INTERNET-DRAFT   D. Eastlake
Intended status: Proposed Standard   Futurewei Technologies
Expires: May 26, 2022   November 27, 2022

              Service Function Chaining (SFC) Parallelism and Diversions
                      <draft-eastlake-sfc-parallel-04.txt>

Abstract
   Service Function Chaining (SFC) is the processing of packets through
   a sequence of Service Functions (SFs) within an SFC domain by the
   addition of path information and metadata on entry to that domain,
   the use and modification of that path information and metadata to
   step the packet through a sequence of SFs, and the removal of that
   path information and metadata on exit from that domain. The IETF has
   standardized a method for SFC using the Network Service Header
   specified in [RFC 8300](RFC 8300).

   There are requirements for SFC to process packets through parallel
   sequences of service functions and to easily splice in additional
   service functions or splice service functions out of a service chain.
   This document provides use cases for these requirements and
   extensions to SFC to support them.

Status of This Memo

   Distribution of this document is unlimited. Comments should be sent
   to the SFC Working Group mailing list <sfc@ietf.org> or to the
   authors.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

Table of Contents

1. Introduction

Service Function Chaining (SFC) is the processing of packets through
a sequence of Service Functions (SFs) within an SFC domain by the
addition of path information and metadata on entry to that domain,
the use and modification of that path information and metadata to
step the packet through a sequence of SFs, and the removal of that
path information and metadata on exit from that domain. The IETF has
standardized a method for SFC using the Network Service Header
specified in [RFC8300].

There are requirements for SFC to process packets through parallel
sequences of service functions and to easily splice in additional
service functions or splice service functions out of a service chain.
This document provides use cases for these requirements and
extensions to SFC to support them.

1.1 Conventions Used in This Document

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.

Acronyms and terms:

   downstream - The direction from ingress to egress.

   diversion - A reclassification of an SFC packet into one or
      multiple parallel packets with difference SPIs where this
      reclassified packet or packets are, in the normal case,
      combined at a downstream rendezvous point which restores the
      original SPI.

   DP - Diversion Point - An SF implementing a diversion.

   MD - Metadata - Part of the NSH.

   NSH - Network Service Header [RFC8300].

   rendezvous - The process of taking one or more corresponding SFC
      packets that have been diverted at an upstream DP, combining
      the packets if there are more than one, and restoring the
      original SPI.

   RePIn - Rendezvous Point Information. Metadata included in an SFC

packet for use at an RP.

RP - Rendezvous Point - An SF implementing a rendezvous.

SF - Service Function [RFC7665].

SFC - Service Function Chaining [RFC7665].

SFF - Service Function Forwarder [RFC7665] - A type of node that
     forwards packets based on the NSH.

SFP - Service Function Path.

SI - Service Index - Part of the NSH.

SPI - Service Path Identifier - Part of the NSH.

TLV - Type Length Value.

upstream - The direction from egress to ingress.

VLCH - Variable Length Context Header - A type of NSH header
     metadata.

2. **Service Function Chaining Background**

   Service Function Chaining (SFC) calls for the encapsulation of
   traffic within a service function chaining domain using a Network
   Service Header (NSH [RFC8300]) added by the "Classifier" (ingress
   node) on entry to the domain and the NSH being removed on exit from
   the domain at the downstream egress node as shown in Figure 1. The
   NSH controls the path of a packet in an SFC domain and includes
   additional information.

```
                    |
                    v
              +----------+
       . .|Classifier|. .  .  .  .  .  .  .  .  . .
       .  +----------+                            .
       .        |            +----+               .
       .        |          --+ SF |     Service   .
       .        |         /  +----+     Function  .
       .        v      ---             Chaining   .
       .    +-----+/        +----+     domain      .
       .    | SFF |--------+ SF |                  .
       .    +-----+\        +----+                  .
       .        |     ---                          .
       .        |       \  +----+                   .
       .        |        --+ SF |                   .
       .        v           +----+                   .
       .    +-----+                  +----+          .
       .    | SFF |-----------------+ SF |           .
       .    +-----+                  +----+          .
       .        |          +----+                   .
       .        |        --+ SF |                   .
       .        |       /  +----+                   .
       .        v    ---                            .
       .    +-----+/        +----+                   .
       .    | SFF |--------+ SF |                    .
       .    +-----+\        +----+                    .
       .        |     ---                            .
       .        |       \  +----+                     .
       .        |        --+ SF |                     .
       .        v           +----+                     .
       .    +------+                                   .
       . . .| Exit |. . .  .  .  .  .  .  .  .  .  . .
            +------+
                |
                v
```
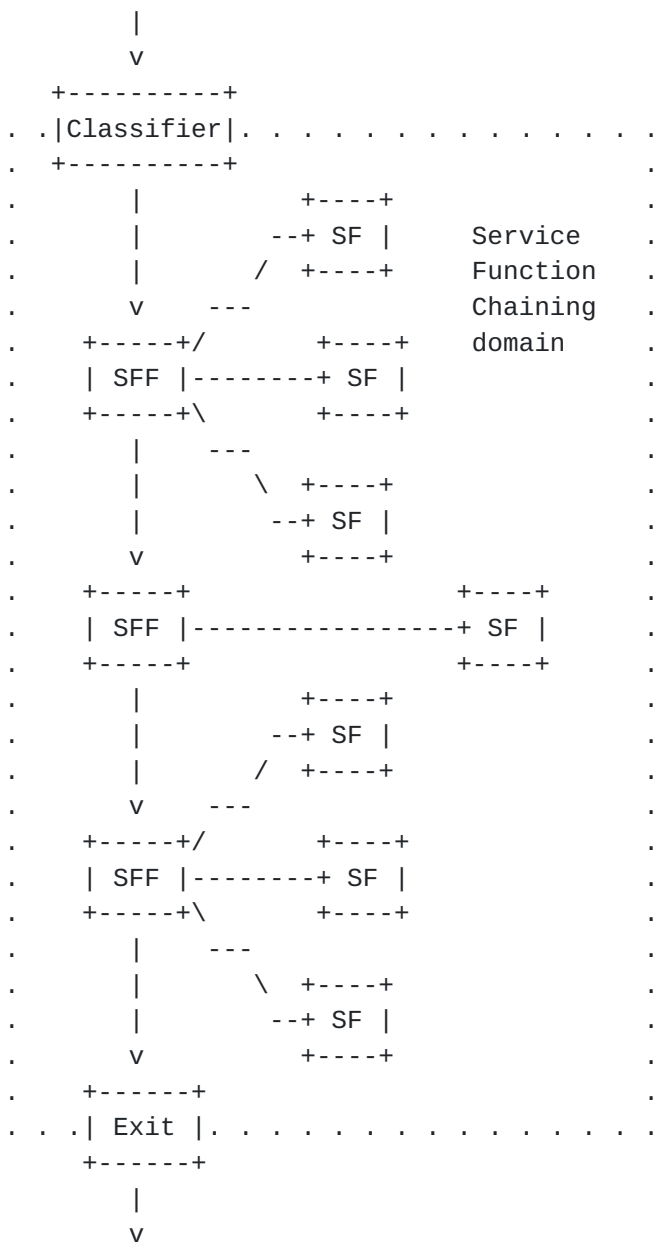
                  Figure 1. Example SFC Path Forwarding Nodes

Traffic passes through a sequence of Service Function Forwarders

(SFFs) each of which sends the traffic to one or, sequentially, more
than one Service Functions (SFs). Each SF performs some operation on
the traffic, for example firewall or Network Address Translation
(NAT) or load balancer, and then returns it to the SFF from which it
was received. There may be multiple instances of SFs performing the
same function attached to the same or different SFFs.

Logically, during the transit of an SFF, the outer transport header
that got the packet to the SFF is stripped (see Figure 2), the SFF
decides on the next forwarding step, either (1) adding a new
transport header or (2) in case of error discarding or logging the
packet and not forwarding it or, (3) if the SFF is the exit/egress,
removing the NSH header and then adding a new transport header. The
transport used may be different in different regions of the SFC
domain. For example, a version of the Internet Protocol (IP) could be
used in some part and Multi-Protocol Label Switching (MPLS) used in
other part of the SFC domain.

```
            +-----------------------------------+
            |   Outer Transport Header          |
            +-----------------------------------+
            |   Network Service Header (NSH)     |
            | +-------------------------------+  |
            | | Base Header                   |  |
            | +-------------------------------+  |
            | | Service Path Header           |  |
            | +-------------------------------+  |
            | | Metadata (Context Header(s))  |  |
            | +-------------------------------+  |
            +-----------------------------------+
            | Original Packet / Frame / Payload |
            +-----------------------------------+
```
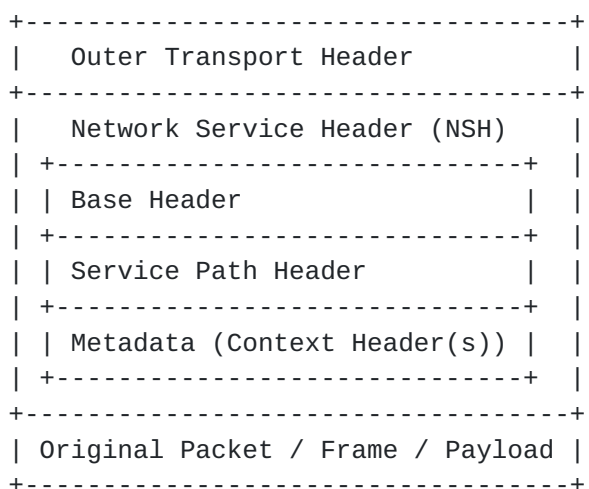
Figure 2. Data Encapsulation with the NSH

An SF can receive one or more SFC packets from an SFF and return to
it a larger or smaller number of SFC packets; that is to say, SFC
packets can be discarded or created by an SF.

2.1 The Network Service Header (NSH)

The NSH header is used to encapsulate and control the subsequent path
of traffic. It consists of three parts, the initial 32-bit Base
Header, the 32-bit Service Path Header, and any Context Headers
holding metadata, as shown in more detail in Figure 3 and specified
in [RFC8300].

The Base Header includes a Length field whose value is the overall
NSH length. Because the Base Header and Service Path Header are fixed
length, the length of the Metadata can be computed from this Length
field. The Base Header also includes a field indicating the type of
metadata in the NSH.

The Service Path Header consists of a Service Path Identifier (SPI)
and a Service Index (SI). The SPI identifies the logical path the
packet should follow while the SI indicates which step along that
path the packet is at.

An SF anywhere along a Service Function Path can re-classify an SFC
packet by replacing the Service Path Identifier (SPI) and Service
Index (SI) in the NSH. SFFs can also insert, delete, or change
metadata (Context Header(s)) in the NSH.

```
 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Ver|O|U|    TTL    |   Length  |U|U|U|U|MD Type| Next Protocol |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Service Path Identifier                     | Service Index |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Optional Context Headers / Metadata                 ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
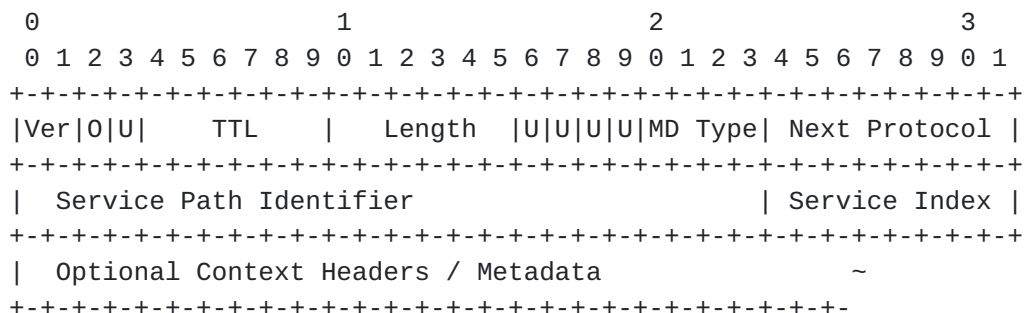
       Figure 3. Network Service Header Details from [RFC8300]

2.2 NSH Metadata and Variable Length Context Headers

If the MD Type field in the NSH Base Header has the value 1, there is
a single fixed length 128-bit Context Header whose format is not
further defined by the IETF. In that case, the NSH Length field has
the value 6.

If the MD Type field has the value 2, there are zero or more
Variable-Length Context Headers (VLCHs) as shown in Figure 4 at the
end of the NSH. The absence of any Context Headers is indicated by
using MD Type 2 and an NSH Length of 2. MD Type 0 is reserved and MD
Types 3 through 15 are unassigned.

```
 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Metadata Class                | Type          |U| Length     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Variable Length Metadata                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

                 Figure 4. Variable Length Context Header

   The minimum size for a VLCH is 32 bits consisting of the Metadata
   Class, Type, one unused bit, and Length as shown in Figure 4.

   The Metadata Class field is 16 bits, and its value specifies the
   organization under which the particular of VLCHs specified. Metadata
   Class zero is the IETF base class. The 8-bit Type field's value,
   along with the Metadata Class value, indicates the meaning of the
   Context Header and its Variable-Length Metadata. The size of the
   Length field is 7 bits and its value gives the length in octets of
   the Variable-Length Metadata that follows the initial fixed length
   portion of the VLCH. A VLCH with no Variable-Length Metadata is
   indicated by a Length field whose value is zero. VLCHs are padded so
   that they always start and end at a multiple of 4 bytes from the
   beginning of the NSH.

3. **Requirements for Parallelism and Diversions**

   There are requirements to split a Service Function Chain (SFC) into
   two or more parallel Service Function Paths (SFPs) that later rejoin
   as shown in Figure 5.

```
                         +------+      +-----+
                    ----->|SFF 2a|<---->| SF 2|
                   /       +------+\     +-----+
                  /                 \____
                 /                       \
      +-----+/         +------+        ->+-----+
  ---->|SFF 1|--------->|SFF 2b|-------->|SFF 3|----->
      +-----+\         +------+        ->+-----+
       /|\    \         /|\          /    /|\
        |      \         |          /      |
       \|/      \       \|/        |      \|/
      +-----+    \      +-----+    |     +-----+
      | SF 1|     \     | SF 3|    /     | SF 5|
      | DP  |      \    +-----+   /      | RP  |
      +-----+       \            /       +-----+
                     \  +------+/      +-----+
                      ->|SFF 2c|<---->| SF 4|
                        +------+      +-----+
```
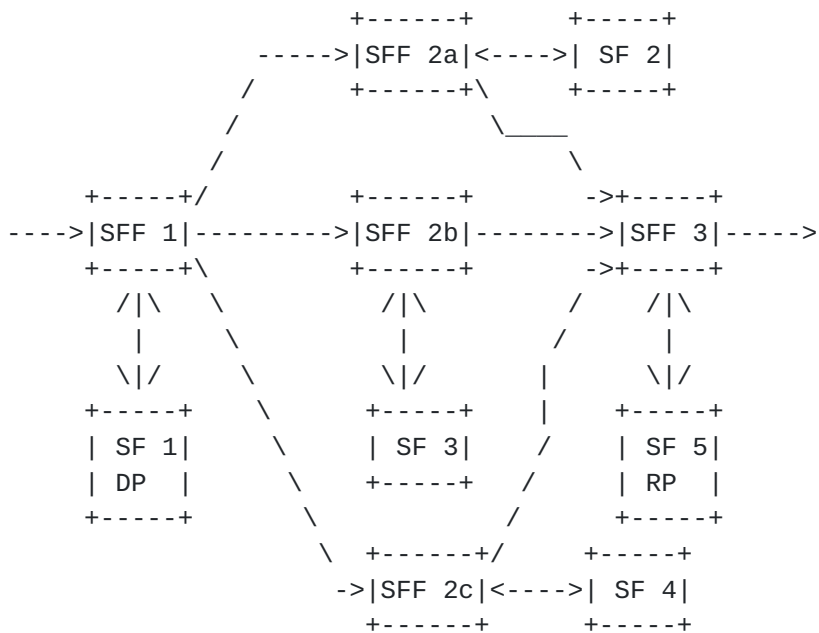
                   Figure 5: Parallel Service Function Paths

   For example, there may be two or more Service Functions (SFs) that
   can be performed in parallel with the goal, for time critical traffic
   such as some financial or gaming traffic, of delaying the stream of
   packets only by the amount of time taken by the slowest single SF; if
   the packets went through the SFs sequentially, the delay would be the
   sum of the times taken by each of the SFs. An example of such
   potential parallel processing might be that the SFs operate of
   different parts of the packet such as one SF operating on packet
   addressing while another operates on the information payload. Another
   example might be that one SF creates a signature or integrity code
   over parts of the packet to be inserted into the packet payload while
   another SF encrypts parts of the packet (or alternatively, they
   verify and decrypt in parallel).

   Another example of desirable parallelism would be improved
   reliability or accuracy if the SFs executed in parallel were
   unreliable or were different implementations of doing the same
   processing. For example, some quantum computers are currently
   unreliable so it would be desirable to perform some quantum process
   several times and compare the results to pick the most common value,

or a vote could be taken between the results of different
implementations of some process.

In Figure 5 it could be that any of the parallel paths could have
more or less than one SFF, although exactly one is shown in the
example for simplicity and any of the SFFs in any of the parallel
SFPs could process a packet through more than one SF, although they
are shown using only one SF in Figure 5 for simplicity. (Note that
while SFFs implement an SFP, SFPs logically consists of the sequence
of SFs. Thus, for example, an SFP could divert into multiple parallel
SFPs that rejoin at an RP all implemented through SFs off of one and
the same SFF.) It could also be that one or more of the parallel
paths would themselves further split into parallel paths and so on.

There are cases where it is desirable to divert an SFP so as to
splice one or more added SFs into that SFP or to divert it so as to
slice out one or more sequential SFs that were downstream in that
SFP, as shown in Figures 6 and 7. Although SF 3 in each of those two
cases could re-classify the packet with a new SPI and SI that include
the remainder of the new diverted path, this would require that a new
SFP with this new SPI already be configured in all the SFFs for the
remainder of the Service Function Path after a diversion. In the case
of Figure 7, SF 3 could possibly just adjust the Service Index, but
this would require relaxing any checking at SF 5 of the SPI/SI or
source address of packets on the main SFP or may otherwise be
undesirable.

```
                         +-----+        +-----+
                        ->|SFF 3| <---->| SF 4|
                       /   +-----+\      +-----+
                      /           \
      +-----+        +-----+          ->+-----+       +-----+
   --->|SFF 1|----->|SFF 2| - - - - - >|SFF 4|----->|SFF 5|--->
      +-----+        +-----+           +-----+       +-----+
       /|\            /|\               /|\           /|\
        |              |                 |             |
       \|/            \|/               \|/           \|/
      +-----+        +-----+           +-----+       +-----+
      | SF 1|        | SF 3|           | SF 5|       | SF 6|
      +-----+        | DP  |           | RP  |       +-----+
                     +-----+           +-----+
```

                    Figure 6: Splicing in One or More SFFs

```
                           ------------\
                         /              \
                       /                  \
         +-----+      +-----+      +-----+   \->+-----+
   --->|SFF 1|----->|SFF 2|- - ->|SFF 4|- - ->|SFF 5|--->
         +-----+      +-----+      +-----+      +-----+
          /|\          /|\          /|\          /|\
           |            |            |            |
          \|/          \|/          \|/          \|/
         +-----+      +-----+      +-----+      +-----+
         | SF 1|      | SF 3|      | SF 5|      | SF 5|
         +-----+      | DP  |      +-----+      | RP  |
                      +-----+                   +-----+
```
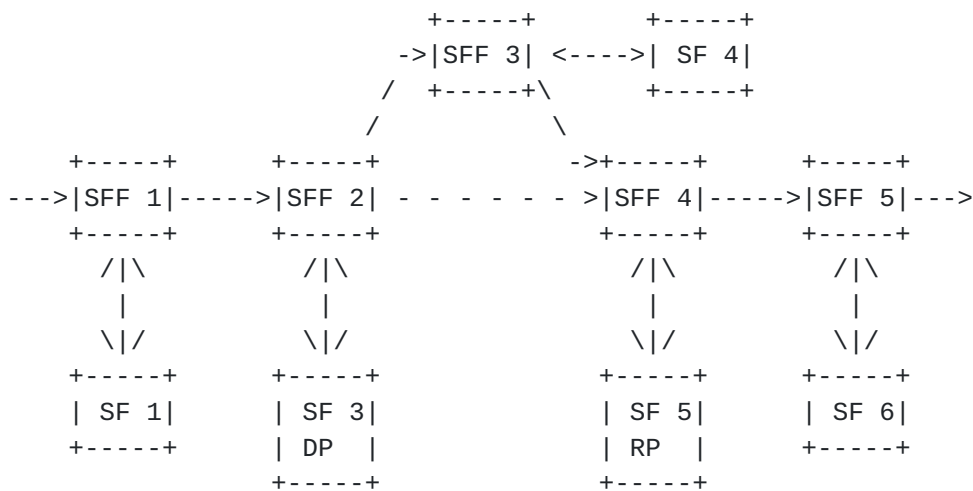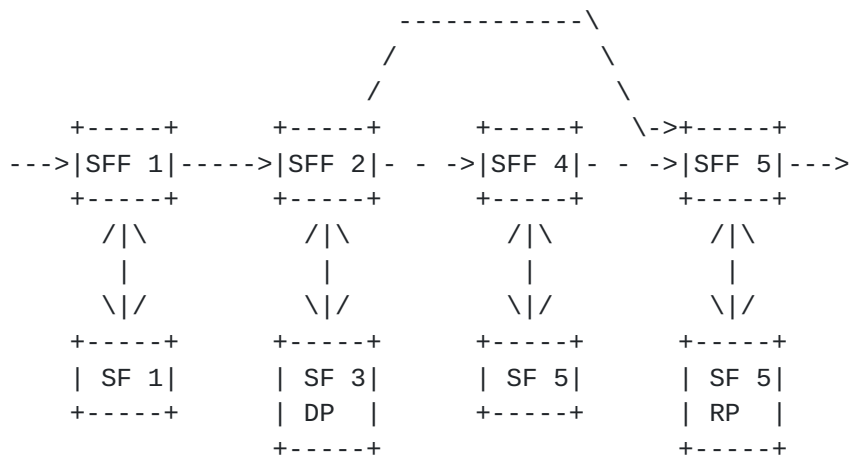
Figure 7: Splicing out One or More SFFs

Combinations of the cases shown in Figures 5, 6, and 7 may be needed
where a diversion such as in Figures 6 or 7 occur within a parallel
path as in Figure 5 or parallel paths as in Figure 5 occur within a
diversion as in Figure 6. Generalizing Figure 6 and 7, a diversion
might splice in a path with some number of SFs that cuts out a
portion of the original SFP that had some number of SFs.

Although DPs and RPs are logically Service Functions (SFs) and shown
as separate boxes in the above figures, like any other SF they can be
implemented as co-located with an SFF.

4. Diversion Points and Rendezvous Points

SF 1 in Figure 5 and SF 3 in Figures 6 and 7 are referred to as
Diversion Points (DPs) because they are nodes at which an SFP is
diverted to one or more SFPs with new SPIs that are intended to
rejoint/return to the original SPI at a downstream Rendezvous Point.
SF 5 in Figures 5, 6, and 7 is referred to as a Rendezvous Point (RP)
because it is the node at which one or more SFPs from an upstream DP
rejoin an original SFP with an original SPI. The coresponding packets
so received at an RP are merged or coordinated.

In general, an RP needs to be configured to expect SFC packets to
arrive at that RP on diverted SFPs. An RP may need additional
information in the SFC packets, as discussed in Section 4.1, to be
included in their NSH. Divergence point behavior is discussed in
Section 4.2 and RP behavior is discussed in Section 4.3.

4.1 Rendezvous Point Information (RePIn)

To recombine packets from divergent SFP(s) at a Rendezvous Point
(RP), or rejoin a diverted SFP to the original SFP, additional
information may be needed in the packets. This is accomplished
through inclusion of the RP Information (RePIn) Variable Length
Context Header (VLCH), as shown in Figure 8, in the packet's NSH.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Metadata Class = 0x0000       | Type=TBD      |U| Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Saved Service Path Identifier                 | Saved SI      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Variable Length Sub-TLVs                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
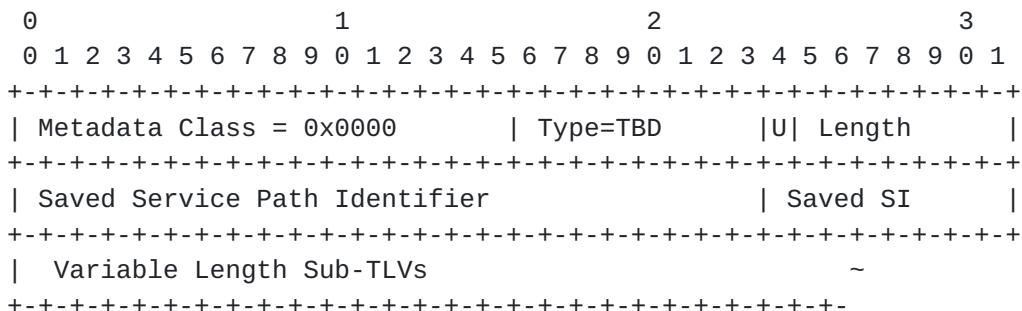
Figure 8: RePIn VLCH

The Length field is the total length of the Variable Length Sub-TLVs
in octets.

The Saved Service Path Identifier and Saved SI are the SPI and SI in
the NSH of the SFC packet being diverted after entry to the DP SF and
the SI has been decremented.

The Variable Length Sub-TLVs consist of zero or more RePIn VLCH Sub-
TLVs. The format of a RePIn VLCH Sub-TLV is as shown in Figure 9

except for Sub-Type 1 as discussed in [Section 4.1.1](); however, all

RePIn VLCH Sub-TLVs are padded at the end up to an even multiple of 4
octets.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type      |X|Sub-Length   | Variable Length Metadata    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
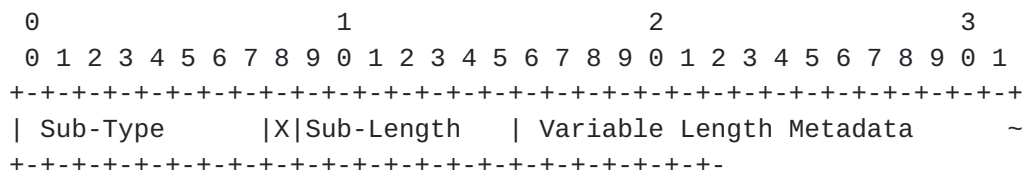
                Figure 9: General RePIn VLCH Sub-TLV

The Sub-Type field is an 8-bit unsigned integer that is always
present and indicates the format of the rest of the Sub-TLV. The X
bit may be assigned a meaning for particular Sub-Types; if no such
meaning is assigned for a particular Sub-Type, the X bit MUST be sent
as zero and ignored on receipt. Sub-Length is an unsigned integer
giving the length of the variable length metadata in octets.

Unless the specification for a RePIn VLCH Sub-TLV Sub-Type specifies
that there may be multiple occurrences of that Sub-TLV, it may only
be included once. If there are multiple instances, the first
occurrence is used with any subsequent occurrences being ignored.

4.1.1 Packet Identifier

When an SFC packet is replicated and diverted to more than one
parallel path to be merged back together at a Rendezvous Point (RP),
a method of matching packets is needed such as labeling each copy
that originated with the same packet before the split using a packet
identifier such as a packet counter, fine grained time stamp, or hash
code. Such an identifier might already exist in the packet, for
example a TCP sequence number. If not, it is included as a VLCH
special Sub-TLV as shown below. Use of a packet counter is
RECOMMENDED.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type=1    | Packet Identifier                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
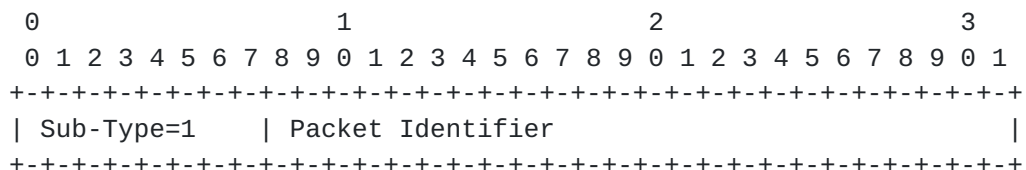
               Figure 9: Packet Identifier Special Sub-TLV

Because this is expected to be a common RePIn VLCH Sub-TLV, in order
to save octets, for this Sub-Type only, the "Sub-TLV" X and Sub-

Length fields are subsumed into the Packet Identifier field.

4.1.2 **Packet Extent Modified**

   If two or more SFPs used in parallel have modified parts of a packet,
   the RP may need additional information to be able to recombine the
   different copies of the packet it will be receiving. As an example of
   the complexities involved, an SF could change the length of part of a
   packet in a way dependent on the content of that part such as by
   applying a data compression or de-compression algorithm to part of
   the packet or by conditionally inserting or removing a VLAN tag
   depending on addressing information.

   In simple cases such as parallel SFPs that modify fixed size disjoint
   parts of a packet without changing the size of those parts, it may be
   possible for an RP to be configured to recombine the results without
   added information. But in more complex or variable length cases,
   parallel SFPs need to add information as to what part of the original
   packet they modified and how this may have changed the length of that
   part. Also, with such additional information, in some cases only one
   of the parallel SFPs would need to forward all of the original packet
   with modifications to the RP; one or more other parallel SFPs could
   just forward their modified part and the RP would be able to
   recombine the results thus saving communications link capacity that
   would be used if they all sent full packets.

```
    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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Sub-Type=2    |X|Sub-Length=6 | Offset                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Priority      | Original Size         | Modified Size         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                  Figure 10: Packet Extent Modified Sub-TLV

   If the X bit is zero, the entire modified packet is present in the
   SFC packet. If the X bit is a one, only the modified portion appears
   in the packet which requires any SFs between the SF that modified the
   packet and the RP to be capable of handling such an abbreviated
   packet.

   Offset is the number of bits between the end of the NSH or the last
   NSH if there are multiple stacked NSHs and the portion of the packet
   being modified. Original Size and Modified Size are the size in bits
   of the portion being modified before and after that modification. Any
   of the Offset, Original Size, and Modified Size fields may have the
   value zero.

   If parallel SFPs have modified the same or overlapping parts of a

packet, the RP may need some way to resolve this conflict which could

include a relative priority for changes made by different SFs
configured at the RP and/or indicated in the RP Information (RePIn)
or from other sources. The Priority field MAY be used for this
purpose; it contains an unsigned integer where a larger magnitude
value indicates a higher priority that would prevail over a lower
priority. If not used, the Priority field MUST be sent as zero and
ignored on receipt.

If a path has modified more than one portion of a packet, multiple
instance of the Packet Extent Modified Sub-TLV can be included in the
RePIn VLCH. If any Packet Extent Modified Sub-TLV Sub-Lenth is any
value other than 6, the meta-data is corrupt, the packet is silently
discarded, and an error SHOULD be logged.

4.1.3 Saved Metadata

A DP may need to save Metadata so it will not be seen inside a
diversion and will be restored at the RP. This Sub-TLV is used for
that purpose. See Section 4.2 and 4.3 for further details on the use
of this sub-TLV.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Sub-Type=3    |X| Sub-Length  | MBZ                   |MetaTyp|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Variable Length Saved Metadata                 ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
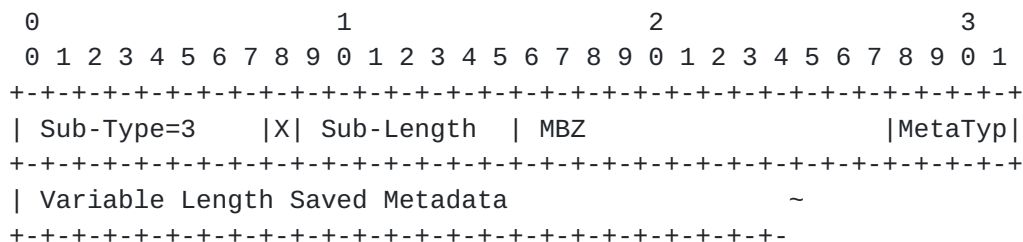
                  Figure 10: Saved Metadata Sub-TLV

The X and MBZ bits MUST be sent as zero and ignored on receipt.
MetaTyp is the MD Type of Metadata saved. The presence of the MBZ
field causes the saved metadata to be aligned on a multiple of 4
octets.

4.1.4 Saved TTL

The TTL limits the number of SFs that can be traversed between
ingress and egress. The packet is discarded if the TTL is exhausted.
This is a safety measure to defend against infinite or very large
loops due to malfunctions, configuration error, or other reasons.
Thus, the RECOMMENDED mode of operation is to use a TTL value that is
decremented continuously from original SFC domain ingress to final

SFC egress including throughout any diversions. If the TTL is reset

on entry to a diversion, then the Saved TTL Sub-TLV MUST be used so
that the previous TTL can be restored at the diversion's RP.

Note that resetting the TTL on entry to a diversion opens the
possibility for loop where a diversion diverts to itself or there are
two diversions X and Y where X diverts to Y and Y diverts to X or
more complex scenarios all of which are made safe by using a
continuous TTL and unsafe by resetting the TTL on diversion entry.
Such loops will result in a growing amount of metadata which might
safely lead to packet discard or unsafely cause repeated
fragmentation.

If, despite the above warning, it is desired to reset the TTL at the
DP and restore it at the RP, the Saved TTL Sub-TLV as shown below is
used.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Sub-Type=4    |X| Sub-Length=2| MBZ           | Saved TTL     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
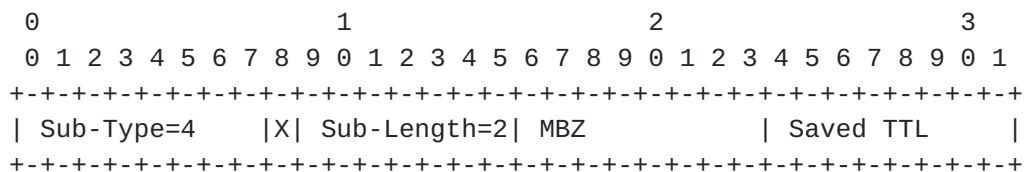
Figure 11: Saved TTL Sub-TLV

The X and MBZ bits MUST be sent as zero and ignored on receipt. The
Saved TTL is the value of the TTL field copied from the NSH after its
initial decrement on entering the DP SF. If the Saved TTL Sub-TLV
Sub-Lenth is any value other than 2, the meta-data is corrupt, the
packet is silently discarded, and an error SHOULD be logged.

4.2 Diversion Point (DP) Behavior

If it is desired to simply skip some SFs in an SFP, a diversion may
not be necessary. The SI can simply be decreased to that for the next
SF desired if the SFF to which the SF/DP that reduces the SI returns
the packet can handle that reduced SI value and forward the packet to
the appropriate SFF. Otherwise, the procedure below in this section
is used and this procedure MAY be used even in cases where simple
reduction of the SI would work.

If the RP can recognize diverted SFC packets and modify/merge them
appropriately to restore them to the original SFP with appropriate
Metadata, then inserting a RePIn VLCH might not be needed. In other
cases take the steps below. This is a logical procedure and any
procedure can be used that results in the same diverted SFC
packet(s).

1. Construct a RePIn VLCH containing the SPI and SI from the NSH and
   then change those fields in the NSH to the diversion SPI and SI.
   If diversion is to multiple parallel SFPs, make a copy of the SFC
   packet for each diversion, then construct a VLCH and modify the
   NSH as in the previous sentence for each one of the parallel
   paths. Then perform the following steps to each modified copy and
   its corresponding RePIn VLCH.

2. If it will be necessary for the RP to merge the modified copies of
   the original SFC packet sent over parallel paths or if the RP
   needs to restore a particular ordering to packets, then add a
   Packet Identification Sub-TLV to the RePIn VLCH unless sufficient
   information is available in the packet payload for the RP to do so
   without a Packet Identification Sub-TLV.

3. Take any Metadata from the original NSH that should be restored at
   the RP and add it to the RePIn VLCH as a Saved Metadata Sub-TLV.
   If the NSH already has any RePIn VLCHs, they need not be saved as
   they will be masked by the new RePIn VLCH that will be inserted
   before them (this indicates that a diversion from a diversion is
   being created). To save space, any Metadata that has been saved in
   the RePIn VLCH and is not needed in the diversion SFP SHOULD be
   removed from the NSH if MD Type 2 and MUST be removed from the NSH
   if MD Type 1. (In the Type 1 case, this converts the NSH to MD
   Type 2 with no Metadata.)

4. If it is desired to use a new value for the NSH TTL in the
   diversion, with the old value restored at the RP, add a Saved TTL
   Sub-TLV to the RePIn VLCH and set the TTL in the NSH to a
   configured value which may be dependent on the diversion being
   entered. This is NOT RECOMMENDED as discussed in [Section 4.1.4](#).

5. The RePIn VLCH constructed as above is inserted into an NSH as in
   the subpoints below:
   5.a If, after the above step 3, the initial NSH in the SFC packet
       is MD Type 2, insert the RePIn VLCH constructed above before
       any existing RePIn VLCH in the NSH.
   5.b If, after the above step 3, the initial NSH in the SFC packet
       is MD Type 1, this implies that there is Type 1 metadata that
       may be needed by one or more SFs in the diversion. If the SFs
       in the diversion can handle stacked NSHs, insert an MD Type 2
       NSH copied from the initial NSH except for metadata, after the
       initial MD Type 1 NSH to hold the RePIn VLCH. Handling stacks
       NSHs means the SF (or its proxy) can parse through them to
       find the needed metadata and the payload to operate on and, if
       the SF generates packets, can create them with appropriate
       stacked NSHs. If the SFs in the diversion cannot handle
       stacked NSHes, the creation of the diversion is beyond the

scope of this document.

6. Perform such other functions or modifications to the metadata or
   other parts of the SFC packet as are appropriate based on the
   saved or new SPI or other factors.

7. Transmit the modified packets.

The addition of Metadata and possible additional NSH header (see step
5.b above) may lead to fragmentation or decreased payload Maximum
Transmission Unit (MTU) in some networks.

[4.3](#) Rendezvous Point (RP) Behavior

A RP will have been configured to know the SFC packet SPI and SI
values in diverted packets for which it is to perform the RP service.
The SI is decremented when an SFC packet is received by an SF; for an
RP this might decrement the SI to zero. The RP performs the steps
below. If the RP can restore diverted SFC packets to their former SFP
and, to the extent necessary, match and merge diverted packets
received over parallel paths and correctly order the resulting SFC
packets, without the presence of a RePIn VLCH, it does so and the
remainder of this section is inapplicable. If not, the following
logical procedure or any procedure resulting in the same SFC packet
is used.

1. Steps 2 and 3 below are performed on each diverted packet received
   by the RP. If the RP is merging parallel diversions, step 4 is
   then performed on the set of matching packets. In this case and
   any case where the RP should restore packet order, the RP must be
   prepared to buffer packets until they can be processed and
   forwarded. Overflow of such a buffer will result in lost packets
   and SHOULD be logged as an error. How long to wait for missing
   diverted packets and what action to take if it is decided they
   have been lost are application and implementation dependent.
   Finally, Step 5 is performed.

2. Find and remove the first RePIn VLCH in the diverted packet. This
   is referred to below as the removed VLCH. It might be in a second
   stacked NSH if the initial NSH has MD Type 1.

3. Restore the packet from the diversion through the sub-steps listed
   below.
   3.a If there is a Saved TTL in the removed VLCH, restore the old
       TTL into the initial NSH.
   3.c Restore the saved SPI and SI from the removed RePIn VLCH into
       the initial NSH.
   3.d Restore metadata as follows:
       3.d.1 Restore any MD Type 2 Saved Metadata from the removed

VLCH into the NSH from which that VLCH was removed.

          3.d.2 If there is MD Type 1 Saved Metadata in the removed VLCH
               and there is an initial MD Type 1 NSH in the packet,
               replace the MD Type 1 metadata with the saved MD Type 1
               metadata.
          3.d.3 If there is MD Type 1 Saved Metadata in the removed VLCH
               and there is an initial MD Type 2 NSH in the packet,
               insert a new initial NSH into the packet which is a copy
               of that MD Type 2 NSH except that it is MD Type 1 with
               the saved MD Type 1 metadata.
      3.e If, at this point, the packet starts with an MD Type 1 NSH
           followed by an MD Type 2 NSH with no metadata, remove that 2nd
           NSH.

  4. Match up SFC packets arriving at the RP through parallel paths
     using the Packet Identification Sub-TLV in the removed RePIn VLCH
     or using some other technique. For each matching set, perform the
     sub-steps below. Arbitrarily select one of the matching diverted
     packets to modify into the merged packet unless configured to use
     some particular diverted packet such as the one received over a
     particular diversion. This is referred to below as the merged
     packet even before the merger is complete.
     4.a For error checking, any saved SPI and SI in the matching
        packets SHOULD be compared and an error logged if they are not
        identical.
     4.b For safety, it is RECOMMENDED that the minimum NSH TTL from
        the parallel SFC packets be copied into the merged packet.
     4.c Depending on the application and implementation, the remaining
        metadata in the merged packet may be used or updated based on
        the remaining Metadata in the other packets being merged. How
        to do this is beyond the scope of this document.
     4.d The payloads of the other packets being merged, that is the
        portion after any NSHs, are used to update the payload in the
        merged packet. This may be based on RP configuration for the
        application or Packet Extent Modified Sub-TLVs in the removed
        RePIn VLCHs or a combination of these.

   5. Perform such other functions or modifications to the metadata
      or other parts of the SFC packet as are appropriate based on
      the saved or new SPI or other factors.

   6. Trasnmit the merged packet.

5. **IANA Considerations**

   The following subsections provide IANA assignment considerations.

5.1 **Variable Length Context Header Type**

   IANA is requested to assign a variable length context header type
   from the "NSH IETF-Assigned Optional Variable-Length Metadata Types"
   registry as follows:

Value	Description	Reference
TDB	Rendezvous Point Information (RePIn)	[this document]

5.2 **RePIn VLCH Sub-Types**

   IANA is requested to create a sub-registry under the "NSH IETF-
   Assigned Optional Variable-Length Metadata Types" registry as
   follows:

   Name: Sub-TLVs under the Type TBD Variable Length Context Header
   Registration Procedure: Expert Review
   Reference: [this document]

Sub-Type	Description	Reference
0	reserved	[this document]
1	Packet Identifier	[this document]
2	Packet Extent Modified	[this document]
3	Saved Metadata	[this document]
4	Saved TTL	[this document]
5-254	unassigned	[this document]
255	reserved	[this document]

6. Security Considerations

For general SFC and NSH security considerations, see [RFC8300].

More TBD...

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, <http://www.rfc-editor.org/info/rfc2119>.

    [RFC8174] - Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
          2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May
          2017, <http://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>.

Informative References

    [ITU_Liaison] - ITU-T SG11, "LS on recent service function chaining
          related developments in Q4/SG11: two new draft Supplements",
          ITU-T SG11-LS-179, March 2021,
          <https://datatracker.ietf.org/liaison/1736/>.

    [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>.

    [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>.

Appendix: Relation to Hierarchical SFC

   Experimental [RFC8459] describes "Hierarchical SFC" in which SFs in a
   higher level SFP can be entire lower level SFPs with a different SPI
   and where the higher level SPI is restored at the end of the lower
   level SFP. This is similar to a diversion in this document. The
   Internal Boundary Nodes (IBNs) in [RFC8459] that transition an SFC
   packet between the higher and lower levels are similar to DPs/RPs in
   the terminology of this document.

   Experimental [RFC8459] discusses a wide variety of mechanisms to
   implement Hierarchical SFC while this document looks toward
   specifying a more specific set of mechanisms as a Proposed Standard
   to support parallelism and other types of diversions.

Acknowledgements

   The authors gratefully acknowledge the comments and suggestions of
   the following persons:

      TBD

Authors' Addresses

      Donald E. Eastlake, 3rd
      Futurewei Technologies
      2386 Panoramic Circle
      Apopka, FL 32703 USA

      Tel: +1-508-333-2270
      Email: d3e3e3@gmail.com

Copyright and IPR Provisions