

MPLS NETWORK ACTION (MNA) HEADER ENCODINGS (DRAFT-JAGS-MPLS-MNA-HDR-00)

Jaganbabu Rajamanickam (jrajan@cisco.com) - Presenter
Rakesh Gandhi (rgandhi@cisco.com)
Jisu Bhattacharya (jisu@cisco.com)
Bruno Decraene (bruno.dekraene@orange.com)
Royi Zigler (royi.zigler@broadcom.com)
Weiqiang Cheng (chengweiqiang@chinamobile.com)
Luay Jalil (luay.jalil@verizon.com)
Haoyu Song (haoyu.song@futurewei.com)
Xiao Min (xiao.min2@zte.com.cn)
Bin Wen (Bin_Wen@cable.comcast.com)
Sami Boutros (sboutros@ciena.com)

ABBREVIATIONS

Abbreviations	Meaning
AD	Ancillary Data
ADL	Additional Data Length
I2E	Ingress To Egress
HPI	Hop By Hop Post-Stack Network Action Presence Indicator
ISD	In-Stack Data
IS-NAI-Opcode	In-Stack Network Action Indicator Opcode
INE	In-Stack Network Action Extension Presence Indicator
INI	In-Stack Network Action Presence Indicator
MNA	MPLS Network Action
NAI	Network Action Indicator
NAI-OP	Network Action Indicator Opcode
NASI	Network Action Sub-Stack Indicator
PNI	Post-Stack Network Action Presence Indicator
PSD	Post-Stack Data
bSPL	Base Special Purpose Label
MSD	Maximum Stack Depth

AGENDA

- Scope
- High Level Solution
- Network Action Indicators
- In-Stack Network Action Encoding [Formats](#)
- In-Stack Network Action Encoding [Examples](#)
- Backwards Compatibility
- Advantages
- Next Steps

SCOPE

- Solution for In-Stack and Post-Stack MNA Encoding Formats
- Address MNA requirements [draft-ietf-mpls-mna-requirements]
- Align with MNA framework [draft-andersson-mpls-mna-fwk]

HIGH-LEVEL SOLUTION

Extending the existing MPLS Header for MPLS Network Action (MNA) needs the following to be defined:

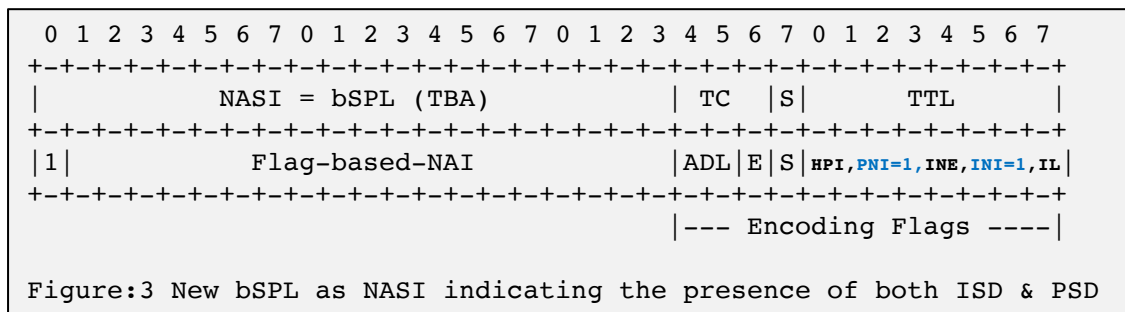
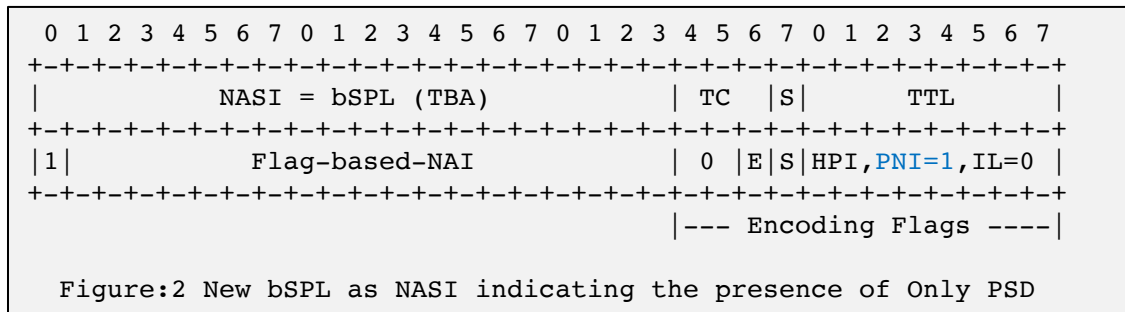
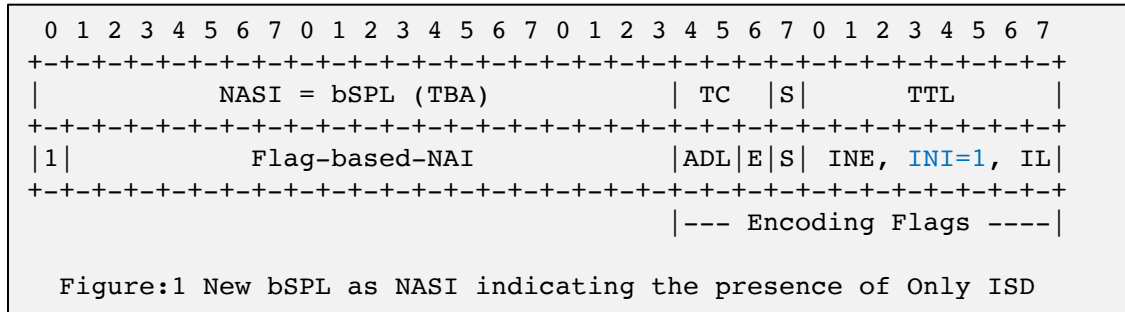
1. **MPLS Network Action Indicators:**

- a. Label to indicate the presence of Network Action Sub-Stack (NASI).
 - b. Flags to indicate the presence of In-Stack and Post-Stack Network Actions.
- Useful for the Network Processors to know the presence of Network Actions encoded in the packet and make decision to process or skip the Network Action.

2. **MPLS Network Action Encoding Formats:**

- The formats in which the Network Actions can be carried in the MPLS packet.
- This includes the information of both In-stack and Post-Stack Network Actions.

NETWORK ACTION PRESENCE INDICATORS



NASI: This is a label to indicate the presence of MPLS Network Action sub-stack.

- A new bSPL will be allocated by IANA for this purpose.
- TC and TTL fields of this new bSPL cannot be re-purposed for the backward compatibility, so the MNA encoding flags (presence of the In-Stack/Post-Stack NA indicators, etc.) are defined in the TC and TTL fields of the next LSE.

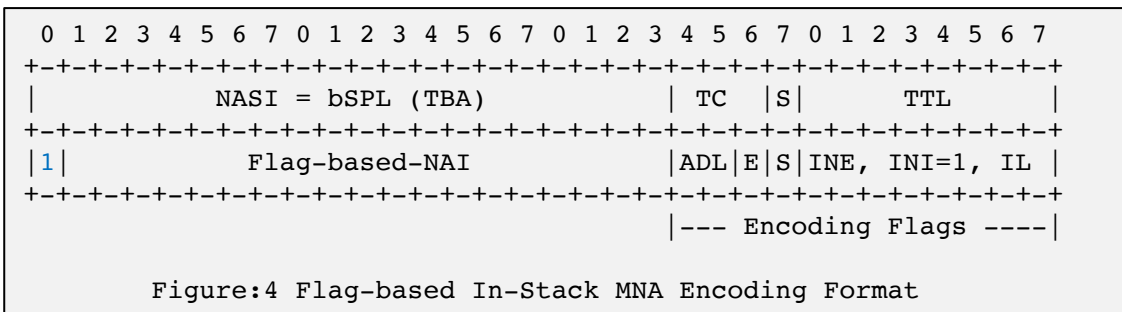
In-Stack Network Action Presence Indicator (INI): Indicates the presence of the In-Stack Network Actions that are encoded in the packet as shown in Figure-2.

- The In-Stack Network Action can be Flag-based that does not require Ancillary Data or Opcode-based that requires Ancillary data.

Post-Stack Network Action Presence Indicator (PNI): Indicates the presence of the Post-Stack Network Action that are encoded in the packet as shown in Figure-3.

MNA header may contain both In-Stack and Post-Stack Network Actions in the same header as shown in Figure-4.

IN-STACK MPLS NETWORK ACTION ENCODING FORMAT (FLAG-BASED)



In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** - Flag in the TTL field indicates the presence of In-Stack MPLS Network Action.
- **IL (In-Stack Data Length)** - 3-bit value in the TTL field indicates the total length of the In-Stack Extension Header encoded by the specific NASI indicator. This length is represented in the unit of words. This does not include this header word.
- **INE (In-Stack NA Extension indicator)** - Flag in the TTL field indicates the presence of another set of NA sub-stack available in the packet for processing. The last NA sub-stack will set the Flag as “0”. This is used to extend the NA sub-stack to extend the In-Stack MNA encoding that exceeds 7 words.

Flag-based In-Stack Network Action Encoding Format:

- **ADL (Additional Data Length)**: This is the 2-bit field in the TC to indicate the length of Additional words that are encoded to carry the Additional Flag-based Network Action Indications (NAI).
- **E (End to End Processing)**: Flag in TC field that indicates whether the Flag-based In-Stack Network Action to be processed Hop-By-Hop or End-To-End. For End-To-End processing this will be set to “1”.
- **Flag-based-NAI**: Flag-based In-Stack NAs are encoded in the 19-bit Label Field. The MSB bit of the Label field MUST set to “1” to prevent aliasing with other SPLs. This bit position value is assigned by IANA.

IN-STACK MPLS NETWORK ACTION ENCODING FORMAT (OPCODE-BASED)

The In-Stack MPLS Network Actions encoding for Opcode-based as shown in the figure 6.

In-Stack Network Action with AD Encoding Format:

- **IS-NAI-Opcode:** This is the 8-bit value in the Label field. This indicates the In-Stack Network Actions opcode that has been assigned for a specific application by IANA.
- The characteristics of the opcode **MUST** be defined while the solution is developed and requesting for the value from IANA.
- **Ancillary Data:** This is 20-bit field. 12-bits are encoded in the Label field and other 8-bits are encoded in the TTL field.
- **ADL (Additional Data Length):** Some applications might need to encode more than 20-bit of ancillary data. Those applications can use this field to encode additional ancillary data.
- **E (End-to-End Processing):** This flag in TC field indicates whether the Flag-based In-Stack Network Action to be processed Hop-By-Hop or End-To-End. For End-To-End processing this will be set to “1”.

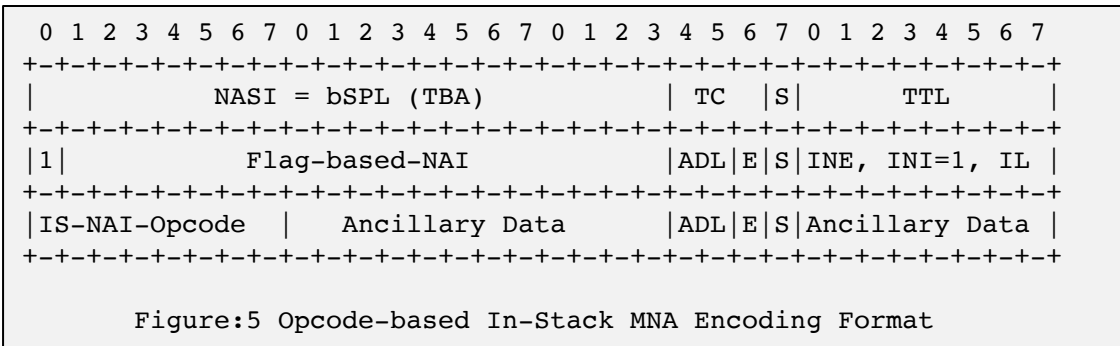
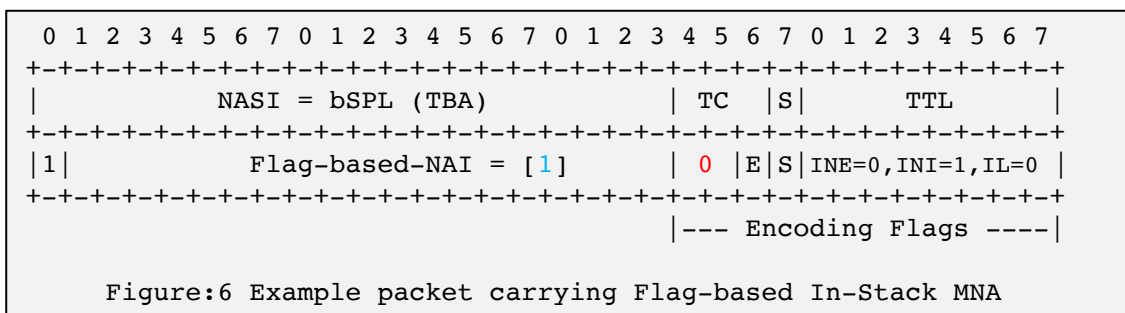


Figure:5 Opcode-based In-Stack MNA Encoding Format

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-1 (FLAG-BASED)



In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to "1" to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – Since there are no additional data is encoded, this field MUST be set to "0"
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encoded in the packet, this value MUST be set to "0"

Flag-based In-Stack Network Action Encoding Format:

- **Flag-based-NAI:** In this example, an application has set Flag-based bit position "1" (Assigned by IANA)
- **ADL (Additional Data Length):** The flag at bit position "1" is accommodated in the first 19-bit field, this value MUST be set to "0"
- **E (End to End Processing):** Based on the application, this value can be set to "1" or "0"

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-2 (MULTIPLE FLAG-BASED)

```

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
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          NASI = bSPL (TBA)          | TC | S |          TTL          |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| 1 |          Flag-based-NAI = [ 1 ]          | 1 | E | S | INE=0, INI=1, IL=1 |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| 1 |          Flag-based-NAI = [ 1 ]          | E | S | Flag-based-NAI |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Figure:7 Example packet carrying Multiple Flag-based In-Stack MNA

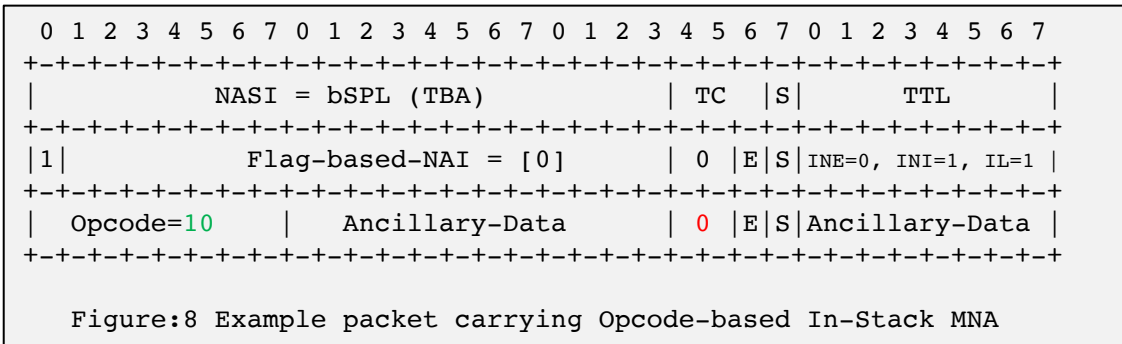
In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to "1" to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – This value MUST be set to "1", as we are using additional one word to encode In-Stack MNA
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encoded in the packet, this value MUST be set to "0"

Flag-based In-Stack Network Action Encoding Format:

- **Flag-based-NAI:** In this example, applications have set Flag-based bit positions "1" and "20" (Assigned by IANA)
- **ADL (Additional Data Length):** The flag at bit position "20" cannot be accommodated in the first 19-bit field, so this is encoded in the second word. Hence, this field MUST be set to "1"
- **E (End to End Processing):** Based on the application, this value can be set to "1" or "0"

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-3 (OPCODE-BASED)



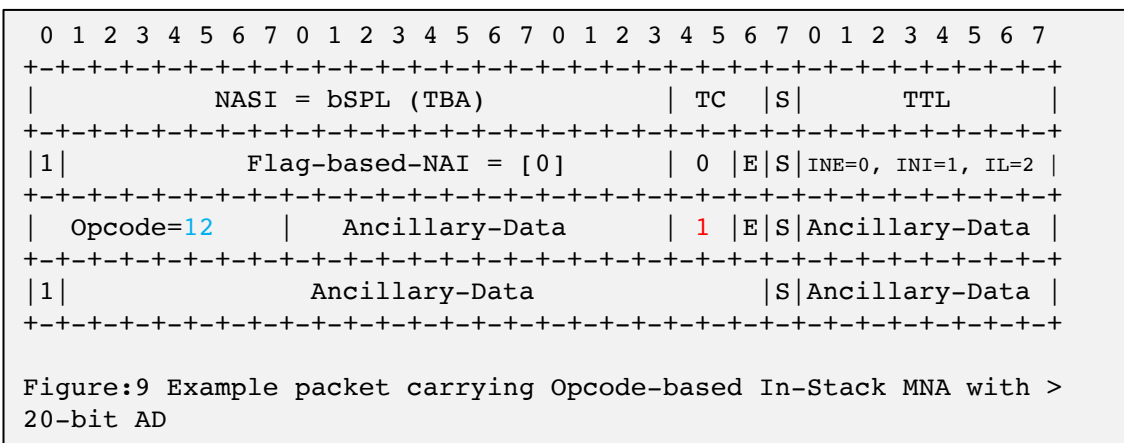
In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to "1" to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – This value MUST be set to "1", as we are using additional one word to encode In-Stack MNA
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encoded in the packet, this value MUST be set to "0"

In-Stack Network Action with AD Encoding Format:

- **Opcode:** In this example, an application has set the In-Stack opcode value "10" (Assigned by IANA)
- **Ancillary-Data:** In this example, the application opcode "10" is adding a 20-bit length of ancillary data
- **ADL (Additional Data Length):** In this example, since the ancillary data of length are within 20-bit boundary, this field MUST be set to "0"
- **E (End to End Processing):** Based on the application, this value can be set to "1" or "0"

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-4 (OPCODE-BASED WITH > 20-BIT ANCILLARY DATA)



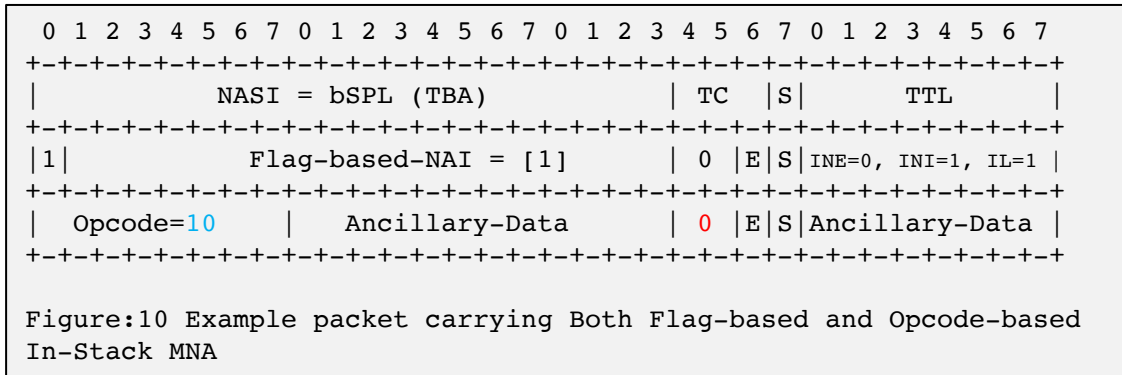
In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – This value MUST set to “2”, as we are using additional two words to encode In-Stack MNA
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encode in the packet this value MUST be set to “0”

In-Stack Network Action with AD Encoding Format:

- **Opcode:** In this example, the application has set the In-Stack opcode value “12” (Assigned by IANA)
- **Ancillary-Data:** In this example, an application with opcode “12” is adding a 32-bit length of ancillary data
- **ADL (Additional Data Length):** In this example, since the ancillary data of length is more than 20-bit boundary, this field MUST be set to “1”, indicating that additional word has been used to encode the Ancillary data corresponding to the opcode. **If the opcode is not recognized by the MNA capable devices, then this field would make it be easy for the ASICs to skip this opcode processing and move to the next opcode in the stack**
- **E (End to End Processing):** Based on the application, this value can be set to “1” or “0”

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-5 (BOTH FLAG-BASED & OPCODE-BASED)



In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – This value MUST be set to “1”, as we are using additional one word to encode In-Stack MNA
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encode in the packet this value MUST be set to “0”

In-Stack Network Action with Flag-based & AD Encoding Format:

- **Flag-based-NAI:** In this example, the application has set Flag-based bit position “1” (Assigned by IANA)
- **Opcode:** In this example, the application has set the In-Stack opcode value “10” (Assigned by IANA)
- **Ancillary-Data:** In this example, the application opcode “10” is adding a 20-bit length of ancillary data
- **ADL (Additional Data Length):** In this example, since the ancillary data of length is within 20-bit boundary, this field MUST be set to “0”
- **E (End to End Processing):** Based on the application, this value can be set to “1” or “0”

IN-STACK MPLS NETWORK ACTION ENCODING EXAMPLE-6 (MULTIPLE OPCODE-BASED AND FLAG-BASED)

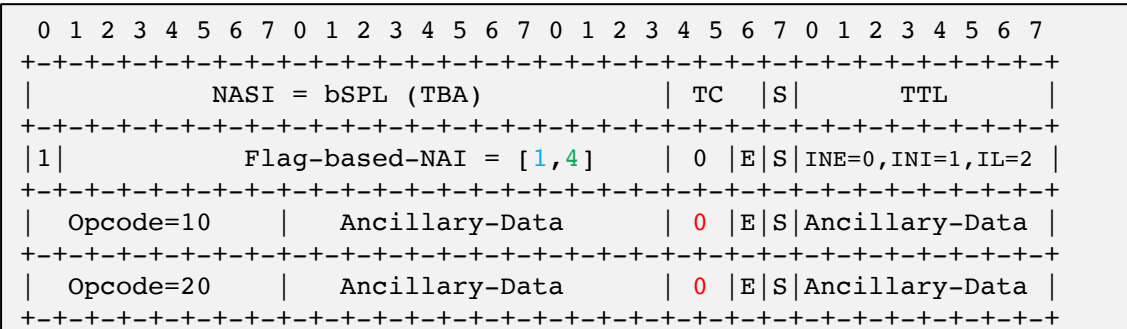


Figure:11 Example packet carrying Multiple Opcode-based and Flag-based In-Stack MNA

In-Stack Network Action Header:

- **INI (In-Stack NA presence Indication)** – This bit MUST be set to indicate the presence of In-Stack MNA
- **IL (In-Stack Data Length)** – This value MUST be set to “1”, as we are using additional one word to encode In-Stack MNA
- **INE (In-Stack NA Extension indicator)** – Since this is the only MNA sub-stack encode in the packet this value MUST be set to “0”

In-Stack Network Action with Flag-based & AD Encoding Format:

- **Flag-based-NAI:** In this example, two applications has set Flag-based bit position “1” and “4” (Assigned by IANA)
- **Opcode:** In this example, two application has set their In-Stack opcode values “10” and “20” respectively (Assigned by IANA)
- **Ancillary-Data:** In this example, the application opcode “10” and “20” are adding their 20-bit length of their corresponding ancillary data. Since the opcode and its corresponding AD are residing next to each other, it is easier for the ASICs to process
- **ADL (Additional Data Length):** In this example, since the ancillary data of length are within 20-bit boundary, this field MUST be set to “0”
- **E (End to End Processing):** Based on the application, this value can be set to “1” or “0”

BACKWARDS COMPATIBILITY

- Node that needs to process MNA MUST signal the MNA capability
 - Worst-case, the packet will be dropped if the MNA Label is exposed at the top on the non-supporting node
- ECMP behavior is not adversely affected by the MNA encoding
 - Labels in the Label stack does not change for a flow
- Does not alias with an existing reserved bSPL
- Penultimate node with TTL propagation behavior does not corrupt the MNA encoding
- Can co-exist with existing G-ACh
 - Opcode to specify the byte-offset for Post-Stack Network Action after the BOS

ADVANTAGES

- MNA encoding follows the MNA framework and requirements - both In-Stack and Post-Stack Data
- MNA encoding has **flexibility** to encode Network Actions in a desired order
- MNA encoding is **extensible** by defining new Network Action opcodes and Flag-based LSEs
 - Easy to extend for user-defined Network Actions
- MNA encoding is **hardware parser friendly**
 - Length field of the MNA sub-stack allows to easily skip sub-stack
 - Network Action Opcode, Ancillary Data, Scope and Length in a single LSE are easy to process in hardware- **does not require the parser to traverse back-and-forth**
 - Network Action Length allows to carry Ancillary Data of different lengths (including > 20-bits)
 - Network Action Length allows node to easily skip not-supported Network Actions
- **MSD efficient** encoding when enabling Network Action using only MNA Flags
- Backwards compatible in an existing network
- MNA encoding is **ECMP friendly**
 - Labels in the label stack does not change for a flow
 - Encoding allows to change AD in the TTL field for a flow if there is such a need

NEXT STEPS

- Welcome review comments and feedbacks
- Strong interest from multiple Vendors and Operators
- Requesting MPLS WG adoption

THANK YOU!

APPENDIX

HARDWARE ANALYSIS

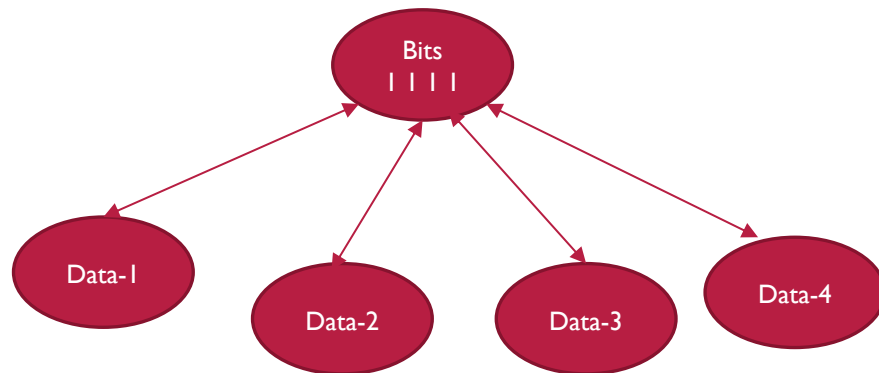
HARDWARE ANALYSIS

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.

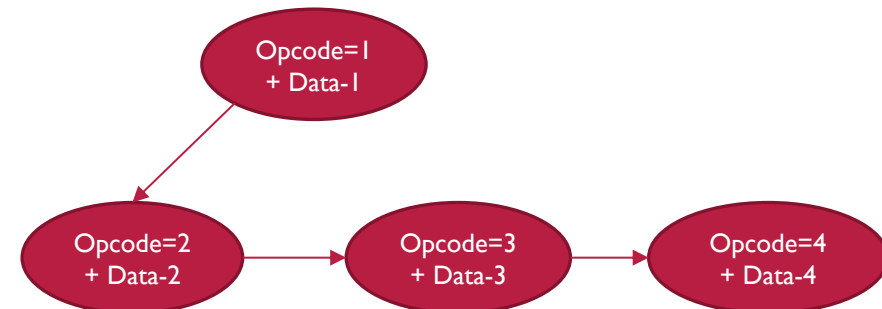
HARDWARE ANALYSIS - PARSING HEADER CHAINS

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

THANK YOU!