Internet Engineering Task Force (IETF)
Request for Comments: 8595
Category: Standards Track
ISSN: 2070-1721

A. Farrel
Old Dog Consulting
S. Bryant
Futurewei
J. Drake
Juniper Networks

June 2019

An MPLS-Based Forwarding Plane for Service Function Chaining

Abstract

This document describes how Service Function Chaining (SFC) can be achieved in an MPLS network by means of a logical representation of the Network Service Header (NSH) in an MPLS label stack. That is, the NSH is not used, but the fields of the NSH are mapped to fields in the MPLS label stack. This approach does not deprecate or replace the NSH, but it acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract Service Functions (SFs) before being delivered to the intended destination. An architecture for SFC is defined in [RFC7665].

When applying a particular service function chain to the traffic selected by a service classifier, the traffic needs to be steered through an ordered set of SFs in the network. This ordered set of SFs is termed a Service Function Path (SFP), and the traffic is passed between Service Function Forwarders (SFFs) that are responsible for delivering the packets to the SFs and for forwarding them onward to the next SFF.

In order to steer the selected traffic between SFFs and to the correct SFs, the service classifier needs to attach information to each packet. This information indicates the SFP on which the packet is being forwarded and hence the SFs to which it must be delivered. The information also indicates the progress the packet has already made along the SFP.

The Network Service Header (NSH) [RFC8300] has been defined to carry the necessary information for SFC in packets. The NSH can be inserted into packets and contains various information, including a Service Path Identifier (SPI), a Service Index (SI), and a Time To Live (TTL) counter.

Multiprotocol Label Switching (MPLS) [RFC3031] is a widely deployed forwarding technology that uses labels placed in a packet in a label stack to identify the forwarding actions to be taken at each hop through a network. Actions may include swapping or popping the labels as well as using the labels to determine the next hop for forwarding the packet. Labels may also be used to establish the context under which the packet is forwarded. In many cases, MPLS will be used as a tunneling technology to carry packets through networks between SFFs.

This document describes how SFC can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack. This approach is applicable to all forms of MPLS forwarding (where labels are looked up at each hop and are swapped or popped [RFC3031]). It does not deprecate or replace the NSH, but it acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks. The mechanisms described in this document are a compromise between the full function that can be achieved using the NSH and the benefits of reusing the existing

MPLS forwarding paradigms (the approach defined here does not include the O bit defined in [RFC8300] and has some limitations to the use of metadata as described in Section 12).

Section 4 provides a short overview of several use case scenarios that help to explain the relationship between the MPLS label operations (swapping, popping, stacking) and the MPLS encoding of the logical NSH described in this document.

It is assumed that the reader is fully familiar with the terms and concepts introduced in [RFC7665] and [RFC8300].

Note that one of the features of the SFC architecture described in [RFC7665] is the "SFC proxy", which exists to include legacy SFs that are not able to process NSH-encapsulated packets. This issue is equally applicable to the use of MPLS-encapsulated packets that encode a logical representation of an NSH. It is discussed further in Section 9.

2. 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.

3. Choice of Data-Plane SPI/SI Representation

While [RFC8300] defines the NSH that can be used in a number of environments, this document provides a mechanism to handle situations in which the NSH is not ubiquitously deployed. In this case, it is possible to use an alternative data-plane representation of the SPI/SI by carrying the identical semantics in MPLS labels.

In order to correctly select the mechanism by which SFC information is encoded and carried between SFFs, it may be necessary to configure the capabilities and choices either within the whole Service Function Overlay Network or on a hop-by-hop basis. It is a requirement that both ends of a tunnel over the underlay network (i.e., a pair of SFFs adjacent in the SFP) know that the tunnel is used for SFC and know what form of NSH representation is used. A control-plane signaling approach to achieve these objectives is provided using BGP in [BGP-NSH-SFC].

Note that the encoding of the SFC information is independent of the choice of tunneling technology used between SFFs. Thus, an MPLS representation of the logical NSH (as defined in this document) may be used even if the tunnel between a pair of SFFs is not an MPLS tunnel. Conversely, MPLS tunnels may be used to carry other encodings of the logical NSH (specifically, the NSH itself).

4. Use Case Scenarios

There are five scenarios that can be considered for the use of an MPLS encoding in support of SFC. These are set out in the following subsections.

4.1. Label Swapping for Logical NSH

The primary use case for SFC is described in [RFC7665] and delivered using the NSH, which, as described in [RFC8300], uses an encapsulation with a position indicator that is modified at each SFC hop along the chain to indicate the next hop.

The label-swapping use case scenario effectively replaces the NSH with an MPLS encapsulation as described in Section 6. The MPLS labels encode the same information as the NSH to form a logical NSH. The labels are modified (swapped per [RFC3031]) at each SFC hop along the chain to indicate the next hop. The processing and the forwarding state for a chain (i.e., the actions to take on a received label) are programmed into the network using a control plane or management plane.

4.2. Hierarchical Encapsulation

[RFC8459] describes an architecture for hierarchical encapsulation using the NSH. It facilitates partitioning of SFC domains for administrative reasons and allows concatenation of service function chains under the control of a service classifier.

The same function can be achieved in an MPLS network using an MPLS encoding of the logical NSH, and label stacking as defined in [RFC3031] and described in Section 7. In this model, swapping is used per Section 4.1 to navigate one chain, and when the end of the chain is reached, the final label is popped, revealing the label for another chain. Thus, the primary mode is swapping, but stacking is used to enable the ingress classifier to control concatenation of service function chains.

4.3. Fine Control of Service Function Instances

It may be that a service function chain (as described in Section 4.1) allows some leeway in the choice of service function instances along the chain. However, it may be that a service classifier wishes to constrain the choice and this can be achieved using chain concatenation so that the first chain ends at the point of choice, the next label in the stack indicates the specific service function instance to be executed, and the next label in the stack starts a new chain. Thus, a mixture of label swapping and stacking is used.

4.4. Micro Chains and Label Stacking

The scenario in Section 4.2 may be extended to its logical extreme by making each concatenated chain as short as it can be: one SF. Each label in the stack indicates the next SF to be executed, and the network is programmed through the control plane or management plane to know how to route to the next (i.e., first) hop in each chain just as it would be to support the scenarios in Sections 4.1 and 4.2.

This scenario is functionally identical to the use of Segment Routing (SR) in an MPLS network (known as SR-MPLS) for SFC, as described in Section 4.5, and the discussion in that section applies to this section as well.

4.5. SFC and Segment Routing

SR-MPLS uses a stack of MPLS labels to encode information about the path and network functions that a packet should traverse. SR-MPLS is achieved by applying control-plane and management-plane techniques to program the MPLS forwarding plane and by imposing labels on packets at the entrance to the SR-MPLS network. An implementation proposal for achieving SFC using SR-MPLS can be found in [SR-Srv-Prog] and is not discussed further in this document.

5. Basic Unit of Representation

When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used. This unit comprises two MPLS labels, as shown below. The unit may be present one or more times in the label stack as explained in subsequent sections.

In order to convey the same information as is present in the NSH, two MPLS label stack entries are used. One carries a label to provide context within the SFC scope (the SFC Context Label), and the other carries a label to show which SF is to be actioned (the SF Label). This two-label unit is shown in Figure 1.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           SFC Context Label           | TC  |S|      TTL      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           SF Label                    | TC  |S|      TTL      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: The Basic Unit of MPLS Label Stack for SFC

The fields of these two label stack entries are encoded as follows:

Label: The Label fields contain the values of the SFC Context Label and the SF Label encoded as 20-bit integers. The precise semantics of these Label fields are dependent on whether the label stack entries are used for MPLS label swapping (see Section 6) or MPLS label stacking (see Section 7).

TC: The TC bits have no meaning in this case. They SHOULD be set to zero in both label stack entries when a packet is sent and MUST be ignored on receipt.

S: The "Bottom of Stack" bit has its usual meaning in MPLS. It MUST be clear in the SFC Context Label stack entry. In the SF Label stack entry, it MUST be clear in all cases except when the label is the bottom of the stack, when it MUST be set.

TTL: The TTL field in the SFC Context Label stack entry SHOULD be set to 1. The TTL in the SF Label stack entry (called the SF TTL) is set according to its use for MPLS label swapping (see Section 6) or MPLS label stacking (see Section 7) and is used to mitigate packet loops.

The sections that follow show how this basic unit of MPLS label stack may be used for SFC in the MPLS label-swapping case and in the MPLS label-stacking case. For simplicity, these sections do not describe the use of metadata; that topic is covered separately in Section 12.

6. MPLS Label Swapping

This section describes how the basic unit of MPLS label stack for SFC (introduced in Section 5) is used when MPLS label swapping is in use. The use case scenario for this approach is introduced in Section 4.1.

As can be seen in Figure 2, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used (i.e., MPLS, MPLS in UDP, MPLS in GRE, and MPLS in Virtual Extensible Local Area Networks

(VXLANs) or the Generic Protocol Extension for VXLAN (GPE)); thus, the tunnel technology does not need to be MPLS, but MPLS is shown here for simplicity.

An entropy label [RFC6790] may also be present, as described in Section 11.

```
    ---------------
   ~ Tunnel Labels ~
   +---------------+
   ~   Optional    ~
   ~ Entropy Label ~
   +---------------+ - - -
   |   SPI Label   |
   +---------------+  Basic unit of MPLS label stack for SFC
   |   SI Label    |
   +---------------+ - - -
   |               |
   ~    Payload    ~
   |               |
    ---------------
```

Figure 2: The MPLS SFC Label Stack

Under these labels (or other encapsulation) comes a single instance of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries (provided in Section 5), the following meanings are applied:

SPI Label: The Label field of the SFC Context Label stack entry contains the value of the SPI encoded as a 20-bit integer. The semantics of the SPI are exactly as defined in [RFC8300]. Note that an SPI as defined by [RFC8300] can be encoded in 3 octets (i.e., 24 bits), but that the Label field allows for only 20 bits and reserves the values 0 through 15 as "special-purpose labels" [RFC7274]. Thus, a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16.

SI Label: The Label field of the SF Label stack entry contains the value of the SI exactly as defined in [RFC8300]. Since the SI requires only 8 bits, and to avoid overlap with the special-purpose label range of 0 through 15 [RFC7274], the SI is carried in the top (most significant) 8 bits of the Label field with the low-order 12 bits set to zero.

TC: The TC fields are as described in Section 5.

S: The S bits are as described in Section 5.

TTL: The TTL field in the SPI Label stack entry SHOULD be set to 1 as stated in Section 5. The TTL in the SF Label stack entry is decremented once for each forwarding hop in the SFP, i.e., for each SFF transited, and so mirrors the TTL field in the NSH.

The following processing rules apply to the Label fields:

- When a classifier inserts a packet onto an SFP, it sets the SPI Label to indicate the identity of the SFP and sets the SI Label to indicate the first SF in the path.

- When a component of the SFC system processes a packet, it uses the SPI Label to identify the SFP and the SI Label to determine which SFF or instance of an SF (an SFI) to deliver the packet to. Under normal circumstances (with the exception of branching and reclassification -- see [BGP-NSH-SFC]), the SPI Label value is preserved on all packets. The SI Label value is modified by SFFs and through reclassification to indicate the next hop along the SFP.

The following processing rules apply to the TTL field of the SF Label stack entry and are derived from Section 2.2 of [RFC8300]:

- When a classifier places a packet onto an SFP, it MUST set the TTL to a value between 1 and 255. It SHOULD set this according to the expected length of the SFP (i.e., the number of SFs on the SFP), but it MAY set it to a larger value according to local configuration. The maximum TTL value supported in an NSH is 63, and so the practical limit here may also be 63.

- When an SFF receives a packet from any component of the SFC system (classifier, SFI, or another SFF), it MUST discard any packets with TTL set to zero. It SHOULD log such occurrences but MUST apply rate limiting to any such logs.

- An SFF MUST decrement the TTL by one each time it performs a lookup to forward a packet to the next SFF.

- If an SFF decrements the TTL to zero, it MUST NOT send the packet and MUST discard the packet. It SHOULD log such occurrences but MUST apply rate limiting to any such logs.

- SFIs MUST ignore the TTL but MUST mirror it back to the SFF unmodified along with the SI (which may have been changed by local reclassification).

- If a classifier along the SFP makes any change to the intended path of the packet, including for looping, jumping, or branching (see [BGP-NSH-SFC]), it MUST NOT change the SI TTL of the packet. In particular, each component of the SFC system MUST NOT increase the SI TTL value; otherwise, loops may go undetected.

7. MPLS Label Stacking

This section describes how the basic unit of MPLS label stack for SFC (introduced in Section 5) is used when MPLS label stacking is used to carry information about the SFP and SFs to be executed. The use case scenarios for this approach are introduced in Section 4.

As can be seen in Figure 3, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used.

```
       -------------------
      ~   Tunnel Labels   ~
      +-------------------+
      ~     Optional      ~
      ~   Entropy Label   ~
      +-------------------+ - - -
      | SFC Context Label |
      +-------------------+  Basic unit of MPLS label stack for SFC
      |     SF Label      |
      +-------------------+ - - -
      | SFC Context Label |
      +-------------------+  Basic unit of MPLS label stack for SFC
      |     SF Label      |
      +-------------------+ - - -
      ~                   ~
      +-------------------+ - - -
      | SFC Context Label |
      +-------------------+  Basic unit of MPLS label stack for SFC
      |     SF Label      |
      +-------------------+ - - -
      |                   |
      ~      Payload      ~
      |                   |
       -------------------
```

Figure 3: The MPLS SFC Label Stack for Label Stacking

An entropy label [RFC6790] may also be present, as described in Section 11.

Under these labels comes one or more instances of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries (provided in Section 5), the following meanings are applied:

SFC Context Label: The Label field of the SFC Context Label stack entry contains a label that delivers SFC context. This label contains the SPI, encoded as a 20-bit integer using the semantics exactly as defined in [RFC8300]. Note that in this case a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16. This label may also be used to convey other SFC context-specific semantics, such as indicating how to interpret the SF Label or how to forward the packet to the node that offers the SF if so configured and coordinated with the controller that programs the labels for the SFP.

SF Label: The Label field of the SF Label stack entry contains a value that identifies the next SFI to be actioned for the packet. This label may be scoped globally or within the context of the preceding SFC Context Label and comes from the range 16 ... 2^20 - 1.

TC: The TC fields are as described in Section 5.

S: The S bits are as described in Section 5.

TTL: The TTL fields in the SFC Context Label stack entry and in the SF Label stack entry SHOULD be set to 1 as stated in Section 5 but MAY be set to larger values if the label indicated a forwarding operation towards the node that hosts the SF.

The following processing rules apply to the Label fields:

- When a classifier inserts a packet onto an SFP, it adds a stack comprising one or more instances of the basic unit of MPLS label stack for SFC. Taken together, this stack defines the SFs to be actioned and so defines the SFP that the packet will traverse.
- When a component of the SFC system processes a packet, it uses the top basic unit of label stack for SFC to determine to which SFI to next deliver the packet. When an SFF receives a packet, it examines the top basic unit of MPLS label stack for SFC to determine where to send the packet next. If the next recipient is a local SFI, the SFF strips the basic unit of MPLS label stack for SFC before forwarding the packet.

8. Mixed-Mode Forwarding

The previous sections describe homogeneous networks where SFC forwarding is either all label swapping or all label popping (stacking). This simplification helps to clarify the explanation of the mechanisms.

However, as described in Section 4.2, some use cases may use label swapping and stacking at the same time. Furthermore, it is also possible that different parts of the network utilize swapping or popping such that an end-to-end service chain has to utilize a combination of both techniques. It is also worth noting that a classifier may be content to use an SFP as installed in the network by a control plane or management plane and so would use label swapping, but that there may be a point in the SFP where a choice of SFIs can be made (perhaps for load balancing) and where, in this instance, the classifier wishes to exert control over that choice by use of a specific entry on the label stack as described in Section 4.3.

When an SFF receives a packet containing an MPLS label stack, it checks from the context of the incoming interface, and from the SFP indicated by the top label, whether it is processing an {SPI, SI} label pair for label swapping or a {context label, SFI index} label pair for label stacking. It then selects the appropriate SFI to which to send the packet. When it receives the packet back from the SFI, it has four cases to consider.

- If the current hop requires an {SPI, SI} and the next hop requires an {SPI, SI}, it sets the SPI Label according to the SFP to be traversed, selects an instance of the SF to be executed at the next hop, sets the SI Label to the SI value of the next hop, and tunnels the packet to the SFF for that SFI.

- If the current hop requires an {SPI, SI} and the next hop requires a {context label, SFI Label}, it pops the {SPI, SI} from the top of the MPLS label stack and tunnels the packet to the SFF indicated by the context label.

- If the current hop requires a {context label, SFI Label}, it pops the {context label, SFI Label} from the top of the MPLS label stack.

  * If the new top of the MPLS label stack contains an {SPI, SI} label pair, it selects an SFI to use at the next hop and tunnels the packet to the SFF for that SFI.

  * If the new top of the MPLS label stack contains a {context label, SFI Label}, it tunnels the packet to the SFF indicated by the context label.

9. A Note on Service Function Capabilities and SFC Proxies

The concept of an "SFC proxy" is introduced in [RFC7665]. An SFC proxy is logically located between an SFF and an SFI that is not "SFC aware". Such SFIs are not capable of handling the SFC encapsulation (whether that be NSH or MPLS) and need the encapsulation stripped from the packets they are to process. In many cases, legacy SFIs that were once deployed as "bumps in the wire" fit into this category until they have been upgraded to be SFC aware.

The job of an SFC proxy is to remove and then reimpose SFC encapsulation so that the SFF is able to process as though it was communication with an SFC-aware SFI, and so that the SFI is unaware of the SFC encapsulation. In this regard, the job of an SFC proxy is no different when NSH encapsulation is used and when MPLS encapsulation is used as described in this document, although (of course) it is different encapsulation bytes that must be removed and reimposed.

It should be noted that the SFC proxy is a logical function. It could be implemented as a separate physical component on the path from the SFF to the SFI, but it could be co-resident with the SFF or it could be a component of the SFI. This is purely an implementation choice.

Note also that the delivery of metadata (see Section 12) requires specific processing if an SFC proxy is in use. This is also no different when NSH functionality or the MPLS encoding defined in this document is in use, and how it is handled will depend on how (or if) each non-SFC-aware SFI can receive metadata.

10. Control-Plane Considerations

In order that a packet may be forwarded along an SFP, several functional elements must be executed.

- Discovery/advertisement of SFIs.

- Computation of the SFP.

- Programming of classifiers.

- Advertisement of forwarding instructions.

Various approaches may be taken. These include a fully centralized model where SFFs report to a central controller the SFIs that they support, the central controller computes the SFP and programs the classifiers, and (if the label-swapping approach is taken) the central controller installs forwarding state in the SFFs that lie on the SFP.

Alternatively, a dynamic control plane may be used, such as that described in [BGP-NSH-SFC]. In this case, the SFFs use the control plane to advertise the SFIs that they support, a central controller computes the SFP and programs the classifiers, and (if the label-swapping approach is taken) the central controller uses the control plane to advertise the SFPs so that SFFs that lie on the SFP can install the necessary forwarding state.

11. Use of the Entropy Label

Entropy is used in ECMP situations to ensure that packets from the same flow travel down the same path, thus avoiding jitter or reordering issues within a flow.

Entropy is often determined by hashing on specific fields in a packet header, such as the "five-tuple" in the IP and transport headers. However, when an MPLS label stack is present, the depth of the stack could be too large for some processors to correctly determine the entropy hash. This problem is addressed by the inclusion of an entropy label as described in [RFC6790].

When entropy is desired for packets as they are carried in MPLS tunnels over the underlay network, it is RECOMMENDED that an entropy label be included in the label stack immediately after the tunnel labels and before the SFC Labels, as shown in Figures 2 and 3.

If an entropy label is present in an MPLS payload, it is RECOMMENDED that the initial classifier use that value in an entropy label inserted in the label stack when the packet is forwarded (on the first tunnel) to the first SFF. In this case, it is not necessary to remove the entropy label from the payload.

12. Metadata

Metadata is defined in [RFC7665] as providing "the ability to exchange context information between classifiers and SFs, and among SFs." [RFC8300] defines how this context information can be directly encoded in fields that form part of the NSH encapsulation.

Sections 12.1 and 12.2 describe how metadata is associated with user data packets, and how metadata may be exchanged between SFC nodes in the network, when using an MPLS encoding of the logical representation of the NSH.

It should be noted that the MPLS encoding is less functional than the direct use of the NSH. Both methods support metadata that is "per-SFP" or "per-flow" (see [RFC8393] for definitions of these terms), but "per-packet" metadata (where the metadata must be carried on each packet because it differs from one packet to the next even on the same flow or SFP) is only supported using the NSH and not using the mechanisms defined in this document.

12.1. Indicating Metadata in User Data Packets

Metadata is achieved in the MPLS realization of the logical NSH by the use of an SFC Metadata Label, which uses the extended special-purpose label construct [RFC7274]. Thus, three label stack entries are present, as shown in Figure 4:

- The Extension Label (value 15).

- An extended special-purpose label called the Metadata Label Indicator (MLI) (value 16).

- The Metadata Label (ML).

```
 ----------------
| Extension = 15 |
+----------------+
|      MLI       |
+----------------+
| Metadata Label |
 ----------------
```

Figure 4: The MPLS SFC Metadata Label

The Metadata Label value is an index into a table of metadata that is programmed into the network using in-band or out-of-band mechanisms. Out-of-band mechanisms potentially include management-plane and control-plane solutions (such as [BGP-NSH-SFC]) but are out of scope for this document. The in-band mechanism is described in Section 12.2.

The SFC Metadata Label (as a set of three labels as indicated in Figure 4) may be present zero, one, or more times in an MPLS SFC packet. For MPLS label swapping, the SFC Metadata Labels are placed immediately after the basic unit of MPLS label stack for SFC, as shown in Figure 5. For MPLS label stacking, the SFC Metadata Labels are placed at the bottom of the label stack, as shown in Figure 6.

```
                          ----------------
                         ~ Tunnel Labels  ~
                         +----------------+
                         ~   Optional     ~
                         ~ Entropy Label  ~
                         +----------------+
                         |   SPI Label    |
                         +----------------+
                         |   SI Label     |
                         +----------------+
                         | Extension = 15 |
                         +----------------+
                         |     MLI        |
                         +----------------+
                         | Metadata Label |
                         +----------------+
                         ~     Other      ~
                         |    Metadata    |
                         ~  Label Triples ~
                         +----------------+
                         |                |
                         ~    Payload     ~
                         |                |
                          ----------------
```

Figure 5: The MPLS SFC Label Stack for Label Swapping with Metadata Label

```
                           -------------------
                          ~   Tunnel Labels   ~
                          +-------------------+
                          ~     Optional      ~
                          ~   Entropy Label   ~
                          +-------------------+
                          | SFC Context Label |
                          +-------------------+
                          |     SF Label      |
                          +-------------------+
                          ~                   ~
                          +-------------------+
                          | SFC Context Label |
                          +-------------------+
                          |     SF Label      |
                          +-------------------+
                          |   Extension = 15  |
                          +-------------------+
                          |        MLI        |
                          +-------------------+
                          |  Metadata Label   |
                          +-------------------+
                          ~       Other       ~
                          |      Metadata     |
                          ~   Label Triples   ~
                          +-------------------+
                          |                   |
                          ~      Payload      ~
                          |                   |
                           -------------------
```

Figure 6: The MPLS SFC Label Stack for Label Stacking with Metadata Label

12.2. In-Band Programming of Metadata

A mechanism for sending metadata associated with an SFP without a payload packet is described in [RFC8393]. The same approach can be used in an MPLS network where the NSH is logically represented by an MPLS label stack.

The packet header is formed exactly as previously described in this document so that the packet will follow the SFP through the SFC network. However, instead of payload data, metadata is included after the bottom of the MPLS label stack. An extended special-purpose label is used to indicate that the metadata is present. Thus, three label stack entries are present:

o The Extension Label (value 15).

o An extended special-purpose label called the Metadata Present Indicator (MPI) (value 17).

o The Metadata Label (ML) that is associated with this metadata on this SFP and can be used to indicate the use of the metadata as described in Section 12.

The MPI, if present, is placed immediately after the last basic unit of MPLS label stack for SFC. The resultant label stacks are shown in Figure 7 for the MPLS label-swapping case and Figure 8 for the MPLS label-stacking case.

```
                 ---------------
                ~ Tunnel Labels ~
                +---------------+
                ~   Optional    ~
                ~ Entropy Label ~
                +---------------+
                |   SPI Label   |
                +---------------+
                |   SI Label    |
                +---------------+
                | Extension = 15|
                +---------------+
                |      MPI      |
                +---------------+
                | Metadata Label|
                +---------------+
                |               |
                ~    Metadata   ~
                |               |
                 ---------------
```

Figure 7: The MPLS SFC Label Stack for Label Swapping Carrying Metadata

```
                            ------------------
                           ~   Tunnel Labels  ~
                           +------------------+
                           ~     Optional     ~
                           ~   Entropy Label  ~
                           +------------------+
                           | SFC Context Label|
                           +------------------+
                           |     SF Label     |
                           +------------------+
                           | SFC Context Label|
                           +------------------+
                           |     SF Label     |
                           +------------------+
                           ~                  ~
                           +------------------+
                           | SFC Context Label|
                           +------------------+
                           |     SF Label     |
                           +------------------+
                           |   Extension = 15 |
                           +------------------+
                           |        MPI       |
                           +------------------+
                           |  Metadata Label  |
                           +------------------+
                           |                  |
                           ~    Metadata      ~
                           |                  |
                            ------------------
```

Figure 8: The MPLS SFC Label Stack for Label Stacking Carrying Metadata

In both cases, the metadata is formatted as a TLV, as shown in Figure 9.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Length              |        Metadata Type          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                         Metadata                              ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: The Metadata TLV

The fields of this TLV are interpreted as follows:

Length: The length of the metadata carried in the Metadata field in octets, not including any padding.

Metadata Type: The type of the metadata present. Values for this field are taken from the "NSH MD Types" registry maintained by IANA and defined in [RFC8300] and encoded with the most significant bit first.

Metadata: The actual metadata formatted as described in whatever document defines the metadata. This field is end-padded with zero to 3 octets of zeroes to take it up to a 4-octet boundary.

12.2.1. Loss of In-Band Metadata

Note that in-band exchange of metadata is vulnerable to packet loss. This is both a risk arising from network faults and an attack vulnerability.

If packets that arrive at an SFF use an MLI that does not have an entry in the metadata table, an alarm can be raised and the packet can be discarded or processed without the metadata according to local configuration. This provides some long-term mitigation but is not an ideal solution.

Further mitigation of loss of metadata packets can be achieved by retransmitting them at a configurable interval. This is a relatively cheap, but only partial, solution because there may still be a window during which the metadata has not been received.

The concern of lost metadata may be particularly important when the metadata applicable to a specific MPI is being changed. This could result in out-of-date metadata being applied to a packet. If this is a concern, it is RECOMMENDED that a new MPI be used to install a new entry in the metadata table, and the packets in the flow should be marked with the equivalent new MLI.

Finally, if an application that requires metadata is sensitive to this potential loss or attack, it SHOULD NOT use in-band metadata distribution but SHOULD rely on control-plane or management-plane mechanisms, because these approaches can use a more sophisticated protocol that includes confirmation of delivery and can perform verification or inspection of entries in the metadata table.

13. Worked Examples

This section reverts to the simplified descriptions of networks that rely wholly on label swapping or label stacking. As described in Section 4, actual deployment scenarios may depend on the use of both mechanisms and utilize a mixed mode as described in Section 8.

Consider the simplistic MPLS SFC overlay network shown in Figure 10. A packet is classified for an SFP that will see it pass through two SFs (SFa and SFb) that are accessed through two SFFs (SFFa and SFFb, respectively). The packet is ultimately delivered to the destination, D.

```
            +---------------------------------------------------+
            |                   MPLS SFC Network                |
            |                                                   |
            |            +---------+       +---------+          |
            |            |   SFa   |       |   SFb   |          |
            |            +----+----+       +----+----+          |
            |               ^ | |             ^ | |             |
            |            (2)| | |(3)       (5)| | |(6)          |
            |       (1)     | | V     (4)     | | V    (7)      |
       +----------+ ---> +----+----+ ----> +----+----+ ---> +-------+
       |Classifier+------+  SFFa   +-------+  SFFb   +------+   D   |
       +----------+      +---------+       +---------+      +-------+
            |                                                   |
            +---------------------------------------------------+
```

Figure 10: Service Function Chaining in an MPLS Network

Let us assume that the SFP is computed and assigned an SPI value of 239. The forwarding details of the SFP are distributed (perhaps using the mechanisms of [BGP-NSH-SFC]) so that the SFFs are programmed with the necessary forwarding instructions.

The packet progresses as follows:

1. The classifier assigns the packet to the SFP and imposes two label stack entries comprising a single basic unit of MPLS SFC representation:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 255.

Further labels may be imposed to tunnel the packet from the classifier to SFFa.

2. When the packet arrives at SFFa, SFFa strips any labels associated with the tunnel that runs from the classifier to SFFa. SFFa examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFa. The packet is forwarded to SFa unmodified.

3. SFa performs its designated function and returns the packet to SFFa.

4. SFFa modifies the SI in the lower label stack entry (to 254) and uses the SPI/SI to look up the forwarding instructions. It sends the packet with two label stack entries:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 254.

   Further labels may be imposed to tunnel the packet from SFFa to SFFb.

5. When the packet arrives at SFFb, SFFb strips any labels associated with the tunnel from SFFa. SFFb examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFb. The packet is forwarded to SFb unmodified.

6. SFb performs its designated function and returns the packet to SFFb.

7. SFFb modifies the SI in the lower label stack entry (to 253) and uses the SPI/SI to look up the forwarding instructions. It determines that it is the last SFF in the SFP, so it strips the two SFC Label stack entries and forwards the payload toward D using the payload protocol.

Alternatively, consider the MPLS SFC overlay network shown in Figure 11. A packet is classified for an SFP that will see it pass through two SFs (SFx and SFy) that are accessed through two SFFs (SFFx and SFFy, respectively). The packet is ultimately delivered to the destination, D.

```
           +---------------------------------------------------+
           |                   MPLS SFC Network                |
           |                                                   |
           |            +---------+       +---------+          |
           |            |   SFx   |       |   SFy   |          |
           |            +----+----+       +----+----+          |
           |               ^ | |             ^ | |             |
           |            (2)| | |(3)       (5)| | |(6)          |
           |       (1)     | | V     (4)     | | V    (7)      |
      +----------+ ---> +----+----+ ----> +----+----+ ---> +-------+
      |Classifier+------+  SFFx   +-------+  SFFy   +------+   D   |
      +----------+      +---------+       +---------+      +-------+
           |                                                   |
           +---------------------------------------------------+
```

Figure 11: Service Function Chaining Using MPLS Label Stacking

Let us assume that the SFP is computed and assigned an SPI value of 239. However, the forwarding state for the SFP is not distributed and installed in the network. Instead, it will be attached to the individual packets using the MPLS label stack.

The packet progresses as follows:

1. The classifier assigns the packet to the SFP and imposes two basic units of MPLS SFC representation to describe the full SFP:

   * The top basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SFx.

  - The lower basic unit comprises two label stack entries as follows:

    - The higher label stack entry contains a label carrying the SFC context.

    - The lower label stack entry contains a label carrying the SF indicator for SFy.

  Further labels may be imposed to tunnel the packet from the classifier to SFFx.

2. When the packet arrives at SFFx, SFFx strips any labels associated with the tunnel from the classifier. SFFx examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFx. The packet is forwarded to SFx unmodified.

3. SFx performs its designated function and returns the packet to SFFx.

4. SFFx strips the top basic unit of MPLS SFC representation, revealing the next basic unit. It then uses the revealed context/SF values to determine how to route the packet to the next SFF, SFFy. It sends the packet with just one basic unit of MPLS SFC representation comprising two label stack entries:

   - The higher label stack entry contains a label carrying the SFC context.

   - The lower label stack entry contains a label carrying the SF indicator for SFy.

   Further labels may be imposed to tunnel the packet from SFFx to SFFy.

5. When the packet arrives at SFFy, SFFy strips any labels associated with the tunnel from SFFx. SFFy examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFy. The packet is forwarded to SFy unmodified.

6. SFy performs its designated function and returns the packet to SFFy.

7. SFFy strips the top basic unit of MPLS SFC representation, revealing the payload packet. It forwards the payload toward D using the payload protocol.

14. Implementation Notes

It is not the job of an IETF specification to describe the internals of an implementation, except where that directly impacts upon the bits on the wire that change the likelihood of interoperability or where the availability of configuration or security options directly affects the utility of an implementation.

However, in view of the objective of this document to acknowledge that there may be a need for an interim deployment of SFC functionality in brownfield MPLS networks, this section provides some observations about how an SFF might utilize MPLS features that are available in existing routers. This section is not intended to be definitive or technically complete; rather, it is indicative.

Consider the mechanism used to indicate to which Virtual Routing and Forwarding (VRF) system an incoming MPLS packet should be routed in a Layer 3 Virtual Private Network (L3VPN) [RFC4364]. In this case, the top MPLS label is an indicator of the VRF system that is to be used to route the payload.

A similar approach can be taken with the label-swapping SFC technique described in Section 6 such that the SFC Context Label identifies a routing table specific to the SFP. The SF Label can be looked up in the context of this routing table to determine to which SF to direct the packet and how to forward it to the next SFF.

Advanced features (such as metadata) are not inspected by SFFs. The packets are passed to SFIs that are MPLS-SFC aware or to SFC proxies, and those components are responsible for handling all metadata issues.

Of course, an actual implementation might make considerable optimizations on this approach, but this section should provide hints about how MPLS-based SFC might be achieved with relatively small modifications to deployed MPLS devices.

15. Security Considerations

Discussion of the security properties of SFC networks can be found in [RFC7665]. Further security discussion for the NSH and its use is provided in [RFC8300]. Those documents provide analysis and present a set of requirements and recommendations for security, and the normative security requirements from those documents apply to this specification. However, it should be noted that those documents do not describe any mechanisms for securing NSH systems.

It is fundamental to the SFC design that the classifier is a fully trusted element. That is, the classification decision process is not visible to the other elements, and its output is treated as accurate. As such, the classifier has responsibility for determining the processing that the packet will be subject to, including, for example, firewall functions. It is also fundamental to the MPLS design that packets are routed through the network using the path specified by the node imposing the labels and that the labels are swapped or popped correctly. Where an SF is not encapsulation aware, the encapsulation may be stripped by an SFC proxy such that a packet may exist as a native packet (perhaps IP) on the path between the SFC proxy and the SF; however, this is an intrinsic part of the SFC design, which needs to define how a packet is protected in that environment.

SFC components are configured and enabled through a management system or a control plane. This document does not make any assumptions about what mechanisms are used. Deployments should, however, be aware that vulnerabilities in the management plane or control plane of an SFC system imply vulnerabilities in the whole SFC system. Thus, control-plane solutions (such as [BGP-NSH-SFC]) and management-plane mechanisms must include security measures that can be enabled by operators to protect their SFC systems.

An analysis of the security of MPLS systems is provided in [RFC5920], which also notes that the MPLS forwarding plane has no built-in security mechanisms. Some proposals to add encryption to the MPLS forwarding plane have been suggested [MPLS-Opp-Sec], but no mechanisms have been agreed upon at the time of publication of this document. Additionally, MPLS does not provide any cryptographic integrity protection on the MPLS headers. That means that procedures described in this document rely on three basic principles:

- The MPLS network is often considered to be a closed network such that insertion, modification, or inspection of packets by an outside party is not possible. MPLS networks are operated with closed boundaries so that MPLS-encapsulated packets are not admitted to the network, and MPLS headers are stripped before packets are forwarded from the network. This is particularly pertinent in the SFC context because [RFC7665] notes that "The architecture described herein is assumed to be applicable to a single network administrative domain." Furthermore, [RFC8300] states that packets originating outside the SFC-enabled domain MUST be dropped if they contain an NSH and packets exiting the SFC-enabled domain MUST be dropped if they contain an NSH. These constraints apply equally to the use of MPLS to encode a logical representation of the NSH.

- The underlying transport mechanisms (such as Ethernet) between adjacent MPLS nodes may offer security mechanisms that can be used to defend packets "on the wire".

- The SFC-capable devices participating in an SFC system are responsible for verifying and protecting payload packets and their contents as well as providing other security capabilities that might be required in the particular system.

Additionally, where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how the tunnel is secured.

Thus, this design relies on the component underlying technologies to address the potential security vulnerabilities, and it documents the necessary protections (or risk of their absence) above. It does not include any native security mechanisms in-band with the MPLS encoding of the NSH functionality.

Note that configuration elements of this system (such as the programming of the table of metadata; see Section 12) must also be adequately secured, although such mechanisms are not in scope for this protocol specification.

No known new security vulnerabilities over the SFC architecture [RFC7665] and the NSH specification [RFC8300] are introduced by this design, but if issues are discovered in the future, it is expected that they will be addressed through modifications to control/ management components of any solution or through changes to the underlying technology.

16. IANA Considerations

IANA has made allocations from the "Extended Special-Purpose MPLS Label Values" subregistry of the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry as follows:

Value	Description	Reference
16	Metadata Label Indicator (MLI)	RFC 8595
17	Metadata Present Indicator (MPI)	RFC 8595

17. References

17.1. Normative References

[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>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, DOI 10.17487/RFC7274, June 2014, <https://www.rfc-editor.org/info/rfc7274>.

[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>.

[RFC8393] Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", RFC 8393, DOI 10.17487/RFC8393, May 2018, <https://www.rfc-editor.org/info/rfc8393>.

17.2. Informative References

[BGP-NSH-SFC]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", Work in Progress, draft-ietf-bess-nsh-bgp-control-plane-11, May 2019.

[MPLS-Opp-Sec]
Farrel, A. and S. Farrell, "Opportunistic Security in MPLS Networks", Work in Progress, draft-ietf-mpls-opportunistic-encrypt-03, March 2017.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8459] Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", RFC 8459, DOI 10.17487/RFC8459, September 2018, <https://www.rfc-editor.org/info/rfc8459>.

[SR-Srv-Prog]
Clad, F., Ed., Xu, X., Ed., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", Work in Progress, draft-xuclad-spring-sr-service-programming-02, April 2019.

Acknowledgements

This document derives ideas and text from [BGP-NSH-SFC]. The authors are grateful to all those who contributed to the discussions that led to that work: Loa Andersson, Andrew G. Malis, Alexander (Sasha) Vainshtein, Joel Halpern, Tony Przygienda, Stuart Mackie, Keyur Patel, and Jim Guichard. Loa Andersson provided helpful review comments.

Thanks to Loa Andersson, Lizhong Jin, Matthew Bocci, Joel Halpern, and Mach Chen for reviews of this text. Thanks to Russ Mundy for his Security Directorate review and to S Moonesamy for useful discussions. Thanks also to Benjamin Kaduk, Alissa Cooper, Eric Rescorla, Mirja Kuehlewind, Alvaro Retana, and Martin Vigoureux for comprehensive reviews during IESG evaluation.

The authors would like to be able to thank the authors of [SR-Srv-Prog] and [RFC8402] whose original work on service chaining and the identification of services using Segment Identifiers (SIDs), and conversation with whom, helped clarify the application of SR-MPLS to SFC.

Particular thanks to Loa Andersson for conversations and advice about working group process.

Contributors

The following individual contributed text to this document:

Andrew G. Malis
Email: agmalis@gmail.com

Authors' Addresses

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Stewart Bryant
Futurewei

Email: stewart.bryant@gmail.com

John Drake
Juniper Networks

Email: jdrake@juniper.net