

Redefining ELI considered harmful; NPL considered harmful

Stewart Bryant

John Drake

Tony Li

Existing proposals

- Two presentations on re-using ELI for MIAD:
 1. draft-jags-mpls-ext-hdr-00
 - Predecessor discussed 2/10/2022 in Open DT
 2. draft-gandhi-mpls-ioam-03
 - Discussed 2/24/2022 in Open DT
- Base proposal: draft-decraene-mpls-slid-encoded-entropy-label-id-03

Claims of Advantages with ELI

1. Faster deployment since EL already deployed
2. Single label for Entropy -> smaller label stack
3. No hardware support necessary for another SPL
4. Saves an SPL and no new signaling for the SPL
5. ECMP still works
6. Label stack is smaller if we need MIAD and EL

Claim 1

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

- What is the deployment critical path?
 - WG convergence: we need to agree on what we're doing
 - Software development: we go write forwarding plane and signaling code
 - Testing & Deployment
- Expected time frame: 3 years
- Incremental signaling changes are stipulated. SPL signaling is not critical path.
- Claim 1 is inconsequential.

Claim 2

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

- Any proposal that includes Entropy in the MIAD extension has the same property.
- This is NOT specific to reusing ELI.
- Claim 2 is false.

Claim 3

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

- New platforms that could have hardcoded MIAD support are years away. Having ubiquitous deployment of these is decades away.
- Most new platforms will have software driven NPUs.
- MIAD must be supportable on legacy platforms.
- Legacy platforms that will add MIAD support will not require hardware changes. That’s why they’re legacy.
- Claim 3 is irrelevant. MIAD cannot wait for new hardware.

Claim 4

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

- a) Reusing ELI would save an SPL, but creates risk because it’s not backwards compatible. Adding new LSEs for ISD after ELI is not compatible.
- b) Adding signaling extensions is already stipulated.
- Claim 4a is true, but is more than offset by the risk.
- Claim 4b is inconsequential.

Claim 5

“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.”

- Assuming MIAD includes EL, this is always true, regardless of SPL.
- Claim 5 is false.

Claim 6

“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.”

- Assuming MIAD includes EL, this is always true, regardless of SPL.
- Claim 6 is false.

Using Network Programming Labels (NPL)

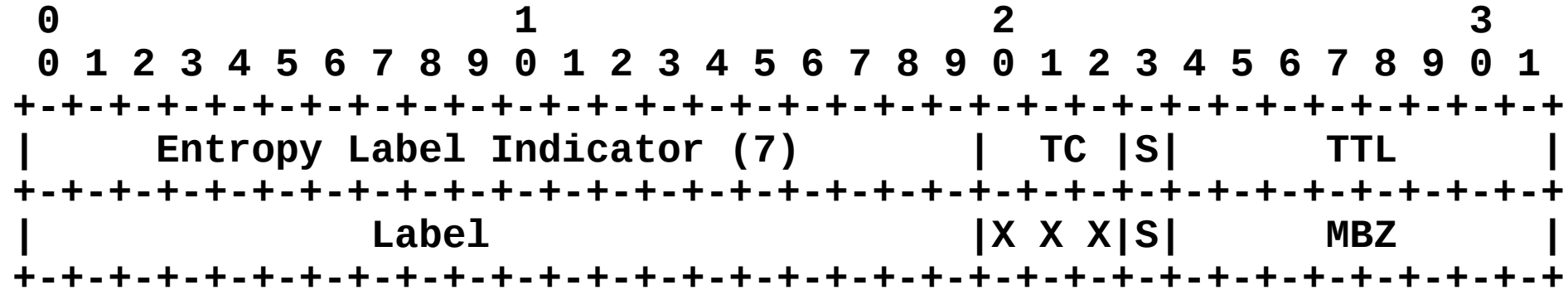
- NPL not defined in any IETF document we can find
- External documents suggest that NPL is SR specific
- MIAD is not SR specific
 - See MIAD requirements (draft-bocci-mpls-miad-adi-requirements-02), section 3.1.1, requirement 12

Summary

- Reusing ELI
 - Claims of the benefits of reusing ELI are either false or inconsequential
 - Risk of backward compatibility problems is high
 - This is a poor choice
- Using NPL
 - Does not meet our requirements
- Conclusion: We should use a new SPL for MIAD.

Backup slides
Backward compatibility

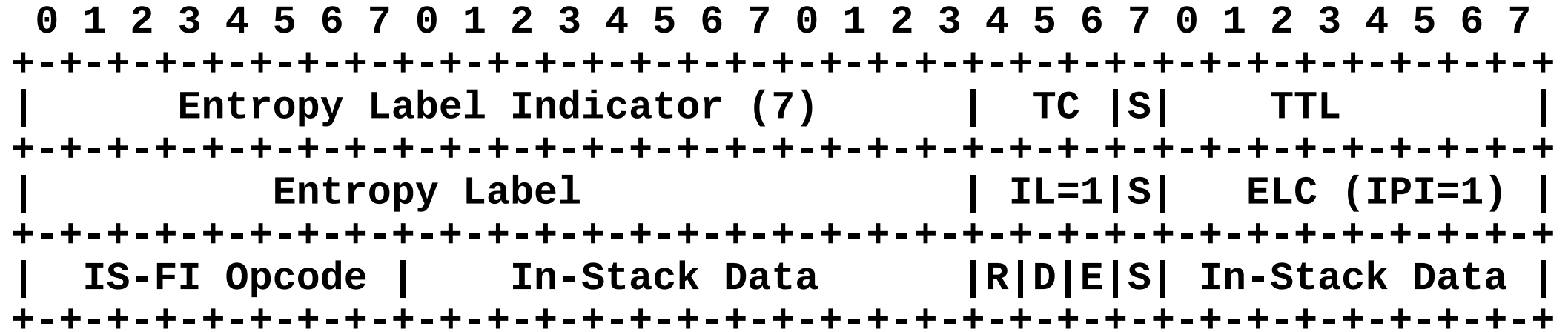
Legacy entropy labels (RFC 6790)



MBZ: Must be zero

**“The TTL for the EL MUST be zero to ensure that it is not used inadvertently for forwarding. The TC for the EL may be any value.”
(RFC 6790)**

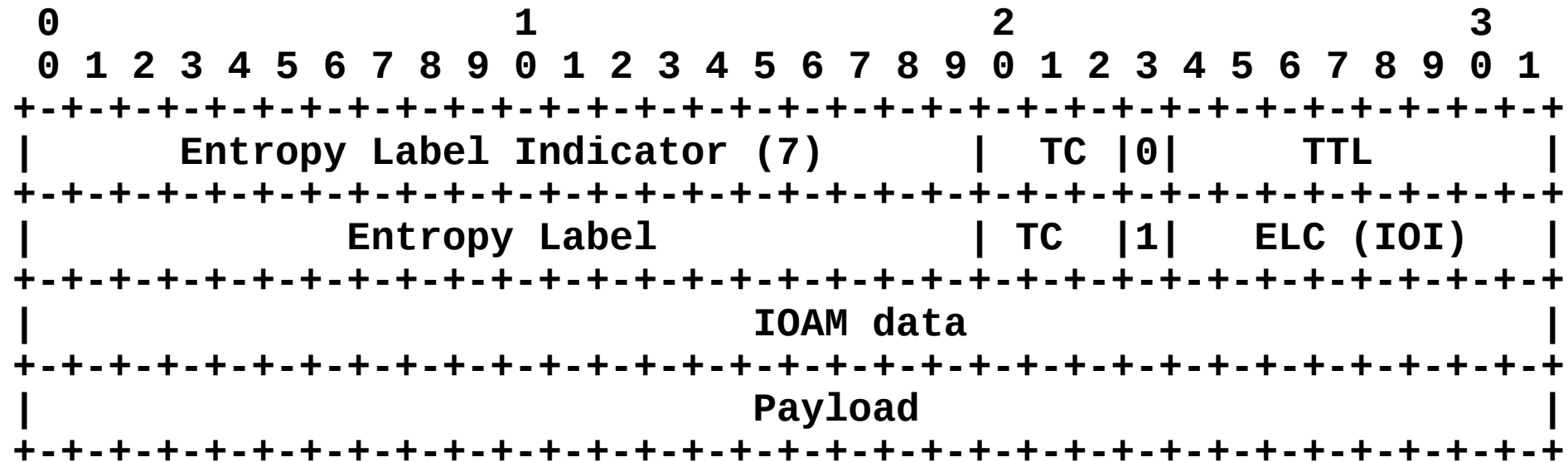
Proposal 1 encoding



Backward compatibility

- Suppose that a legacy hard-coded LSR sees this in the stack, how will it interpret this?
- LSR expects to see ELI + EL. It instead gets ELI + EL + <stuff>
- What happens if LSR pops ELI? It removes ELI + EL.
- We end up with <stuff> at top of stack.

Proposal 2 encoding



Backward compatibility

- ELI + EL now appear at bottom of stack, then IOAM, then payload
- What happens if legacy LSR goes to pop ELI + EL?
- IOAM is treated as payload

Summary

- At best, claims of backward compatibility apply to draft-decreaene-mpls-slid-encoded-entropy-label-id-03
- That draft only redefines bits within $\langle \text{ELI} + \text{EL} \rangle$
- Adding more LSEs onto $\langle \text{ELI} + \text{EL} \rangle$ is not backward compatible.