MPLS Extension Header for In-network Services

draft-song-mpls-extension-header-00

Haoyu Song
Zhenbin Li
Tianran Zhou
Stewart Bryant
Andrew Malis
“No idea is so bad that it won’t be proposed over and over again”

George Swallow
From IPv4 to IPv6

- Extend the address space
  - Not so successful due to NAT

- Extension headers!
  - Offer huge innovation possibilities
    - Security
    - Segment routing
    - In network services (SFC, In-situ OAM)
    - Network programming!

- However, there are still issues unsolved
  - IPv6 header overhead (40-byte base header)
  - Can’t quickly skip the extension headers to access the upper layer protocols
Can we do the same thing on MPLS?

• MPLS is imperfect
  • No indicator for the upper layer protocols
  • Difficult to encapsulate new headers and metadata

• Many not so successful attempts to fix MPLS
  • The room is tiny
  • Backward compatibility!

• “Case by case” patching is not good
  • Difficult to combine multiple special cases
  • Difficult to extend
  • Difficult to standardize
  • Difficult to support future needs

• Designing a general mechanism to solve a lot of problem and create a lot of innovation opportunities is plausible
  • Learn experience from the other protocols!
Time is coming!

• In-network services need to be supported by MPLS
  • INT/IOAM
  • Network Programming
  • DDOS prevention
• Multiple services may need to be stacked together
• Need to be backward compatible if needed
• Performance considerations
  • Avoid unnecessary label stack scanning
  • Allow quick access to the inner packet
Ways to achieve that

• A Special “Extension Header Label”
  • Use case is significant enough to deserve one
  • 8 unallocated so far (4-6 and 8-12)

• Two-label scheme: XL(15) + EHL
  • Still okay, but need one more label
  • No need to go this way, otherwise it’s tempting to play with the EHL encoding

• Dual FEC labels to indicate the existence of EH
  • Avoid the trouble to introduce a new special label
  • Complicate the control plane

• We prefer the option #1
Packet Format

- EHL = TBD
- Header Length
- Next Hdr = h2
- Header Length
- Next Header = h1
- MPLS Label Stack
- Extension Header Label
- Header of extension headers
- N extension headers
- Original inner packet
Some Details

• Extension Header Label (EHL) can be in any location in the label stack
  • For backward compatibility, it needs to be at BOS
  • For upgraded networks, it can be at any location in the stack
    • Preferred to be close at the top for performance reasons

• Next Header values
  • The Next Header field in the last extension header can have two special values:
    • “NONE” – no any header and payload after this header
    • “UNKNOWN” – the header or payload type after this header is unknown
      • To be compatible with the original MPLS design

• How about load balancing?
  • Use ELI+EL and put the label pair anywhere in the label stack
  • Can quickly skip the extension headers and use the upper layer protocol for LB
Conclusions

• MPLS is widely deployed
• MPLS has low overhead
• But it is difficult for MPLS to support in-network services
• A flexible and extensible solution - MPLS extension header
  • We believe the use cases are strong enough to deserve a special label for MPLS extension headers
  • We are developing new applications and use cases which can take advantage of the MPLS extension headers
• Feedback is welcome!