MPLS Extension Headers: Enabling Extensible In-Network Services in MPLS Networks

draft-song-mpls-extension-header
draft-song-mpls-eh-indicator
draft-andersson-mpls-eh-architecture
draft-andersson-mpls-eh-label-stack-operations

Haoyu Song, Robin Li, Tianran Zhou, Loa Andersson, Jeffery Zhang, Jim Guichard, Stewart Bryant

Version History

- MPLS Extension Header (EH) draft -00 published in July 2018, evolves to -06 today
- MPLS EH Indicator (EHI) draft -00 published in February 2019, evolves to -04 today
- Two MPLS EH Architecture and Operation drafts -00 published in February 2019, evolves to -02 today

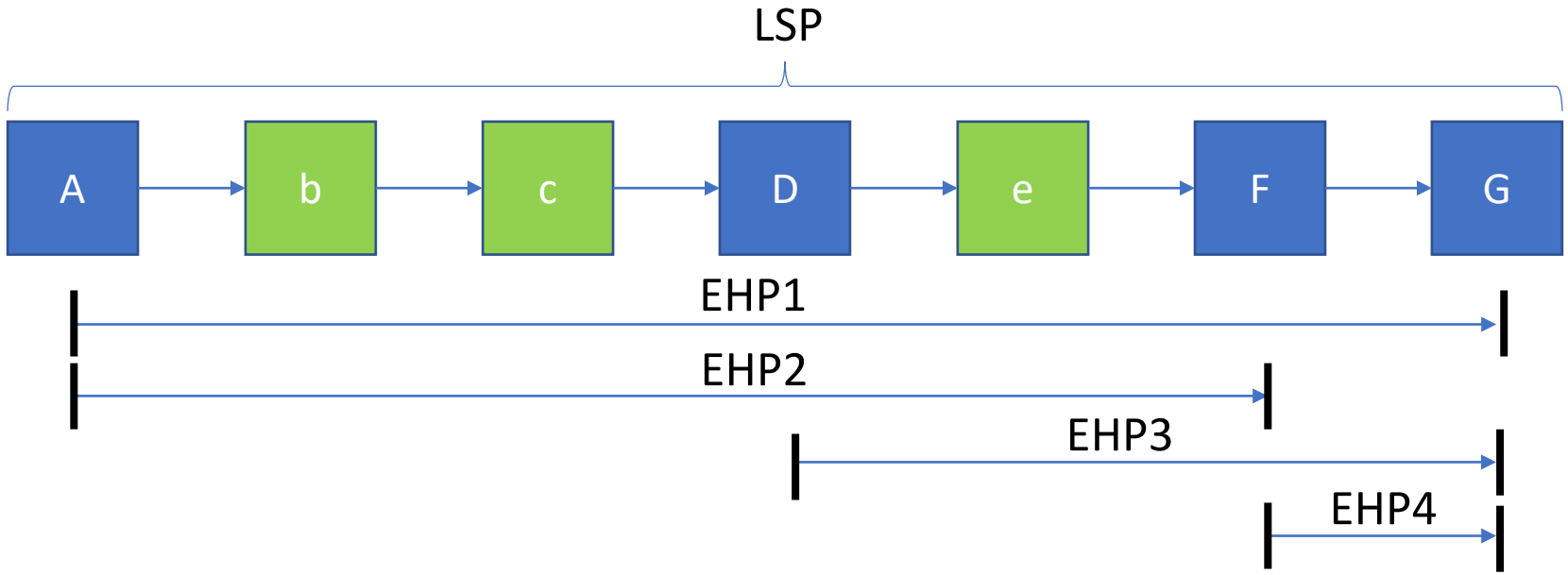
Motivation

- In-Network Services over user packets
  - In-situ OAM
  - Network Slicing
  - Service Function Chaining (SFC)
  - Bier
  - Segment Routing/Network Programming
  - Network security, network telemetry …
- Requirements
  - User packet to encapsulate extra instruction header or metadata
  - Add, process, and remove instruction header or metadata in a network
  - Possibly stack multiple coexisting services on one packet
  - Should support fast data-plane processing
- Supporting In-Network Services in MPLS

Solution – MPLS Extension Headers

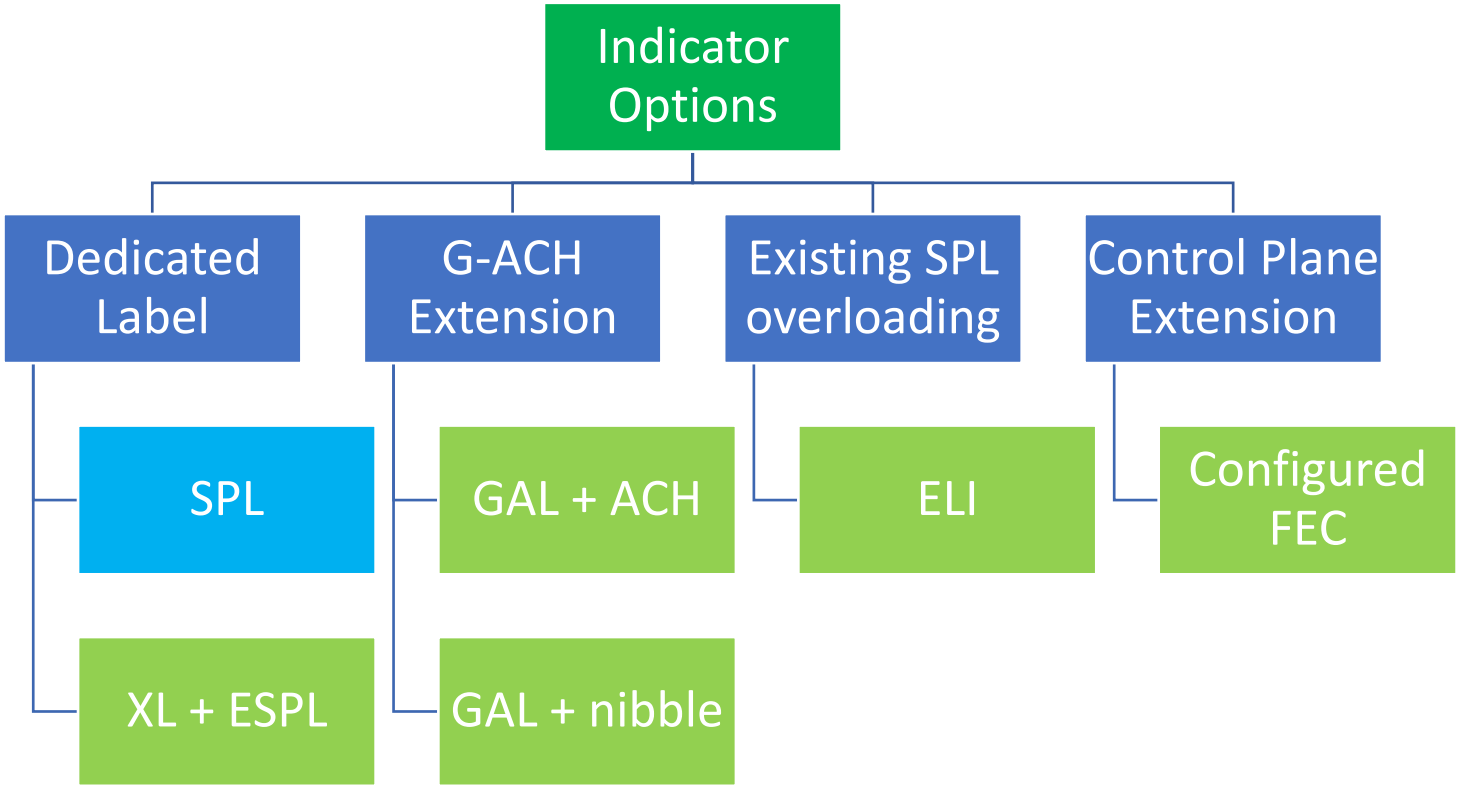
- Stop designing piecemeal and incompatible solutions which compete the same resource (e.g., SPL, the location after the label stack)
- Instead, a generic framework once for all: Extension Headers (EH) between MPLS label stack and payload with an in-stack indicator
- Learn the lessons from IPv6 EH!
  - Only end hosts are allowed to add/remove EHs → in network operations
  - Only one HBH header allowed, forcing a hierarchical structure to support multiple HBH options → allow multiple chained HBH headers
  - Need to scan through all the EHs to access the original L4 headers → allow skipping EHs in one step
  - Drop packets with unknown EHs → ignore unknown EHs
  - Not necessarily for fast path → optimized for data plane fast path processing

Requirements



- Flexibility
- Extensibility
- Performance
- Backward compatibility

The In-stack EH indicator (EHI) options



- MPLS Open DT has decided to not go to the GAL/GACH path
- SPL is our preferred method
- Proposal has been made to encode the EHI with other information

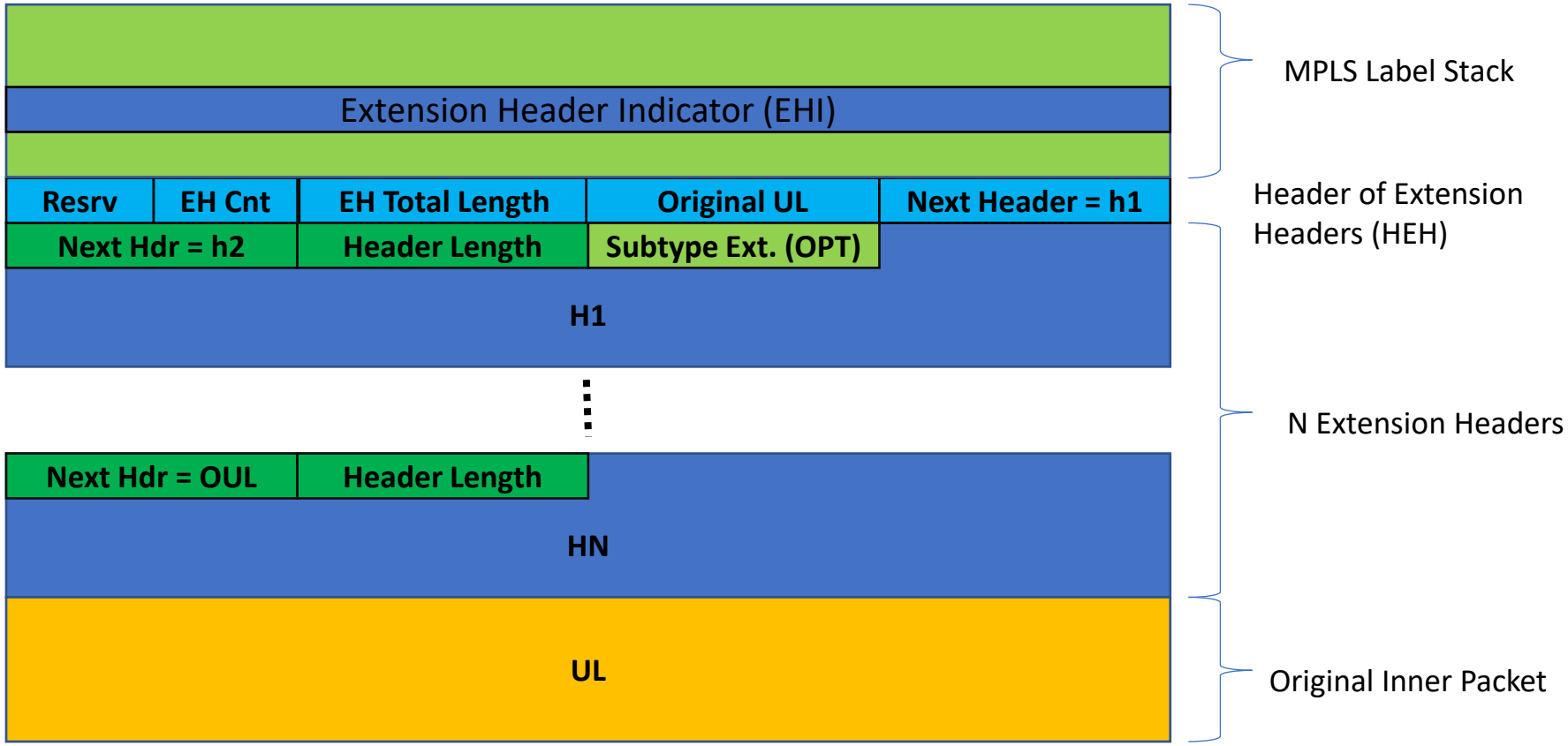
EHI SPL

- Redefine unused CoS/TTL field in the EHI SPL
- “H” flag indicates the existence of HBH EH(s)
  - Existence of EHI itself indicates the existence of EHs
  - Help non-EHP-edge-nodes to avoid unnecessary EH checking
- “EH offset” provides the offset of the HEH from the current location
  - Only useful if EHI is not at BoS
  - Could use fewer bits and save some bits for other purpose

MPLS Extension Header (EH)

- Multiple Extension Header(s) can be chained together
  - Each EH indicates the length of itself and the type of the next EH
  - EH type could adopt the standard Internet protocol numbers
  - For better extensibility, an EH could have subtypes, specified in a subfield
- Special Next Header types
  - “NONE”: no next EH and payload, for special packets (e.g., probe)
  - “UNKNOWN”: only in last EH, indicate the payload type is unknown
  - “MPLS”: another MPLS label stack follows
- EHs are located after BoS
  - If GAL/GACH is present, located after GACH
- All EHs can be jumped in one step
  - A Header of EH summarize the EH stack
- Support E2E and HBH types
  - E2E EHs are located below HBH EHs

MPLS EH Format Details



- Up to 15 EHs in a packet allowed
- Maximum lengths of EHs is 1K Bytes
- Allow HEH + 0 EH

Performance Optimization using FEC labels

- The need to find EHI below ToS could be a performance concern
- When establishing an LSP, two FEC labels are advertised, and one of it means “No EH in the packet”
- EH-incapable nodes do the regular forwarding
- EH-capable nodes
  - If regular label is received, need to examine if there are EHs in the packet
    - If yes, use regular label to forward the packet
    - If not, use “No EH” FEC label to forward the packet
  - If “No EH” label is received
    - If the node doesn’t add EH to the packet, no need to examine EH, continue to use “No EH” label to forward the packet
    - Otherwise, use regular label to forward the packet

Hardware Implementation Considerations

- Design for simplicity and performance
  - A simple post-stack EH header chain with a simple in-stack indicator can minimize the parser FSM size (storage) and depth (latency)
  - HBH EHs listed before E2E EHs to maximize the usability given limited header buffer size
  - Prefer to put the EH indicator at the BoS for simplicity and backward compatibility
    - Could use FEC labels to avoid unnecessary label stack scanning

Summary

- EH is a generic solution for MPLS in-network services
  - Built on common industry practices
  - Keep performance, flexibility, and extensibility in mind
- EH is especially compelling for MPLS
  - MPLS label stack overhead is much smaller than IPv6
  - MPLS is protocol independent, can encapsulate various protocols
  - No too much history burden. More freedom for innovations