Authors: J. Rajamanickam, Ed.   R. Gandhi, Ed.
         Cisco Systems, Inc.   Cisco Systems, Inc.
         J. Bhattacharya   B. Decraene   R. Zigler
         Cisco Systems, Inc.   Orange   Broadcom
         W. Cheng   L. Jalil
         China Mobile   Verizon

MPLS Extension Header Encodings

Abstract

This document uses the Multiprotocol Label Switching (MPLS) Entropy
Label (EL) extensions defined in draft-decraene-mpls-slid-encoded-
entropy-label-id or a new Special Purpose Label to indicate the
presence of MPLS Extension Header (MEH) in an MPLS label stack. It
defines different MPLS Extension Header encoding formats to carry
additional data in the MPLS label stack that can influence
forwarding decision and to carry additional data after the Bottom of
the MPLS label stack.

Status of This Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six
months and may be updated, replaced, or obsoleted by other documents
at any time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 September 2022.

Copyright Notice

Table of Contents

1. Introduction

[[RFC3032](#)] defines MPLS Header for carrying a stack of MPLS labels
which are used to forward packets in an MPLS network. Today's new
applications require the MPLS packets to carry some additional
indicators and associated ancillary data that would be used in MPLS
packet forwarding decision or for OAM purpose.

Each application requires a separate Extended Special Purpose Label
(eSPL) to address its problem that adds 2 extra labels (extension
label 15 + eSPL) in the MPLS label stack. This approach does not
scale, as it increases the label stack depth with multiple eSPLs
that need to be imposed by the encapsulation node and scanned by the
intermediate nodes. Also, currently there are no solutions defined
to add ancillary data in a label stack or add multiple ancillary
data after the Bottom Of Stack (BOS) in an MPLS packet. Ancillary
data can be used to carry additional information, for example, a
network slice identifier, In-Situ OAM (IOAM) data presence
indicator, etc. Some of these use-cases are described in [[I-D.saad-mpls-miad-usecases](#)].

This document defines a new MPLS data plane extension header format
to efficiently encode forwarding and OAM instructions those are easy
to process in hardware. The instructions are encoded in the form of
flags and opcodes and can be carried without associated ancillary
data or with short in-stack ancillary data or with one or more
ancillary data after the BOS.

MPLS Entropy Label (EL) standard is defined in [[RFC6790](#)]. This
document uses the Entropy Label extensions defined in [[I-D.decraene-mpls-slid-encoded-entropy-label-id](#)] or a new Special Purpose Label
(SPL) to indicate the presence of MPLS Extension Header (MEH) in an
MPLS label stack. It defines different MPLS Extension Header
encoding formats to carry additional data in the MPLS label stack
that can influence forwarding decision and to carry additional data
after the Bottom of the MPLS label stack.

1.1. Requirements

This document defines different MPLS Extension Header encoding
formats to support the following requirements:

1. MPLS packet to carry additional data in the MPLS label stack to
   influence forwarding. This can be of two types:

   1a. Forwarding Instruction Flags (FIF) that does not use
   additional data.

   1b. Forwarding Instruction (FI) that needs additional data.

2. MPLS packet to carry additional data after the Bottom of the MPLS
   Label Stack.

3. Any combination of (1) and (2) in the same MPLS packet.

When MPLS Extension Header is added in an MPLS Label stack, the
extension header MUST NOT contain the label field that can conflict
with any previously allocated reserved label value. [I-D.bocci-mpls-
miad-adi-requirements] describes additional requirements for MPLS
Extension Header.

2. Conventions Used in This Document

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119] [RFC8174]
when, and only when, they appear in all capitals, as shown here.

2.2. Terminology

BOS (Bottom Of Stack): Bottom of the MPLS label stack.

BOS-FI (Bottom Of Stack Forwarding Instruction): This is the
Forwarding Instruction that is encoded after Bottom of MPLS Stack.

BPI (Bottom of the Stack MPLS Extension Header Presence Indicator):
This is the flag to indicate the presence of MPLS Extension Header
after the bottom of the MPLS label stack.

E2E (Edge-To-Edge): Edge to Edge.

EL (Entropy Label): Entropy Label defined as per [RFC6790].

ELC (Entropy Label Control): EL TTL field re-purposed to carry
Entropy Label control bits defined in [I-D.decraene-mpls-slid-
encoded-entropy-label-id].

FI (Forwarding Instruction): Forwarding Instruction is the
instruction that expresses the forwarding behaviour. This can result

in changing the forwarding decision or adding some information or important data to the packet.

FIF (Forwarding Instruction Flags): A bitwise flag that influences the forwarding behaviour. This flag does not need any additional data to execute its FI.

FIOC (Forwarding Instruction Opcode): A Opcode value that refers to a specific Forwarding Instruction.

HBI (Hop-By-Hop Bottom of the Stack MPLS Extension Header Presence Indicator): This is the flag to indicate the presence of MPLS Extension Header after the bottom of the MPLS label stack that require Hop-By-Hop processing.

IS-FI (In-Stack Forwarding Instruction): This is the Forwarding Instruction that is encoded in the MPLS label stack.

IPI (In-Stack MPLS Extension Header Presence Indicator): This is the flag to indicate the presence of MPLS Extension Header in the MPLS label stack.

MEI (MPLS Extension Indicator): This is the Indicator MPLS Label which indicates the presence of MPLS Extension Header in the MPLS Label stack.

MEH (MPLS Extension Header): MPLS Extension Header encoding carried in the MPLS Label stack.

MPLS (Multiprotocol Label Switching): Multiprotocol Label Switching.

NPL (Network Programming Label): Network Programming Label provisioned by user.

SPI (Slice ID Presence Indicator): This is the flag to indicate the presence of Slice ID in the Entropy Label field.

SPL (Special Purpose Label): IANA Allocated Special Purpose Label in the range of 0-15. Extended Special Purpose Label (eSPL) uses label value 15.

TC (Traffic Class): Traffic Class.

TTL (Time-To-Live): Time To Live.

3. Overview

Extending existing MPLS Header needs two main parts.

   *MPLS Extension Header Indicator (MEI) - This is a way to indicate
    the presence of MPLS Extension Header in the packet. This could
    be done using two different methods. Each method has its own
    advantages and disadvantages. This document describes both
    options of MEI. The encoding formats defined in this document are
    compatible with both options of MEI.

    Option 1. MEI by extending ELI/EL

    Option 2. MEI by using a New Special Purpose Label (SPL)
    allocated by IANA

    Option 3. MEI by using New Network Programming Label (NPL)
    provisioned by user

   *MPLS Extension Header Format - The format in which the MPLS
    Extension Header could be carried in the MPLS packet. This
    includes both In-stack Extension Header and BOS Extension Header.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | TC  |S|      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                      Figure 1: MPLS Label Format

  New In-Stack (IS) MPLS Extension Header format is defined in this
  document to carry the In-stack Forwarding Instruction and
  corresponding data in the MPLS label stack.

   *It uses MPLS Label field to carry the Forwarding Instruction
    Opcode.

   *It uses Traffic Class (TC) field to identify the Length of the
    MPLS In-Stack Extension Header size.

   *It uses MPLS Label and Time-To-Live (TTL) fields to carry the In-
    Stack data (can be Flags or data).

  A new Bottom Of Stack (BOS) MPLS Extension Header format is defined
  in this document to carry the BOS Forwarding Instruction and
  corresponding data after the MPLS Label stack.

The MPLS Extension Header encoding formats defined in this document
are flexible and allow to stack multiple In-Stack and BOS MPLS
Extension Headers in a desired order in the same MPLS packet.

3.1. Option 1 - ELC as MPLS Extension Header Indicator

As described in [I-D.decraene-mpls-slid-encoded-entropy-label-id],
the EL's 8-bit TTL field is re-purposed as Entropy Label Control
(ELC) field. One bit from ELC is requested for the Slice ID Presence
Indicator (SPI) and the 7 bits are available for use. From the ELC,
3 bits (for IPI, BPI and HBI) are allocated to indicate the presence
of MPLS Extension Header.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Entropy Label Indicator (7)       | TC  |S|      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Slice ID    |  Entropy Label         | IL  |S|  ELC (SPI=1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ELI/EL Packet Format

TTL (in ELC) bit allocations are defined by user as follows:

Bit Position	Description
TBD0	SPI - Slice ID Presence Indicator: Indicate the presence of Slice ID in the Entropy label as defined in [I-D.decraene-mpls-slid-encoded-entropy-label-id].
TBD1	IPI - In-Stack Extension Header Presence Indicator: Indicate the presence of In-Stack MPLS Extension Header after this label.
TBD2	BPI - Bottom Of Stack Extension Header Presence Indicator: Indicates the presence of MPLS Extension Header after the Bottom Of Stack (BOS).
TBD3	HBI - Hop-By-Hop Bottom Of Stack Extension Header Indicator: Indicates the MPLS Extension Header after the Bottom Of Stack requires Hop-By-Hop processing.
TBD4 - TBD7	Unassigned Bits.

Table 1: Bit Fields

IL - In-Stack Extension Header Length - The 3-bit TC field in the EL
is used to indicate the length of the In-Stack MPLS Extension Header
(excluding the ELI and EL labels) in terms of number of 32-bit
labels. If more that 7 labels are needed in an MPLS extension

header, the node can either use a BOS extension header to carry the
data or use an additional In-stack MPLS Extension Header with MEI
Label.

For backwards compatibility, an intermediate and decapsulating nodes
MUST only read the length from the TC field when the IPI (In-Stack
Extension Presence Indicator) is set to "1".

3.1.1. Advantages with ELC

Faster deployment in an existing network that has EL already
deployed with an incremental benefit (e.g., incremental signaling
extension for ELI capability).

Single label for Entropy in the MPLS header which helps with keeping
label stack size smaller.

When EL is already enabled in the network, the proposed scheme does
not require hardware to support an additional SPL indicator.

Save a new Special Purpose Label and related protocol extensions to
signal its capability in LDP, RSVP-TE, BGP, IS-IS, OSPF, BGP-LS,
etc.

An intermediate node can compute ECMP hash with the EL field and
avoid inconsistent load-balancing of traffic flow that can happen
when MPLS Extension Header alters the label stack.

Reduce MPLS Label stack size when EL is enabled for ECMP hashing
when MPLS Extension Header is also used. As there is only one field
for EL in the MPLS Header, it simplifies the MPLS header processing.

3.2. Option 2 - New SPL as MPLS Extension Header Indicator

The MPLS Extension Header encoding formats defined in this document
is equally applicable when using a new Special Purpose Label (SPL)
(value TBA1) allocated by IANA.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MEI=SPL (TBA1)                 | IL  |S| IPI,BPI,HBI   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

        Figure 3: New SPL as MPLS Extension Header Indicator

The TTL field in the SPL (value TBA1) is used to encode FI Flags
including IPI, HBI and BPI flags defined in this document. The

definition and meaning of these flags and IL field are exactly the
same as those in ELC field.

3.3. Option 3 - NPL as MPLS Extension Header Indicator

The MPLS Extension Header encoding formats defined in this document
is equally applicable when using a Network Programming Label (NPL)
configured by an operator.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MEI=NPL                          | IL  |S| IPI,BPI,HBI   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

             Figure 4: NPL as MPLS Extension Header Indicator

The TTL field in the NPL is used to encode FI Flags including IPI,
HBI and BPI flags defined in this document. The definition and
meaning of these flags and IL field are exactly the same as those in
ELC field.

4. In-Stack MPLS Extension Header Encoding

This section describes the encoding format of the MPLS Extension
Header carried as part of the MPLS label stack. The encoding format
defined is flexible (e.g., stackable opcodes in desired order),
extensible (by defining new Opcodes) and ASIC friendly (by using
Extension Header Length, Opcode+Data in the same field).

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Entropy Label or SPL or NPL     | IL=1|S| IPI=1          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  IS-FI Opcode |    In-Stack Data    |R|D|E|S| In-Stack Data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

               Figure 5: In-Stack Extension Header Format

IPI flag is set to "1" to indicate the presence of In-Stack MPLS
Extension Header.

Since In-Stack MPLS Extension Header is present as part of the MPLS
Label stack, the 32-bit MPLS Label is redefined to encode the MPLS
Extension Header as follows:

Label Field:

The first 8 bits are used to define the In-Stack Forwarding
Instruction (IS-FI) Opcode. Next 12 bits in the Label field and the
8 bits from the TTL field are used to carry In-Stack data
corresponding to the IS-FI opcode. This opcode ranges from 1 to 255.
IS-FI Opcode value of 0 is marked as invalid to avoid the label
value aliasing with the reserved SPLs.

  *IS-FI Opcode Value:1 - IANA Allocated to carry the Forwarding
   Instruction Flags (FIF).

  *IS-FI Opcode Value:2 - IANA Allocated to indicate the offset in
   terms of number of bytes for the start of the BOS data after the
   MPLS Label Bottom of the Stack. This can allow to carry Generic
   Control Word (0000b) [RFC4385] and G-ACh (0001b) [RFC5586] fields
   immediately after the BOS. Adding of this opcode is not required
   when the BOS data starts immediately after the Bottom of the
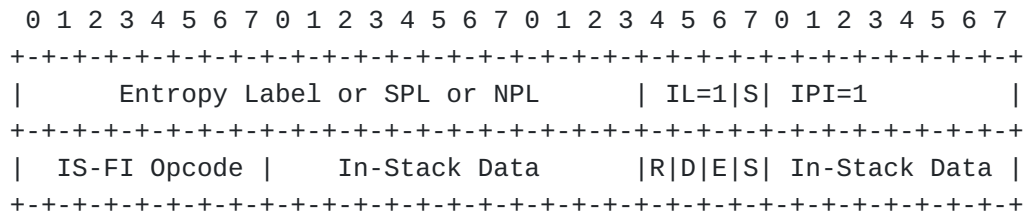   Label Stack (i.e. when offset is 0).

  *IS-FI Opcode Value:3-254 - MUST be assigned by IANA.

  *IS-FI Opcode Value:255 - IANA Allocated for IS-FI Opcode range
   extension. This gives the extensibility for opcode range beyond
   255.

IS-FI Opcode MUST define the following procedure before it can be
used:

  1. Define the Data format encoded in the MPLS extension header.

  2. Define the Hop-By-Hop or Edge-To-Edge (only on the
  decapsulation node) processing scope.

  3. The Hop-By-Hop IS-FI opcodes MUST be placed before the Edge-
  To-Edge IS-FI Opcodes in the MPLS Extension Header of the packet
  to optimize the Hop-By-Hop processing in hardware.

TC Field:

This field is used to indicate the MPLS Extension Header stacking
and In-Stack Data stacking.

  E (E2E-Bit): MPLS Extension Header In-Stack Data requires E2E
  processing. If this is set to "1", then this 4-byte MPLS
  Extension Header requires Edge-To-Edge processing. If this is set
  to 0, then this 4-byte MPLS Extension Header requires Hop-By-Hop
  processing. Note that E2E-Bit is not used with the Entropy Label
  TC field.

D (DS-Bit): Data Stacking Bit. This is used to encode more than
20 bits of data for this IS-FI Opcode. If this is set to "1",
then this is the end of the data for the IS-FI Opcode.

R (Reserved Bit): MUST be set to "0" on transmit and ignored on
receive.

TTL Field:

This 8-bit field is used to carry In-Stack data apart from the 12
bits in the Label field.

NOTE:

An intermediate node may use the full MPLS label stack for ECMP
hash computation hence the In-Stack MPLS extension header MUST
NOT change the Label Field part of the IS-FI data within the same
traffic flow. But the TTL part of IS-FI data can change for the
same traffic flow without affecting the ECMP hash. The In-Stack
Extension Header encoding defined above ensures this.

5. Bottom Of Stack MPLS Extension Header Encoding

This section describes the encoding format of the MPLS Extension
Header which is present after the bottom of the MPLS label stack.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL     | TC  |1|  BPI=1, HBI   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 1 0|Reserve|  BOS-FI Opcode| Length=1(word)|   BOS-Flags   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           BOS-Data                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Payload                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: BOS Extension Header Format

BPI flag is set to "1" to indicate the presence of MPLS Extension
Header after the bottom of MPLS label stack.

HBI flag is set to "1" to indicate the MPLS Extension Header after
the Bottom Of Stack that requires Hop-By-Hop processing.

A new generic 4-byte header is defined to carry the information
about the Forwarding Instruction and its corresponding data that is
carried after the bottom of label stack. This generic header is

added to each Forwarding Instruction that is encoded after the MPLS bottom of the stack. This generic header gives the flexibility to add multiple Forwarding Instruction after the BOS in any desired order.

The 4-byte BOS Extension Header is described below:

Bit Position	Description
0 - 3	This 4-bit nibble MUST be set to "0010b". This is to avoid aliasing with an IPv4/IPv6 header.
4 - 7	This 4-bit nibble defines the version of the generic header format. The current version value is "0".
8 - 15	This 8-bit field indicates the BOS FI Opcode value. This opcode values will be allocated by IANA.
16 - 23	This 8-bit field indicates the length of the data encoded in units of 4 bytes excluding the current header.
24 - 31	This 8-bit field carries the BOS-Flags. 0 - NH bit (Next-Header Presence Bit): Indicates the presence of next BOS extension header. 1 - H bit (Hop-By-Hop Bit): Hop-By-Hop processing is required for this Bottom Of Stack data. 7 - 2 bits: Unassigned bits.

Table 2: BOS MPLS Extension Header Format

BOS-FI Opcode value of 0 is marked as invalid.

BOS-FI Opcode Value:1-254 - MUST be assigned by IANA.

BOS-FI Opcode Value:255 - IANA Allocated for BOS-FI Opcode range extension. This gives the extensibility for opcode range beyond 255.

If an application requires to add its own data TLV, then the TLV can be added as part of BOS-Data.

BOS-FI Opcode MUST define the following procedure before it can be used:

1. Define the Data format encoded in the MPLS extension header.

2. Define the Hop-By-Hop or Edge-To-Edge (only on the decapsulation node) processing scope.

3. The Hop-By-Hop BOS-FI opcodes MUST be placed before the Edge-To-Edge BOS-FI Opcodes in the MPLS Extension Header of the packet to optimize the Hop-By-Hop processing in hardware.

6. MPLS Extension Header Encoding Example Use-case-1.a - Carrying FI without data in the MPLS label stack

The TTL field can support only up to 8-bit flags. This is the use-case to extend the TTL flags and carry additional Forwarding Instruction Flags (FIF) in the MPLS label stack. These forwarding instructions do not require any additional data to be carried with this FI.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL      | IL=1|0|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IS-FI Opcode=1 |  Flags                |R|1|E|1|   Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Example In-Stack Extension Header Carrying Forwarding Instruction Flags

IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

Label Field:

   In this case the FI opcode value is set to "1". FI Opcode value "1" is reserved for extending the TTL flags. This indicates the presence of additional flags in the Label field and TTL fields

TC Field:

   DS-Bit - This bit is set to "1" to indicate that the flags are not extended further.

TTL Field:

   8-bit field is used to encode the Forwarding Instruction Flags apart from 12 bits Label field.

The FIF bit position and its meaning MUST be defined by IANA.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Entropy Label or SPL or NPL     | IL=2|0|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IS-FI Opcode=1 |  Flags              |R|1|E|1|   Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|   Flags                           |R|1|E|1|   Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Example In-Stack Extension Header carrying more than 20 bits
FI Flags

More than 20 bits of data can be encoded as part of IS-FI opcode. In
this specific case, the FI flags which are more than 20 bits are
encoded in next 4 bytes of the MPLS header.

While encoding the additional data, the Most Significant bit of the
Label Field MUST be set to "1" to prevent from aliasing with the
reserved SPLs in the case of legacy devices.

7. MPLS Extension Header Encoding Example Use-case-1.b - Carrying FI with data in the MPLS label stack

This is the use-case where the MPLS Label stack to carry the
Forwarding Instruction with a corresponding data.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Entropy Label or SPL or NPL     | IL=1|0|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode  |       Data          |R|1|E|1|   Data        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
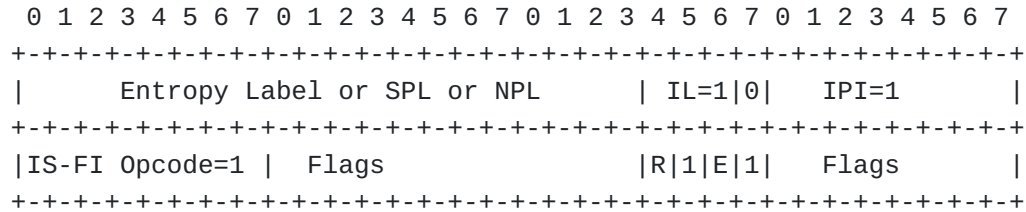
Figure 9: Example In-Stack Extension Header Carrying FI with the data

IPI flag is set to "1" to indicate the presence of In-Stack MPLS
Extension Header.

Label Field:

First 8 bits encodes the In-Stack forwarding opcode. In this case
the FI opcode value ranges from 1 to 254. This value is assigned
by IANA. This opcode value defines data format carried in the
Label field and the TTL field.

TC Field:

>   DS-Bit - This bit is set to "1" to indicate that the data is
>   encoded in the 19-bit Label field and does not exceed 19 bits.
>
>   R-Bit - Reserved bit and MUST be set to "0" on transmit and
>   ignored when received.

TTL Field:

>   8-bit field is used to encode the In-Stack data apart from 12-bit
>   Label field.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Entropy Label or SPL or NPL       | IL=2|0|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode  |      Data             |R|0|E|0|   Data        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|                     Data            |R|1|E|1|   Data        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Example In-Stack Extension Header Carrying FI with the data
more than 20 bits

More than 20 bits of data can be encoded as part of IS-FI opcode. In
this specific case, the In-Stack data which are more than 20 bits
are encoded in next 4 bytes of the MPLS header.

While encoding the additional data, the Most Significant bit of the
Label Field MUST be set to "1" to prevent from aliasing with the
reserved SPLs in the case of legacy devices.

8. MPLS Extension Header Encoding Example Use-case-2 - Carrying FI with data after the MPLS label stack

This is the use-case where the Forwarding Instruction with a
corresponding data is carried after the MPLS bottom of label stack.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL      | TC  |1| BPI=1,HBI=1   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 1 0|Reserve|BOS-FI Opcode=1| Length=1(word)|Flags NH=1,H=1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          BOS-Data1                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 1 0|Reserve|BOS-FI Opcode=2| Length=2(word)|Flags NH=0,H=0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          BOS-Data2                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          BOS-Data2                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Payload                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Example BOS Extension Header Carrying FI with data

BPI flag is set to "1" to indicate the presence of BOS MPLS
Extension Header. Also, HBI flag is set to 1 to indicate the presece
of BOS MPLS Extension Header that requires Hop-By-Hop processing.

In this case, the MPLS packet is encoding two different types of BOS
FI (Opcode 1 and Opcode 2) after the bottom of MPLS label stack.

The first BOS MPLS Extension Header has the Length value as "1",
this indicates that the data corresponding to this FI opcode "Type1"
is 4 bytes following this header. Also the Next-Header (NH) flag in
BOS-Flags is set to "0x1", this indicates the presence of next BOS
MPLS Extension Header. The H flag is set to "0x1" that indicates the
Hop-By-Hop processing is required.

The second BOS MPLS Extension Header has the Length value as "2",
this indicates that the data corresponding to the FI opcode "Type2"
is 8 bytes following this header. In this case the Next-Header flag
in BOS-Flags is set to "0x0", this indicates that this is the last
BOS MPLS Extension Header encoded. The H flag is set to "0x0" that
indicates the Hop-By-Hop processing is not required.

9. MPLS Extension Header Encoding Example Use-case-3 - Carrying use-case 1.a, 1.b and 2 in MPLS packet

This is the use-case where the same MPLS packet carrie the use-cases
"1.a", "1.b" and "2".

```
  0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Entropy Label or SPL or NPL      | IL=3|0| IPI=BPI=HBI=1 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | IS-FI Opcode=1|  Flags               |R|0|E|0| Flags         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | IS-FI Opcode=2|        0             |R|1|E|0| Offset = 1    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | IS-FI Opcode=3|       Data           |R|1|E|1|     Data      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0 0 1 0|Reserve|BOS-FI Opcode=1| Length=1(word)|Flags NH=1,H=1 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         BOS-Data1                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0 0 1 0|Reserve|BOS-FI Opcode=2| Length=2(word)|Flags NH=0,H=0 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         BOS-Data2                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         BOS-Data2                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Payload                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

         Figure 12: MPLS Packet Carrying 1.a, 1.b and 2 Use-cases

   IPI and BPI flags are set to "1" to indicate the presence of both
   In-Stack and BOS MPLS Extension Header as mentioned in the above
   use-cases. IS-FI Opcode 2 is added to indicate the offset of 1 word
   after the MPLS header BOS and start of the BOS Extension Header.

10. Node Capability Signaling

   The node capability for the MPLS Extension Header must be signaled
   before the MPLS Encapsulating node can add the necessary MPLS
   Extension Header in the MPLS label stack. The capability signaling
   will be added in LDP, RSVP-TE, BGP, IS-IS, OSPF, BGP-LS, etc. This
   is outside the scope of this document.

11. Security Considerations

   The security considerations in [RFC3032] also apply to the
   extensions defined in this document. The MPLS Extension header MUST
   NOT be exposed to the node which does not support the new MPLS
   Extension Header.

12. Backward Compatibility

12.1. Backward Compatibility With ELC as MEH Indicator

As specified in [RFC6790], the TTL field of the EL MUST be "0". On the Node which is capable of processing the MPLS Extension Header when it finds that this TTL value is non-zero, then only it will start processing the MPLS Extension header.

In addition, the TC field will be interpreted as the In-Stack MPLS Header Extension Length only when the TTL field's IPI Flag is set to "1".

For the legacy node that does not advertise the MPLS Extension Header capability, the Encapsulating node MUST make sure that the MPLS Extension header is not at the top of the MPLS label stack to avoid misforwarding the packets by misinterpreting In-Stack Extension Header as a label.

The MPLS Extension Header Encoding format is designed to make sure that it does not alias with any reserved SPL.

The MPLS extension does not affect the existing GAL / G-ACh [RFC5586] based encoding of data in the MPLS packet. This MPLS extension can co-exist with the existing GAL / G-ACh based encoding of data.

12.2. Backward Compatibility With SPL as MEH Indicator

For the legacy node that does not advertise the MPLS Extension Header capability, the Encapsulating node MUST make sure that the MPLS Extension header is not at the top of the MPLS label stack to avoid dropping the packets.

The MPLS Extension Header Encoding format is designed to make sure that it does not alias with any reserved SPL.

The MPLS extension does not affect the existing GAL / G-ACh [RFC5586] based encoding of data in the MPLS packet. This MPLS extension can co-exist with the existing GAL / G-ACh based encoding of data.

13. Processing In-Stack MPLS Extension Header

Encapsulating Node:

   *MUST NOT add In-Stack MPLS Extension header if the decapsulation
    node is not capable of In-Stack MPLS Extension header.

*SHOULD NOT change the IS-FI Opcode and the first 12 bits of the
    In-Stack Data for the same packet flow avoid ECMP path change.

   *MAY change In-Stack data part present only in the TTL field for
    the same packet flow.

   *MUST ensure that the penultimate node does not remove the MPLS
    extension header.

 Intermediate Node:

   *MUST ignore the IS-FI Opcode that are not supported.

   *MUST NOT add In-Stack MPLS Extension header if the decapsulation
    node is not capable of In-Stack MPLS Extension header.

   *SHOULD NOT change the IS-FI Opcode and the first 12 bits of the
    In-Stack Data for the same packet flow.

   *MAY change In-Stack data part present only in the TTL field for
    the same packet flow.

   *MAY remove the IS-FI opcode and its corresponding data for all
    matching packet flow.

 Decapsulating Node:

   *MUST remove the In-Stack MPLS Extension header.

14. **Processing BOS MPLS Extension Header**

 Encapsulating Node:

   *MUST NOT add BOS MPLS Extension header if the decapsulation node
    is not capable of BOS MPLS Extension header.

   *MUST ensure that the penultimate node does not remove the MPLS
    extension header.

 Intermediate Node:

   *MAY add additional data to the existing BOS-FI encoded.

   *MAY add a new BOS-FI and its corresponding data if the
    decapsulation node supports BOS MPLS Extension header.

 Decapsulating Node:

   *MUST remove the BOS MPLS Extension header.

15. IANA Considerations

Below are the IANA actions which this document is requesting.

15.1. IANA Considerations for Forwarding Instruction Flags

IANA is requested to create a new registry to assign the bit position and the meaning to the Forwarding Instruction Flags based on the user request.

Bit Position	Description	Reference
19-0	Unassigned	This document

Table 3: Forwarding Instruction Flags
Registry

15.2. IANA Considerations for IS-FI Opcode

IANA is requested to create a new registry to assign IS-FIOC opcode values. All code-points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code-points are allocated according to Table 4:

Value	Description	Reference
1 - 175	IETF Review	This document
176 - 239	First Come First Served	This document
240 - 251	Experimental Use	This document
252 - 254	Private Use	This document

Table 4: In-Stack Forwarding Instruction Opcode
Registry

Following IS-FIOC Opcode values are assigned from this registry.

Value	Description	Reference
0	Invalid value	This document
1	Forwarding Instruction Flags	This document
2	Offset of start of Bottom Of Stack Data after BOS Label	This document
255	Opcode Range Extension Beyond 255	This document

Table 5: In-Stack Forwarding Instruction Opcode Values

15.3. IANA Considerations for BOS-FI Opcode

IANA is requested to create a new registry to assign BOS-FIOC opcode
values.

Value	Description	Reference
1 - 175	IETF Review	This document
176 - 239	First Come First Served	This document
240 - 251	Experimental Use	This document
252 - 254	Private Use	This document

Table 6: Bottom-Of-Stack Forwarding Instruction
Opcode Registry

Following BOS-FIOC Opcode values are assigned from this registry.

Value	Description	Reference
0	Invalid value	This document
255	Opcode Range Extension Beyond 255	This document

Table 7: Bottom-Of-Stack Forwarding Instruction Opcode
Values

The application that requires an Opcode for the Forwarding
Instruction (IS-FIOC or BOS-FIOC) or a Flag must request the code-
point and its meaning from IANA.

15.4. IANA Considerations for New Special Purpose Label

IANA is requested to allocate a value TBA1 for the MEI SPL label
from the "Base Special-Purpose MPLS Label Values" registry to
indicate the presence of MPLS Header Extension.

16. Appendix

16.1. Alternate approach for In-Stack Extension Header Encoding

In the above In-Stack Extension Header Encoding the Label field is
used to encode the FI Opcode. So just for completeness, here is the
alternate way of In-Stack Extension Header Encoding is provided.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL      | IL=1|S|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|                 Data                |R|D|E|S|   FI Opcode   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Alternate In-Stack Extension Header Format

   IPI flag is set to "1" to indicate the presence of In-Stack MPLS
   Extension Header.

   Since In-Stack MPLS Extension Header is present as part of the MPLS
   Header, the MPLS Header is redefined to encode the MPLS Extension
   Header.

   Label Field:

      Most significant bit is always set to "1" to avoid aliasing with
      the reserved SPLs.

      Rest of the 19 bits and the "R" bit from the TC bit can be used
      by the application. So total of 20 bits can be used to carry the
      data corresponding to IS-FI opcode.

   TC Field:

   This carries data stacking bits. They are as follows:

      D (DS-Bit): Data Stacking Bit. This is used to encode more than
      19 bits of extended data in the MPLS Label stack. If this is set
      to "1", then this is the end of extended data.

      R (Reserved Bit): This is used to encode the IS-FI data.

   TTL Field:

   This carries In-Stack Forwarding Instruction opcode.

16.2. MPLS Extension Header Example for Entropy Label using New SPL

   The MPLS Extension Header encoding formats defined in this document
   is applicable when using a new Special Purpose Label (SPL) or using
   a Network Programming Label (NPL) configured by an operator.

   The TTL field in the SPL (value TBA1) is used to encode FI Flags
   including IPI, HBI and BPI flags defined in this document.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MEI=SPL (value TBA1)           | IL=1|S|    IPI=1        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode=3|  Entropy Label     |R|D|E|S|    SLID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: MPLS Extension Header Encoding Example for Entropy Label
using New SPL

The FI Opcode value 3 as an example indicates encoding of Entropy
Label and Slice ID as shown in the above Figure.

16.3. MPLS BOS Extension Header Example with IOAM Data Fields

The Bottom Of Stack (BOS) Extension Header is used with BOS Opcode
for IOAM.

Bottom Of Stack Presence Indicator (BPI) flag in TTL is set to "1"
to indicate the presence of BOS Extension Header. HBI flag in TTL is
set to "1" to indicate the BOS Extenion Header requires Hop-By-Hop
processing.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Entropy Label or SPL or NPL      | TC |1| BPI=1, HBI    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|0 0 1 0|Reserve|BOS Opcode=IOAM|Length (words) | Flags (NH, H) |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
| IOAM-OPT-Type | IOAM HDR Len  | Block Number  | Reserved      |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  I
|                                                             |  O
|                                                             |  A
~                  IOAM Option and Data Space                 ~  M
|                                                             |  |
|                                                             |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|                                                             |
|                                                             |
|                  Optional Payload + Padding                 |
|                                                             |
|                                                             |
|                                                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
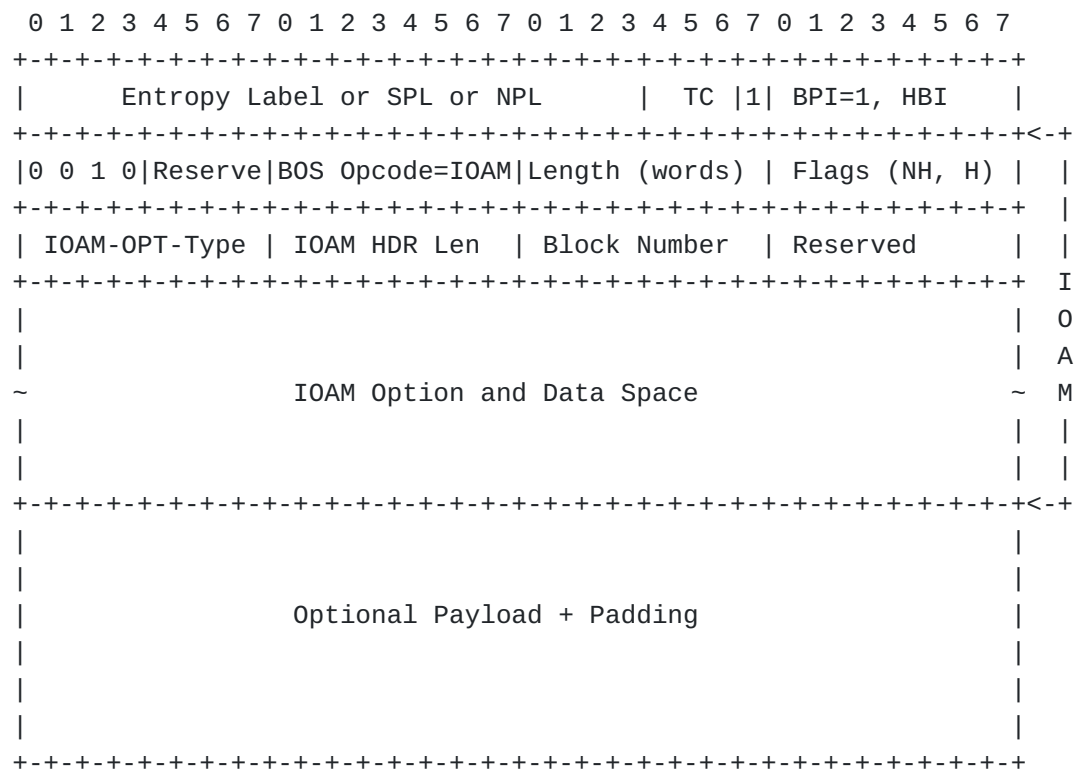
Figure 15: Example MPLS Encapsulation for IOAM using BOS Extension
Header

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>.

[RFC3032]
Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[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>.

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

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

17.2. Informative References

[I-D.decraene-mpls-slid-encoded-entropy-label-id]
Decraene, B., Filsfils, C., Henderickx, W., Saad, T., Beeram, V. P., and L. Jalil, "Using Entropy Label for Network Slice Identification in MPLS networks.", Work in Progress, Internet-Draft, draft-decraene-mpls-slid-encoded-entropy-label-id-03, 11 February 2022, <https://www.ietf.org/archive/id/draft-decraene-mpls-slid-encoded-entropy-label-id-03.txt>.

[I-D.saad-mpls-miad-usecases] Saad, T., Makhijani, K., and H. Song, "Usecases for MPLS Indicators and Ancillary Data", Work in Progress, Internet-Draft, draft-saad-mpls-miad-usecases-00, 7 January 2022, <https://www.ietf.org/archive/id/draft-saad-mpls-miad-usecases-00.txt>.

[I-D.bocci-mpls-miad-adi-requirements] Bocci, M. and S. Bryant, "Requirements for MPLS Label Stack Indicators and Ancillary Data", Work in Progress, Internet-Draft, draft-bocci-mpls-miad-adi-requirements-01, 24 January 2022, <https://www.ietf.org/archive/id/draft-bocci-mpls-miad-adi-requirements-01.txt>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI

10.17487/RFC5586, June 2009, <https://www.rfc-editor.org/info/rfc5586>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <https://www.rfc-editor.org/info/rfc4385>.

Acknowledgments

Contributors

TBD

Authors' Addresses

Jaganbabu Rajamanickam (editor)
Cisco Systems, Inc.
Canada

Email: jrajaman@cisco.com

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Jisu Bhattacharya
Cisco Systems, Inc.

Email: jisu@cisco.com

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Royi Zigler
Broadcom

Email: royi.zigler@broadcom.com

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com