Stream: Internet Engineering Task Force (IETF)
RFC: 9790
Updates: 4928
Category: Standards Track
Published: July 2025
ISSN: 2070-1721
Authors:
K. Kompella, *Juniper Networks*
S. Bryant, *University of Surrey 5GIC*
M. Bocci, *Nokia*
G. Mirsky, Ed., *Ericsson*
L. Andersson, *Huawei Technologies*
J. Dong, *Huawei Technologies*

RFC 9790
IANA Registry and Processing Recommendations for the First Nibble Following a Label Stack

Abstract

This document creates a new IANA registry (called the "Post-Stack First Nibble" registry) for the first nibble (4-bit field) immediately following an MPLS label stack. Furthermore, this document presents some requirements for registering new values and making the processing of MPLS packets easier and more robust.

The relationship between the IANA "Post-Stack First Nibble" registry and the "IP Version Numbers" registry (RFC 2780) is described in this document.

This document updates RFC 4928 by deprecating the heuristic method for identifying the type of packet encapsulated in MPLS.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9790.

Copyright Notice

Table of Contents

1. Introduction

An MPLS packet consists of a label stack, an optional Post-Stack Header (PSH), and an optional embedded packet (in that order). Examples of PSHs include existing headers such as control words [RFC4385], BIER (Bit Index Explicit Replication) headers [RFC8296] and the like, as well as new types of PSH being discussed by the MPLS Working Group. However, in the data plane, there are very few clues regarding the PSH and no clue as to the type of embedded packet; this information is communicated via other means, such as the routing protocols that signal the labels in the stack. Nonetheless, in order to better handle an MPLS packet in the data plane, it is common practice for network equipment to "guess" the type of embedded packet. Such equipment may also need to process the PSH. Both of these require parsing the data after the label stack. To do this, the "first nibble" (the top four bits of the first octet following the label stack) is often used. Although some existing network devices may use such a method, it needs to be stressed that the correct interpretation of the Post-stack First Nibble (PFN) in a PSH can be made only in the context established through the control or management plane with the Label Stack Entry (LSE) or group of LSEs in the preceding label stack that characterizes the type of the PSH. Any attempt to rely on the value in any other context is unreliable. Because the PFN value should not be used to deduce the type of PSH by itself and the space of PFN values is limited, the reuse of PFN values is encouraged when possible.

The semantics and usage of the first nibble are not well documented, nor are the assignments of values. This document serves four purposes:

- To document the values already in use.
- To provide a mechanism to document future assignments through the creation of a new IANA "Post-Stack First Nibble" registry and describe the relationship between it and the IANA "IP Version Numbers" registry [RFC2780].
- Provide a method for tracking usage by requiring more detailed documentation.
- To stress that any MPLS packet not carrying plain IPv4 or IPv6 packets contains a PSH. This also applies to packets of any new version of IP (see Section 2.4).

Section 2.1.1 of this document includes an analysis of load-balancing techniques; based on this, Section 2.1.1.1 introduces a requirement that deprecates the use of the heuristic method for identifying the type of packet encapsulated in MPLS and recommends using a dedicated label value for load balancing. The intent is for legacy routers to continue operating as they have, with no new problems introduced as a result of this document. However, new implementations that follow this document enable more robust network operation.

Furthermore, this document updates [RFC4928] by deprecating the heuristic method for identifying the type of packet encapsulated in MPLS. This document clearly states that the type of encapsulated packet cannot be determined based on the PFN alone.

1.1. Requirements 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.

1.2. Definitions

Deprecation: Regardless of how the deprecation is understood in other IETF documents, the interpretation in this document is that if a practice has been deprecated, that practice should not be included in new implementations or deployed in new deployments.

Embedded Packet: A packet that follows immediately after the MPLS label stack and an optional PSH. The embedded packet could be an IPv4 or IPv6 packet, an Ethernet packet (for Virtual Private LAN Service (VPLS) [RFC4761] [RFC4762] or EVPN [RFC7432]), or some other type of Layer 2 frame [RFC4446].

Label Stack: A label stack is represented as a consecutive sequence of "label stack entries" (four-octet fields) after the Layer 2 header but before any network layer header. The last label stack entry of a label stack has its Bottom of Stack bit set.

MPLS Packet: A packet whose Layer 2 header declares the type to be MPLS. For example, the Ethertype is 0x8847 or 0x8848 for Ethernet, and the Protocol field is 0x0281 or 0x0283 for PPP.

MPLS Payload: All data after the label stack and any optional PSHs. It is possible that more than one type of PSH may be present in a packet, and some PSH specifications might allow multiple PSHs of the same type. The presence rules for multiple PSHs are a matter for the documents that define those PSHs, e.g., [MNA-PS-HDR].

Post-stack First Nibble (PFN): The most significant four bits of the first octet following the label stack.

Post-Stack Header (PSH): A field containing information that may be of interest to the egress Label Switching Router (LSR) or transit LSRs. Examples include a control word [RFC4385] [RFC8964] or an associated channel header [RFC4385] [RFC5586] [RFC9546].

1.3. Abbreviations

BIER: Bit Index Explicit Replication

FAT: Flow-Aware Transport

LSE: Label Stack Entry

LSR: Label Switching Router

MNA: MPLS Network Action

PFN: Post-stack First Nibble

PSH: Post-Stack Header

PW: Pseudowire

SPL: Special-Purpose Label

1.4. Reference Figures

Figure 1 echoes the format of MPLS packets as defined in [RFC3032] where TC indicates the Traffic Class field [RFC5462] that replaced the EXP (Experimental Use) field, S is the Bottom of Stack flag, and TTL is the Time to Live field.

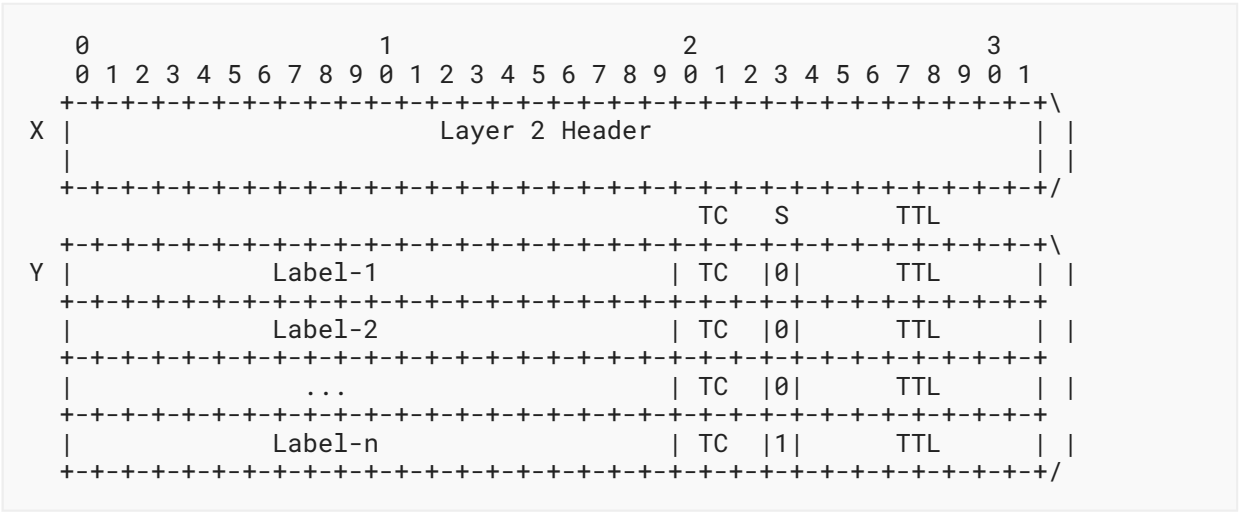

Figure 1: Example of an MPLS Packet with Label Stack

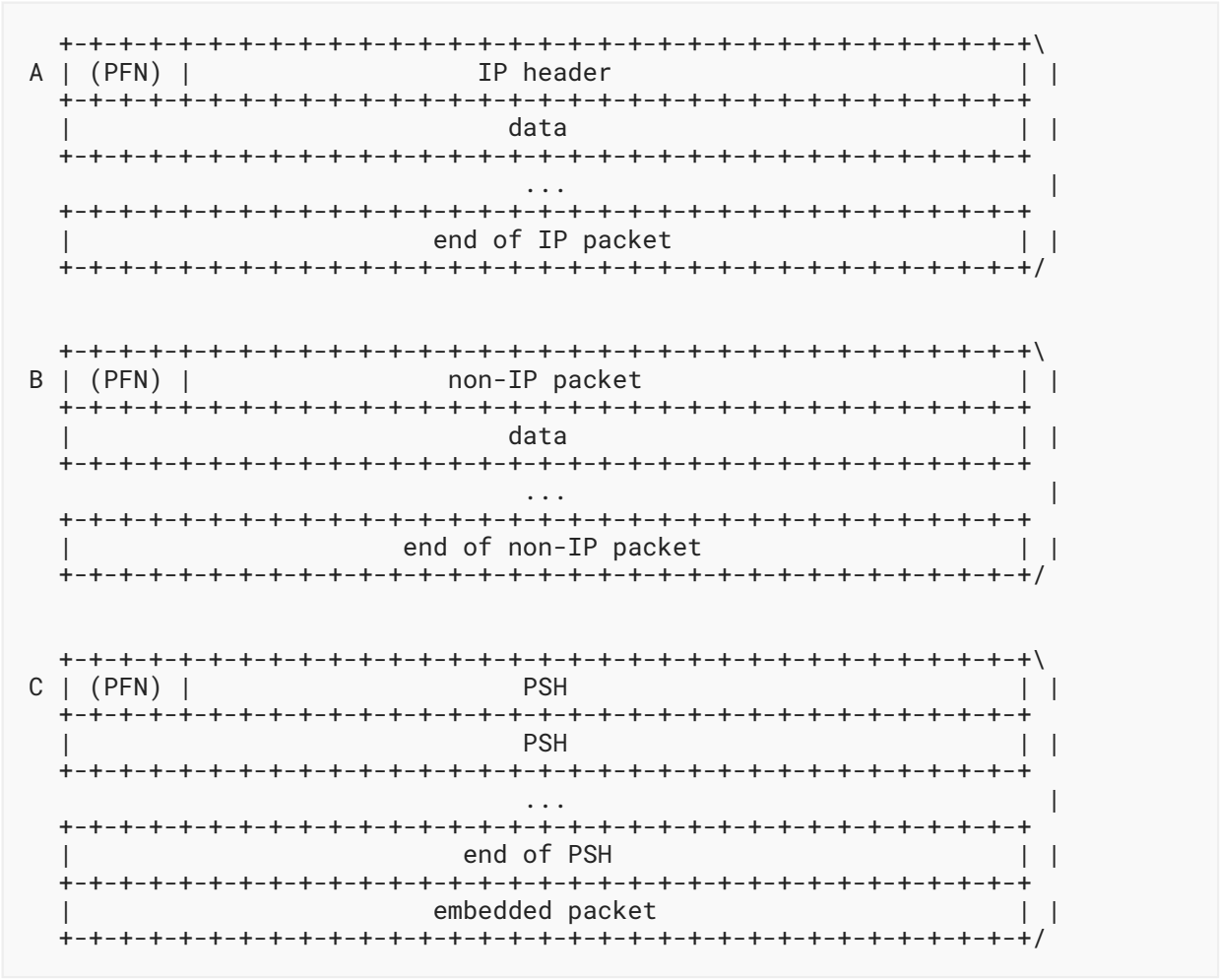

Figure 2: Examples of an MPLS Packet Payload With and Without a Preceding Post-Stack Header

Figure 1 shows an MPLS packet with a Layer 2 header X and a label stack Y ending with Label-n. Figure 2 displays three examples of an MPLS payload:

Example A: The first payload is a bare IP packet, i.e., no PSH. The PFN in this case overlaps with the IP version number.

Example B: The next payload is a bare non-IP packet; again, no PSH. The PFN here is the first nibble of the payload, whatever it happens to be.

Example C: This example is an MPLS Payload that follows a PSH. Here, the embedded packet could be IP or non-IP.

Thus, the complete MPLS packet would consist of [X Y A], [X Y B], or [X Y C].

2. Rationale

2.1. Why Look at the First Nibble

An MPLS packet can contain one of many types of embedded packets. Three common types are:

1. An IPv4 packet (whose IP header has version number 4).
2. An IPv6 packet (whose IP header has version number 6).
3. A Layer 2 Ethernet frame (i.e., not including the Preamble or the Start frame delimiter), starting with the destination Media Access Control (MAC) address.

Many other packet types are possible; in principle, any Layer 2 embedded packet is permissible. Indeed, at some points in time, packets of the Point-to-Point Protocol, Frame Relay, and Asynchronous Transfer Mode were reasonably common and may become so again.

In addition, there may be a PSH ahead of the embedded packet. The value of PFN is considered to ensure that the PSH can be correctly parsed.

2.1.1. ECMP Load Balancing

There are four common ways to load balance an MPLS packet:

1. Use the top label alone.
2. Use all of the non-SPLs [RFC7274] in the stack. This is better than using the top label alone.
3. "Divine" the type of embedded packet and use fields from the guessed header. The ramifications of using this load-balancing technique are discussed in detail in Section 2.1.1.1. This way is better than the two ways above.
4. Use either an Entropy Label [RFC6790] or a Flow-Aware Transport (FAT) Pseudowire Label [RFC6391] (see Section 2.1.1.1). This is the best way.

Load balancing based on just the top label means that all packets with that top label will go the same way, which is far from ideal. Load balancing based on the entire label stack (not including SPLs) is better, but it may still be uneven. However, if the embedded packet is an IP packet, then the combination of (<source IP address>, <dest IP address>, <transport protocol>, <source port>, and <dest port>) from the IP header of the embedded packet forms an excellent basis for load balancing. This is what is typically used for load balancing IP packets.

An MPLS packet doesn't, however, carry a payload type identifier. There is a simple (but risky) heuristic that is commonly used to guess the type of the embedded packet. The first nibble of an IP header, i.e., the four most significant bits of the first octet, contains the IP version number. That, in turn, indicates where to find the relevant fields for load balancing. The heuristic goes roughly as described in Section 2.1.1.1.

2.1.1.1. Heuristic for ECMP Load Balancing

1. If the PFN is 0x4 (0100b), treat the payload as an IPv4 packet, and find the relevant fields for load balancing on that basis.
2. If the PFN is 0x6 (0110b), treat the payload as an IPv6 packet, and find the relevant fields for load balancing on that basis.
3. If the PFN is anything else, the MPLS payload is not an IP packet; fall back to load balancing using the label stack.

This heuristic has been implemented in many (legacy) routers and performs well in the case of example A in Figure 2. However, this heuristic can work very badly for the non-IP packet as shown in example B in Figure 2. For example, if payload B is an Ethernet frame, then the PFN is the first nibble of the Organizationally Unique Identifier of the destination MAC address, which can be 0x4 or 0x6. This would lead to the packet being treated as an IPv4 or IPv6 packet such that data at the offsets of specific relevant fields would be used as input to the load-balancing heuristic, resulting in unpredictable load balancing. This behavior can happen to other types of non-IP payloads as well.

That, in turn, led to the idea of inserting a PSH (e.g., a pseudowire control word [RFC4385], a Deterministic Networking (DetNet) control word [RFC8964], a Network Service Header (NSH) [RFC8300], or a BIER header [RFC8296]) where the PFN is not 0x4 or 0x6; this explicitly prevents forwarding engines from confusing the MPLS payload with an IP packet. [RFC8469] recommends the use of a control word when the embedded packet is an Ethernet frame. [RFC8469] was published at the request of the operator community and the IEEE Registration Authority Committee as a result of operational difficulties with pseudowires that did not contain the control word.

Where load balancing of MPLS packets is desired, it is **RECOMMENDED** that the load-balancing mechanism use the value of a dedicated label, for example, either an Entropy Label [RFC6790] or a FAT Pseudowire Label [RFC6391]. Furthermore, the heuristic of guessing the type of the embedded packet, as discussed above, **SHOULD NOT** be used.

A consequence of the heuristic approach is that while legacy routers may look for a PFN of 0x4 [RFC0791] or 0x6 [RFC8200], no legacy router will look for any other PFN for load-balancing purposes, regardless of what future IP version numbers will be. This means that the values 0x4 and 0x6 are used to (sometimes incorrectly) identify IPv4 and IPv6 packets, but no other PFN values will be used to identify IP packets.

This document creates a new registry for all 16 possible values (see Section 3).

2.2. Updates to RFC 4928

The text in RFC 4928 [RFC4928] concerning the first nibble after the MPLS label stack has been updated by this document, and the heuristic for snooping this nibble has been deprecated. Section 3 of [RFC4928] is updated as follows:

OLD TEXT:

> It is **REQUIRED**, however, that applications depend upon in-order packet delivery restrict the first nibble values to 0x0 and 0x1. This will ensure that their traffic flows will not be affected if some future routing equipment does similar snooping on some future version(s) of IP.

NEW TEXT:

> Network equipment **MUST** use a PSH (Post-Stack Header) with a PFN (Post-stack First Nibble) value that is neither 0x4 nor 0x6 in all cases where the MPLS payload is neither an IPv6 nor an IPv4 packet.

The following requirement (discussed is Section 2.1.1.1) replaces paragraph 4 in Section 3 of [RFC4928] as follows:

OLD TEXT:

> This behavior implies that if in the future an IP version is defined with a version number of 0x0 or 0x1, then equipment complying with this BCP would be unable to look past one or more MPLS headers, and loadsplit traffic from a single LSP across multiple paths based on a hash of specific fields in the IPv0 or IPv1 headers. That is, IP traffic employing these version numbers would be safe from disturbances caused by inappropriate loadsplitting, but would also not be able to get the performance benefits.

NEW TEXT:

> The practice of deducing the payload type based on the PFN value is deprecated to avoid inaccurate load balancing. This **MUST NOT** be part of new implementations or deployments. This also means that concerns about load balancing for future IP versions with a version number of 0x0 or 0x1 are no longer relevant.

Furthermore, the following text is appended to Section 1.1 of [RFC4928]:

NEW TEXT:

> PSH: Post-Stack Header
>
> PFN: Post-stack First Nibble

2.3. Why Create a Registry

The framework for MPLS Network Actions (MNAs) is described in [RFC9789] and is an enhancement to the MPLS architecture. The use of Post-Stack Data (PSD) to encode the MNA indicators and ancillary data (described in Section 3.6 of [RFC9789]) might place data in the PFN, which could conflict with other uses of that nibble. This issue is described in Section 3.6.1 of [RFC9789] and is further illustrated by the PFN value of 0x0, which has two different formats depending on whether the PSH is a pseudowire control word or a DetNet control word; disambiguation requires the context of the service label.

With a registry, PSHs become easier to identify and parse. In addition, they do not need a means outside the data plane to interpret them correctly, and their semantics and usage are documented and referenced in the registry.

2.4. IP Version Numbers Versus Post-Stack First Nibble Values

The use of the PFN stemmed from the desire to heuristically identify IP packets for load-balancing purposes. It was then discovered that non-IP packets, misidentified as IP when the heuristic failed, were being badly load balanced, leading to the scenario described in [RFC4928]. This situation may confuse some as to the relationship between the "Post-Stack First Nibble" registry and the "IP Version Numbers" registry. These registries are quite different:

1. The explicit purpose of the "IP Version Numbers" registry is to track IP version numbers in an IP header.
2. The purpose of the "Post-Stack First Nibble" registry is to track PSH types.

The only intersection points between the two registries are the values 0x4 and 0x6 (for backward compatibility).

2.5. Next Step to More Deterministic Load Balancing in MPLS Networks

Network evolution is impossible to control, but it develops over a period of time determined by various factors.

This document discourages further proliferation of the implementations that could lead to undesired effects on data flows. In doing so, it limits the scope of future protocol developments and thus helps to ensure that future network evolution will be smoother.

Section 2 of [RFC4385] suggests the use of a PSH solely for the purpose of avoiding IP ECMP treatment of non-IP payloads encapsulated in MPLS. Obsoleting this use of a PSH would assist with the progress toward a simpler, more coherent system of MPLS data encapsulation. (Other uses of a PSH may still be valid.) However, before that can be done, it is important to collect sufficient evidence that there are no marketed or deployed implementations using the heuristic practice to load balance MPLS data flows.

Therefore, the next steps toward more deterministic load balancing in MPLS networks are to gradually deprecate non-PSH MPLS encapsulations of non-IP data, to cease using heuristic load balancing, and to survey the available and deployed implementations to determine when obsoletion may be achieved.

3. IANA Considerations

Per this document, IANA has created a registry group called "Post-Stack First Nibble" that consists of a single registry called the "Post-Stack First Nibble" registry. The initial contents of the registry are shown in Table 1. The assignment policy is Standards Action [RFC8126]. It is important to note that the same PFN value can be used in more than one protocol. The correct interpretation of the PFN in a PSH can be made only in the context of the LSE or group of LSEs in the preceding label stack that characterizes the type of the PSH and, consequently, the PFN.

Protocol	Value	Description	Reference
DetNet	0x0	DetNet Control Word	[RFC8964]
NSH	0x0	NSH Base Header, payload	[RFC8300]
PW	0x0	PW Control Word	[RFC4385]
DetNet	0x1	DetNet Associated Channel	[RFC9546]
MPLS	0x1	MPLS Generic Associated Channel	[RFC5586]
PW	0x1	PW Associated Channel	[RFC4385]
NSH	0x2	NSH Base Header, OAM	[RFC8300]
	0x3	Unassigned	
	0x4	Reserved	RFC 9790
BIER	0x5	BIER Header	[RFC8296]
	0x6	Reserved	RFC 9790
	0x7 - 0xF	Unassigned	

Table 1: Post-Stack First Nibble Registry

4. Security Considerations

This document creates a new IANA registry for PFNs and specifies changes to the treatment of packets in the data plane based on the first nibble of data beyond the MPLS label stack. One intent of this is to reduce or eliminate errors in determining whether or not a packet being transported by MPLS is IP. While such errors have primarily caused unbalanced, and thus inefficient, multipathing, they have the potential to cause more severe security problems.

For general security considerations involving the MPLS label stack, see [RFC3032].

5. References

5.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[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>.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, DOI 10.17487/RFC2780, March 2000, <https://www.rfc-editor.org/info/rfc2780>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <https://www.rfc-editor.org/info/rfc4385>.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, DOI 10.17487/RFC4928, June 2007, <https://www.rfc-editor.org/info/rfc4928>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <https://www.rfc-editor.org/info/rfc6391>.

[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>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[RFC8469] Bryant, S., Malis, A., and I. Bagdonas, "Recommendation to Use the Ethernet Control Word", RFC 8469, DOI 10.17487/RFC8469, November 2018, <https://www.rfc-editor.org/info/rfc8469>.

[RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <https://www.rfc-editor.org/info/rfc8964>.

[RFC9789] Andersson, L., Bryant, S., Bocci, M., and T. Li, "MPLS Network Actions (MNAs) Framework", RFC 9789, DOI 10.17487/RFC9789, July 2025, <https://www.rfc-editor.org/info/rfc9789>.

5.2. Informative References

[MNA-PS-HDR] Rajamanickam, J., Ed., Gandhi, R., Ed., Zigler, R., Li, T., and J. Dong, "Post-Stack MPLS Network Action (MNA) Solution", Work in Progress, Internet-Draft, draft-ietf-mpls-mna-ps-hdr-01, 30 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mpls-mna-ps-hdr-01>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <https://www.rfc-editor.org/info/rfc4446>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

[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>.

[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>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC9546] Mirsky, G., Chen, M., and B. Varga, "Operations, Administration, and Maintenance (OAM) for Deterministic Networking (DetNet) with the MPLS Data Plane", RFC 9546, DOI 10.17487/RFC9546, February 2024, <https://www.rfc-editor.org/info/rfc9546>.

Acknowledgements

The authors express their appreciation and gratitude to Donald E. Eastlake 3rd for the review, insightful questions, and helpful comments. Also, the authors are grateful to Amanda Baber for helping organize the IANA registry in a clear and concise manner.

Éric Vyncke, John Scudder, Warren Kumari, Murray Kucherawy, and Gunter Van de Velde provided helpful comments during IESG review.

The authors would also like to thank Adrian Farrel for his patient and careful shepherding and for helping to finalize the text.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: kireeti.ietf@gmail.com

Stewart Bryant
University of Surrey 5GIC
Email: sb@stewartbryant.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Greg Mirsky (EDITOR)
Ericsson
Email: gregimirsky@gmail.com

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: jie.dong@huawei.com