# MPLS EXTENSION HEADER ENCODINGS (DRAFT-JAGS-MPLS-EXT-HDR-00)

JAGANBABU RAJAMANICKAM (JRAJAMAN@CISCO.COM)

RAKESH GANDHI (RGANDHI@CISCO.COM)

JISU BHATTACHARYA (JISU@CISCO.COM)

BRUNO DECRAENE (BRUNO.DECRAENE@ORANGE.COM)

ROYI ZIGLER (ROYI.ZIGLER@BROADCOM.COM)

WEIQIANG CHENG (CHENGWEIQIANG@CHINAMOBILE.COM)

LUAY JALIL (LUAY.JALIL@VERIZON.COM)

113th IETF

# ABBREVIATIONS

| Abbreviations | Meaning |
|---|---|
| FI | Forwarding Instruction |
| IS-FI | In-Stack Forwarding Instruction |
| ELC | Entropy Label Control |
| ELI | Entropy Label Indicator (value 7) |
| BOS | Bottom Of MPLS Stack |
| BOS-FI | Bottom Of MPLS Stack Forwarding Instruction |
| IL | In-Stack Data Length |
| DS | Data Stacking |
| IPI | In-Stack MPLS Extension Header Presence Indicator |
| BPI | BOS MPLS Extension Header Presence Indicator |
| HBI | Hop-By-Hop BOS MPLS Extension Header Presence Indicator |
| MEH | MPLS Extension Header |
| MEI | MPLS Extension Header Indicator |
| SPL | Special Purpose Label |
| MSD | Maximum Stack Depth |

# AGENDA

- Problem Statement and Requirements
- MEI Options
- In-Stack MPLS Extension Header Encoding Format
- Bottom Of Stack MPLS Extension Header Encoding Format
- Encoding Examples
- Next Steps

# PROBLEM STATEMENT & REQUIREMENTS

**PROBLEM STATEMENT**:

Today's new applications (such as IETF Network Slicing, In-Situ OAM, In-band Performance-Measurement, In-band Telemetry, etc.) require MPLS/SR-MPLS packet to carry some indicators and associated ancillary data that is used in the MPLS packet forwarding decision or for OAM purpose. This will require MPLS packets to carry more SPLs/eSPLs and thus increase the MSD size.

**REQUIREMENTS**:

1. MPLS packets header to carry additional data in the label stack to influence forwarding or more. This could be of two types:
   A. Flag based Forwarding Instruction that does not need additional data
   B. 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 the above can co-exist in the same MPLS packet
4. All the above must be backwards compatible

# HIGH-LEVEL SOLUTION

Extending the existing MPLS Header basically needs the following:

1. **MPLS Extension Header Indicator (MEI)** - Indicates the presence of MPLS Extension Header in the packet

2. **MPLS Extension Header (MEH) 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

# MPLS EXTENSION HEADER INDICATOR (MEI) - OPTIONS

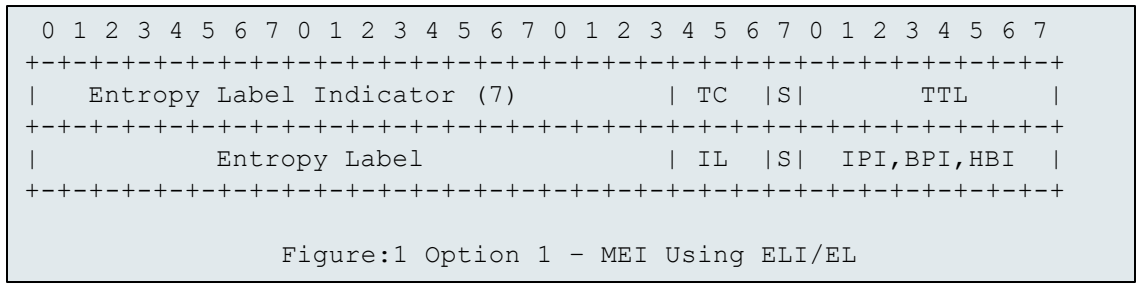



Figure:1 Option 1 – MEI Using ELI/EL

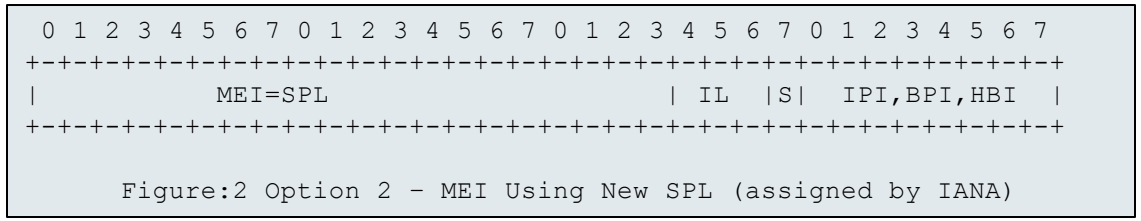



Figure:2 Option 2 – MEI Using New SPL (assigned by IANA)

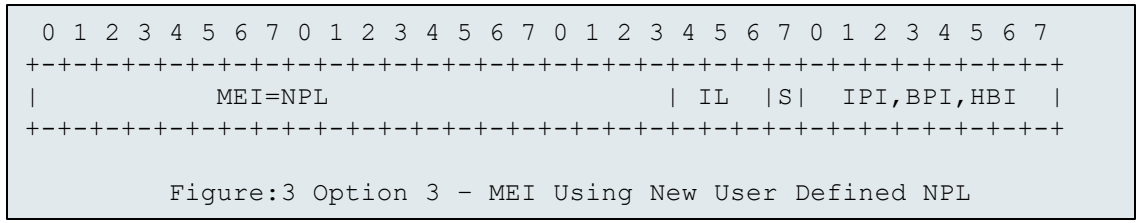



Figure:3 Option 3 – MEI Using New User Defined NPL

**MEI:** This is the way to indicate the presence of MPLS Extension Header. This could be done in three different ways

- **Option 1**: MEI by extending ELI/EL – Re-purposing TC, TTL fields of EL
- **Option 2**: MEI by using a new Special Purpose Label (SPL) – Assigned by IANA
- **Option 3**: MEI by using a new Network Programming Label (NPL) – Provisioned by user

Each options has its own advantages and disadvantages. One or more options can be selected by the IETF WG.

- **IL (In-stack Data Length):** TC field is used to indicate the length of the In-Stack data length. In the order of four bytes.
- **IPI (In-Stack MPLS Extension Header Presence Indicator):** A Bit in the TTL field indicates the presence of the In-Stack MPLS Extension Header.
- **BPI (Bottom Of Stack MPLS Extension Header Presence Indicator):** A Bit in the TTL field indicates the presence of the BOS MPLS Extension Header.
- **HBI (Hop-By-Hop BOS MPLS Extension Header Indicator):** A Bit in the TTL field indicates the BOS MPLS Extension Header needs to be processed Hop by Hop.

# IN-STACK MPLS EXTENSION HEADER ENCODING FORMAT

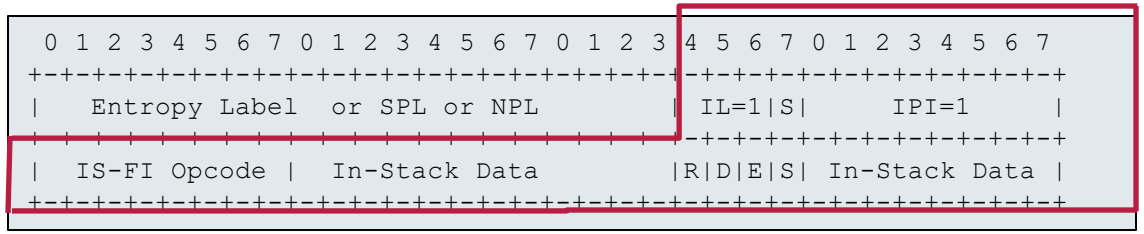



Figure 4:In-Stack MPLS Extension Header Format

**In-Stack Extension Header Indicator**:

---

- **IPI** flag in the TTL field indicates the presence of In-Stack MPLS Extension Header.
- **IL**- In-Stack Data Length. TC field is re-purposed to indicate the length of the In-Stack Extension Header.

**In-Stack-Data Format**:

---

- **IS-FI Opcode**: This is 8-bit opcode defines the Forwarding Instruction. This is encoded as the first 8-bits of the Label field.
- **In-Stack Data**: This is 20 bits field data corresponding to the IS-FI Opcode. This uses 12 bits from the Label field and the 8 bits from the TTL field.
- **D – (DS-Bit)**: Data Stacking Bit: This is used to encode more than 20 bits of data for this IS-FI Opcode.
- **E – (E2E-Bit)**: MPLS Extension Header In-Stack Data requires E2E processing only. If this is “0” then it requires Hop-By-Hop processing.
- **R – (Reserved)**: Not used currently.

**IS-FI Opcodes Assigned**:

- Value:1 - Carry Forwarding Instruction Flags
- Value:2 - Byte offset of the BOS-data location from BOS
- Value:3 - 254 - Must Assigned by IANA
- Value:255 - Extending the opcode range beyond 255

# BOTTOM OF STACK MPLS EXTENSION HEADER ENCODING FORMAT

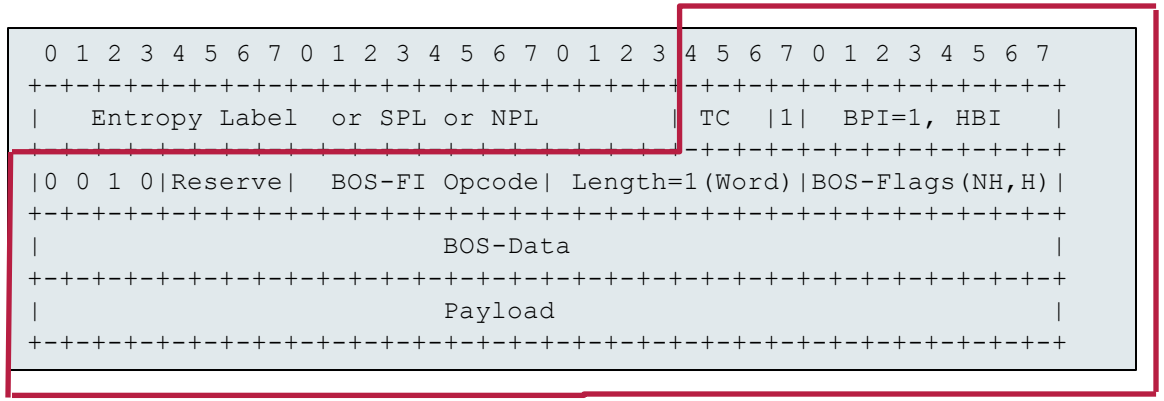



Figure 5: BOS MPLS Extension Header Format

4-Byte BOS Extension Header is defined to carry the information about the Forwarding Instruction and its corresponding data that is carried after the bottom of stack.

**BOS Extension Header Indicator**:

---

- **BPI**: BOS MPLS Extension Header Presence Indicator. This flag indicates the presence of MPLS Extension Header after the BOS.
- **HBI**: Hop-By-Hop BOS Extension Header Indicator. This field is used to indicate the presence of Extension header present after the BOS that needs Hop-By-Hop processing.

**BOS-Data Format:**

---

- **0 0 1 0 b**: This is a fixed 4-bit nibble to avoid aliasing with an IPv4/IPv6 header.
- **BOS-FI Opcode**: This 8-bit field indicates the BOS FI Opcode value. This opcode values will be allocated by IANA.
- **Length**: Length of BOS Data is in the units of 4 bytes. This BOS data can have its own TLV and sub-TLV.
- **BOS-Flags**: This is the Flags used to process this data.
  - 0$^{th}$ bit - NH bit: Next Header Presence Bit. If set, then there is another BOS extension header is followed.
  - 1$^{st}$ bit - H bit: Hop-By-Hop Bit. If this Bit is set, then this BOS data needs to be processed on all the Hops.

# ENCODING EXAMPLE-1: IN-STACK EXTENSION HEADER CARRYING FLAG-BASED FI

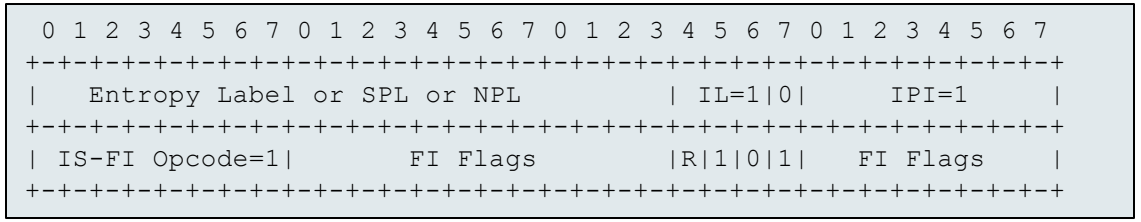



Figure 6: In-Stack FI Opcode "1" encoded to carry Flag based FI

The IS-FI **Opcode "1"** is reserved for carrying the Forwarding Instruction that does not need any ancillary data. This is also called "Flag-based Forwarding Instruction". The bit-position of the flag for each application is assigned by IANA.

**First Word**:
**IPI=1** → Indicates the presence of In-Stack MPLS Extension Header
**IL=1** → Indicates that In-Stack data length is of four bytes (i.e., words)

**Second Word**:
**IS-FI Opcode=1** → Indicates that, it is carrying Flag-based Forwarding Instructions
**FI Flags** → Flags based Forwarding instructions, which does not need any additional ancillary data
**D - (DS-Bit)=1** → This indicates the end of the Flags field
**E - (E2E-Bit)**=0 -> Requires Hop-By-Hop processing

# ENCODING EXAMPLE-2: IN-STACK EXTENSION HEADER CARRYING MORE THAN 20-BITS DATA

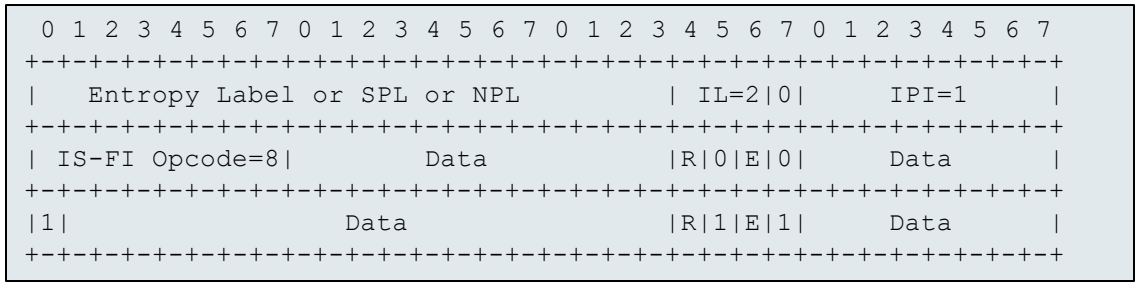



Figure 7: In-Stack FI opcode 8 encoding more than 20 bits of ancillary data

**First Word**:
**IPI=1** → Indicates the presence of In-Stack MPLS Extension Header
**IL=2** → Indicates that In-Stack data length is of eight bytes (two words)

**Second Word**:
**IS-FI Opcode=8** → Indicates the Forwarding Instruction Opcode value 8. This value is assigned by IANA for a specific type of Forwarding Instruction.
**Data** → Ancillary data with respective to FI Opcode
**DS-Bits=0** → This indicates that the ancillary data continued in next four bytes

**Third Word**:
**"1"** → First bit of the MSB Must be set to "1" to prevent the ancillary data value from aliasing with the SPLs in the case of Legacy devices.
**Data** → This is the continuation of the ancillary data.
**DS-Bit=1** → This indicates the end of the ancillary data corresponding to the IS-FI Opcode "8".

# ENCODING EXAMPLE-3: BOS EXTENSION HEADER CARRYING MULTIPLE BOS-FI OPCODES

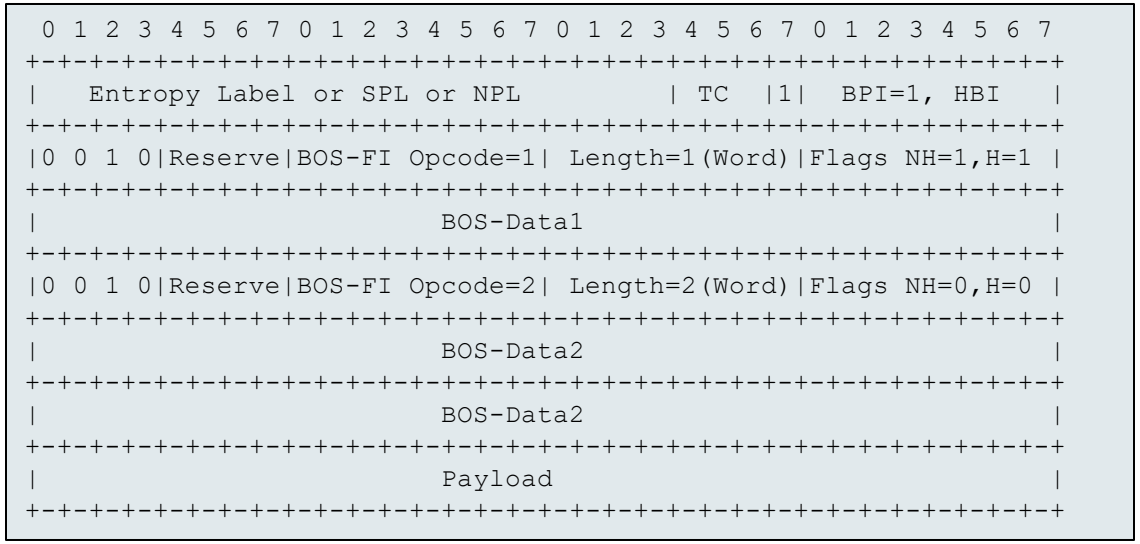



Figure 8: BOS FI opcode 1 & 2 carrying their ancillary BOS data

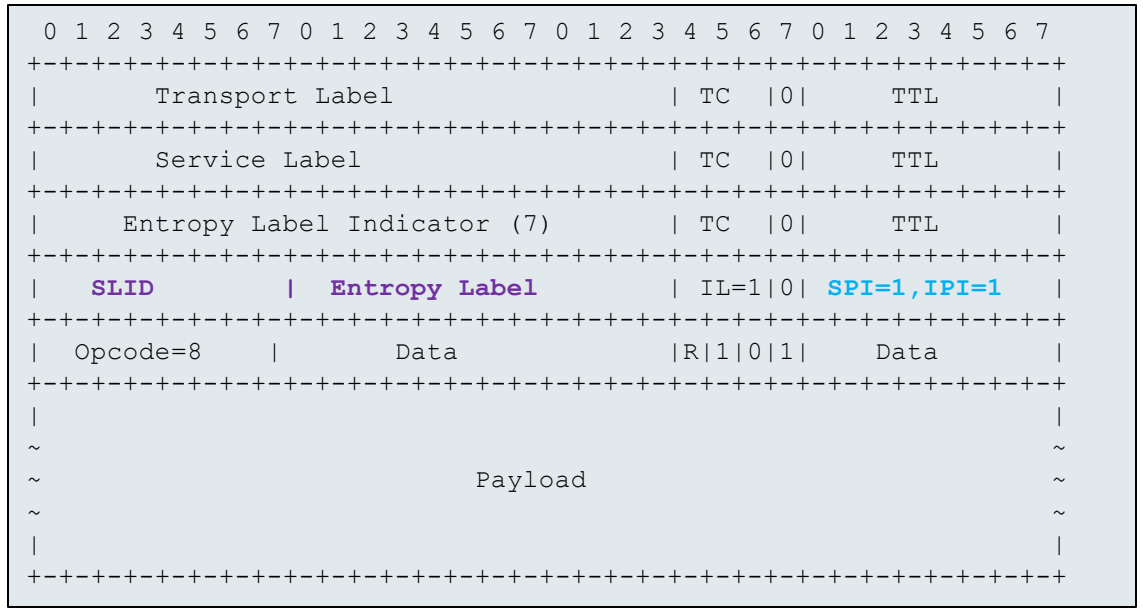
**First Word**:
**BPI=1** → Indicates the presence of BOS MPLS Extension Header
**HBI=2** → Indicates that this BOS MPLS Extension Header should be processed Hop-By-Hop

**Second Word**:
**"0 0 1 0 b"**: This is a fixed 4-Bits nibble to avoid aliasing with an IPv4/IPv6 header
**BOS-FI Opcode=1** → Indicates the BOS Forwarding Instruction Opcode value 1. This value is assigned by IANA for a specific type of Forwarding Instruction.
**Length=1** → The Length of the ancillary data that is encoded with respect to the BOS-FI Opcode "1".
**Flags NH=1** → This indicates that, there is another BOS-FI opcode encoded
**Flags H=1** → This indicates that the BOS-FI opcode should be processed Hop-By-Hop

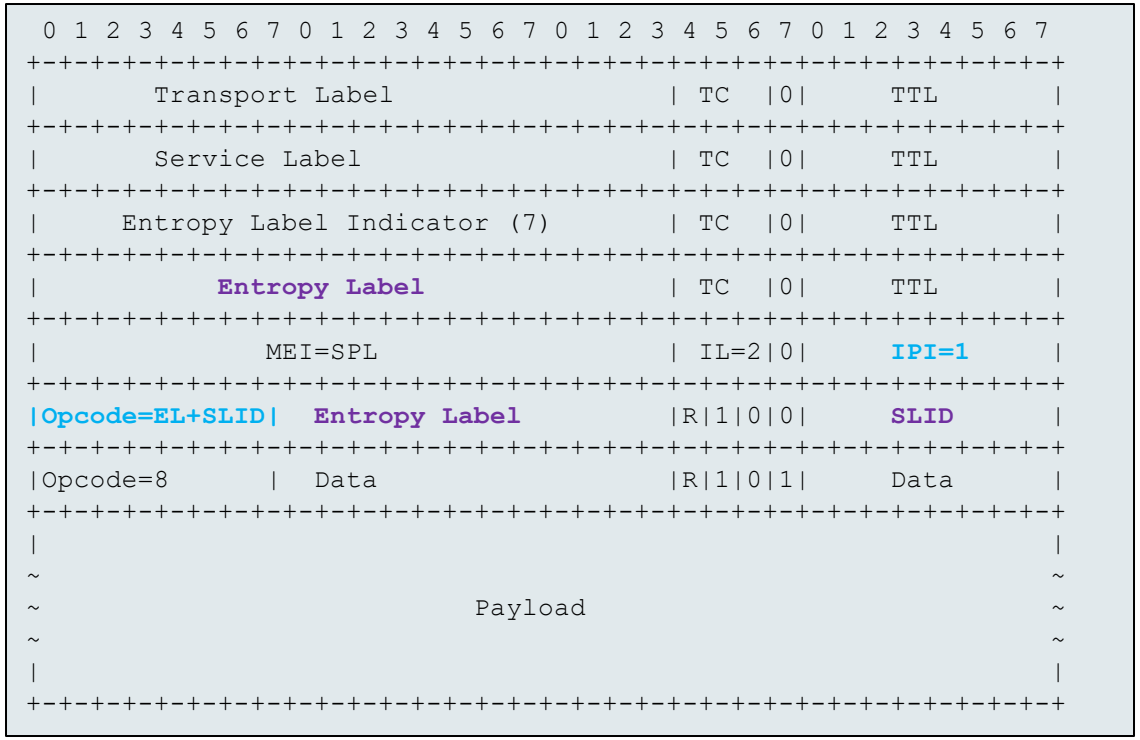
**Fourth Word**:
**BOS-FI Opcode=2** → Indicates the BOS Forwarding Instruction Opcode value 2. This value is assigned by IANA for a specific type of Forwarding Instruction.
**Length=2** → The Length of the ancillary data that is encoded with respect to the BOS-FI Opcode "2".
**Flags NH=0 ->** This indicates that, there is no further BOS-FI opcode encoded
**Flags H=0** -> This indicates that the BOS-FI opcode should be processed on edge node only.

# SAMPLE PACKET CARRYING IN-STACK MPLS EXTENSION HEADER WITH OPTION-1 AND OPTION-2

**OPTION 1 - Encoding using ELI/EL MEI with entropy + Example Opcode=8 (MSD = 5)**




**Option 2 - Encoding using NEW SPL MEI with entropy + Example Opcode=8 (MSD = 7)**



# NEXT STEPS

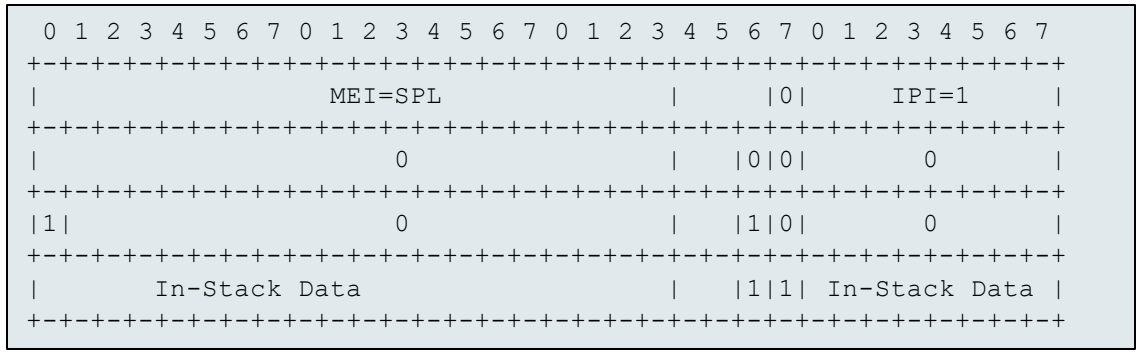
- Welcome review comments and feedbacks
  - Feedback on MEI Label options
- Requesting MPLS WG adoption

# THANK YOU!

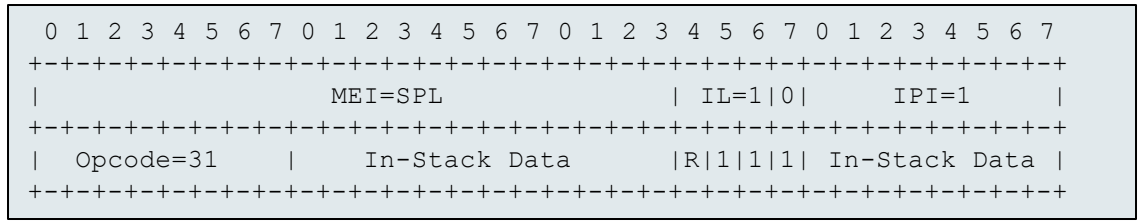
# APPENDIX

# BITWISE VS. OPCODE-BASED FORWARDING INSTRUCTIONS

**Bitwise-Based Forwarding Instructions**




**Opcode-Based Forwarding Instructions (Approach in this Draft)**




For Example: Let us assume, that in future, an application that needs to carry ancillary data in the MPLS header has been assigned an Instruction number 31. If that application needs to carry its ancillary data in the MPLS packet, then

- In the case of Bitwise-based instruction, it needs to set 31$^{st}$ bit and it must encode the second word as empty.
- In the case of Opcode-based instruction, it needs to use value 31 in the opcode field. This reduces the MPLS stack size

# HARDWARE ANALYSIS

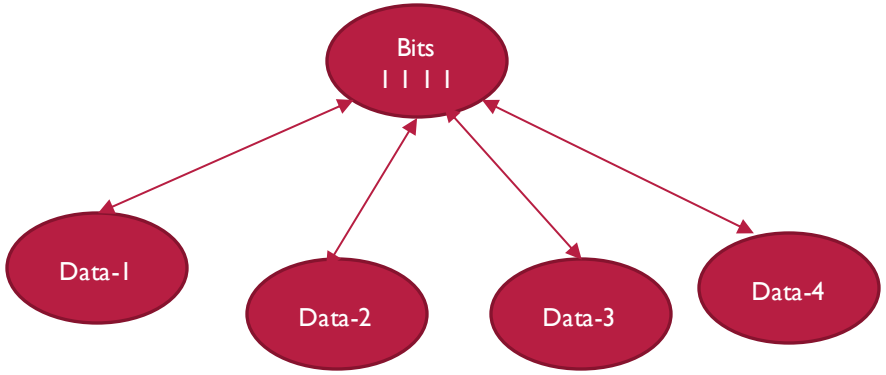
# HARDWARE ANALYSIS

1. Do not parse everything in parser. Parser should only delineate the layer offsets and layer types. The subsequent termination or forwarding macros/functions can dig deeper in the layer header and take appropriate actions. Layer types are limited.
2. In rare cases, parser can provide some layer attributes to termination/forwarding macros. Attribute bits are limited.
3. Parsers do not have huge stack depths. Try not to break one header into multiple unnecessarily.
4. Parsers do not have big TCAMs. Reduce dependency on TCAMs to identify special header field values.

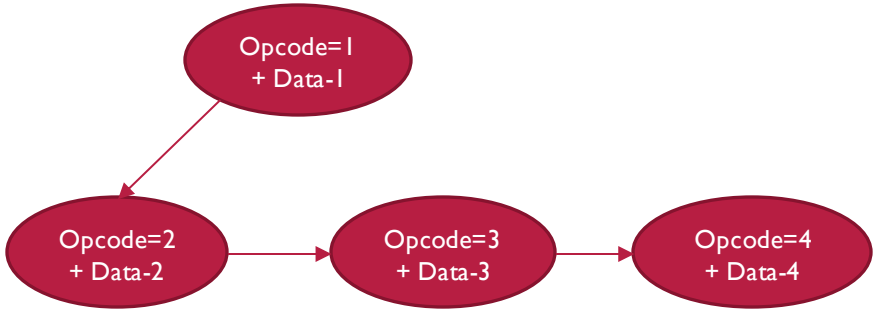
# HARDWARE ANALYSIS - PARSING HEADER CHAINS

Opcode-Based Header chains are preferred over bitmap catalog:

1. Most parsers are well suited for handling Opcode-Based header chains, because that's how most IP headers are defined e.g., IPv6 next headers, VLAN headers etc.
2. Parsers are **NOT well-suited for traversing back** on a header to find the next header. In the case of Bitmap catalog, it would require the parser to traverse back and forth.




Bitmap Catalog – Complex for parsers




Opcode based - Preferred Header Chain

# HARDWARE ANALYSIS - MPLS LABEL PROCESSING

1. Give information so that parser can skip labels that it does not care about
2. Do not force an inflexible order on IS-FI.
3. Stick to only one method for indicating entropy. EL/ELI method is sufficient. NPUs do not need/want to support more reserve labels to avoid complexity.
4. Avoid aliasing with SPL. All in-stack data should start with a 1’b1 or someway to make sure that the Label value is more than 15. A legacy device can simply scan stack and mis-identify IS-FI as SPL
5. Avoid aliasing with 4’b0000 , 4’b0100 and 4’b0110 in BOS data. Many legacy devices implement speculative parsing after MPLS header for L2VPN-Psuedocode, IPv4 and IPv6
6. In many cases, packets with IS-FI just need to be punted to CPU. Do not waste time parsing them in NPU.

# THANK YOU!