Multi-Purpose Special Purpose Label for Encoding Actions with Associated Data

Kireeti Kompella | Tarek Saad | Vishnu Pavan Beeram | Israel Meilik

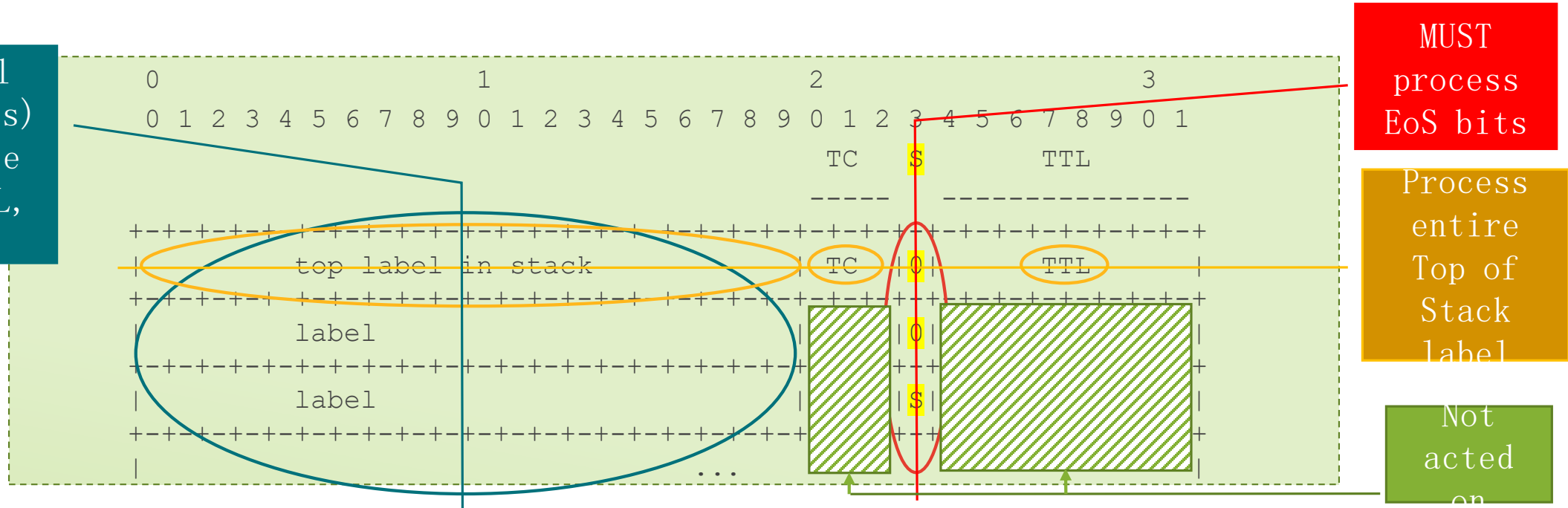Colby Barth | Chandra Ramachandran | Srihari Sangli | Swamy SRK

JUNIPER NETWORKS | Engineering Simplicity

Motivation

- There are multiple MPLS based features that require one or more actions to be applied in the forwarding plane along the path of an LSP

- Each of these features requires specific data to be carried in the MPLS packet

  - Some suffice in themselves, with no additional data

    - E.g., NFFRR - effectively, a boolean

  - Some features require further data to be carried

    - Which Slice-Aggregate does this packet belong to? (slide identifier)

    - Which SR path/Flow does this packet belong to? (path/flow ID)

    - Are there Generic Delivery Functions? (frag, etc.)

    - Is there OAM data within or beyond the MPLS label stack? (multiple types)

  - In the future, there may be other forwarding actions to consider

    - "Opaque" data offers a way to capture these

- Each of these would like one or more bSPLs as their "indicator"

Key Insight: Behavior of Forwarding Engines on Label Stack

Process label values (20 bits) for accessible labels for SPL, entropy

MUST process EoS bits

Process entire Top of Stack label

Not acted on

```
 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
                                        TC    S        TTL
                                       -----  -  --------------
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                top label in stack             | TC  |0|     TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    label                          |0|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    label                          |S|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      ...
```

- For labels <u>below the top of stack</u>, forwarding engines <u>look only at the label values</u> (to compute a hash, or to look for special purpose labels) <u>and the S bit</u> (to find end of stack)
- They *don't* look at the TC bits or the TTL value of labels below top of stack
- In which case, we can repurpose these bits on bSPLs and eSPLs
- Corollary: such labels MUST NOT reach the top of stack (by popping labels above them)

Approach

- ~~Defining a new special-purpose / extended-special-purpose label for every new feature is not ideal~~

  - This is expensive, generally involving a hardware re-spin or microcode update

- Proposal: use a <u>single bSPL</u> to *compactly* encode *multiple* forwarding actions in MPLS labels

  - This would serve as bSPL for EL, GISS, NFFRR, OAM, Path Identifier and Generic Delivery Function

  - EL already has an ELI, but the *rest are active requests* (except for GDF, which may join in)

  - This bSPL (called FAI) uses the TC/TTL bits to encode more data

  - FAI is accompanied by forwarding actions data (FAD), also compactly encoded

  - This approach can serve as the blueprint for future bSPLs, again using the TC/TTL fields to encode more information

- A key new use case is the Global Identifier for Slice Selector (GISS)

  - Enables MPLS-based transport network slices (cf. draft-bestbar-teas-ns-packet)

Theory of Operation

```
                  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
                                                          TC     S        TTL

                                                         -----   --------------
                  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                  |          (previous forwarding label   | TC  |0|       TTL       |
                  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ──▶             |          Multi-purpose bSPL           |H B Q|0|b b b b b b b b| action bits
                  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ──▶             |          Associated Data ...                |S|      ...       |
                  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ──▶             |                            ...              |S|                |
                  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- The bSPL is identified by the label field (20 bits)

- The "TC" bits a further header (H), data beyond the LS (B) and Opaque data (Q)

- The "TTL" bits define the action bits in variously sized bit fields

  - Each of these actions may have associated data, which appear in the following labels
  - Each Associated Data word carries 31 usable bits, i.e., all bits but the S bit

Auxiliary Information

- This bSPL may occur anywhere in the label stack (except at the top)

  - In particular, it may be repeated for reasons of "readable stack depth"

- This technique is not limited to bSPLs – eSPLs can use it as well

- The number of Associated Data words and their parsing depends on the action bits

- The H bit allows for an expanded header (more action bits); the following word is the expanded header

- The data beyond end of stack is assumed to be self-identifying (e.g., TLVs)

  - The B bit just says that there is some interesting data beyond the stack

- The Q bit indicates that there are further associated data beyond those identified by the action bits

Specific Instance: FAI/FAD

- Use a bSPL called "Forwarding Actions Indicator" (FAI, value TBD) to signal that certain forwarding actions must be taken, and provide data for them (FAD)

- The data is divided into those that are part of the <u>label stack</u> (LS FAD), and those that come after the label stack (in the <u>payload</u>) (PL FAD)
  - Each section has a well-known part and an opaque part
  - Ignore the "opaque" part for now

- We suggest that the FAI be used for: Entropy, GISS, NFFRR, FlowID, …

- … as well as "interesting" information beyond the end of stack: OAM, GDF, PathID

Proposed Next Steps (MPLS WG)

We propose a new bSPL be allocated for FAI, and the process of retiring the ELI be started

- " make before break"

- The " break" should be on the books, whether implemented or not

We propose that any future requests for bSPLs be examined as to whether this technique applies

We further propose that this approach be considered for existing bSPLs, if applicable, with a view to optimizing their use and effectiveness
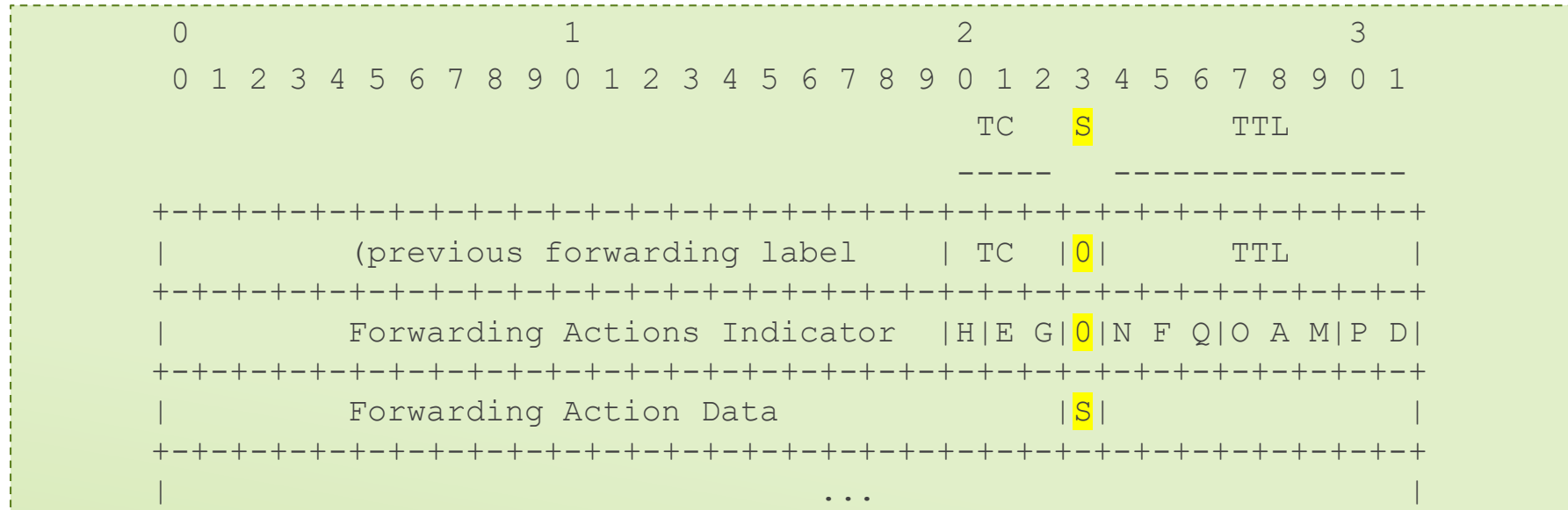
- Again, "make before break"

We will include considerations for eSPLs in our next revision

We STRONGLY SUGGEST ASIC vendors look at this proposal and send the authors feedback

Forwarding Actions Ind (FAI) / Fwding Actions Data (FAD)

```
        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
                                                TC     S        TTL

                                               -----    --------------
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |          (previous forwarding label    | TC  |0|      TTL      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |          Forwarding Actions Indicator  |H|E G|0|N F Q|O A M|P D|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |          Forwarding Action Data                |S|            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
```

- Forwarding Actions Indicator: This is an MPLS base Special Purpose Label (bSPL). It indicates the presence of LS Forwarding Actions Data (FAD) immediately below this label stack entry in the label stack.
  - The TC and TTL fields of the FAI encode the types of FAD
  - The "end of stack" bit is sacrosanct and must reflect the end of label stack correctly
- Forwarding Actions Data: This includes "well-known" data and opaque data

Example: FAI/FAD in Label Stack (LS) & in Payload (PL)

```
Flag  Meaning
----  -------

H     Header
E     Entropy
G     GISS          ⎫
N     NFFRR         ⎬  LS FAD
F     Flow ID       ⎪
Q     FFS           ⎭
=============       ⎫
OAM   FFS           ⎬  PL FAD
P     Path ID       ⎪
D     GDFH          ⎭
```

Header: For Further Study
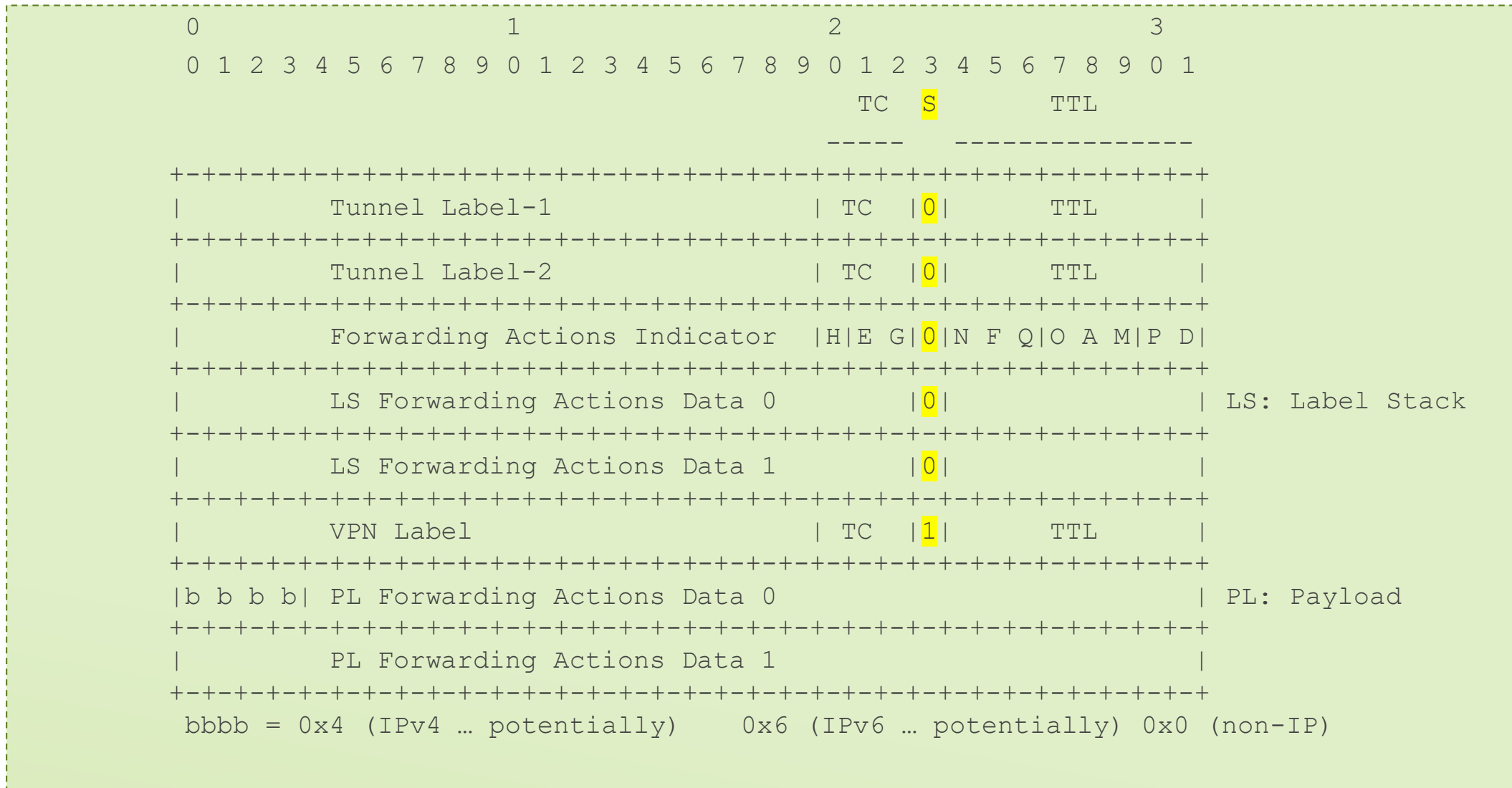Entropy: RFC 6790
GISS: Global Identifier for Slice Selector
NFFRR: No Further Fast ReRoute
Flow ID: RFC 8321
OAM: Operations, Administration and Management
GDFH: Generic Delivery Function Header

Example: FAI/FAD in Label Stack (LS) & in Payload (PL)

```
                    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
                                                            TC   S      TTL
                                                            ----- --------------
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |         Tunnel Label-1                | TC  |0|      TTL      |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |         Tunnel Label-2                | TC  |0|      TTL      |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |      Forwarding Actions Indicator     |H|E G|0|N F Q|O A M|P D|
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |      LS Forwarding Actions Data 0     |0|                     | LS: Label Stack
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |      LS Forwarding Actions Data 1     |0|                     |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |         VPN Label                     | TC  |1|      TTL      |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |b b b b| PL Forwarding Actions Data 0                          | PL: Payload
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   |         PL Forwarding Actions Data 1                          |
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                    bbbb = 0x4 (IPv4 … potentially)    0x6 (IPv6 … potentially) 0x0 (non-IP)
```

Notes

- The H bit is for an extended Forwarding Actions Header (FAH) - more flags, etc. (TBD)

- The contents of the LS FAD is determined by the LS FAD flag bits HEGNFQ
  - Ignore the Q flag (Opaque LS FAD) for now
  - These flags indicate the presence and order of items in the LS FAD

- The R bit is reserved for future use

- The contents of the PL FAD is determined by the PL FAD flag bits OAMPD
  - Ignore q (Opaque PL FAD) for now
  - These flags indicate the presence and order of items in the PL FAD

- The first nibble of the payload MUST NOT be 0x4 or 0x6 if used for
  - Hardware in the field may use this for load balancing info
  - If the first field in the payload is iOAM, this nibble is 0x1

What about the Control Word?

How the LS FAD Flag Bits Are 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
                                                TC   S       TTL

                                                -----   --------------
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |          Forwarding Actions Indicator   |H|E G|0|N F Q|- - - - -|
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |          LS Forwarding Actions Data 0         |0|               | LS: Label Stack
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |          LS Forwarding Actions Data 1         |0|               |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |          LS Forwarding Actions Data 2         |0|               |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                               ...                        1
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |b b b b|            (payload/control word)                      |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

How the LS FAD Flag Bits Are Used

```
uint32_t *NF = (&FAI)+1 ("next FAD word")

if (H): hdr = *NF++;
switch (EG): // Ent & GISS
  case 00: break;
  case 01: Ent = *NF >> 12; GISS = *++NF & 0x0eff; break;
  case 10: Ent = *NF >> 16; GISS = *++NF & 0xfeff; break;
  case 11: Ent = *NF++ & 0xfffffeff; GISS = *NF++ & 0xfffffeff; break;

NFFRR = (N == 1);
if (F): FID = *NF++; // Flow ID

H  =   1: FAD 0 is a header (for expansion)
EG =  00: no Entropy/GISS data
EG =  01: LS FAD 0 is a combo: top 20 bits are Entropy, bottom 12 bits are GISS + S
EG =  10: LS FAD 0 is a combo: top 16 bits are Entropy, bottom 16 bits are GISS + S
EG =  11: LS FAD 0 is a 31-bit Entropy field; LS FAD 1 is a 31-bit GISS
N  = 0/1: do FRR as usual if needed/do NFFRR (drop packet if FRR is needed) (no LS FAD)
F  =   1: LS FAD contains a Flow ID (with bits for Alternate Marking)
```

EG Field

Entropy is used for load balancing (from RFC 6790)

GISS = Global Identifier for Slice Selector: data field used for QoS classification and path steering

A GISS value of 0 means "default slice" – use the TC bits for QoS

If the EG field is 00, the FAD carries neither an Entropy field nor a GISS

If the EG field is 11, the FAD carries both an Entropy field and a GISS, each 31 bits long
- This means that FAD is at least 2 words

The other values are a combo, where a single word contains both GISS and Entropy, with different splits

If you want NFFRR and Entropy and no GISS: set N to 1 and EG to 01 or 10 (20 or 16 bits of entropy)
- set the GISS to 0 (default)
- FAI + FAD = 2 words (can replace ELI + EL and (eventually) deprecate ELI)

Added Depth to Label Stack

Desired Actions	FAI + length of LS FAD	Dedicated bSPL per FA
NFFRR	1	1
EL	2	2
NFFRR + EL	2	3
NFFRR + EL + GISS	2/3	6
NFFRR + FI	2	3
FI	2	2
FI + EL	3	4

Repeating the FAI/FAD for "Readable Label Depth"

To accommodate routers that have a limited "lookahead" for reading labels, the FAI can be repeated

There are a few alternatives here: the repeated FAIs can set EG to 00, so it's a single word

- The "last" FAI has these bits set correctly, with the correct GISS and Entropy

- Other routers can search for the last FAI to get the real GISS and Entropy

Alternatively, the repeated FAI can set the GE correctly, at the cost of repeating the GISS/EL

Needs more thinking

How the PL FAD Flag Bits Are 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
                                                   TC  S         TTL

                                                  -----    ---------------
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             |       Forwarding Actions Indicator  |-|- -|0|- - -|O A M|P D|
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                      ...                  1
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             |b b b b| Control Word/PL Forwarding Actions Data 0           | PL: Payload
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             |           PL Forwarding Actions Data 1                       |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

OAM = bbb: various types of OAM may exist
P   =   1: a Path Identifier exists (used for Path accounting)
D   =   1: Generic Delivery Function Header exists

PL data MUST be self-identifying (especially the length)
PL MUST occur in the order of the flag bits that are set
PL starts with a CW if CW was signaled end to end in the control plane (?)
PL FAD 0 MUST respect bbbb != 0x4 or 0x6

References

Entropy Label: RFC 6790

GISS: https://tools.ietf.org/html/draft-bestbar-teas-ns-packet-01

NFFRR: https://tools.ietf.org/html/draft-kompella-mpls-nffrr-01

Alt Mark: https://tools.ietf.org/html/draft-cheng-mpls-inband-pm-encapsulation-04 ; RFC 8321

iOAM: https://tools.ietf.org/html/draft-gandhi-spring-ioam-sr-mpls-00

Path Identifier: https://tools.ietf.org/html/draft-hegde-spring-traffic-accounting-for-sr-paths-02

Generic Delivery Header: https://tools.ietf.org/html/draft-zzhang-tsvwg-generic-transport-functions-00