A Primer on the Development of MPLS
draft-bryant-mpls-dev-primer-01

Abstract

There has been significant recent interest in developing MPLS to address new needs. This memo collects together various documents that together describe the key aspects of the MPLS architecture together with the development proposals that the author is aware of.

The purpose of this document is to bring everyone up to speed on the rational for the existing design and to alert them to the new proposals.

Table of Contents

1. Introduction

There has been significant recent interest in developing the MPLS data plane to address new needs. This memo collects together various documents that together describe the key aspects of the MPLS architecture together with the development proposals that the author is aware of.

The intent of this work is to bring everyone up to speed on the rational for the existing design and to alert them to the new proposals. If I have missed any proposals, then please accept my apologies for the oversight, let me know and I will include it in the list.

I do not anticipate that this memo will progress to become an RFC.

The interest in this work to develop MPLS was noted by the Chairs of the DETNET, MPLS, PALS, and SPRING IETF working groups who called a joint meeting at IETF 110. The agenda, slides notes etc of the meeting are currently at https://datatracker.ietf.org/meeting/agenda/

Editor's note the above URL will change when the meeting is included in the IETF Proceedings.

A video recording of the meeting is to be found at https://youtu.be/rt_vQTToO1s

2. Background Documents

2.1. Multiprotocol Label Switching Architecture (RFC 3031)

[RFC3031] describes the base architecture of MPLS. This is updated by:

- o [RFC6178] which describes label edge router forwarding of IPv4 option packets and is not relevant to this discussion.

- o [RFC6790] which describes the use of entropy labels in MPLS forwarding. The entropy label is a two stack entry tuple that consists of a special purpose label (SPL) followed by a label stack entry (LSE) who's label is pure entropy to be applied to the selection of a path from the Equal Cost Multi-Path (ECMP) set. Its use in choosing the ECMP member is optional. This was the first formal introduction of the concept of an MPLS forwarder needing to parse below the top of the label stack.

Prior to the introduction of RFC 6790 it was already common practice for LSRs to scan the label stack to the bottom of stack (a simple task requiring the checking of a single bit in each LSE) to heuristically check that the packet was IP and if so to hash the five tuple to determine the ECMP path to use.

This approach worked well until Ethernet addresses were (legitimately) deployed that confused these forwarders into thinking that Ethernet packets were IP packets ((RFC8469}}.

2.2. MPLS Label Stack Encoding (RFC 3032)

[RFC3032] Specifies the encoding of an MPLS LSE. This has been updated by:

- [RFC3443] describes the two models of TTL handling in MPLS. In addition that it should be noted that some LSR decrement the TTL of the TOP label _before_ inspecting the label in the top LSE.

- [RFC3270] describes the application of diffserv to MPLS and the pipe vs uniform models. It may apply to this work if we are popping xSPLs.

EDITOR'S NOTE need to think about [RFC3270] some more.

- [RFC5129] describes Explicit Congestion Marking in MPLS. It is not clear how widely this is deployed. This is something that needs to be thought through when re-purposing the TC bits as has been proposed in some of the drafts listed below.

- [RFC5462] renames the EXP field to TC field. This is a naming convention and not a technology change.

- [RFC5586] defines the Generic Associated Channel (see later).

- [RFC7274] defines the process for allocating and retiring special purpose labels (SPLs) (formerly known as Reserved Labels). It also creates the extended special purpose labels (eSPLs). eSPLs are another instance of a twin label in the stack, with the first label (15) from the SPL range followed by another label that specifies the function of the twin labels.

2.3. Control Word for Use over an MPLS PSN (RFC5586)

[RFC5586] describes the first use of a type of meta-data between the bottom of the MPLS label stack and the packet payload. The pseudowire (PW) control word (CW) is used to carry additional information that the egress Provider Edge (PE) LSR needs to construct the outgoing packet. It also tells the forwarder that the packet is not an IP packet and so should not be subjected to ECMP.

[RFC8469] is an update to the Ethernet PW specification [RFC4448] to strongly encourage the use of the CW for Ethernet PWs. All other PWs that have been defined require the use of the CW.

2.4. Avoiding Equal Cost Multipath (ECMP) Treatment in MPLS Network

[RFC4928] describes the issue of accidental ECMP of packets not carry IP. This arises because some forwarders accidentally mistake a non-IP packet for an IP packet. They use a heuristic method of determining that the packet is IP. They sometimes get this wrong causing a misordered the delivery of some of the packets. [RFC4928] formally introduces the concept of using the first nibble of 0b0000 to avoid this. Note that the technique is actually older than this having been introduced in [RFC4385].

2.5. Synonymous Flow Labels

[RFC8957] describes a technique whereby additional labels are introduced to an MPLS network that mimic the behavior of other MPLS labels but also introduce new properties to the FEC. The main use is to trigger OAM actions, but the method can be used to trigger other agreed actions.

2.6. Residence Time Measurement in MPLS Networks

[RFC8169] describes a method of measuring the dwell or residence time of a packet in a router. The time is accumulated in a G-ACh packet carried below the label stack. Note that the G-ACh uses first nibble = 0b0001. This is the first instance of a G-ACh packet being specified for operation on a user data packet.

The data packet is carried over an RSVP-TE path and thus the top of stack label indicates to the forwarder both the next-hop and outgoing label, but also indicates the presence of the G-ACh and the need to perform the residence time accumulation.

The RFC predates segment routing (SR) and does not mention Software Defined Networks (SDNS), but the method could be used in those environments.

3. Other Observations Received

The following comments were received and are included for the benefit of the reader.

Some of these comments should probably me moved to a requirements draft.

For better of worse, passive taps and splitters are universally deployed for various monitoring purposes, the resulting feed is a direct copy of the traffic on the wire; active monitoring within the network element is equally widely deployed. Neither of these

mechanisms have insight into the FEC context. This would imply that the encapsulation is deterministic - i.e. it needs to be possible for the packet to be unambiguously decoded by such a side observer.

Filters for monitored traffic also need to be created somehow, this implies that an LSE and the new header wire format needs to be unambiguous in the context that does not have access to endpoints.

The amount of state that needs to be distributed across the domain, the rate of such state change and state accumulation points needs to be understood. If information needs to be signaled end to end, what is the impact on the transit nodes and what is the impact of the total state that is kept within a domain. An example is MSD propagation - while it is on the order of a rounding error for an IGP to propagate, a generic realization would require maintaining all of that state for whole domain in every node, and realistically that will be on the order of a few tens of MSD entries per page. While not a problem for platforms with wide address space, 32 bit address space systems are widely deployed and will stay in a foreseeable future - this increase of state is not free in their context.

From the practical implementation point of view, there appear to be three large groups

o hardware implementations

o scalar software

o vector software implementations

Vector implementations are the fastest growing ones, both by the addressable market and by the performance benefits. However, those benefits can easily be written off if data formats are in direct opposition to vector processing rules. vertical processing within a lane is what vector implementations assume, with reasonably good support for horizontal operations within lanes/ However support for anything that does not align uniformly to all lanes is quite poor. Variable length headers with fields being spread across various lane positions are a bad match for this technology. If it is within a cache line the penalty is still reasonable but quickly approaches the point where the benefits of vector processing are over-shadowed by the additional processing required to get data in the right order.

While MPLS-TP and MPLS have the same data plane encapsulation, they do not necessarily make practical use of the same data plane instance - it is a basic network design aspect to separate different classes of traffic in different layers.

There are many design issues to look at, but before we go too far with the new proposals we need to understand the practical uses of the technology and the practical limitations of the methods of implementation.

EDITOR'S NOTE add in illustration label stacks for major encapsulations - single and multiple label IPv4/IPv6, and packet pseudowires - the encapsulations that make the dominant part of traffic. Something similar to what extended headers document has for showing wire image, but for current encapsulation - to have an illustratory view of which node needs to lookup what and how deep.

4. New Proposals

This section catalogues new proposals for how metadata is carried and how its presence is indicated to the forwarder.

4.1. MPLS Extension Header Architecture

[I-D.andersson-mpls-eh-architecture] specifies an architecture for the extension of MPLS to include Extension Headers (EH). The proposal is for the EHs to carry information on in-network services and functions in an MPLS network. The extension headers are carried after the MPLS Label Stack, and the presence of EHs are indicated in the label stack by an Extension Header Indicator (EHI).

Proposed use cases are:

- In-situ OAM
- Network Telemetry and Measurement
- Network Security
- Segment Routing
- Network Programming

The draft calls two types of EH:

- "hop-by-hop" (HBH)
- "End to end" (E2E).

The draft proposes to indicate the presence of the EH via the FEC. The ability of a router on the LSP to process a packet correctly is advertised in the routing protocol.

4.2. MPLS Label Operations in MPLS EH capable networks

[I-D.andersson-mpls-eh-label-stack-operations] 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 this describes how 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 Extended Special Purpose label called Extension Header Indicator (EHI) in the label stack.

The EH(s) are carried over a G-ACh. Three ACHs are suggested E2E, HBH, Both. A number of EHs can be accommodated with the number being indicated by a parameter in the ACH.

The draft considers the stack structure in a number of cases such as VPN (and presumably PW) and non-VPN (native IP payload) cases.

The draft shows how RSVP-TE signaling would work.

4.3. Encapsulation For MPLS Performance Measurement with Alternate Marking

[I-D.ietf-mpls-inband-pm-encapsulation] shows how a flow ID can be carried in a packet. The application is Alternate Marking (AM) for performance monitoring of the network.

It proposes the use of an eSPL (two LSEs) preceding a third LSE which the flow ID in its label field.

This LSE triplet can occur more than once in the label stack and can occur at any position within the label stack. If it is used for two different purposes in the stack the Flow ID must be different.

Where Alternate Marking is used two method of creating the alternating pairs are proposed, using two Flow IDs which will have ECMP implications possibly requiring the including of an entropy label pair, or using the TC bits which may effect queue priority on the egress LSR when the Flow ID is bottom of stack.

Considerations of maximum stack depth apply.

4.4. MPLS Data Plane Encapsulation for In-situ OAM Data

[I-D.gandhi-mpls-ioam-sr] shows how the IOAM data fields defined in [I-D.ietf-ippm-ioam-data] could be carried in MPLS. It carries the OAM data in an G-Ach and specifies both hop-by-hop and end-to-end versions.

The OAM present/type indicator is an e(SPL) at the bottom of the MPLS label stack requiring a P-router to scan the stack to find the label.

The draft proposes the stacking of G-Ach blocks at the bottom of stack with the IOAM G-Ach first and any subsequent G-Ach located through the use of a length field in the IOAM G-Ach.

4.5. Multi-purpose Special Purpose Label for Forwarding Actions

[I-D.kompella-mpls-mspl4fa] notes that the forwarder does not need to use the TC, or TTL fields in an LSE that does not become top of stack. It proposes to exploit these fields as indicators of forwarding actions, by modifying the semantics of these fields.

There are a number of key proposals in the draft:

- Using the "spare bits" as forwarding indicator flags to specify actions or in some cases inactions
- Using the method to multi-purpose SPLs and thus expand the number of single label SPLs available to the IETF.
- Reusing the Entropy Label fields to carry additional data needed by the forwarder. This latter point could be adopted by any eSPL. One use for this additional data that was proposed (certainly in discussion but I cannot see it in the draft) was the use of this facility to carry a network slice identifier.

4.6. No Further Fast Reroute

[I-D.kompella-mpls-nffrr] proposes the use of an SPL (note not an eSPL) to indicate that a fast re-route action is not to be undertaken on the packet.

Uses an SPL for this single purpose

4.7. Carrying Virtual Transport Network Identifier in MPLS Packet

[I-D.li-mpls-enhanced-vpn-vtn-id] is a method of carrying a virtual network identifier in an MPLS packet. It does this by carrying meta-

data below the MPLS label stack. It does not use the G-ACh but instead a new design with a first nibble value of 0b0011.

Note that when we define new first nibbles we are technically taking IP versions away from the IETF Internet Area. When PWE3 first proposed this we agreed with the IETF of the day that we would only take 0b0000 and 0b0001. I am looking to see if this agreement was documented.

The presence of the VTN is indicted by an SPL (note not an eSPL) somewhere in the label stack. The draft discusses how multiple VTNs can be placed in the packet, but not how multiple types of meta-data are to be carried.

4.8. Segment Routed Time Sensitive Networking

[I-D.stein-srtsn] describes how information can be encoded in the MPLS label stack to inform the forwarder when a time sensitive packet should be sent. Each LSE becomes 64 bits, the first 32 bits a conventional MPLS label and the second part contains dispatch time information.

Note that as far as I can see there is no provision for an S bit making label stack scanning for other information liable to make a mistake.

There is no information that I can see stating how the LSR knows that the LSE is in this format and so I assume that it knows from the FEC.

4.9. Options for MPLS Extension Header Indicator

[I-D.song-mpls-extension-header] provides a catalogue of methods of identifying the presence of presence of an extension header after the label stack. The methods could of course be used for identifying the presence of some other structure after the label stack.

The methods listed are:

- A special purpose label
- An extended special purpose label pair
- A GAL and an associated channel header
- A GAL followed by a structure with a different first nibble value
- The use of a new FEC

4.10. MPLS Extension Header

[I-D.song-mpls-extension-header] describes a design for an MPLS extension header to be placed after the MPLS label stack. The Header of Extension Headers (HEH) specifies the number of extension headers that follow. The HEH has the four bit ECMP defeat nibble, a count of number of extension headers, the length of the set of extension headers and the type of the following extension header. An Extension Header (EH) starts with the type of the header that follows this EH, the length of this EH followed by the EH data/payload.

Two generic types of EH as specified, End to End and Hop by Hop.

4.11. MPLS Payload Protocol Identifier

[I-D.xu-mpls-payload-protocol-identifier] describes a method of adding a protocol identifier (PID) to an MPLS packet.

A 16 bit PID is carried in a 32 bit structure following the label stack. The structure just has an ECMP defeat nibble 0b000 and the PID.

Presence of the PID is indicated by an SPL or an eSPL at the bottom of stack.

An alternative method of indicating the PIL is also proposed by using a first nibble of 0b1111. Note that this might be defeated by an MPLS payload other than IP. For reasons discussed in [RFC8469] in this arrangement the PIL could not be used at a mid-point, but would be safe at an endpoint. The first nibble comments in {#VTN} also apply to this proposal.

No provision is made for carrying other data beyond the bottom of stack, and there is no discussion on how this works with VPNs and PWs.

4.12. Generic Transport Functions

[I-D.zzhang-tsvwg-generic-transport-functions] describes a method of adding fragmentation to a number of protocols including MPLS. The fragmentation header follows the end of stack. It does not take any ECMP precautions through the first nibble. Some mitigation could be achieved by the use of the ELI/EL where the P routers can support this.

Indication of the fragmentation header is indicated by the FEC.

Note that the draft referenced pseudowires (PWs) and that PWs have a fragmentation method [RFC4623]. However this feature is not thought to be widely implemented.

4.13. Use of an MPLS LSE as an Ancillary Data Pointer

[I-D.bryant-mpls-aux-data-pointer] described how Label Stack Entries (LSEs) can be used to point to ancillary data carried below the MPLS label stack. This allows the stack to explicitly direct the forwarder to specific items of ancillary (meta) data, thereby reducing the ambiguity of the various implicit systems proposed. Thus, as an example it is possible to specify a latency requirement on a path segment rather than requiring the forwarder to determine which of several latency specifations are applicable to it.

A difficulty with the pointer approach occurs if the packet is ever expanded, for example as a result of the use of an iOAM incremental trace approach [I-D.ietf-ippm-ioam-data] adapted for MPLS. It is not clear how widely deployed such an approach will be. A mitigation approach is expected to be proposed in he next version of [I-D.bryant-mpls-aux-data-pointer].

5. Security Considerations

Any changes to the MPLS security model as a result of a change will need to be considered within the proposals themselves. This document is a catalog of existing RFCs and design proposals and does not itself modify the security of MPLS networks.

6. IANA Considerations

This document has no IANA requests.

7. References

7.1. Normative References

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

7.2. Informative 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.andersson-mpls-eh-label-stack-operations]
Andersson, L., Guichard, J. N., Song, H., and S. Bryant, "MPLS Label Operations in MPLS EH capable networks", draft-andersson-mpls-eh-label-stack-operations-02 (work in progress), October 2021.

[I-D.bryant-mpls-aux-data-pointer]
Bryant, S., Clemm, A., and T. Eckert, "Use of an MPLS LSE as an Ancillary Data Pointer", draft-bryant-mpls-aux-data-pointer-00 (work in progress), May 2021.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-06 (work in progress), February 2021.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-17 (work in progress), December 2021.

[I-D.ietf-mpls-inband-pm-encapsulation]
Cheng, W., Min, X., Zhou, T., Dong, X., and Y. Peleg, "Encapsulation For MPLS Performance Measurement with Alternate Marking Method", draft-ietf-mpls-inband-pm-encapsulation-02 (work in progress), October 2021.

[I-D.kompella-mpls-mspl4fa]
Kompella, K., Beeram, V. P., Saad, T., and I. Meilik, "Multi-purpose Special Purpose Label for Forwarding Actions", draft-kompella-mpls-mspl4fa-01 (work in progress), July 2021.

[I-D.kompella-mpls-nffrr]
Kompella, K. and W. Lin, "No Further Fast Reroute", draft-kompella-mpls-nffrr-02 (work in progress), July 2021.

[I-D.li-mpls-enhanced-vpn-vtn-id]
Li, Z. and J. Dong, "Carrying Virtual Transport Network Identifier in MPLS Packet", draft-li-mpls-enhanced-vpn-vtn-id-01 (work in progress), April 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.

[I-D.stein-srtsn]
Stein, Y. (., "Segment Routed Time Sensitive Networking", draft-stein-srtsn-01 (work in progress), August 2021.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X., Assarpour, H., Ma, S., and F. Clad, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-09 (work in progress), September 2021.

[I-D.zzhang-tsvwg-generic-transport-functions]
Zhang, Z., Bonica, R., and K. Kompella, "Generic Transport Functions", draft-zzhang-tsvwg-generic-transport-functions-00 (work in progress), November 2020.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <https://www.rfc-editor.org/info/rfc3270>.

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <https://www.rfc-editor.org/info/rfc3443>.

[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>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <https://www.rfc-editor.org/info/rfc4448>.

[RFC4623] Malis, A. and M. Townsley, "Pseudowire Emulation Edge-to-Edge (PWE3) Fragmentation and Reassembly", RFC 4623, DOI 10.17487/RFC4623, August 2006, <https://www.rfc-editor.org/info/rfc4623>.

[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>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <https://www.rfc-editor.org/info/rfc5129>.

[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>.

[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>.

[RFC6178] Smith, D., Mullooly, J., Jaeger, W., and T. Scholl, "Label Edge Router Forwarding of IPv4 Option Packets", RFC 6178, DOI 10.17487/RFC6178, March 2011, <https://www.rfc-editor.org/info/rfc6178>.

[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>.

[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>.

[RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <https://www.rfc-editor.org/info/rfc8169>.

[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>.

[RFC8957] Bryant, S., Chen, M., Swallow, G., Sivabalan, S., and G. Mirsky, "Synonymous Flow Label Framework", RFC 8957, DOI 10.17487/RFC8957, January 2021, <https://www.rfc-editor.org/info/rfc8957>.

Author's Address

Stewart Bryant
University of Surrey

Email: sb@stewartbryant.com