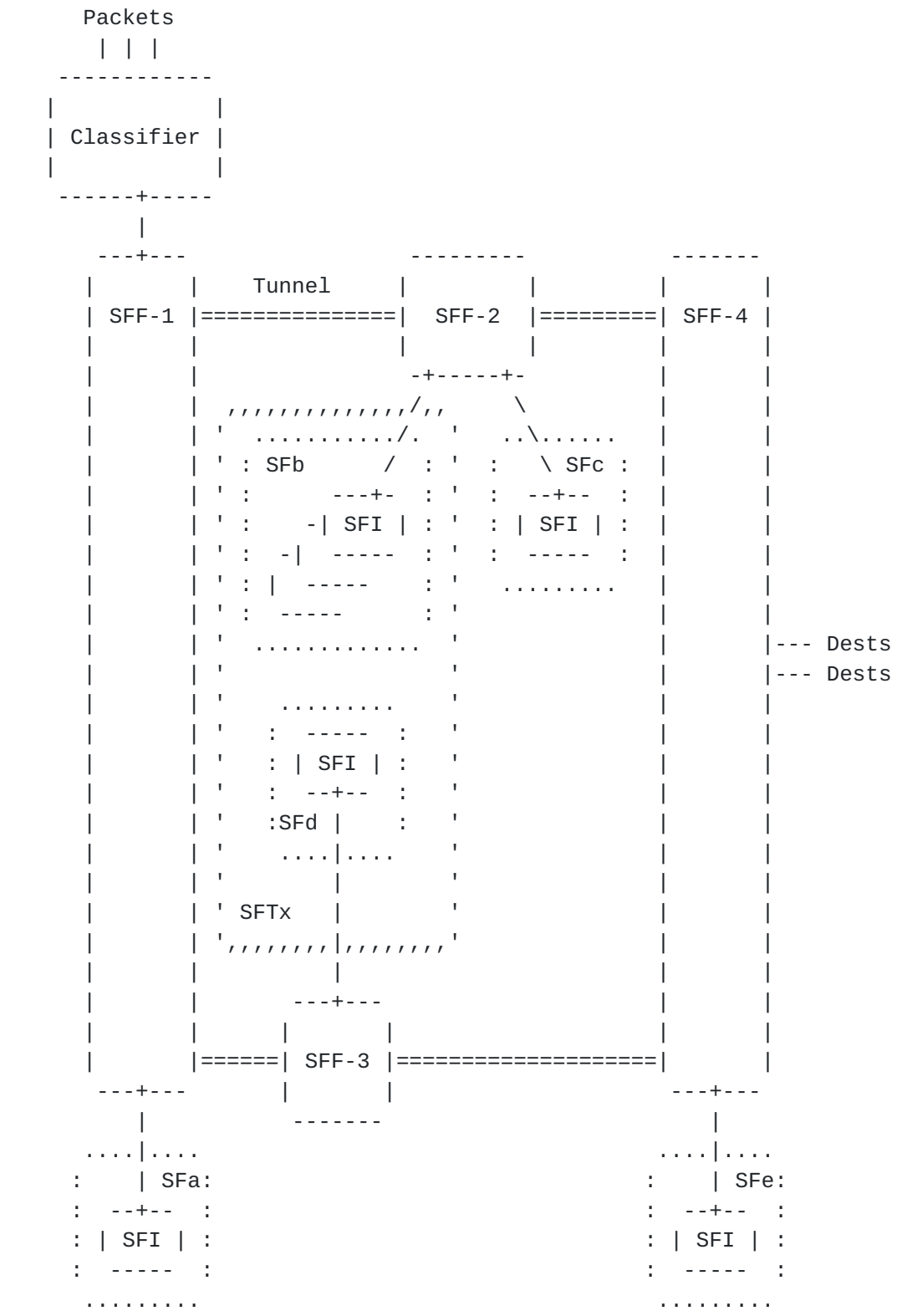


Figure 1: The SFC Architecture Reference Model

As previously noted, [RFC8300] makes it clear that the mechanisms it defines are intended for use within a single provider's operational domain. This reduces the requirements on the control plane function.

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.

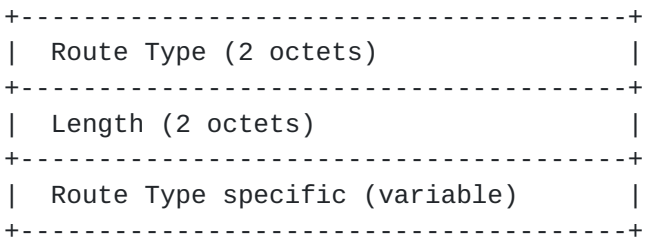


Figure 2: The Format of the SFC NLRI

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.

The nexthop field of the MP_REACH_NLRI attribute of the SFC NLRI MUST be set to loopback address of the advertising SFF.

3.1. Service Function Instance Route (SFIR)

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

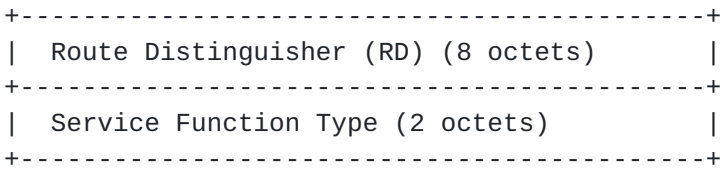


Figure 3: SFIR Route Type specific NLRI

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 the functions/features of service function can offer, 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 type of SF that it hosts, and it may advertise an SFIR for each instance of each type of SF. The minimal advertisement allows construction of valid SFPs and leaves the selection of SFIs to the local SFF; the detailed advertisement may have scaling concerns, but allows a Controller that constructs an SFP to make an explicit choice of SFI.

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 MUST 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.

Note that the Tunnel Encapsulation attribute MUST contain sufficient information to allow the advertising SFF to identify the overlay or VPN network which a received packet is transiting. This is because the [SPI, SI] in a received packet is specific to a particular overlay or VPN network.

3.1.1. SFI Pool Identifier Extended Community

This document defines a new transitive extended community of type TBD6 with Sub-Type 0x00 called the SFI Pool Identifier extended community. It MAY be included in SFIR advertisements, and is used to indicate the identity of a pool of SFIRs to which an SFIR belongs. Since an SFIR may be a member of multiple pools, multiple of these extended communities may be present on a single SFIR advertisement.

SFIR pools allow SFIRs to be grouped for any purpose. Possible uses include control plane scalability and stability. A pool identifier may be included in an SFPR to indicate a set of SFIs that are acceptable at a specific point on an SFP (see Section 3.2.1.3 and Section 4.3).

The SFI Pool Identifier extended community is encoded in 8 octets as shown in Figure 4.

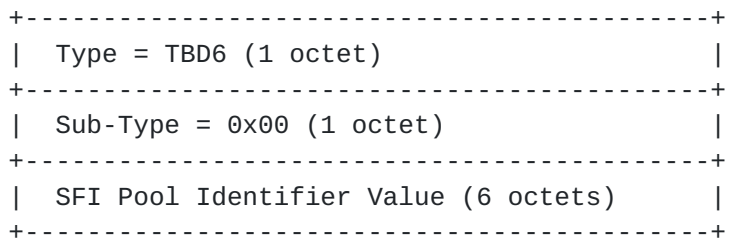


Figure 4: The SFI Pool Identifier Extended Community

The SFI Pool Identifier Value is encoded in a 6 octet field in network byte order, and is a globally unique value. This means that pool identifiers need to be centrally managed, which is consistent with the assignment of SFIs to pools.

3.1.2. MPLS Mixed Swapping/Stacking Extended Community

This document defines a new transitive extended community of type TBD7 with Sub-Type 0x00 called the MPLS Mixed Swapping/Stacking Labels. The community is encoded as shown in Figure 5. It contains a pair of MPLS labels: an SFC Context Label and an SF Label as described in [RFC8595]. Each label is 20 bits encoded in a 3-octet (24 bit) field with 4 trailing bits that MUST be set to zero.

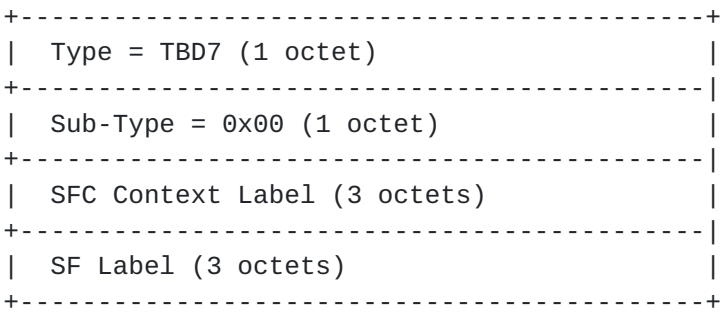


Figure 5: The MPLS Mixed Swapping/Stacking Extended Community

Note that it is assumed that each SFF has one or more globally unique SFC Context Labels and that the context label space and the SPI address space are disjoint.

If an SFF supports SFP Traversal with an MPLS Label Stack it MUST include this extended community with the SFIRs that it advertises.

See Section 7.6 for a description of how this extended community is used.

3.2. Service Function Path Route (SFPR)

Figure 6 shows the Route Type specific NLRI of the SFPR.

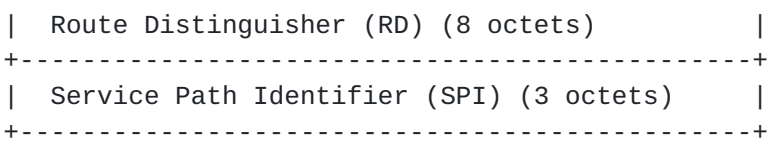


Figure 6: SFPR Route Type Specific NLRI

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 SFPRs are originated from different Controllers they MUST have different RDs. Additionally, SFPRs 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 SFPRs.

The Service Path Identifier is defined in [RFC8300] 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 Optional Transitive 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:

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

- Type: A single octet indicating the type of the SFP attribute TLV. Values are taken from the registry described in Section 10.3.
- Length: A two octet field indicating the length of the data following the Length field counted in octets.
- 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.

- 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).

- 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.

- 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).

Section 6 of [RFC4271] describes the handling of malformed BGP attributes, or those that are in error in some way. [RFC7606] revises BGP error handling specifically for the for UPDATE message, provides guidelines for the authors of documents defining new attributes, and revises the error handling procedures for a number of existing attributes. This document introduces the SFP attribute and so defines error handling as follows:

- When parsing a message, an unknown Attribute Type code or a length that suggests that the attribute is longer than the remaining message is treated as a malformed message and the "treat-as-withdraw" approach used as per [RFC7606].

- When parsing a message that contains an SFP attribute, the following cases constitute errors:

  1. Optional bit is set to 0 in SFP attribute.

  2. Transitive bit is set to 0 in SFP attribute.

  3. Unknown TLV type field found in SFP attribute.

  4. TLV length that suggests the TLV extends beyond the end of the SFP attribute.

  5. Association TLV contains an unknown SFPR-RD.

  6. No Hop TLV found in the SFP attribute.

7. No sub-TLV found in a Hop TLV.

8. Unknown SFIR-RD found in a Hop TLV.

o The errors listed above are treated as follows:

1., 2., 6., 7.: The attribute MUST be treated as malformed and the "treat-as-withdraw" approach used as per [RFC7606].

3.: Unknown TLVs SHOULD be ignored, and message processing SHOULD continue.

4.: Treated as a malformed message and the "treat-as-withdraw" approach used as per [RFC7606]

5., 8.: The absence of an RD with which to correlate is nothing more than a soft error. The receiver SHOULD store the information from the SFP attribute until a corresponding advertisement is received.

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 7

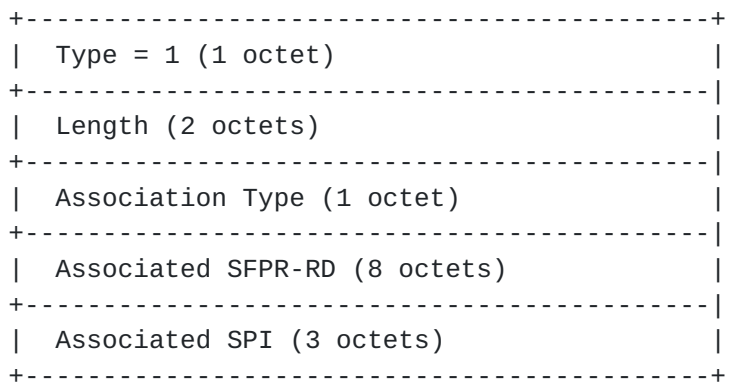


Figure 7: The Format of the Association TLV

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 the associated SFP as advertised in an SFPR.

The Associated SPI contains the SPI of the associated SFP as advertised in an SFPR.

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 7.1.

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 8. At least one Hop TLV MUST be present in an SFP attribute.

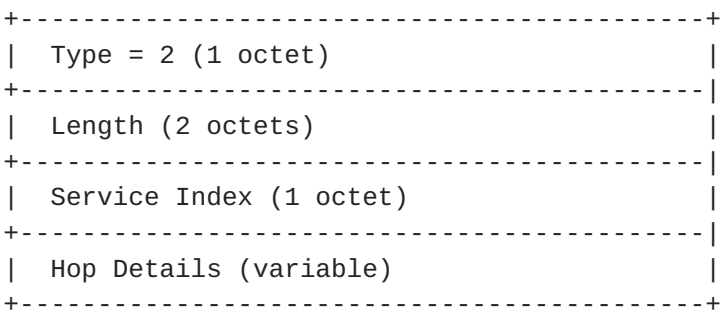


Figure 8: The Format of the Hop TLV

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 [RFC8300] 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 field consists of a sequence of one or more sub-TLVs.

Each hop of the SFP may demand that a specific type of SF is executed, and that type is indicated in sub-TLVs of the Hop TLV. At least one sub-TLV MUST be present. This provides a list of which types of SF are acceptable at a specific hop, and for each type it allows a degree of control to be imposed on the choice of SFIs of that particular type.

If no Hop TLV is present in an SFP Attribute, it is a malformed attribute

3.2.1.3. The SFT TLV

The SFT TLV MAY be included in the list of sub-TLVs of the Hop TLV. The format of the SFT TLV is shown in Figure 9. The TLV contains a list of SFIR-RD values each taken from the advertisement of an SFI. Together they form a list of acceptable SFIs of the indicated type.

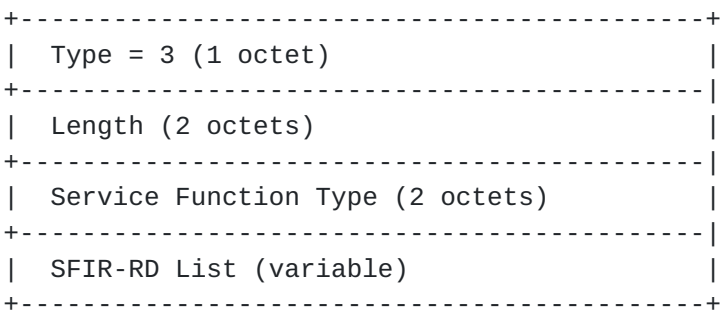


Figure 9: The Format of the SFT TLV

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 value indicates the category (type) of SF that is to be executed at this hop. The types are as advertised for the SFs supported by the SFFs 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 hop description is further qualified beyond the specification of the SFTs by listing, for each SFT in each hop, the SFIs that may be used at the hop. The SFIs are identified using the SFIR-RDs from the advertisements of the SFIs in the SFIRs. Note that if the list contains one or more SFI Pool Identifiers, then for each the SFIR-RD list is effectively expanded to include the SFIR-RD of each SFIR advertised with that SFI Pool Identifier. An SFIR-RD of value zero has special meaning as described in Section 5. Each entry in the list is eight octets long, and the number of entries in the list can be deduced from the value of the Length field.

3.2.1.4. MPLS Swapping/Stacking TLV

The MPLS Swapping/Stacking TLV (Type value 4) is a zero length sub-TLV that is OPTIONAL in the Hop TLV and is used when the data representation is MPLS (see Section 7.5). When present it indicates to the Classifier imposing an MPLS label stack that the current hop is to use an {SFC Context Label, SF label} rather than an {SPI, SF} label pair. See Section 7.6 for more details.

3.2.1.5. SFP Traversal With MPLS Label Stack TLV

The SFP Traversal With MPLS Label Stack TLV (Type value 5) is a zero length sub-TLV that can be carried in the SFP Attribute and indicates to the Classifier and the SFFs on the SFP that an MPLS labels stack with label swapping/stacking is to be used for packets traversing the SFP. All of the SFF specified at each the SFP's hops MUST have advertised an MPLS Mixed Swapping/Stacking Extended Community (see Section 3.1.2) for the SFP to be considered usable.

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:

- 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.

- 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.

- 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) will usually 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:

- The RT as just described.
- A Service Function Type (SFT) that is the type of service function that is provided (such as "firewall").
- 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 within a single VPN instance supported by the underlay network.

The SFPR advertisement comprises:

- An RT as described in Section 4.1.
- A tuple that identifies the SFPR
  - An RD that identifies an advertisement of an SFPR.
  - The SPI that uniquely identifies this path within the VPN instance distinguished by the RD. This SPI also appears in the NSH.
- 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.
- 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 [RFC8300].
- 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. Gaps MAY appear in the sequence as described in Section 4.5.1. Malformed SFPRs MUST be discarded and MUST cause any previous instance of the SFPR (same SFPR-RD and SPI) to be discarded.

Note that if the SFIR-RD list in an SFT TLV contains one or more SFI Pool identifiers, then in the above expression, 'p' is the sum of the number of individual SFIR-RD values and the sum for each SFI Pool Identifier of the number of SFIRs advertised with that SFI Pool Identifier. I.e., the list of SFIR-RD values is effectively expanded to include the SFIR-RD of each SFIR advertised with each SFI Pool Identifier in the SFIR-RD list.

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 component 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 with the SPI of the selected SFP and the SI of its first hop.

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 uses a specific reserved SFT value at a given hop of an SFPR and is described in Section 6.1.

4.5.1. Processing With 'Gaps' in the SI Sequence

The behavior of an SF as described in [RFC8300] is to decrement the value of the SI field in the NSH by one before returning a packet to the local SFF for further processing. This means that there is a good reason to assume that the SFP is composed of a series of SFs each indicated by an SI value one less than the previous.

However, there is an advantage to having non-successive SIs in an SPI. Consider the case where an SPI needs to be modified by the insertion or removal of an SF. In the latter case this would lead to a "gap" in the sequence of SIs, and in the former case, this could only be achieved if a gap already existed into which the new SF with its new SI value could be inserted. Otherwise, all "downstream" SFs would need to be renumbered.

Now, of course, such renumbering could be performed, but would lead to a significant disruption to the SFC as all the SFFs along the SFP were "reprogrammed". Thus, to achieve dynamic modification of an SFP (and even, in-service modification) it is desirable to be able to make these modifications without changing the SIs of the elements that were present before the modification. This will produce much more consistent/predictable behavior during the convergence period where otherwise the change would need to be fully propagated.

Another approach says that any change to an SFP simply creates a new SFP that can be assigned a new SPI. All that would be needed would be to give a new instruction to the Classifier and traffic would be switched to the new SFP that contains the new set of SFs. This approach is practical, but neglects to consider that the SFP may be referenced by other SFPs (through "branch" instructions) and used by many Classifiers. In those cases the corresponding configuration resulting from a change in SPI may have wide ripples and give scope for errors that are hard to trace.

Therefore, while this document requires that the SI values in an SFP are monotonic decreasing, it makes no assumption that the SI values are sequential. Configuration tools may apply that rule, but they are not required to. To support this, an SFF SHOULD process as follows when it receives a packet:

- If the SI indicates a known entry in the SFP, the SFF MUST process the packet as normal, looking up the SI and determining to which local SFI to deliver the packet.

- If the SI does not match an entry in the SFP, the SFF MUST reduce the SI value to the next (smaller) value present in the SFP and process the packet using that SI.

- If there is no smaller SI (i.e., if the end of the SFP has been reached) the SFF MUST treat the SI value as invalid as described in [RFC8300].

SFF implementations MAY choose to only support contiguous SI values in an SFP. Such an implementation will not support receiving an SI value that is not present in the SFP and will discard the packets as described in [RFC8300].

5. Selection within 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.

The choice be may offered for load balancing across multiple SFIs, or for discrimination between different actions necessary at a specific hop in the SFP. Different SFT values may exist at a given hop in an SFP to support several cases:

- There may be multiple instances of similar service functions that are distinguished by different SFT values. For example, firewalls made by vendor A and vendor B may need to be identified by different SFT values because, while they have similar functionality, their behavior is not identical. Then, some SFPs may limit the choice of SF at a given hop by specifying the SFT for vendor A, but other SFPs might not need to control which vendor's SF is used and so can indicate that either SFT can be used.

- There may be an obvious branch needed in an SFP such as the processing after a firewall where admitted packets continue along the SFP, but suspect packets are diverted to a "penalty box". In this case, the next hop in the SFP will be indicated with two different SFT values.

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 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.

Thus, at any point in time when an SFF selects its next hop, it chooses from the intersection of the set of next hop RDs contained in

the SFPR and the RDs contained in its local set of SFIRs. If the intersection is null, the SFPR is unusable. Similarly, when this condition obtains the originator of the SFPR should either withdraw the SFPR or re-advertise it with a new set of RDs for the affected hop.

6. Looping, Jumping, and Branching

As described in Section 2 an SFI or an SFF may cause a packet 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 [RFC8300] 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 to 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 anther 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. 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 [RFC8300]) 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.

7.2. Considerations for Stateful Service Functions

Some service functions are stateful. That means that they build and maintain state derived from configuration or from the packet flows that they handle. In such cases it can be important or necessary that all packets from a flow continue to traverse the same instance of a service function so that the state can be leveraged and does not need to be regenerated.

In the case of bidirectional SFPs, it may be necessary to traverse the same instances of a stateful service function in both directions. A firewall is a good example of such a service function.

This issue becomes a concern where there are multiple parallel instances of a service function and a determination of which one to use could normally be left to the SFF as a load-balancing or local policy choice.

For the forward direction SFP, the concern is that the same choice of service function is made for all packets of a flow under normal network conditions. It may be possible to guarantee that the load balancing functions applied in the SFFs are stable and repeatable, but a controller that constructs SFPs might not want to trust to this. The controller can, in these cases, build a number of more specific SFPs each traversing a specific instance of the stateful SFs. In this case, the load balancing choice can be left up to the Classifier. Thus the Classifier selects which instance of a stateful SF is used by a particular flow by selecting the SFP that the flow uses.

For bidirectional SFPs where the same instance of a stateful SF must be traversed in both directions, it is not enough to leave the choice of service function instance as a local choice even if the load balancing is stable because coordination would be required between the decision points in the forward and reverse directions and this may be hard to achieve in all cases except where it is the same SFF that makes the choice in both directions.

Note that this approach necessarily increases the amount of SFP state in the network (i.e., there are more SFPs). It is possible to mitigate this effect by careful construction of SFPs built from a concatenation of other SFPs.

Section 8.9 includes some simple examples of SFPs for stateful service functions.

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.

7.4. Flow Spec for SFC Classifiers

[RFC5575] and [I-D.ietf-idr-rfc5575bis] define a set of BGP routes that can be used to identify the packets in a given flow using fields in the header of each packet, and a set of actions, encoded as extended communities, that can be used to disposition those packets. This document enables the use of these mechanisms by SFC Classifiers by defining a new action extended community called "Flow Spec for SFC Classifiers" identified by the value TBD4. Note that other action extended communities MUST NOT be present at the same time: the inclusion of the "Flow Spec for SFC Classifiers" action extended community along with any other action MUST be treated as an error which SHOULD result in the Flow Specification UPDATE message being handled as Treat-as-withdraw according to [RFC7606] Section 2.

This extended community is encoded as an 8-octet value, as shown in Figure 10.

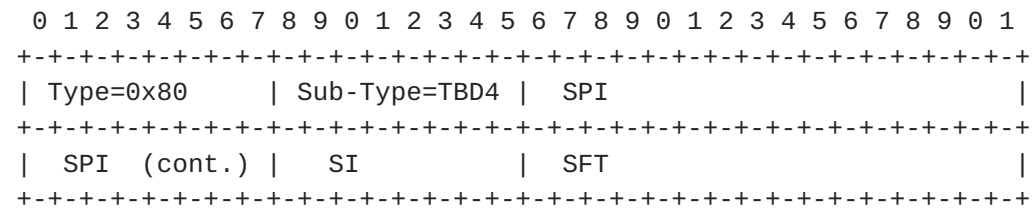


Figure 10: The Format of the Flow Spec for SFC Classifiers Extended Community

The extended community contains the Service Path Identifier (SPI), Service Index (SI), and Service Function Type (SFT) as defined elsewhere in this document. Thus, each action extended community defines the entry point (not necessarily the first hop) into a

specific service function path. This allows, for example, different flows to enter the same service function path at different points.

Note that a given Flow Spec update according to [RFC5575] and [I-D.ietf-idr-rfc5575bis] may include multiple of these action extended communities, and that if a given action extended community does not contain an installed SFPR with the specified {SPI, SI, SFT} it MUST NOT be used for dispositioning the packets of the specified flow.

The normal case of packet classification for SFC will see a packet enter the SFP at its first hop. In this case the SI in the extended community is superfluous and the SFT may also be unnecessary. To allow these cases to be handled, a special meaning is assigned to a Service Index of zero (not a valid value) and an SFT of zero (a reserved value in the registry - see Section 10.5).

- If an SFC Classifiers Extended Community is received with SI = 0 then it means that the first hop of the SFP indicated by the SPI MUST be used.

- If an SFC Classifiers Extended Community is received with SFT = 0 then there are two sub-cases:

  * If there is a choice of SFT in the hop indicated by the value of the SI (including SI = 0) then SFT = 0 means there is a free choice according to local policy of which SFT to use).

  * If there is no choice of SFT in the hop indicated by the value of SI, then SFT = 0 means that the value of the SFT at that hop as indicated in the SPFR for the indicated SPI MUST be used.

One of the filters that the Flow Spec may describe is the VPN to which the traffic belongs. Additionally, note that to put the indicated SPI into context when multiple SFC overlays are present in one network, each FlowSpec update MUST be tagged with the route target of the overlay or VPN network for which it is intended.

7.5. Choice of Data Plane SPI/SI Representation

This document ties together the control and data planes of an SFC overlay network through the use of the SPI/SI which is nominally carried in the NSH of a given packet. However, in order to handle situations in which the NSH is not ubiquitously deployed, it is also possible to use alternative data plane representations of the SPI/SI by carrying the identical semantics in other protocol fields such as MPLS labels [RFC8595].

This document defines a new sub-TLV for the Tunnel Encapsulation attribute, the SPI/SI Representation sub-TLV of type TBD5. This sub-TLV MAY be present in each Tunnel TLV contained in a Tunnel Encapsulation attribute when the attribute is carried by an SFIR. The value field of this sub-TLV is a two octet field of flags, each of which describes how the originating SFF expects to see the SPI/SI represented in the data plane for packets carried in the tunnels described by the Tunnel TLV.

The following bits are defined by this document:

Bit 0: If this bit is set the NSH is to be used to carry the SPI/SI in the data plane.

Bit 1: If this bit is set two labels in an MPLS label stack are to be used as described in Section 7.5.1.

If a given Tunnel TLV does not contain an SPI/SI Representation sub-TLV then it MUST be processed as if such a sub-TLV is present with Bit 0 set and no other bits set. That is, the absence of the sub-TLV SHALL be interpreted to mean that the NSH is to be used.

If a given Tunnel TLV contains an SPI/SI Representation sub-TLV with value field that has no flag set then the tunnel indicated by the Tunnel TLV MUST NOT be used for forwarding SFC packets. If a given Tunnel TLV contains an SPI/SI Representation sub-TLV with both bit 0 and bit 1 set then the tunnel indicated by the Tunnel TLV MUST NOT be used for forwarding SFC packets. The meaning and rules for presence of other bits is to be defined in future documents, but implementations of this specification MUST set other bits to zero and ignore them on receipt.

If a given Tunnel TLV contains more than one SPI/SI Representation sub-TLV then the first one MUST be considered and subsequent instances MUST be ignored.

Note that the MPLS representation of the logical NSH may be used even if the tunnel is not an MPLS tunnel. Conversely, MPLS tunnels may be used to carry other encodings of the logical NSH (specifically, the NSH itself). It is a requirement that both ends of a tunnel over the underlay network know that the tunnel is used for SFC and know what form of NSH representation is used. The signaling mechanism described here allows coordination of this information.

7.5.1. MPLS Representation of the SPI/SI

If bit 1 is set in the in the SPI/SI Representation sub-TLV then labels in the MPLS label stack are used to indicate SFC forwarding and processing instructions to achieve the semantics of a logical NSH. The label stack is encoded as shown in [RFC8595].

7.6. MPLS Label Swapping/Stacking Operation

When a classifier constructs an MPLS label stack for an SFP it starts with that SFP' last hop. If the last hop requires an {SPI, SI} label pair for label swapping, it pushes the SI (set to the SI value of the last hop) and the SFP's SPI onto the MPLS label stack. If the last hop requires a {context label, SFI label} label pair for label stacking it selects a specific SFIR and pushes that SFIR's SFI label and context label onto the MPLS label stack.

The classifier then moves sequentially back through the SFP one hop at a time. For each hop, if the hop requires an {SPI, SI]} and there is an {SPI, SI} at the top of the MPLS label stack, the SI is set to the SI value of the current hop. If there is not an {SPI, SI} at the top of the MPLS label stack, it pushes the SI (set to the SI value of the current hop) and the SFP's SPI onto the MPLS label stack.

If the hop requires a {context label, SFI label}, it selects a specific SFIR and pushes that SFIR's SFI label and context label onto the MPLS label stack.

7.7. Support for MPLS-Encapsulated NSH Packets

[RFC8596] describes how to transport SFC packets using the NSH over an MPLS transport network. Signaling MPLS encapsulation of SFC packets using the NSH is also supported by this document by using the "BGP Tunnel Encapsulation Attribute Sub-TLV" with the codepoint 10 (representing "MPLS Label Stack") from the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry defined in [I-D.ietf-idr-tunnel-encaps], and also using the "SFP Traversal With MPLS Label Stack TLV" and the "SPI/SI Representation sub-TLV" with bit 0 set and bit 1 cleared.

In this case the MPLS label stack constructed by the SFF to forward a packet to the next SFF on the SFP will consist of the labels needed to reach that SFF, and if label stacking is used it will also include the labels advertised in the MPLS Label Stack sub-TLV and the labels remaining in the stack needed to traverse the remainder of the SFP.

8. Examples

Assume we have a service function overlay network with four SFFs (SFF1, SFF3, 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 11.

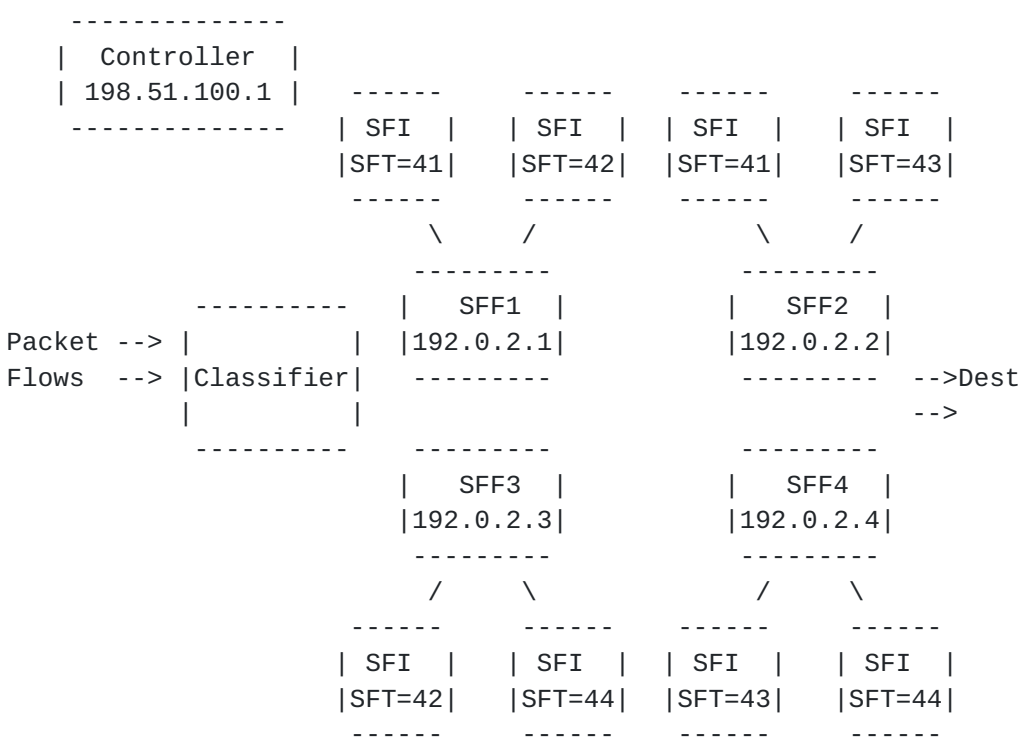


Figure 11: Example Service Function Overlay Network

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 = 192.0.2.1:1],
       [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 7.1.

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 SFPR 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).

8.9. Examples of SFPs with Stateful Service Functions

This section provides some examples to demonstrate establishing SFPs when there is a choice of service functions at a particular hop, and where consistency of choice is required in both directions. The scenarios that give rise to this requirement are discussed in Section 7.2.

8.9.1. Forward and Reverse Choice Made at the SFF

Consider the topology shown in Figure 12. There are three SFFs arranged neatly in a line, and the middle one (SFF2) supports three SFIs all of SFT 42. These three instances can be used by SFF2 to load balance so that no one instance is swamped.

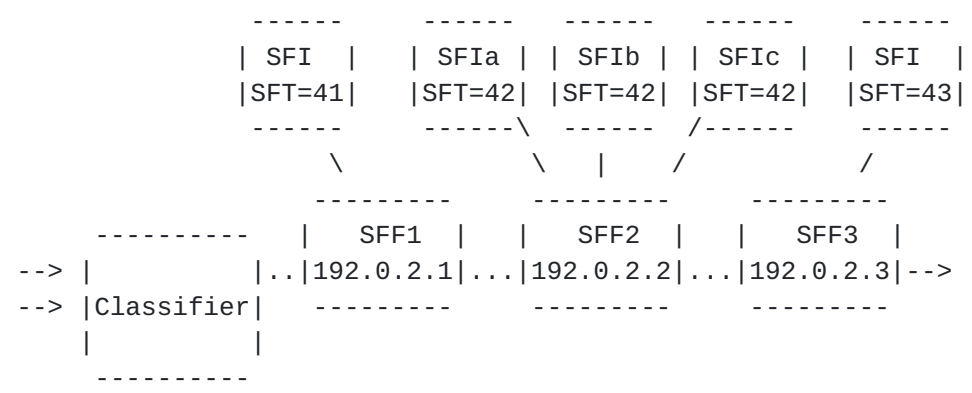


Figure 12: Example Where Choice is Made at the SFF

This leads to the following SFIRs being advertised.

```
RD = 192.0.2.1:11, SFT = 41
RD = 192.0.2.2:11, SFT = 42  (for SFIa)
RD = 192.0.2.2:12, SFT = 42  (for SFIb)
RD = 192.0.2.2:13, SFT = 42  (for SFIc)
RD = 192.0.2.3:11, SFT = 43
```

The controller can create a single forward SFP giving SFF2 the choice of which SFI to use to provide function of SFT 42 as follows. The load-balancing choice between the three available SFIs is assumed to be within the capabilities of the SFF and if the SFs are stateful it is assumed that the SFF knows this and arranges load balancing in a stable, flow-dependent way.

```
SFP12:  RD = 198.51.100.1:112, SPI = 26,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:113, Assoc-SPI = 27,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, {RD = 192.0.2.2:11,
                             192.0.2.2:12,
                             192.0.2.2:13 }],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]
```

The reverse SFP in this case may also be created as shown below using association with the forward SFP and giving the load-balancing choice to SFF2. This is safe, even in the case that the SFs of type 42 are stateful because SFF2 is doing the load balancing in both directions and can apply the same algorithm to ensure that packets associated with the same flow use the same SFI regardless of the direction of travel.

How an SFF knows that an attached SFI is stateful is is out of scope of this document. It is assumed that this will form part of the process by which SFIs are registered as local to SFFs. Section 7.2 provides additional observations about the coordination of the use of stateful SFIs in the case of bidirectional SFPs.

In general, the problems of load balancing and the selection of the same SFIs in both directions of a bidirectional SPF can be addressed by using sufficiently precisely specified SFPs (specifying the exact SFIs to use) and suitable programming of the Classifiers at each end of the SFPs to make sure that the matching pair of SFPs are used.

```
SFP13:  RD = 198.51.100.1:113, SPI = 27,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:112, Assoc-SPI = 26,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, {RD = 192.0.2.2:11,
                             192.0.2.2:12,
                             192.0.2.2:13 }],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]
```

8.9.2. Parallel End-to-End SFPs with Shared SFF

The mechanism described in Section 8.9.1 might not be desirable because of the functional assumptions it places on SFF2 to be able to load balance with suitable flow identification, stability, and equality in both directions. Instead, it may be desirable to place the responsibility for flow classification in the Classifier and let it determine load balancing with the implied choice of SFIs.

Consider the network graph as shown in Figure 12 and with the same set of SFIRs as listed in Section 8.9.1. In this case the controller could specify three forward SFPs with their corresponding associated reverse SFPs. Each bidirectional pair of SFPs uses a different SFI for the SF of type 42. The controller can instruct the Classifier how to place traffic on the three bidirectional SFPs, or can treat them as a group leaving the Classifier responsible for balancing the load.

```
SFP14:  RD = 198.51.100.1:114, SPI = 28,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:117, Assoc-SPI = 31,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:11],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP15:  RD = 198.51.100.1:115, SPI = 29,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:118, Assoc-SPI = 32,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:12],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP16:  RD = 198.51.100.1:116, SPI = 30,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:119, Assoc-SPI = 33,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:13],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP17:  RD = 198.51.100.1:117, SPI = 31,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:114, Assoc-SPI = 28,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:11],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]

SFP18:  RD = 198.51.100.1:118, SPI = 32,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:115, Assoc-SPI = 29,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:12],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]

SFP19:  RD = 198.51.100.1:119, SPI = 33,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:116, Assoc-SPI = 30,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:13],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]
```

8.9.3. Parallel End-to-End SFPs with Separate SFFs

While the examples in Section 8.9.1 and Section 8.9.2 place the choice of SFI as subtended from the same SFF, it is also possible that the SFIs are each subtended from a different SFF as shown in Figure 13. In this case it is harder to coordinate the choices for forward and reverse paths without some form of coordination between SFF1 and SFF3. Therefore it would be normal to consider end-to-end parallel SFPs as described in Section 8.9.2.

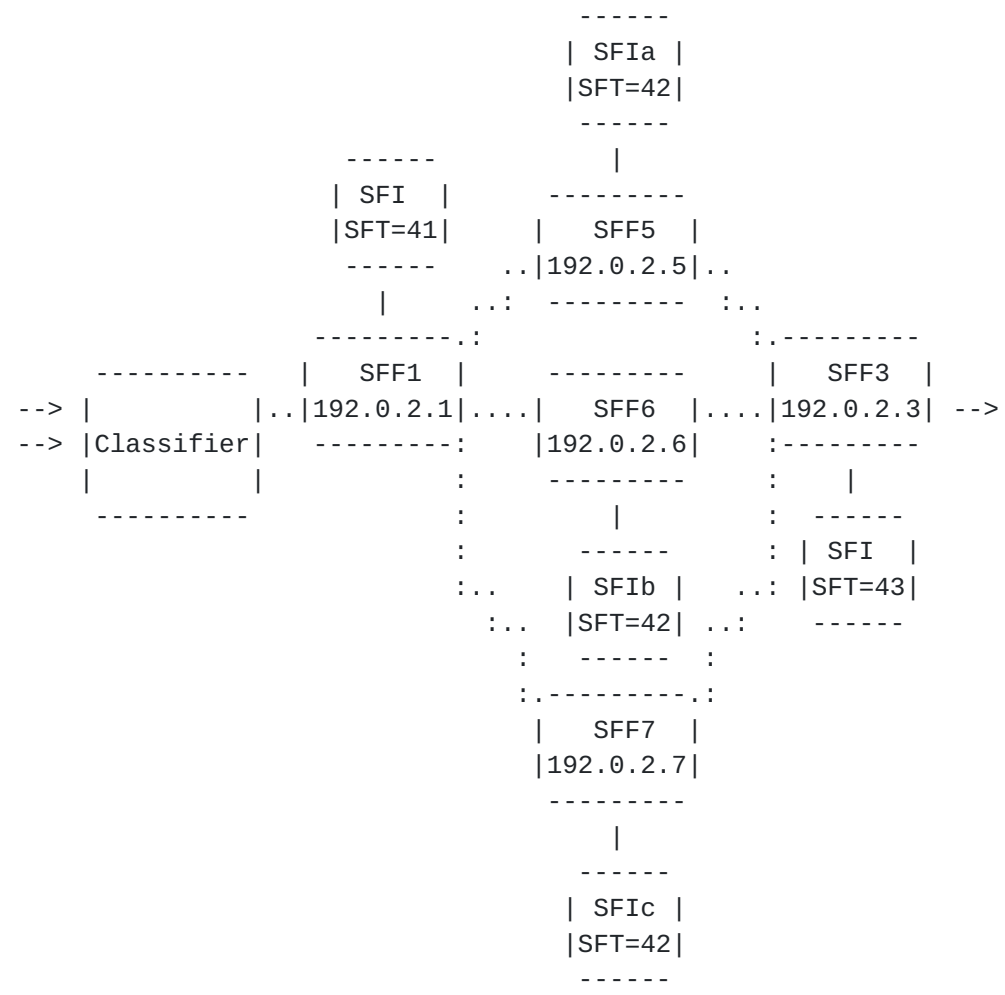


Figure 13: Second Example With Parallel End-to-End SFPs

In this case, five SFIRs are advertised as follows:

```
RD = 192.0.2.1:11, SFT = 41
RD = 192.0.2.5:11, SFT = 42  (for SFIa)
RD = 192.0.2.6:11, SFT = 42  (for SFIb)
RD = 192.0.2.7:11, SFT = 42  (for SFIc)
RD = 192.0.2.3:11, SFT = 43
```

In this case the controller could specify three forward SFPs with their corresponding associated reverse SFPs. Each bidirectional pair of SFPs uses a different SFF and SFI for middle hop (for an SF of type 42). The controller can instruct the Classifier how to place traffic on the three bidirectional SFPs, or can treat them as a group leaving the Classifier responsible for balancing the load.

```
SFP20:  RD = 198.51.100.1:120, SPI = 34,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:123, Assoc-SPI = 37,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.5:11],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP21:  RD = 198.51.100.1:121, SPI = 35,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:124, Assoc-SPI = 38,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.6:11],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP22:  RD = 198.51.100.1:122, SPI = 36,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:125, Assoc-SPI = 39,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.7:11],
       [SI = 253, SFT = 43, RD = 192.0.2.3:11]

SFP23:  RD = 198.51.100.1:123, SPI = 37,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:120, Assoc-SPI = 34,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.5:11],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]

SFP24:  RD = 198.51.100.1:124, SPI = 38,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:121, Assoc-SPI = 35,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.6:11],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]

SFP25:  RD = 198.51.100.1:125, SPI = 39,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:122, Assoc-SPI = 36,
       [SI = 255, SFT = 43, RD = 192.0.2.3:11],
       [SI = 254, SFT = 42, RD = 192.0.2.7:11],
       [SI = 253, SFT = 41, RD = 192.0.2.1:11]
```

8.9.4. Parallel SFPs Downstream of the Choice

The mechanism of parallel SFPs demonstrated in Section 8.9.3 is perfectly functional and may be practical in many environments. However, there may be scaling concerns because of the large amount of state (knowledge of SFPs, i.e., SFPR advertisements retained) if there is a very large amount of choice of SFIs (for example, tens of instances of the same stateful SF), or if there are multiple choices of stateful SF along a path. This situation may be mitigated using SFP fragments that are combined to form the end to end SFPs.

The example presented here is necessarily simplistic, but should convey the basic principle. The example presented in Figure 14 is similar to that in Section 8.9.3 but with an additional first hop.

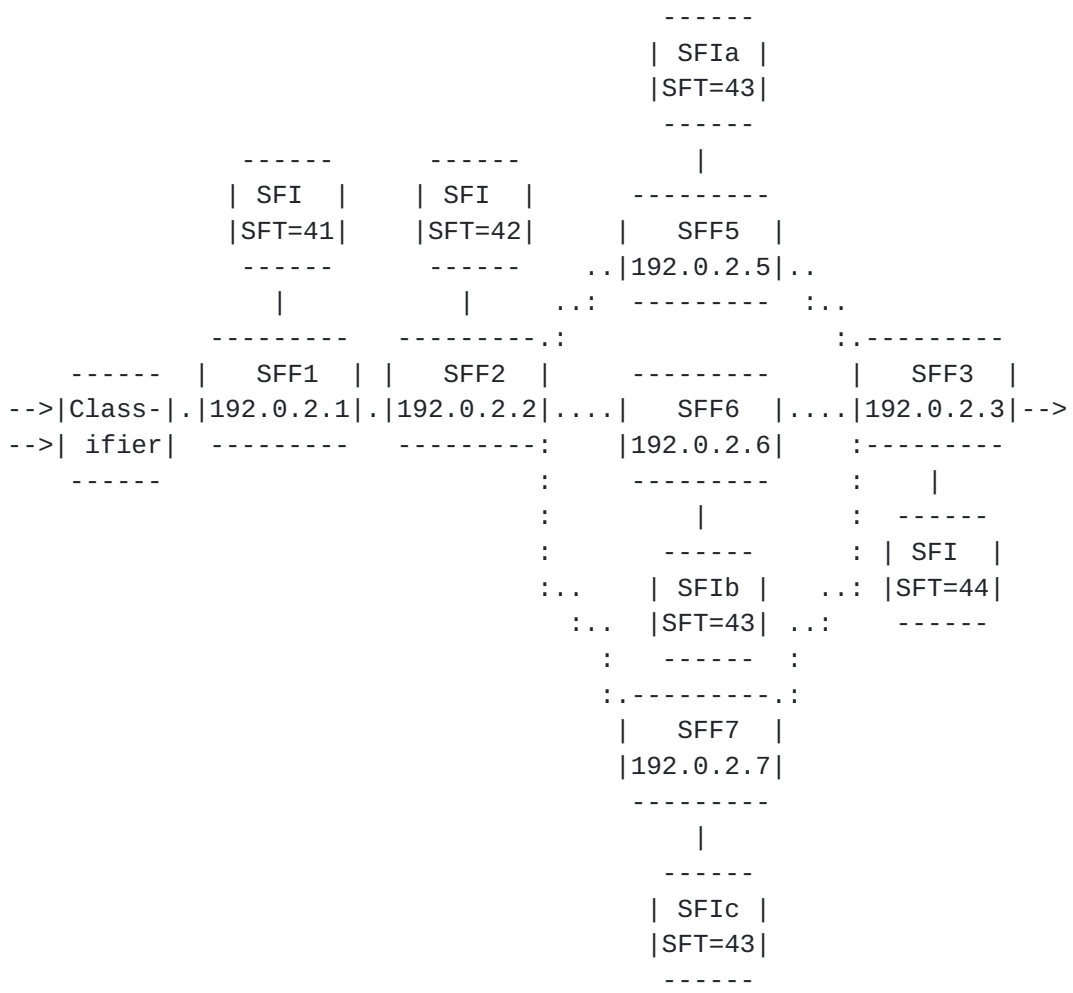


Figure 14: Example With Parallel SFPs Downstream of Choice

The six SFIs are advertised as follows:

```
RD = 192.0.2.1:11, SFT = 41
RD = 192.0.2.2:11, SFT = 42
RD = 192.0.2.5:11, SFT = 43  (for SFIa)
RD = 192.0.2.6:11, SFT = 43  (for SFIb)
RD = 192.0.2.7:11, SFT = 43  (for SFIc)
RD = 192.0.2.3:11, SFT = 44
```

SFF2 is the point at which a load balancing choice must be made. So "tail-end" SFPs are constructed as follows. Each takes in a different SFF that provides access to an SF of type 43.

```
SFP26:  RD = 198.51.100.1:126, SPI = 40,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:130, Assoc-SPI = 44,
       [SI = 255, SFT = 43, RD = 192.0.2.5:11],
       [SI = 254, SFT = 44, RD = 192.0.2.3:11]

SFP27:  RD = 198.51.100.1:127, SPI = 41,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:131, Assoc-SPI = 45,
       [SI = 255, SFT = 43, RD = 192.0.2.6:11],
       [SI = 254, SFT = 44, RD = 192.0.2.3:11]

SFP28:  RD = 198.51.100.1:128, SPI = 42,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:132, Assoc-SPI = 46,
       [SI = 255, SFT = 43, RD = 192.0.2.7:11],
       [SI = 254, SFT = 44, RD = 192.0.2.3:11]
```

Now an end-to-end SFP with load balancing choice can be constructed as follows. The choice made by SFF2 is expressed in terms of entering one of the three "tail end" SFPs.

```
SFP29:  RD = 198.51.100.1:129, SPI = 43,
       [SI = 255, SFT = 41, RD = 192.0.2.1:11],
       [SI = 254, SFT = 42, RD = 192.0.2.2:11],
       [SI = 253, {SFT = 1, RD = {SPI=40, SI=255, Rsv=0},
                            RD = {SPI=41, SI=255, Rsv=0},
                            RD = {SPI=42, SI=255, Rsv=0} } ]
```

Now, despite the load balancing choice being made other than at the initial classifier, it is possible for the reverse SFPs to be well-constructed without any ambiguity. The three reverse paths appear as follows.

```
SFP30:  RD = 198.51.100.1:130, SPI = 44,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:126, Assoc-SPI = 40,
       [SI = 255, SFT = 44, RD = 192.0.2.4:11],
       [SI = 254, SFT = 43, RD = 192.0.2.5:11],
       [SI = 253, SFT = 42, RD = 192.0.2.2:11],
       [SI = 252, SFT = 41, RD = 192.0.2.1:11]

SFP31:  RD = 198.51.100.1:131, SPI = 45,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:127, Assoc-SPI = 41,
       [SI = 255, SFT = 44, RD = 192.0.2.4:11],
       [SI = 254, SFT = 43, RD = 192.0.2.6:11],
       [SI = 253, SFT = 42, RD = 192.0.2.2:11],
       [SI = 252, SFT = 41, RD = 192.0.2.1:11]

SFP32:  RD = 198.51.100.1:132, SPI = 46,
      Assoc-Type = 1, Assoc-RD = 198.51.100.1:128, Assoc-SPI = 42,
       [SI = 255, SFT = 44, RD = 192.0.2.4:11],
       [SI = 254, SFT = 43, RD = 192.0.2.7:11],
       [SI = 253, SFT = 42, RD = 192.0.2.2:11],
       [SI = 252, SFT = 41, RD = 192.0.2.1:11]
```

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 [RFC8300].

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. The control plane mechanisms are very similar to those used for BGP/MPLS IP VPNs as described in [RFC4364], and so the security considerations in that document (Section 23) provide good

guidance for securing SFC systems reliant on this specification. Section 19 of [RFC7432] also provides useful guidance on the use of BGP in a similar environment.

Note that a component of an SFC system that uses the procedures described in this document also requires communications between a controller and the SFC network elements. This communication covers instructing the Classifiers using BGP mechanisms (see Section 7.4) which is covered by BGP security. But it also covers other mechanisms for programming the Classifier and instructing the SFFs and SFs (for example, to bind SFs to an SFF, and to cause the establishment of tunnels between SFFs). This document does not cover these latter mechanisms and so their security is out of scope, but it should be noted that these communications provide an attack vector on the SFC system and so attention must be paid to ensuring that they are secure.

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.

- Values 0 and 65535 are to be marked "Reserved, not to be allocated".

- Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [RFC8126].

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

The new registry should track:

- Type
- Name
- Reference Document or Contact
- 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
4	MPLS Swapping/Stacking	[This.I-D]	Date-to-be-set
5	SFP Traversal With MPLS	[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.

- Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [RFC8126].

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

The new registry should track:

- Association Type
- Name

- Reference Document or Contact
- 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.

- Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- Values 1 through 31 are to be assigned by "Standards Action" [RFC8126] and are referred to as the Special Purpose SFT values.
- Other values (32 through 65534) are to be assigned according to the "First Come First Served" policy [RFC8126].

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

The new registry should track:

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

The registry should initially be populated as follows:

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

10.6. New Generic Transitive Experimental Use Extended Community Sub-Types

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters" with a subregistry of "Generic Transitive Experimental Use Extended Community Sub-Type". IANA is requested to assign a new sub-type as follows:

"Flow Spec for SFC Classifiers" (TBD4 in this document) with this document as the reference.

10.7. New BGP Transitive Extended Community Types

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters" with a subregistry of "BGP Transitive Extended Community Types". IANA is requested to assign new types as follows:

"SFI Pool Identifier" (TBD6 in this document) with this document as the reference.

"MPLS Label Stack Mixed Swapping/Stacking Labels" (TBD7 in this document) with this document as the reference.

10.8. SPI/SI Representation

IANA is requested to assign a codepoint from the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry for the "SPI/SI Representation Sub-TLV" (TBD5 in this document) with this document being the reference.

11. Contributors

Stuart Mackie
Juniper Networks

Email: wsmackie@juinper.net

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Avinash Lingala
AT&T

Email: ar977m@att.com

12. Acknowledgements

Thanks to Tony Przygienda, Jeff Haas, and Andy Malis for helpful comments, and to Joel Halpern for discussions that improved this document. Yuanlong Jiang provided a useful review and caught some important issues. Stephane Litkowski did an exceptionally good and detailed document shepherd review.

Andy Malis contributed text that formed the basis of Section 7.7.

13. References

13.1. Normative References

[I-D.ietf-idr-rfc5575bis]
Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-17 (work in progress), June 2019.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Ramachandra, S., and E. Rosen, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-13 (work in progress), July 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <https://www.rfc-editor.org/info/rfc8595>.

[RFC8596] Malis, A., Bryant, S., Halpern, J., and W. Henderickx, "MPLS Transport Encapsulation for the Service Function Chaining (SFC) Network Service Header (NSH)", RFC 8596, DOI 10.17487/RFC8596, June 2019, <https://www.rfc-editor.org/info/rfc8596>.

13.2. Informative References

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

Authors' Addresses

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

John Drake
Juniper Networks

Email: jdrake@juniper.net

Eric Rosen
Juniper Networks

Email: erosen52@gmail.com

Jim Uttaro
AT&T

Email: ju1738@att.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com