IANA Registry for the First Nibble Following a Label Stack

Abstract

The goal of this memo is to create a new IANA registry (called the
MPLS First Nibble registry) for the first nibble (4-bit field)
immediately following an MPLS label stack. The memo offers a
rationale for such a registry, describes how the registry should be
managed, and provides some initial entries. Furthermore, this memo
sets out some documentation requirements for registering new values.
Finally, it provides some recommendations that makes processing MPLS
packets easier and more robust.

There is an important caveat on the use of this registry versus the
IP version number registry.

This memo, if published, would update [RFC4928] and [RFC8469].

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 29 October 2022.

Copyright Notice

Table of Contents

1. Introduction

An MPLS packet consists of a label stack, an optional "post-stack
header" and an optional embedded packet (in that order). However, in
the data plane, there are scant clues regarding the post-stack
header, 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 post-stack
header. 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.

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

  *To document the assignments already made

*To provide for the clear documentation of future assignments
    through the creation of an "MPLS First Nibble registry"

   *Provide a method to tracking usage by requiring more consistent
    documentation

1.1. Conventions and Definitions

   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
   BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here.

   LSR: label switching router.

   MPLS packet: one whose Layer 2 header declares the type to be MPLS.
      For Ethernet, that means the Ethertype is 0x8847 or 0x8848.

   Label Stack: (of an MPLS packet) all labels (four octet fields)
      after the Layer 2 header, up to and including the label with the
      BoS bit set ([RFC3032]).

   MPLS First Nibble (MFN): the most significant four bits of the
      first octet following the label stack.

   MPLS Payload: all data after the label stack, including the MFN, an
      optional post-stack header and the embedded packet.

   Post-stack Header (PSH): optional field of interest to the egress
      LSR (and possibly to transit LSRs). Examples include a control
      word or an associated channel. The PSH MUST indicate its length,
      so that a parser knows where the embedded packet starts.

   Embedded Packet: All octets beyond the PSH (if any). This could be
      an IPv4 or IPv6 packet (e.g., for traffic engineering of IP
      packets, or for a Layer 3 VPN [RFC4364]), an Ethernet packet (for
      VPLS ([RFC4761], [RFC4762]) or EVPN [RFC7432]), or some other
      type of Layer 2 frame [RFC4446].

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
X |                     Layer 2 Header                           | |
  |                                                             | |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
                                         TC    S      TTL
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
Y |            Label-1                  | TC  |0|      TTL     | |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |            Label-2                  | TC  |0|      TTL     | |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |              ...                    | TC  |0|      TTL     | |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |            Label-n                  | TC  |1|      TTL     | |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
```

Figure 1: Example of an MPLS Packet With Label Stack

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
  A | (MFN) |                 IP packet                          | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           data                            | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           ...                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     end of IP packet                      | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/


     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
  B | (MFN) |               non-IP packet                        | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           data                            | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           ...                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                   end of non-IP packet                    | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/


     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
  C | (MFN) |                   PSH                              | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           PSH                             | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           ...                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                       end of PSH                          | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     embedded packet                       | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
```

             Figure 2: Three Examples of MPLS Payloads

   Figure 1 shows an MPLS packet with Layer 2 header X and a label
   stack Y ending with Label-n. Figure 2 shows three examples of an
   MPLS payload, A, B and C. The full MPLS packets thus are: [X, Y, A],
   [X, Y, B], and [X, Y, C].

   A. The first payload is a bare IP packet, i.e., no PSH. The MFN
   (MPLS First Nibble) in this case overlaps with the IP version
   number.

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

C. The last example is an MPLS Payload that starts with a PSH
followed by the embedded packet.

2. Rationale

2.1. Why Look at the First Nibble

An MPLS packet can contain many types of embedded packet. The most
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 MAC
      address.

Many other packet types are possible, and in principle, any Layer 2
embedded packet is permissible; indeed, in the past, PPP, Frame
Relay and ATM packets were reasonably common.

In addition, there may be a post-stack header ahead of the embedded
packet, and this needs be to parsed. The MPLS First Nibble is
currently used for both of these purposes.

2.1.1. Load Balancing

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

   1. One can use the top label alone.

   2. One can do better by using all the (non-SPL) labels in the
      stack.

   3. One can do even better by "divining" the type of embedded
      packet, and using fields from the guessed header.

   4. One can do best by using either an Entropy Label [RFC6790] or a
      FAT Pseudowire Label [RFC6391]; see Section 2.1.3.)

Load balancing based on just the top label means that all packets
with that top label will go the same way -- this is far from ideal.
Load balancing based on the entire label stack (not including SPLs)
is better, but may still be uneven. If, however, 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 heuristic that is commonly used to guess the type
of the embedded packet. The first nibble, i.e., the four most
significant bits of the first octet, of an IP header contains the IP
version number. This in turn indicates where to find the relevant
fields for load balancing. The heuristic goes roughly as follows:

2.1.1.1. Heuristic for Load Balancing

1. If the MFN is 0x4 (0100b), treat the payload as an IPv4 packet,
   and find the relevant fields for load balancing on that basis.

2. If the MFN is 0x6 (0101b), treat the payload as an IPv6 packet,
   and find the relevant fields for load balancing on that basis.

3. If the MFN 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 Figure 1, A. However, this heuristic
can work very badly for Figure 1, B. For example, if payload B is an
Ethernet frame, then the MFN is the first nibble of the OUI of the
destination MAC address, which can be 0x4 or 0x6, and if so would
lead to very bad load balancing. This behavior can happen to other
types of non-IP payload as well.

This in turn led to the idea of inserting a PSH (e.g., a pseudowire
control word [RFC4385], a DetNet control word [RFC8964] or a BIER
header [RFC8296]) where the MPLS First Nibble is NOT 0x4 or 0x6, to
explicitly prevent 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. RFC 8469 was
published at the request of the operator community and the IEEE RAC
as a result of operational difficulties with pseudowires that did
not contain the control word.

This memo introduces a requirement and a recommendation, the first
building on the above; the second deprecating the use of the
heuristic in Section 2.1.1.1. The intent of both of these is that
legacy routers continue to operate as they have, with no new
problems introduced as a result of this memo. However, new
implementations SHOULD follow these recommendations for more robust
operation.

2.1.2. Requirement

Going forward, network equipment MUST use a post-stack header with
an MPLS First Nibble value that is not 0x4 or 0x6 in all cases when
the MPLS payload is not an IP packet. Effectively, Figure 1, B is
disallowed. [AGREED???]

This replaces the following text from [RFC4928], section 3,
paragraph 3:

"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."

This also replaces the following text from [RFC8469], section 4,
paragraph 1:

"This document updates [RFC4448] to state that both the ingress
provider edge (PE) and the egress PE SHOULD support the Ethernet PW
CW and that, if supported, the CW MUST be used."

2.1.3. Recommendation

It is RECOMMENDED that, going forward, if good load balancing of
MPLS packets is desired, either an Entropy Label or a FAT Pseudowire
Label SHOULD be used; furthermore, going forward, the heuristic in
Section 2.1.1.1 MUST NOT be used. [AGREED???]

A consequence of Recommendation 2 is that, while legacy routers may
look for a MPLS First Nibble of 0x4 or 0x6, no router will look for
a MPLS First Nibble of 0x7 (or whatever the next IP version number
will be) for load balancing purposes. This means that the values 0x4
and 0x6 are used to (sometimes incorrectly) identify IPv4 and IPv6
packets, but no other First Nibble values will be used to identify
IP packets.

This obviates the need for paragraph 4, section 3 in [RFC4928]:

"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."

This also expands the MFN Registry to all 16 possible values, not
just 0x0 and 0x1.

2.1.4. Parsing the Post-stack Header

Given the above recommendations on the use of a post-stack header
and future non-use of the heuristic (Section 2.1.1.1) via the use of

Entropy or FAT Pseudowire Labels, the main reason for creating a
First Nibble registry is to document the types of post-stack headers
that may follow a label stack, and to simplify their parsing.

2.2. Why Create a Registry

The MPLS WG is currently engaged in updating the MPLS architecture;
part of this work involves the use of post-stack headers. This is
not possible if post-stack header values are allocated on an ad hoc
basis, and their parsing and semantics is ill-specified. Consider
that the MPLS First Nibble value of 0x0 has two different formats,
depending on whether the post-stack header is a pseudowire control
word or a DetNet control word; disambiguation requires the context
of the service label. This was a considered decision; documenting
this would be helpful to future implementors.

With a registy, post-stack headers become easier to parse; the
values are unique, not needing means outside the data plane to
interpret them correctly; and their semantics and usage are
documented. (Thank you, IANA!)

2.3. Caveat

The use of the MPLS First Nibble 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
[RFC4928]. This situation may confuse some as to relationship
between the MPLS First Nibble Registry and the IP Version Numbers
registry. These registries are quite different:

1. The IP Version Numbers registry's explicit purpose is to track
   IP version numbers in an IP header.

2. The MPLS First Nibble registry's purpose is to track post-stack
   header types.

The only intersection points between the two registries is for
values 0x4 and 0x6 (for backward compatibility). There is no need to
track future IP version number allocations in the MPLS First Nibble
registry.

3. IANA Considerations

3.1. MPLS First Nibble Registry

This memo recommends the creation of an IANA registry called "The
MPLS First Nibble Registry" with the following values:

Value	Meaning	Reference	Allocation Policy
0x0	PW Control Word	RFC 4385	
0x0	DetNet Control Word	RFC 8964	
0x1	PW Assoc Channel	RFC 4385	
0x2	Unallocated		Standards Action
0x3	Unallocated		Standards Action
0x4	IPv4 header	RFC 791	
0x5	BIER header	RFC 8296	
0x6	IPv6 header	RFC 8200	
0x7-e	Unallocated		Standards Action
0xf	Reserved for expansion		Standards Action

Table 1: MPLS First Nibble Values

3.1.1. Allocation Policy

All new values registered here MUST use the Standards Action policy [RFC8126].

4. References

4.1. Normative References

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

4.2. Informative References

[RFC4364]   Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private
             Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364,
             February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[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>.

[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>.

Authors' Addresses

Kireeti Kompella (editor)
Juniper Networks
1133 Innovation Way
Sunnyvale, 94089
United States of America

Phone: +1-408-745-2000
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
Ericsson

Email: gregimirsky@gmail.com

Lars Olaf (Loa) Andersson
Bronze Dragon Consulting

Email: loa@pi.nu