MPLS Extension Header: 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

Acknowledgements

- Coauthors: Robin Li, Tianran Zhou, Loa Andersson, Jeffery Zhang, Jim Guichard, Stewart Bryant
- Contributors: Kireeti Kompella, Bruno Decraene, Andrew Malis, …
- Thanks to all the chairs and attendees in the MPLS Open DT weekly meetings for fruitful discussions!

- Note: new updates in the following slides are colored in red

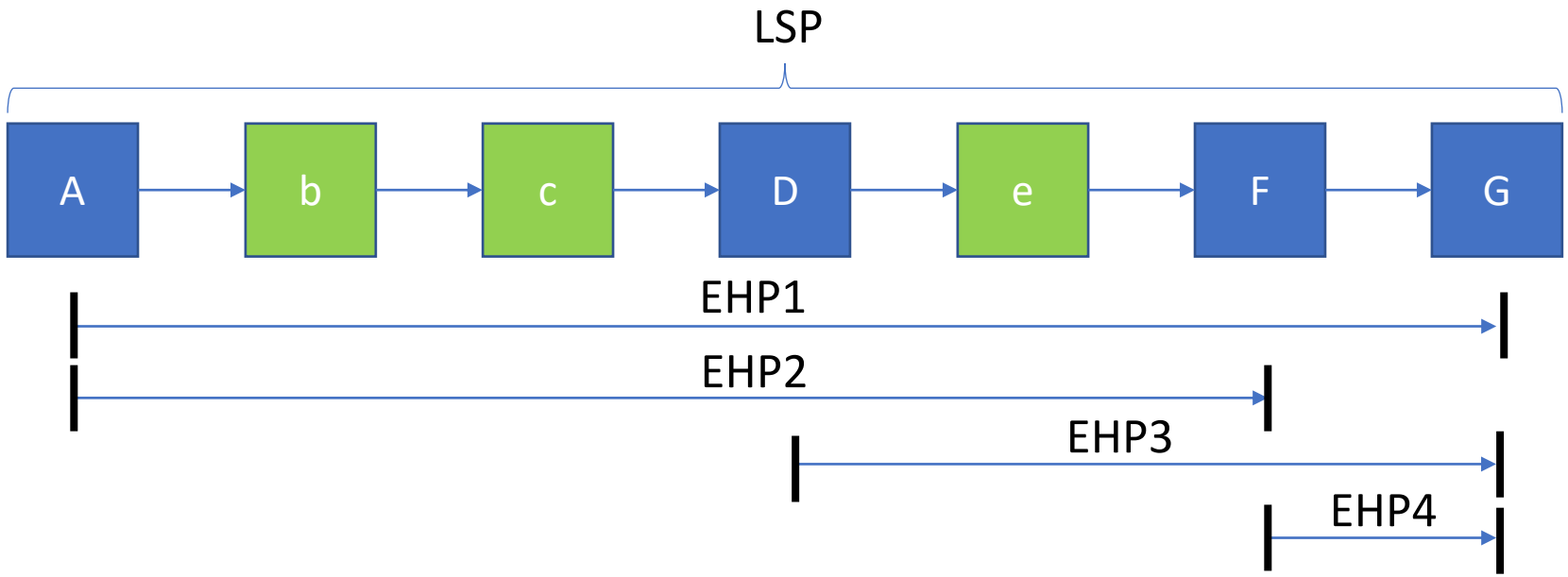
Motivation

- In-Network Services (INS) over user packets
  - In-situ OAM
  - Network Slicing
  - Service Function Chaining (SFC)
  - Bier
  - Segment Routing/Network Programming
  - Network security, network telemetry …
- INS 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
- Supporting INS in MPLS

Solution – MPLS Extension Headers

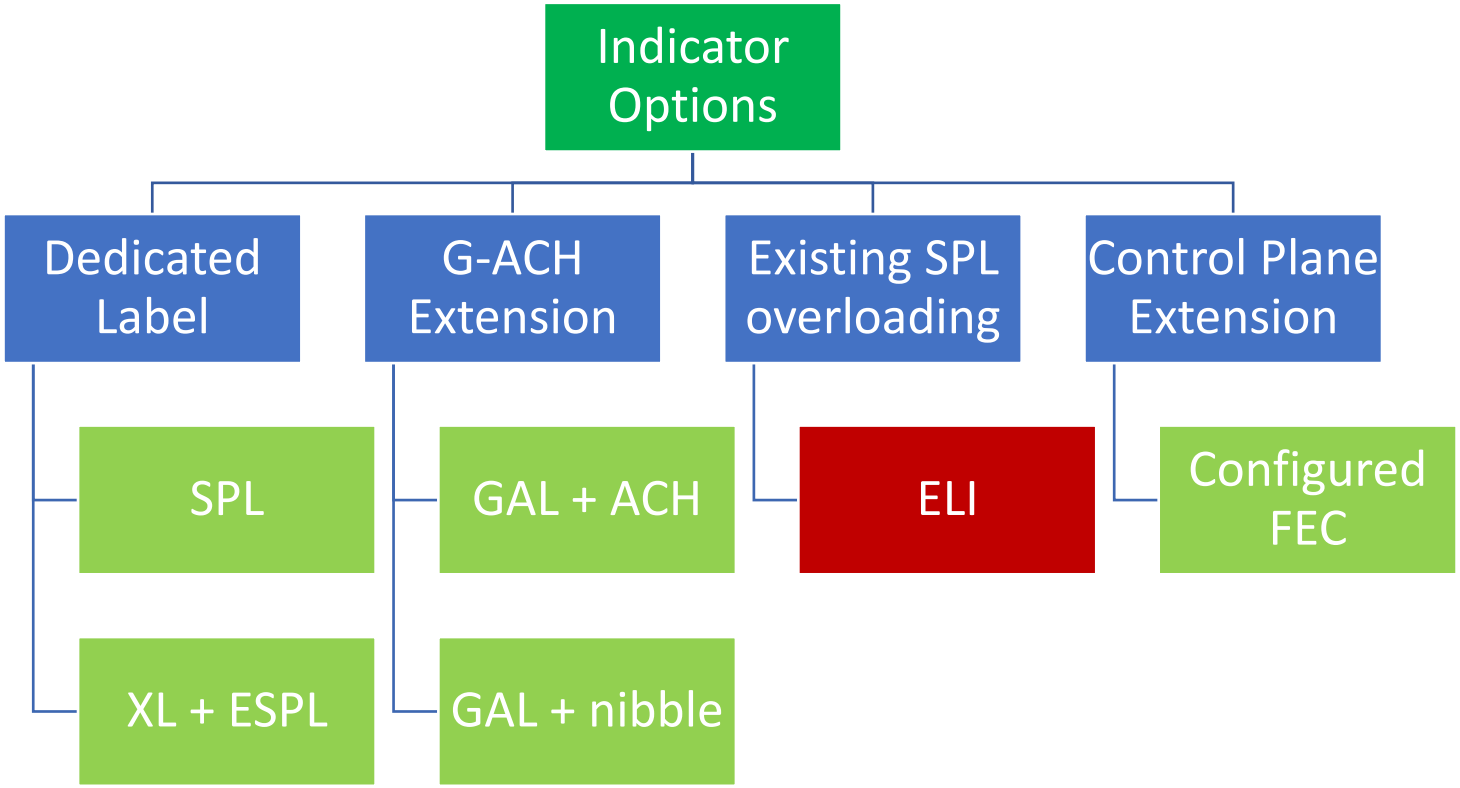
- Stop designing piecemeal and incompatible solutions which compete the same resource (e.g., SPL, the location after the label stack)
- Instead, a generic framework once for all: INS instruction headers/metadata as Extension Headers (EH) between MPLS label stack and payload
- Learn the lessons from IPv6 EH!
  - Only end hosts are allowed to add/remove EHs
  - Only one HBH header allowed, forcing a hierarchical structure to support multiple HBH options
  - Need to scan through all the EHs to access the original L4 headers.

Requirements



- Flexibility
- Extensibility
- Performance
- Backward compatibility

Above BoS: the 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 (e.g., FAI) to overload the EHI with other functions

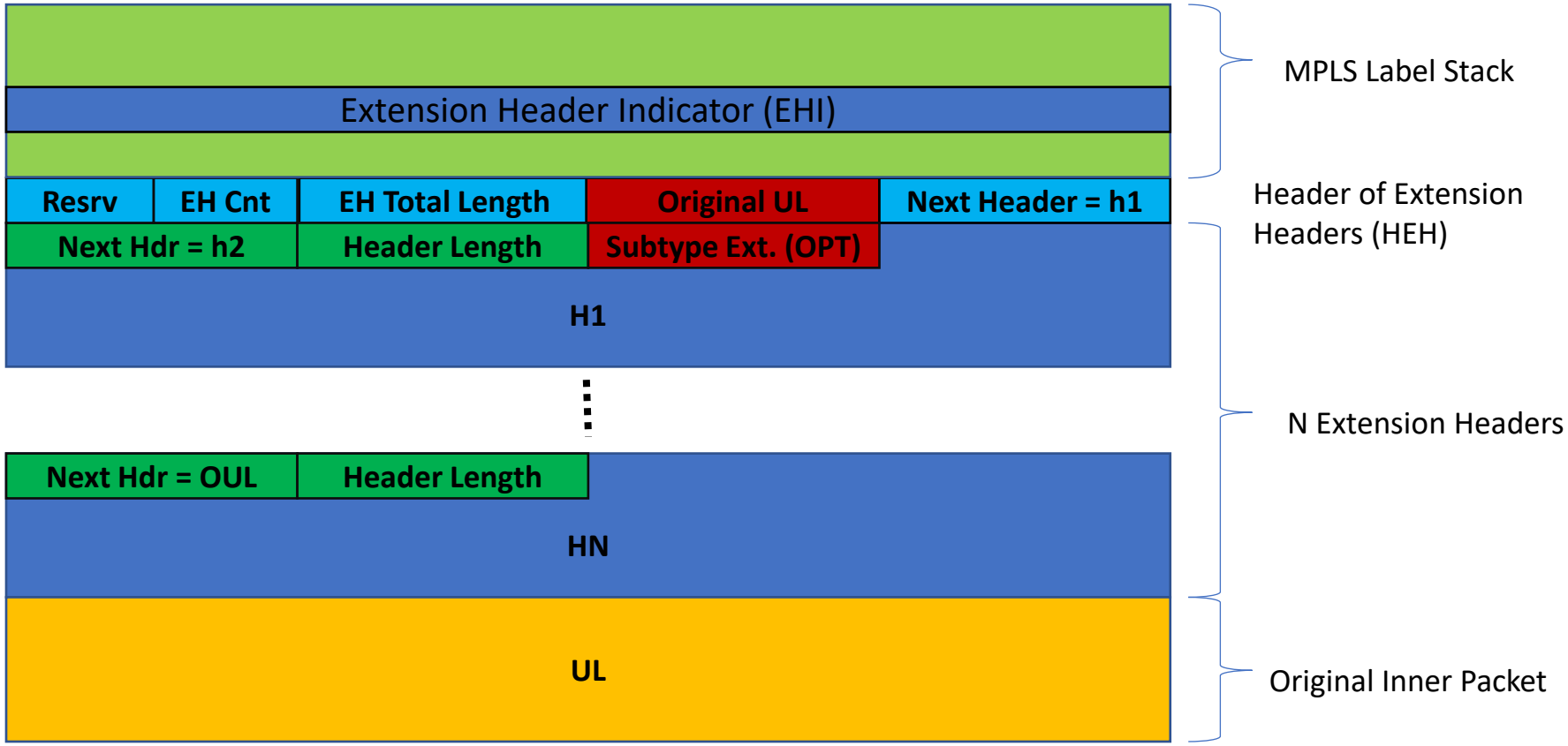
EHI SPL

- Redefine unused CoS/TTL field in the EHI SPL
- “H” flag indicates the existence of HBH EH(s)
  - Help non-EHP-end-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

Below BoS: MPLS Extension Header (EH)

- Multiple Extension Header(s) can be stacked 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



- At most 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

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
- Let’s keep it rolling!