An MPLS-Based Forwarding Plane for Service Function Chaining
draft-farrel-mpls-sfc-04

Abstract

   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 before being delivered to the intended
   destination. An architecture for SFC is defined in RFC7665.

   The Network Service Header (NSH) can be inserted into packets to
   steer them along a specific path to realize a Service Function Chain.

   Multiprotocol Label Switching (MPLS) is a widely deployed forwarding
   technology that uses labels to identify the forwarding actions to be
   taken at each hop through a network. Segment Routing is a mechanism
   that provides a source routing paradigm for steering packets in an
   MPLS network.

   This document describes how Service Function Chaining can be achieved
   in an MPLS network by means of a logical representation of the NSH in
   an MPLS label stack.

Status of This Memo

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 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 Service Functions (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 Service Function Chaining in packets.
The NSH can be inserted into packets and contains various information
including a Service Path Indicator (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 to identify the forwarding
actions to be taken at each hop through a network. In many cases,
MPLS will be used as a tunneling technology to carry packets through
networks between SFFs.

Segment Routing [RFC7855] defines a source routing paradigm for use
in packet switched networks. The application of Segment Routing in
MPLS networks is described in [I-D.ietf-spring-segment-routing-mpls]
and is known as MPLS-SR.

This document describes how Service Function Chaining 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 both classical
MPLS forwarding (where labels are looked up at each hop, and swapped
for the next hop [RFC3031]) and MPLS Segment Routing (where labels
are looked up at each hop, and popped to reveal the next label to
action [I-D.ietf-spring-segment-routing-mpls]). 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.

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

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 SFC) know that the tunnel is used for SFC and know
   what form of NSH representation is used. A control plane signalling
   approach to achieve these objectives is provided using BGP in
   [I-D.ietf-bess-nsh-bgp-control-plane].

   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. 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 service function 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      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
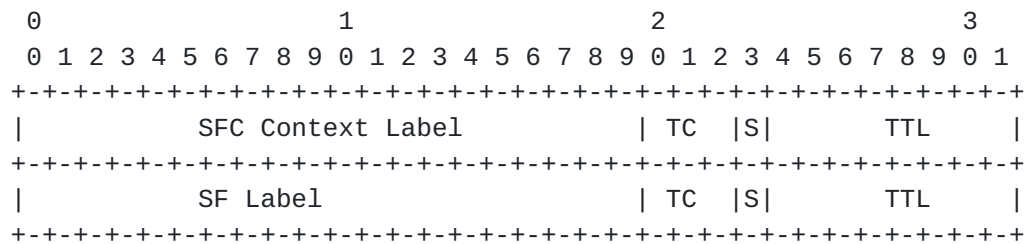
Figure 1: The Basic Unit of MPLS Label Stack for SFC

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 swapping (see Section 5) or MPLS-
   SR (see Section 6).

TC: The TC bits have no meaning. 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 and MAY be set in the
   SF label stack entry depending on whether the label is the bottom
   of stack.

TTL: The TTL field in the SFC Context label stack entry SHOULD be
   set to 1. The TTL in SF label stack entry (called the SF TTL) is
   set according to its use for MPLS swapping (see Section 5) or
   MPLS-SR (see Section 6 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-
SR case. For simplicity, these sections do not describe the use of
metadata: that is covered separately in Section 11.

5. MPLS Label Swapping

This section describes how the basic unit of MPLS label stack for SFC
introduced in Section 4 is used when MPLS label swapping is in use.
As can be seen from 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 VXLAN or GPE), thus the tunnel
technology does not need to be MPLS, but that is shown here for
simplicity.

An entropy label ([RFC6790]) may also be present as described in
Section 10

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 4 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 is 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 though 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 4.

S: The S bits are as described in Section 4.

TTL: The TTL field in the SPI label stack entry SHOULD be set to 1
   as stated in Section 4. The TTL in 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.

```
    --------------
  ~ Tunnel Labels ~
  +--------------+
  ~   Optional   ~
  ~ Entropy Label ~
  +--------------+ - - -
  |   SPI Label  |
  +--------------+  Basic unit of MPLS label stack for SFC
  |   SI Label   |
  +--------------+ - - -
  |              |
  ~    Payload   ~
  |              |
    --------------
```
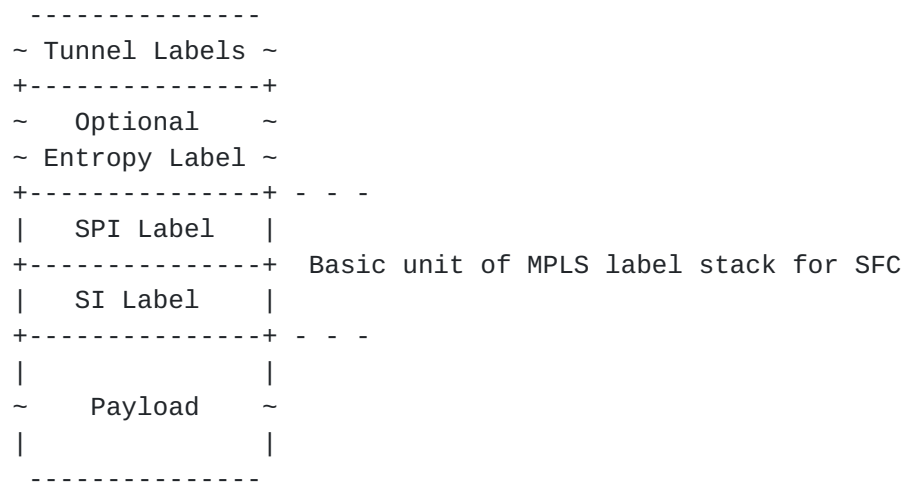
                    Figure 2: The MPLS SFC Label Stack

   The following processing rules apply to the Label fields:

   o 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.

   o 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 to
      which SFF or instance of an SF (an SFI) to deliver the packet.
      Under normal circumstances (with the exception of branching and
      reclassification - see [I-D.ietf-bess-nsh-bgp-control-plane]) 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]:

   o 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.

   o 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.

o An SFF MUST decrement the TTL by one each time it performs a
   forwarding lookup.

o 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.

o 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).

o If a Classifier along the SFP makes any change to the intended
   path of the packet including for looping, jumping, or branching
   (see [I-D.ietf-bess-nsh-bgp-control-plane] 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.

6. MPLS Segment Routing

   This section describes how the basic unit of MPLS label stack for SFC
   introduced in Section 4 is used when in an MPLS-SR network. As can
   be seen 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 and the tunnel technology does not
   need to be MPLS or MPLS-SR, but MPLS-SR is shown here for simplicity.

   An entropy label ([RFC6790]) may also be present as described in
   Section 10

   Under these labels (or other encapsulation) comes one of 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 4 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 may
      be used to indicate the SPI encoded as a 20 bit integer using the
      semantics of the SPI is exactly as defined in [RFC8300] and noting
      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-speific semantics such as indicating, perhaps with a node
      SID (see [I-D.ietf-spring-segment-routing]), how to interpret the
      SF Label.

   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 4.

S: The S bits are as described in Section 4.

TTL: The TTL field in the SFC Context label stack entry SHOULD be
set to 1 as stated in Section 4. The TTL in SF label stack entry
is set according to the norms for MPLS-SR.

```
  ------------------
 ~   MPLS-SR 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       ~
 |                  |
  ------------------
```
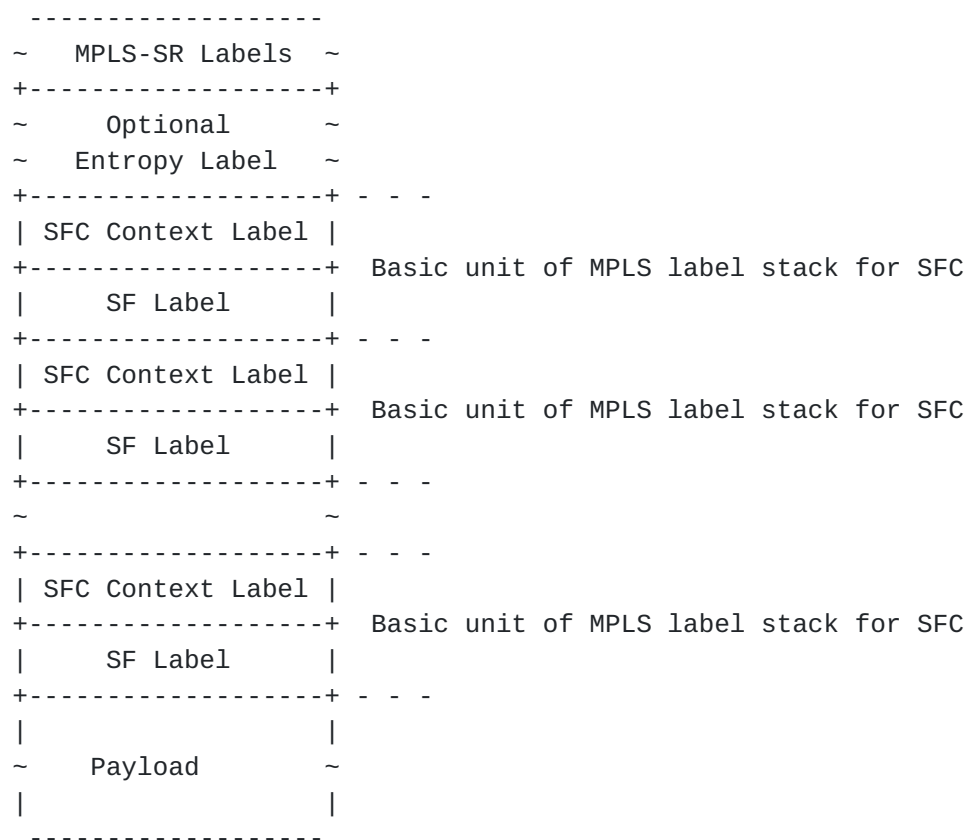
Figure 3: The MPLS SFC Label Stack for Segment Routing

The following processing rules apply to the Label fields:

o 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.

o 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 SFC strips the basic unit of MPLS label stack for
   SFC before forwarding the packet.

7. **Mixed Mode Forwarding**

   The previous sections describe homogeneous networks where SFC
   forwarding is either all label swapping or all label popping. But it
   is also possible that different parts of the network utilize swapping
   or popping for different purposes.

   When an SFF receives a packet containing an MPLS label stack, it
   checks whether it is processing an {SFP, SI} label pair for label
   swapping or a {context label, SFI index} label pair for MPLS-SR. 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.

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

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

   o 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 {SFP, SI}
         label pair, it selects an SFI to use at the next hop, and
         tunnels the packet to SFF for that SFI.

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

8. **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 SFI, but it could be coresident 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 11) requires
   specific processing if an SFC Proxy is in use. This is also no
   different when NSH 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.

9. **Control Plane Considerations**

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

   o Discovery/advertisement of SFIs.

   o Computation of SFP.

   o Programming of Classifiers.

   o 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 [I-D.ietf-bess-nsh-bgp-control-plane]. 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.

10. 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 re-
ordering 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 is included in the label stack immediately after the tunnel
labels and before the SFC labels as shown in Figure 2 and Figure 3.

If an Entropy Label is present in a packet received by an SR-capabale
node (at the end of a tunnel across the underlay network), it is
RECOMMENDED that the value of that label is preserved and used in an
Entropy Label inserted in the label stack when the packet is
forwarded (on the next tunnel) to the next SFF.

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.

11. 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.

The next two sections describe how metadata is associated with user
data packets, and how metadata may is 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 slightly less functional
than the direct use of the NSH. Both methods support metadata that
is "per-SFP" or "per-packet-flow" (see [I-D.farrel-sfc-convent] 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.

11.1. 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:

o The Extension Label (value 15)

o An extended special purpose label called the Metadata Label
   Indicator (MLI) (value TBD1 by IANA)

o 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
[I-D.ietf-bess-nsh-bgp-control-plane]), but are out of scope for this
document. The in-band mechanism is described in Section 11.2

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-SR, the SFC Metadata Labels can be present
zero, one, or more times and 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

```
       ------------------
      ~   MPLS-SR Labels  ~
      +------------------+
      ~      Optional     ~
      ~   Entropy Label   ~
      +------------------+
      | SFC Context Label |
      +------------------+
      |      SF Label     |
      +------------------+
      ~                   ~
      +------------------+
      | SFC Context Label |
      +------------------+
      |      SF Label     |
      +------------------+
      |   Extension = 15  |
      +------------------+
      |        MLI        |
      +------------------+
      |  Metadata Label   |
      +------------------+
      ~       Other       ~
      |     Metadata      |
      ~   Label Triples   ~
      +------------------+
      |                   |
      ~      Payload       ~
      |                   |
       ------------------
```
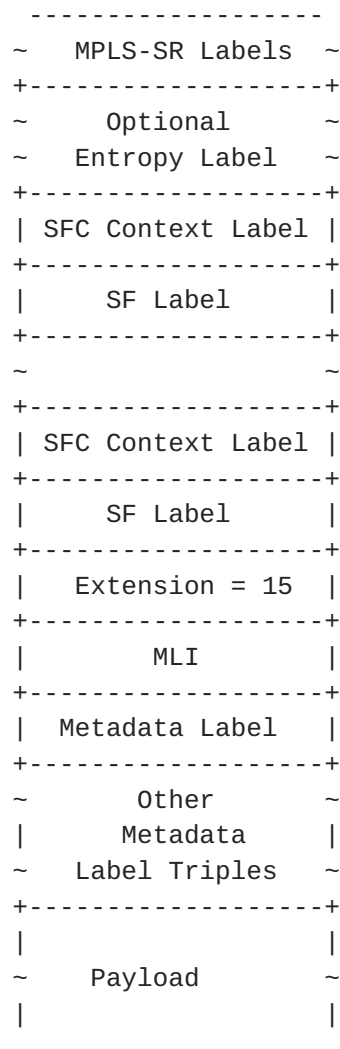
Figure 6: The MPLS SFC Label Stack for MPLS-SR with Metadata Label

11.2. Inband Programming of Metadata

A mechanism for sending metadata associated with an SFP without a
payload packet is described in [I-D.farrel-sfc-convent]. 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 TBD2 by IANA)

   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 11.

   The SFC Metadata Present Label, 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-SR case.

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

            Figure 7: The MPLS SFC Label Stack Carrying Metadata

```
    ------------------
~   MPLS-SR 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     ~
|                  |
    ------------------
```
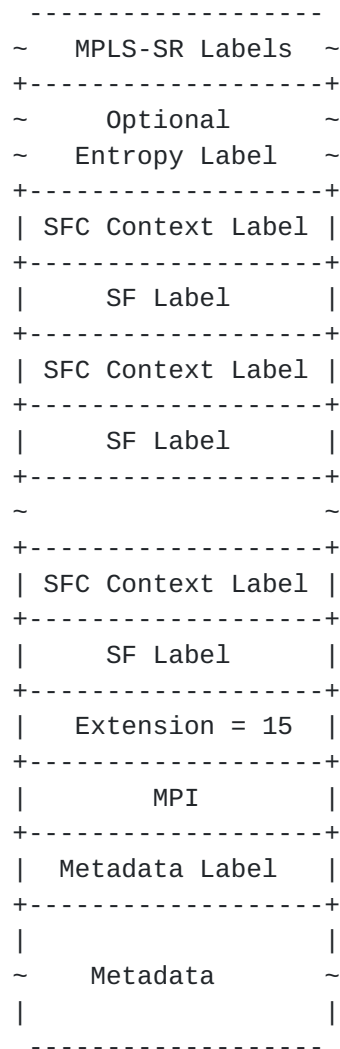
Figure 8: The MPLS SFC Label Stack for MPLS-SR Carrying Metadata

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 "MD Types" registry maintained by IANA
   and defined in [RFC8300].

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

12. Worked Examples

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
Service Functions, SFa and SFb, that are accessed through Service
Function Forwarders SFFa and SFFb respectively. The packet is
ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239.
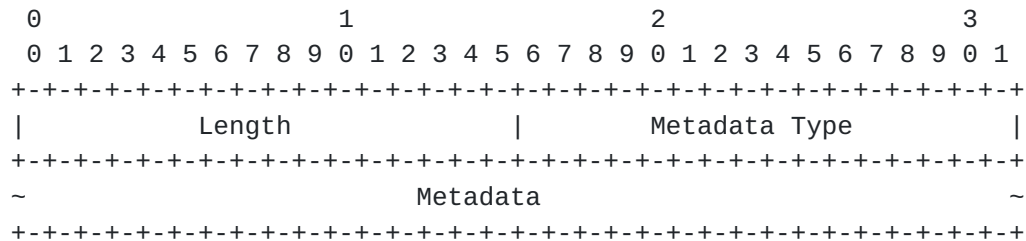The forwarding details of the SFP are distributed (perhaps using the
mechanisms of [I-D.ietf-bess-nsh-bgp-control-plane]) so that the SFFs
are programmed with the necessary forwarding instructions.

The packet progresses as follows:

a. 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.

b. When the packet arrives at SFFa it 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.

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

   d. 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 the SFFa
       to SFFb.

   e. When the packet arrives at SFFb it 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.

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

   g. SFFb modifies the SI in the lower label stack entry (to 253) and
       uses the SPI/SI to lookup 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.

       +---+
       | MPLS SFC Network |
       | |
       | +---------+   +---------+ |
       | | SFa | | SFb | |
       | +----+----+   +----+----+ |
       | ^ | | ^ | | |
       | (b)| | |(c)   (e)| | |(f) |
       | (a) | | V   (d) | | V   (g) |
   +----------+ ---> +----+----+ ----> +----+----+ ---> +-------+
   |Classifier+------+ SFFa   +-------+ SFFb   +------+ D |
   +----------+   +---------+   +---------+   +-------+
       | |
       +---+
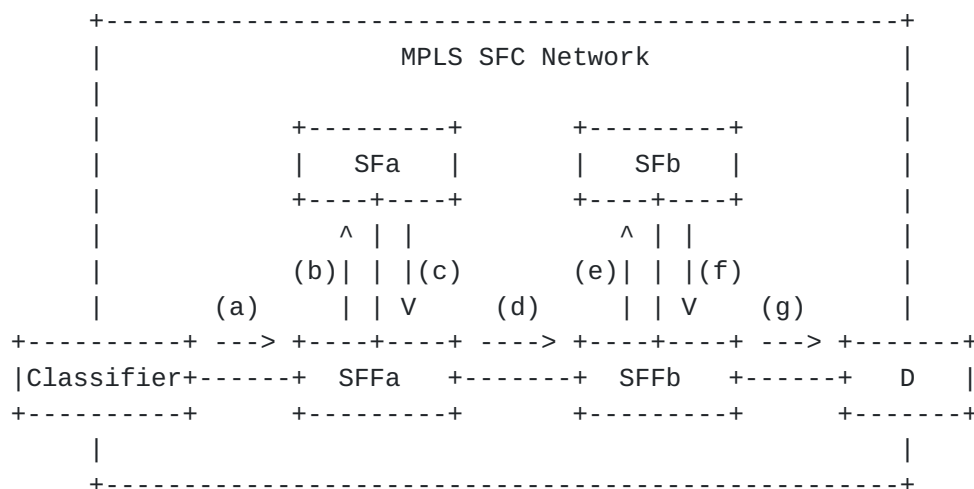

          Figure 10: Service Function Chaining in an MPLS Network

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 Service Functions, SFx and SFy, that are accessed through
Service Function Forwarders SFFx and SFFy respectively. The packet
is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI 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 MPLS-SR.

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 it 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 the SFFx
to SFFy.

5. When the packet arrives at SFFy it 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
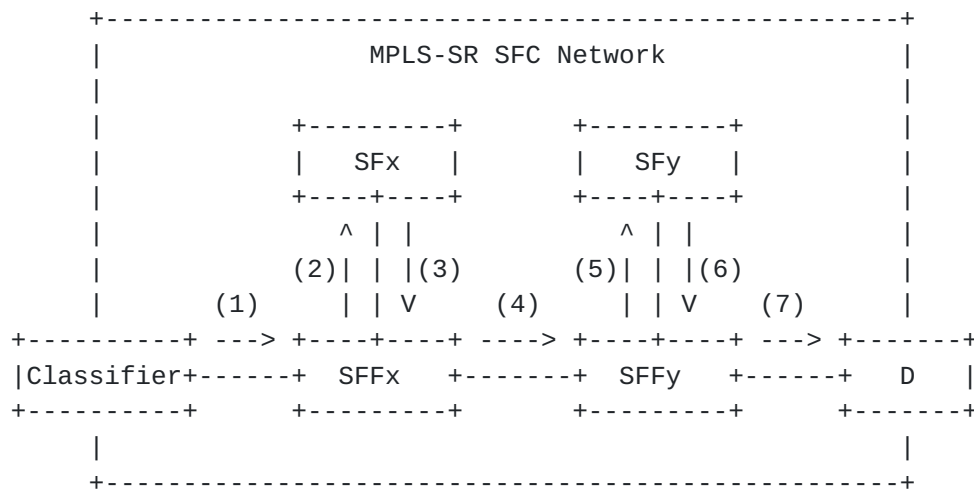   using the payload protocol.

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

Figure 11: Service Function Chaining in an MPLS-SR Network

13. 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
   present in [RFC8300].

   It is fundamental to the SFC design that the classifier is a trusted
   resource which determines the processing that the packet will be
   subject to, including for example the firewall. It is also
   fundamental to the Segment Routing design that packets are routed
   through the network using the path specified by the node imposing the
   SIDs. Where an SF is not encapsulation aware the packet may exist as
   an IP packet, however this is an intrinsic part of the SFC design
   which needs to define how a packet is protected in that environment.
   Where a tunnel is used to link two non-MPLS domains, the tunnel
   design needs to specify how it is secured. Thus the security
   vulnerabilities are addressed in the underlying technologies used by
   this design, which itself does not introduce any new security
   vulnerabilities.

14. IANA Considerations

   This document requests IANA to make 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                        |
      -------+------------------------------------+--------------
      TBD1   | Metadata Label Indicator (MLI)     | [This.I-D]
      TBD2   | Metadata Present Indicator (MPI)   | [This.I-D]
```

15. Acknowledgements

   This document derives ideas and text from
   [I-D.ietf-bess-nsh-bgp-control-plane].

   The authors are grateful to all those who contributed to the
   discussions that led to this work: Loa Andersson, Andrew G. Malis,
   Alexander Vainshtein, Joel M. Halpern, Tony Przygienda, Stuart
   Mackie, Keyur Patel, and Jim Guichard. Loa Andersson provided
   helpful review comments.

   Thanks to Loa Andersson, Lizhong Jin, Matthew Bocci, and Mach Chen
   for reviews of this text.

16. References

16.1. Normative References

[I-D.farrel-sfc-convent]
          Farrel, A. and J. Drake, "Operating the Network Service
          Header (NSH) with Next Protocol "None"", draft-farrel-sfc-
          convent-06 (work in progress), February 2018.

[I-D.ietf-spring-segment-routing-mpls]
          Bashandy, A., Filsfils, C., Previdi, S., Decraene, B.,
          Litkowski, S., and R. Shakir, "Segment Routing with MPLS
          data plane", draft-ietf-spring-segment-routing-mpls-12
          (work in progress), February 2018.

[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>.

[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>.

16.2. Informative References

[I-D.ietf-bess-nsh-bgp-control-plane]
          Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L.
          Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-
          nsh-bgp-control-plane-02 (work in progress), October 2017.

[I-D.ietf-spring-segment-routing]
          Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B.,
          Litkowski, S., and R. Shakir, "Segment Routing
          Architecture", draft-ietf-spring-segment-routing-15 (work
          in progress), January 2018.

   [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>.

   [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>.

   [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>.

   [RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B.,
              Litkowski, S., Horneffer, M., and R. Shakir, "Source
              Packet Routing in Networking (SPRING) Problem Statement
              and Requirements", RFC 7855, DOI 10.17487/RFC7855, May
              2016, <https://www.rfc-editor.org/info/rfc7855>.

Authors' Addresses

   Adrian Farrel
   Juniper Networks

   Email: afarrel@juniper.net

   Stewart Bryant
   Huawei

   Email: stewart.bryant@gmail.com

   John Drake
   Juniper Networks

   Email: jdrake@juniper.net