

MPLS
Internet-Draft
Intended status: Standards Track
Expires: 15 April 2026

A. Farrel
Old Dog Consulting
J. Dong
Huawei Technologies
12 October 2025

A Survey to Determine MPLS Label Stack Inspection Behavior
draft-farrel-mpls-labelstack-inspection-poll-00

Abstract

As part of the work on MPLS Network Actions (MNA) a debate arose concerning how existing MPLS implementations handle Special Purpose Labels (SPLs) especially in the context of processing the MPLS Entropy Label. The question applies to the relative placement of the Entropy Label Indicator (ELI) and the MNA Base SPL in the MPLS label stack. This question is applicable to the use of the ELI and any new base SPL (bSPL), or to the relative placement of any two bSPLs including the Extension Label that precedes any extended SPL.

In order to discover what deployed implementations currently do, it is proposed that the MPLS working group chairs poll participants to answer specific questions about their implementations. This document is a working draft of the questions to be asked.

It is not intended that this document should progress to publication as an RFC.

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 15 April 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Survey Process	4
3. Survey	4
4. Manageability Considerations	7
5. Security Considerations	7
6. IANA Considerations	7
7. Informative References	7
Authors' Addresses	8

1. Introduction

In networks with equal cost paths, such as parallel links, network nodes may load balance traffic across the possible paths. In order to avoid the risk of packets being reordered, all packets from the same flow need to be placed on the same path. Thus, the load balancing decision is made by examining some properties of the packets.

Historically, in IP networks, the decision was based on a hash of the source address, destination address, source port, destination port, and payload protocol Id (the 5-tuple). With the introduction of MPLS, this could still be done, but it might be hard for network nodes to read the 5-tuple, especially with deep label stacks. Therefore, some implementations chose to perform the hash on the first entries in the label stack.

However, in many cases, hashing the label stack does not provide enough discrimination to properly balance the traffic. To aid with this, the MPLS Entropy Label was introduced in [RFC6790]. The Entropy Label contains a value that can be used as the hash to discriminate the traffic flows.

The Entropy Label is indicated in the label stack by the preceding label stack entry which has a special label value, the Entropy Label Indicator (ELI). The ELI is a base Special Purpose Label (bSPL) [RFC9017] so that it will not be accidentally used for forwarding.

Implementations at transit nodes that support the Entropy Label may parse the label stack on a received packet to find the ELI and then read the Entropy Label in the next label stack entry.

Implementations at transit nodes that do not support the Entropy Label can still hash the first entries in the label stack. The ELI and Entropy Label may safely be included in the hash function.

In-stack MPLS Network Actions (MNA) are instructions with optional ancillary data encoded in the label stack as a series of label stack entries, the Network Action Sub-Stack (NAS) [I-D.ietf-mpls-mna-hdr]. The NAS is indicated in the label stack by its first label stack entry containing a bSPL, the MNA-Label.

During the development of the MNA specification, a question was raised about how a legacy node (one that does not support MNA) might react if it was searching for the ELI and encountered the MNA-Label. This was investigated by surveying existing implementations, and the results were published in [I-D.farrel-mpls-forwarder-poll-response]. Additionally, the NAS has been designed such that no label stack entry can be mistaken for a bSPL (specifically, each entry in the NAS begins with seven bits which cannot all be zero).

Further discussion around the interaction of the ELI and the MNA-Label, opened the wider question of how implementations performing ECMP react if they discover an unrecognized bSPL either while hashing the label stack or while searching for the ELI. And beyond that, how an implementation searching the label stack for any specific bSPL including the Extension Label with a specific extended SPL (eSPL) reacts if it encounters an unrecognized bSPL or the Extension Label with an unrecognized eSPL.

This discussion should be held in the context of Section 2.1.1 of [RFC7325] in particular the paragraph that says:

Unknown special-purpose labels and unknown extended special-purpose labels are handled the same. When an unknown special-purpose label is encountered or a special purpose label not directly handled in forwarding hardware is encountered, the packet should be sent to a general-purpose CPU by default. If this capability is supported, there must be an option to either drop or rate limit such packets based on the value of each special-purpose label.

There has been debate about whether the word "encountered" in this text is meant to refer indicate "found at the top of the label stack - possibly after another label has been popped" or whether it extends to parsing a label stack in search of a specific SPL. The survey in this document is also intended to determine current behaviors in this regard.

This document develops a survey with the intention that it be used to determine whether there are any challenges that need to be addressed.

NOTE WELL

At this stage, this is a proposal for the content of the survey. It is produced for discussion and review. Answers should not be submitted for collection.

2. Survey Process

Implementors are invited to complete the survey for each implementation they are aware of.

Responses should be submitted to survey@olddog.co.uk with the subject line "MPLS Label Stack Inspection Survey Response."

Responses will be collated and anonymized, and then posted in an Internet-Draft. If further work is required, it will be brought to the MPLS Working Group for attention.

3. Survey

This is a draft and should not be completed.

Please identify the implementations that this response applies to.

1. Does your implementation support MPLS ECMP?

If so:

- A. What does the implementation do if it encounters a bSPL that it does not recognize at the top of the label stack, or if the top label is the Extension Label and the subsequent label value is an eSPL that it does not recognize?
 - i. Pass the packet to the CPU for processing (possibly followed by one of the following actions)
 - ii. Drop the packet

- iii. Strip the unknown bSPL or the Extension Label and subsequent unknown eSPL and continue processing the packet
 - iv. Some other behavior - please supply a description
- B. Does the implementation search for the bottom of stack to hash on the payload 5-tuple?
- If so, what does the implementation do if it encounters a bSPL it does not recognize in the label stack?
- i. Drop the packet
 - ii. Stop attempting to find the bottom of stack, but continue forwarding the packet
 - iii. Continue searching for the bottom of stack
 - iv. Some other behavior - please supply a description
- C. Does the implementation perform a hash on the top entries in the label stack?
- If so, what does the implementation do if it encounters a bSPL it does not recognize in the label stack?
- i. Drop the packet
 - ii. Stop reading further label stack entries, but hash on the previous label stack entries
 - iii. Stop attempting to perform a hash, but continue forwarding the packet
 - iv. Continue hashing the label stack skipping the unrecognized label
 - v. Continue hashing the label stack including the unrecognized label
 - vi. Some other behavior - please supply a description
- D. Does the implementation support inspecting the label stack to find the ELI and use the subsequent Entropy Label as input to hashing to determine the forwarding next hop?
- If so:

- i. What does the implementation do if it does not find the ELI before it reaches bottom of stack?
 - a. Drop the packet
 - b. Perform some other form of hash
 - c. Forward the packet without load balancing
 - d. Some other behavior - please supply a description
 - ii. What does the implementation do if it does not find the ELI before it reaches the maximum depth it can read into the label stack?
 - a. Drop the packet
 - b. Perform some other form of hash
 - c. Forward the packet without load balancing
 - d. Some other behavior - please supply a description
 - iii. What does the implementation do if it encounters a bSPL (or Extended Label followed by an eSPL) in the label stack it does not recognize before it finds the ELI?
 - a. Drop the packet
 - b. Safely step over the bSPL and continue searching for the ELI
 - c. Some other behavior - please supply a description
2. Does your implementation parse the label stack for any other purpose (for example, to find other specific SPLs or to find the bottom of stack)?

If so, what does the implementation do if it encounters an unrecognized bSPL or the Extended Label followed by an unrecognized eSPL?

- A. Drop the packet
- B. Stop searching, but continue to forward the packet
- C. Step over the label stack entry containing the unrecognized bSPL and continue to search

D. Some other behavior - please supply a description

3. If your implementation parses the label stack in search of one or more bSPLs/eSPLs, what does it do when it has found the label stack entries it was searching for?
 - A. Continue parsing the label until bottom of stack or maximum readable stack depth are reached
 - B. Continue looking through the label stack for additional occurrences of the SPLs where those SPLs are allowed to be present multiple times
 - C. Stop looking through the label stack once one of each type of SPL has been found
4. Is your implementation sensitive to the ordering of SPLs (bSPLs and/or eSPLs) in the label stack?

If so, please elaborate.

4. Manageability Considerations

This document contains no issues for manageability. However, the responses to the survey may uncover additional manageability work that needs to be done. For example, configuration, dynamic exposure of functional behavior, or diagnostics.

5. Security Considerations

Understanding of behaviors around searching the MPLS label stack are important for a stable and secure network.

6. IANA Considerations

This document makes no requests for IANA action.

7. Informative References

[I-D.farrel-mpls-forwarder-poll-response]
Farrel, A., "Anonymized Responses to a Poll on MPLS Forwarder Behavior", Work in Progress, Internet-Draft, draft-farrel-mpls-forwarder-poll-response-01, 6 November 2022, <<https://datatracker.ietf.org/doc/html/draft-farrel-mpls-forwarder-poll-response-01>>.

[I-D.ietf-mpls-mna-hdr]

Rajamanickam, J., Gandhi, R., Zigler, R., Song, H., and K. Kompella, "MPLS Network Action (MNA) Sub-Stack Solution", Work in Progress, Internet-Draft, draft-ietf-mpls-mna-hdr-16, 3 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-mpls-mna-hdr-16>>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.

[RFC7325] Villamizar, C., Ed., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, DOI 10.17487/RFC7325, August 2014, <<https://www.rfc-editor.org/info/rfc7325>>.

[RFC9017] Andersson, L., Kompella, K., and A. Farrel, "Special-Purpose Label Terminology", RFC 9017, DOI 10.17487/RFC9017, April 2021, <<https://www.rfc-editor.org/info/rfc9017>>.

Authors' Addresses

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

Jie Dong
Huawei Technologies
China
Email: jie.dong@huawei.com