

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 9, 2022

L. Andersson
Bronze Dragon Consulting
J. Guichard
H. Song
Futurewei Technologies
S. Bryant
University of Surrey
October 6, 2021

MPLS Label Operations in MPLS EH capable networks
draft-andersson-mpls-eh-label-stack-operations-02

Abstract

Extension Headers (EH) carry information on in-network services and functions in an MPLS network. This document describes the operations on the MPLS label stack when an EH is found in the packet.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 9, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

Internet-Draft

EH Label Stack Operations

October 2021

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Requirement Language	3
2.	Operations on an MPLS Label Stack in an EH capable network .	3
2.1.	Physical Topology	3
2.2.	A day in the life of a packet	5
2.2.1.	Non-VPN Case	6
2.2.1.1.	Non-VPN with the EH in the packet	6
2.2.1.2.	Non-VPN without an EH in the packet	7
2.3.	The VPN case	8
2.3.1.	VPN with EH in the packet	8
2.3.2.	VPN without EH in the packet	9
2.4.	RSVP-TE Tunnel case	10
2.4.1.	RSVP Tunnel and EH present in the packet	11
2.4.2.	RSVP Tunnel and no EH present in the packet	12
2.4.3.	EH capable RSVP-TE tunnel	13
3.	Security Considerations	13
4.	IANA Considerations	13
5.	Acknowledgements	13
6.	References	14
6.1.	Normative References	14
6.2.	Informative References	14
	Authors' Addresses	14

[1.](#) Introduction

This document provides the operating procedures for EH-capable and non-EH-capable LSRs where MPLS Extension Headers (EH) are carried below the MPLS label stack. Further we show that MPLS EHs can be gradually introduced into an existing MPLS network. The capability to handle EHs is announced throughout the MPLS network, and LSRs that don't understand this information simply ignore it.

The extension headers are carried after the MPLS Label Stack, and the presence of EHs are indicate in the label stack by a Extetended Spewcial Purpose label called Extention Header Indicator (EHI) in the label stack.

Extension headers may for example be used when it is required that

the packet carry some metadata, more details will be found in [[I-D.song-mpls-extension-header](#)]. Examples of such cases are In-situ OAM, Network Telemetry and Measurement and Network Security.

Only EH capable LSRs will process EHs, LSRs that are EH non-capable will ignore the EH and forward the packet as if the information was not there.

1.1. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

2. Operations on an MPLS Label Stack in an EH capable network

This document provides a set of examples to show the operations performed on MPLS encapsulated packets in a network where MPLS EHs are used. The document does also illustrated the procedures for processing of the information carried within the MPLS label stack to indicate the presence of EHs below the label stack. For the purpose of illustration, we will assume that the indicator used to point to EHs is a G-ACh Generic Associated Channel Label (GAL) [[RFC5586](#)] + G-Ach Associated Channel Header (ACH) [[RFC5586](#)] with a set of new ACH types to indicate the EH types carried below the MPLS label stack.

As discussed in [[I-D.andersson-mpls-eh-architecture](#)], [[I-D.song-mpls-extension-header](#)] and [[I-D.song-mpls-eh-indicator](#)] there are alternatives to the use of GAL as the indicator; for example an Extension Label (XL) [[RFC7274](#)] + one or more Extended Special Purpose Labels (eSPLs) [[RFC7274](#)] could also be used.

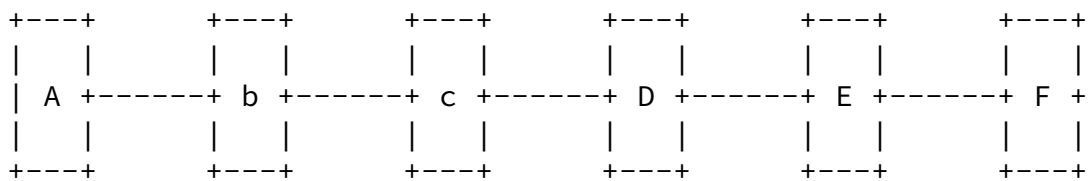
2.1. Physical Topology

Assume a physical topology that includes both EH capable LSRs and non-EH capable LSRs. The topology is intentionall kept quite simple.

Internet-Draft

EH Label Stack Operations

October 2021



Legend:

A, D, E, and F are EH capable LSRs

b and c are non-EH capable LSRs.

Figure 1: EH topology

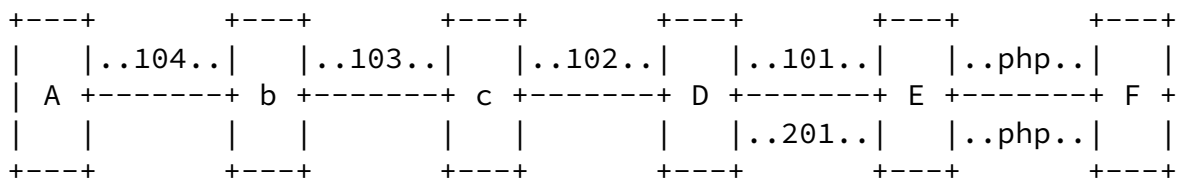
LDP Downstream on Demand (DoD) or Downstream Unsolicited (DU), RSVP-TE, an IGP or a centralized controller could be used to create the label mappings between the LSRs in an EH capable network. Referring to Figure 1, and using LDP DU for illustration, creation of an EH path used by A to send MPLS encapsulated packets with MPLS EHs to F is as show below.

For prefix F reachable at LSR F:

- o F advertises labels F:[ldp: implicit-null, EH-FEC: implicit-null] to E

- o E advertises labels F:[ldp: 101, EH-FEC: 201] to D
- o D advertises label F:[ldp: 102] to c
- o c advertises label F:[ldp: 103] to b
- o b advertises label F:[ldp: 104] to A

This will result in installed labels as shown in Figure 2.



Legend:

A, D, E and F are EH capable nodes.

b and are non-EH capable nodes.

Figure 2: EH topology

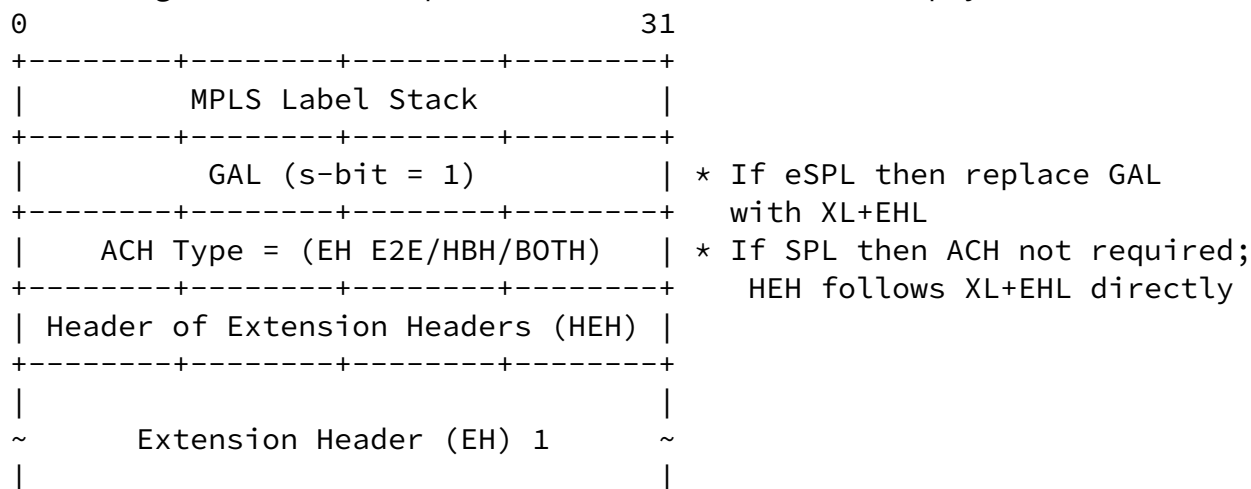
2.2. A day in the life of a packet

This section provides examples of forwarding for some common scenarios in networks with a mix of EH-capable and non-EH-capable LSRs and packets with and without EHs following the MPLS label stack.

All the information processed in the examples below is not strictly a

part of the "label stack"; ACH, EHL, HEH, EH and Payload are carried after the last entry in the label stack.

For reference the following shows the full MPLS EH stack, i.e. including also the EH specific information and the payload.



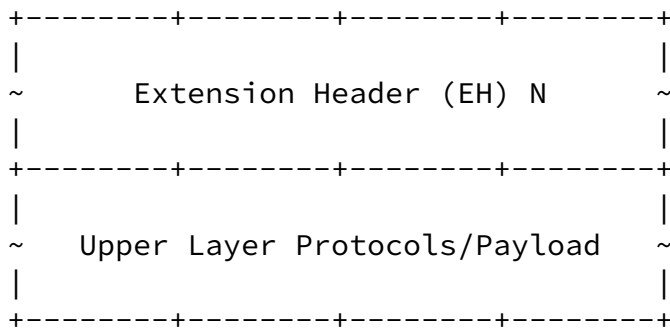


Figure 3: MPLS Extension Header (EH) Stack

[2.2.1.](#) Non-VPN Case

For non-VPN there are two variants, either the EH is present or it is not.

[2.2.1.1.](#) Non-VPN with the EH in the packet

- o A sends packet to b
 - * stack = [104, GAL, ACH, HEH, EH, IP]
- o b is a legacy router so just swaps [104] to [103], and sends the packet to c
 - * stack = [103, GAL, ACH, HEH, EH, IP]
- o c is a legacy router so just swaps [103] to [102], and sends the packet to D

- * stack = [102, GAL, ACH, HEH, EH, IP]
- o D is an EH capable LSR and receives the packet with [102] on top of the stack; D scans the packet for an EH; D finds EH and processes and then swaps the top label to [101] and then sends the packet on to E
 - i Note: this goes on the standard FEC because we only announce

in the packet there is NO EH. In this case EH is present.

- * stack = [101, GAL, ACH, HEH, EH, IP]
- o E receives [101] and scans the packet for EH; finds EH and processes and then pops the top label and send the packet to F
- * stack = [GAL, ACH, HEH, EH, IP]
 - + Note: E is the penultimate hop router so it pops the standard LDP label, and send on the standard FEC to F.
- o F receives the packet and scans the packet for EH; finds EH and processes it. As F is the ultimate hop it pops GAL, and removes ACH, HEH and EH, processes IP and forwards the packet.

2.2.1.2. Non-VPN without an EH in the packet

In this example there is no EH present in the packet.

- o A sends packet to b
 - * stack = [104, IP]
- o b receives the packet, b is a legacy router so it just swaps [104] to [103] and sends the packet to c
 - * stack = [103, IP]
- o c receives the packet, c is a legacy router so it just swaps [103] to [102], and sends the packet to D
 - * stack = [102, IP]
- o D receives the packet, D is an EH capable router, D searches the packet for an EH but finds no EH, D swaps [102] to [201], and sends the packet to E
 - * stack = [201, IP]

- + Note: in this case D sends the packet using the EH-FEC as EH

is **not** present.

- + Note: If downstream is not EH capable then D sends the packet on the standard FEC.
- o E receives the packet [201] and bypasses EH processing (received on the "no EH present" FEC; E is penultimate node so it pops EH-FEC label; and send the packet to F.
 - * stack = [IP]; not exactly a "label stack", but listed here for symmetry
- o F receives [IP] and routes the packet

[2.3.](#) The VPN case

In these two examples there is VPN information in the label stack, in the first there also EHs in the packet.

[2.3.1.](#) VPN with EH in the packet

- o A sends packet to b
 - * stack = [104, VPN, GAL, ACH, HEH, EH, IP]
- o b receives the packet; b is a legacy router and just swaps [104] to [103] and sends the packet to c
 - * stack = [103, VPN, GAL, ACH, HEH, EH, IP]
- o c receives the packet; c is a legacy router and just swaps [103] to [102] and sends the packet to D
 - * stack = [102, VPN, GAL, ACH, HEH, EH, IP]
- o D receives the packet; D is EH capable LSR; D will search the packet for EH and will find and process the EH; D will then swap [102] to [101] and sends the packet to E
 - * stack = [101, VPN, GAL, ACH, HEH, EH, IP]
 - + Note: This packet will be sent normal IP standard FEC; only packets that does not include an EH will be sent on the "no EH present" FEC.

- o E receives the packet; E is EH capable LSR; E will search the packet for EH and will find and process the EH; E will then pop [101] and sends the packet to F
 - * stack = VPN, GAL, ACH, HEH, EH, IP]
 - + Note: E is penultimate hop so pops the LDP label and send the packet on normal IP standard FEC; only packets that does not include an EH will be sent on the "no EH present" FEC.
- o F receives and scans the packet for EH; finds EH and processes it. As F is the ultimate hop it pops the GAL, and removes ACH, HEH and EH, processes the VPN label and forwards the packet.

2.3.2. VPN without EH in the packet

- o A sends packet to b
 - * stack = [104, VPN, IP]
- o b receives the packet; b is a legacy router and just swaps [104] to [103] and sends the packet to c
 - * stack = [103, VPN, IP]
- o c receives the packet; c is a legacy router and just swaps [103] to [102] and sends the packet to D
 - * stack = [102, VPN, IP]
- o D receives the packet; D is EH capable LSR; D will search the packet for EH; D will not find an EH; D will then swap [102] to [201] and sends the packet to E on the "no EH present" FEC. Lo
 - * stack = [101, VPN, IP]
 - + Note: This packet will be sent on the "no EH present" FEC;
- o E receives the packet [201] and bypasses EH processing (received on the "no EH present" FEC; E is the penultimate node so it pops EH- FEC label; and send the packet to F on the "no EH present" FEC.
 - * stack = [VPN, IP]
 - + Note: E is penultimate hop so E pops the "no FEC present"

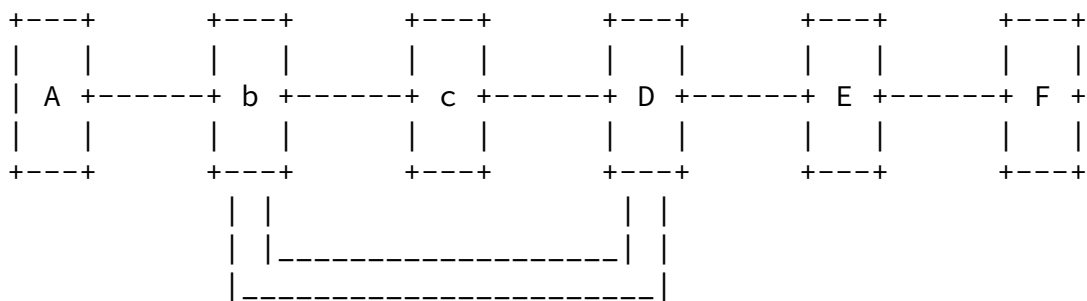
label and send the packet to F.

- o F receives and scans the packet for EH; finds no EH and bypasses EH processing. As F is the ultimate hop it processes the VPN label and forwards the packet.

2.4. RSVP-TE Tunnel case

The RSVP-TE tunnel is not EH capable or the capability has been disabled.

Assume a physical topology that includes both EH capable LSRs and non-EH capable LSRs, as in the earlier examples. This topology also includes a low cost RSVP-TE tunnel between b and D.



Legend:

A, D, E, and F are EH capable LSRs

b and c are non-EH capable LSRs.

Nodes that transport the RSVP-TE tunnel are not EH capable, or the EH capability is disabled.

Figure 4: EH topology

For this example the following assumptions are made:

- o An RSVP-TE tunnel has been established between b and D (packets

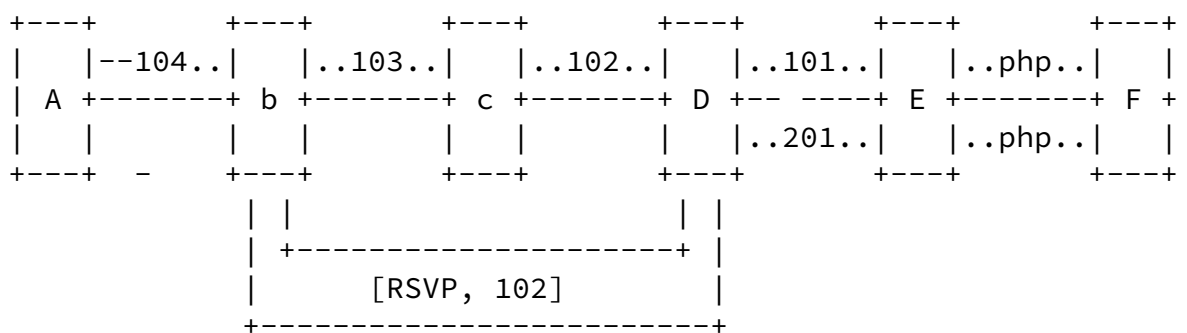
will bypass c)

- o F is reachable at b through RSVP-TE tunnel
- o LDP is enabled on the RSVP-TE tunnel

For prefix [F]: The following label mappings are sent by the LSRs in the network.

- o F advertises labels F: [ldp: implicit-null, EH-FEC: implicit-null] to E
- o E advertises labels F: [ldp: 101, EH-FEC: 201] to D
- o D advertises label F: [ldp: 102] to c and F:[ldp: 102] to b
- o c advertises label F: [ldp: 103] to b
- o b advertises label F: [ldp: 104] to A

This will result in label mappings like this.



Legend:

A, D, E, and F are EH capable LSRs

b and c are non-EH capable LSRs.

Nodes that transport the RSVP-TE tunnel are not EH capable, or the EH capability is disabled. [RSVP] represents the series of tunnel top

lables.

Figure 5: EH topology

To describe the label stack operations in this case the VPN label stack is used, starting with the case where an EH is present in the packet.

2.4.1. RSVP Tunnel and EH present in the packet

- o A sends packet to b

stack = [104, VPN, GAL, ACH, HEH, EH, IP]

Andersson, et al.

Expires April 9, 2022

[Page 11]

Internet-Draft

EH Label Stack Operations

October 2021

- o b receives the packet, since b is a legacy router it swaps [104] to [102], the next-hop reachable through the RSVP-TE tunnel; push the ingress RSVP-TE tunnel label and send it via the tunnel to the tunnel endpoint D

stack = [RSVP, 102, VPN, GAL, ACH, HEH, EH, IP]

- o Intermediate tunnel LSRs will forward (swap) based on the RSVP-TE label.
- o D receives the packet, D will pop the last RSVP-TE label; since D is a EH capable router it will search the stack and find the EH, after processing the EH it will swap [102] to [101], and send the packet to E over the normal FEC

stack = [101, VPN, GAL, ACH, HEH, EH, IP]

Note: this will be forwarded on the standard FEC because since the EH is present in the packet, only packet without an EH is forwarded on the "no EH present" FEC.

- o E receives the packet [101]; since E is a EH capable router it will search the stack and find the EH; after processing the EH it will pop [101], and send the packet to E over the normal FEC

stack = [VPN, GAL, ACH, HEH, EH, IP]

Note: As E is the penultimate hop it will pop the standard LDP label.

- o F receives the packet with the VPN label on top [VPN]; E scans the packet for EH; finds EH and processes. As F is the ultimate hop it pops GAL, and removes ACH, HEH and EH, processes VPN label and forwards the packet.

2.4.2. RSVP Tunnel and no EH present in the packet

- o A sends packet to b
 - * stack = [104, VPN, IP]
- o b receives the packet [104]; b is legacy router and will not search for an EH; b swaps [104] to [102]; pushes [RSVP] sends packet to D over the RSVP-TE tunnel.
 - * stack = [RSVP, 102, VPN, IP]

- o Intermediate tunnel LSRs will forward (swap) based on the RSVP-TE label.
- o D receives pops the tunnel label [RSVP], D is EH capable and scans the packet for EH; D finds no EH is present; pops RSVP-TE label, and then swaps LDP label [102]to [201] and sends the packet to E
 - * stack = [201, VPN, IP]
 - + Note: in this case D sends the packet using the "no EH present" FEC, since there is no EH in the packet.
 - + Note: If the downstream LSR is not EH capable then D will send the packet on the standard FEC.
- o E receives [201] and bypasses EH processing since the packet is received on the "no EH present" FEC; E is the pen-ultimate hop so it EH-FEC label and forward the packet to F

* stack = [VPN, IP]

- o F receives the packet [VPN]; and scans the packet for EH; does not find EH, processes VPN label and forwards the packet.

[2.4.3.](#) EH capable RSVP-TE tunnel

The case where an RSVP-TE tunnel is both EH capable and EH enabled is for further study.

[3.](#) Security Considerations

TBA

[4.](#) IANA Considerations

There are no requests for IANA actions in this document.

Note to the RFC Editor - this section can be removed before publication.

[5.](#) Acknowledgements

TBA

-

[6.](#) References

[6.1.](#) Normative References

[I-D.andersson-mpls-eh-architecture]

Andersson, L., Guichard, J. N., Song, H., and S. Bryant, "MPLS Extension Header Architecture", [draft-andersson-mpls-eh-architecture-02](#) (work in progress), October 2021.

[I-D.song-mpls-eh-indicator]

Song, H., Li, Z., Zhou, T., and L. Andersson, "Options for

MPLS Extension Header Indicator", [draft-song-mpls-eh-indicator-03](#) (work in progress), July 2021.

[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., Andersson, L., and Z. Zhang, "MPLS Extension Header", [draft-song-mpls-extension-header-05](#) (work in progress), July 2021.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", [RFC 5586](#), DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.

[6.2](#). Informative References

[RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", [RFC 7274](#), DOI 10.17487/RFC7274, June 2014, <<https://www.rfc-editor.org/info/rfc7274>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu

James N Guichard
Futurewei Technologies

Email: james.n.guichard@futurewei.com

Haoyu Song
Futurewei Technologies

Email: haoyu.song@futurewei.com

Stewart Bryant
University of Surrey

Email: stewart.bryant@gmail.com