MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 22, 2021

R. Gandhi, Ed.
Z. Ali
C. Filsfils
F. Brockners
Cisco Systems, Inc.
B. Wen
V. Kozak
Comcast

February 18, 2021

MPLS Data Plane Encapsulation for In-situ OAM Data
draft-gandhi-mpls-ioam-sr-06

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the data packet while the packet traverses a path between two nodes in the network. This document defines how IOAM data fields are transported with MPLS data plane encapsulation using new Generic Associated Channel (G-ACh).

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 August 22, 2021.

Copyright Notice

Table of Contents

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than being sent within the probe packets specifically

dedicated to OAM or Performance Measurement (PM). The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data], and can be used for various use-cases for OAM and PM. The IOAM data fields are further updated in [I-D.ietf-ippm-ioam-direct-export] for direct export use-cases and in [I-D.ietf-ippm-ioam-flags] for Loopback and Active flags.

This document defines how IOAM data fields are transported with MPLS data plane encapsulations using new Generic Associated Channel (G-ACh).

2. Conventions

2.1. Requirement Language

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

2.2. Abbreviations

Abbreviations used in this document:

ECMP	Equal Cost Multi-Path
E2E	Edge-To-Edge
G-ACh	Generic Associated Channel
HbH	Hop-by-Hop
IOAM	In-situ Operations, Administration, and Maintenance
MPLS	Multiprotocol Label Switching
OAM	Operations, Administration, and Maintenance
PM	Performance Measurement
POT	Proof-of-Transit
PSID	Path Segment Identifier
PW	PseudoWire
SR	Segment Routing

SR-MPLS Segment Routing with MPLS Data plane

3. MPLS Extensions for IOAM Data Fields

3.1. IOAM Generic Associated Channel

The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. The IOAM data fields are carried in the MPLS header as shown in Figure 1. More than one trace options can be present in the IOAM data fields. G-ACh [RFC5586] provides a mechanism to transport OAM and other control messages over MPLS data plane. The IOAM G-ACh header [RFC5586] with new IOAM G-ACh type is added immediately after the MPLS label stack in the MPLS header as shown in Figure 1, before the IOAM data fields.

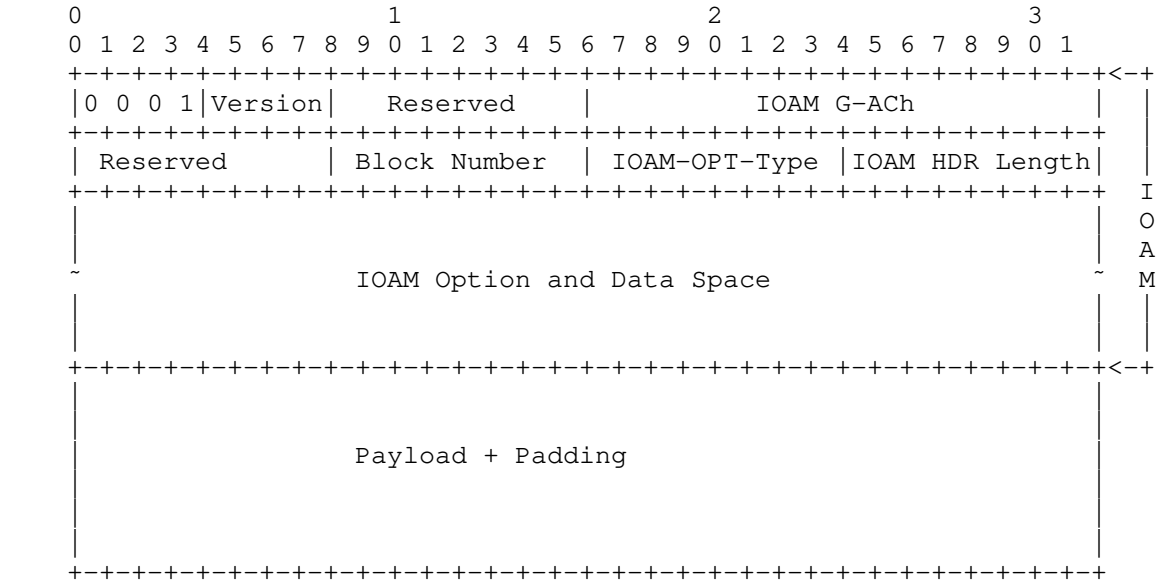

Figure 1: IOAM Generic Associated Channel with IOAM Data Fields

The IOAM data fields are encapsulated using the following fields in the MPLS header:

IP Version Number 0001b: The first four octets are IP Version Field part of a G-ACh header, as defined in [RFC5586].

Version: The Version field is set to 0, as defined in [RFC4385].

IOAM G-ACh: Generic Associated Channel (G-ACh) Type (value TBA3) for IOAM [RFC5586].

Reserved: Reserved Bits MUST be set to zero upon transmission and ignored upon receipt.

Block Number: The Block Number can be used to aggregate the IOAM data collected in data plane, e.g. compute measurement metrics for each block of a flow. It is also used to correlate the IOAM data on different nodes.

IOAM-OPT-Type: 8-bit field defining the IOAM Option type, as defined in Section 8.1 of [I-D.ietf-ippm-ioam-data].

IOAM HDR LEN: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

IOAM Option and Data Space: IOAM option header and data is present as defined by the IOAM-OPT-Type field, and is defined in Section 5 of [I-D.ietf-ippm-ioam-data].

3.2. IOAM Indicator Labels

An IOAM Indicator Label is used to indicate the presence of the IOAM data fields in the MPLS header. There are two IOAM types defined in this document: Edge-to-Edge (E2E) and Hop-by-Hop (HbH) IOAM. If only edge nodes need to process IOAM data then E2E IOAM Indicator Label is used so that intermediate nodes can ignore it. If both edge and intermediate nodes need to process IOAM data then HbH IOAM Indicator Label is used. Different IOAM Indicator Labels allow to optimize the IOAM processing on intermediate nodes by checking if IOAM data fields need to be processed.

4. Edge-to-Edge IOAM

4.1. Edge-to-Edge IOAM Indicator Label

The E2E IOAM Indicator Label is used to indicate the presence of the E2E IOAM data fields in the MPLS header as shown in Figure 2.

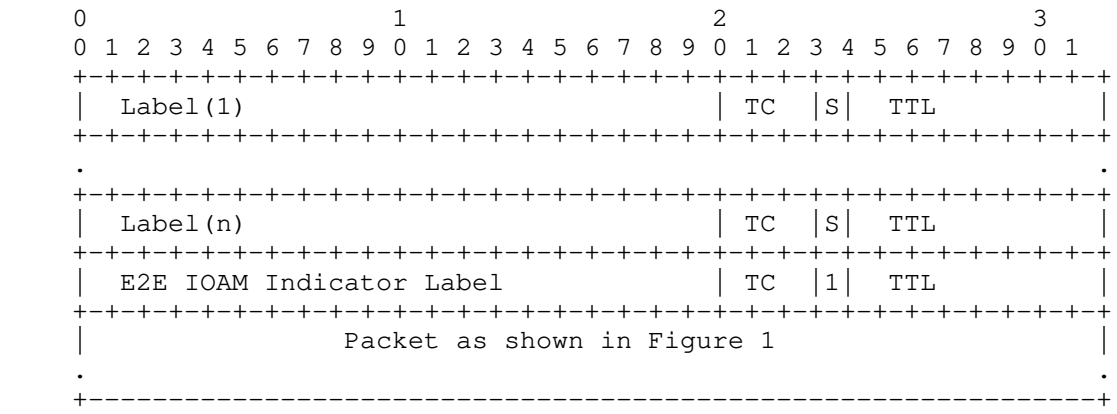

Figure 2: MPLS Encapsulation for E2E IOAM

The E2E IOAM data fields carry the Option-Type(s) that require processing on the encapsulating and decapsulating nodes only. The IOAM Option-Type carried can be IOAM Edge-to-Edge Option-Type [I-D.ietf-ippm-ioam-data]. The E2E IOAM data fields SHOULD NOT carry any IOAM Option-Type that require IOAM processing on the intermediate nodes as it will not be processed by them.

4.2. Procedure for Edge-to-Edge IOAM

The E2E IOM procedure is summarized as following:

- The encapsulating node inserts the E2E IOAM Indicator Label and one or more IOAM data fields in the MPLS header.

- The intermediate nodes do not process IOAM data fields.

- The decapsulating node "punts the timestamped copy" of the received packet as is including the IOAM data fields when the node recognizes the IOAM Indicator Label. The copy of the packet is punted with receive timestamp to the slow path for IOAM data fields processing. The receive timestamp is required by the various E2E OAM use-cases, including streaming telemetry. Note that it is not necessarily punted to the control-plane.

- The decapsulating node processes the IOAM data fields using the procedures defined in [I-D.ietf-ippm-ioam-data]. An example of IOAM processing is to export the data fields, send data fields via streaming telemetry, etc.

- The decapsulating node also pops the IOAM Indicator Label and the IOAM data fields from the received packet. The decapsulated

packet is forwarded downstream or terminated locally similar to the regular data packets.

4.3. Edge-to-Edge IOAM Indicator Label Allocation

The E2E IOAM Indicator Label is used to indicate the presence of the E2E IOAM data fields in the MPLS header. The E2E IOAM Indicator Label can be allocated using one of the following three methods:

- Label assigned by IANA with value TBA1 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].
- Label allocated by a Controller from the global table of the decapsulating node. The Controller provisions the label on both encapsulating and decapsulating nodes.
- Label allocated by the decapsulating node and signalled or advertised in the network. The signaling and/or advertisement extension for this is outside the scope of this document.

5. Hop-by-Hop IOAM

5.1. Hop-by-Hop IOAM Indicator Label

The HbH IOAM Indicator Label is used to indicate the presence of the HbH IOAM data fields in the MPLS header as shown in Figure 3.

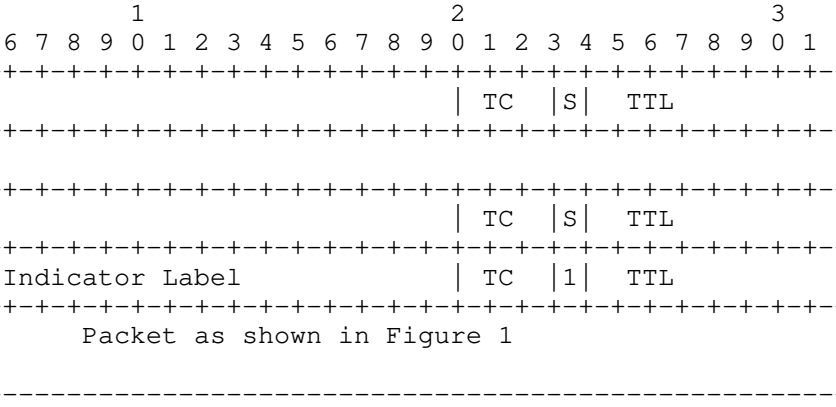

Figure 3: MPLS Encapsulation for HbH IOAM

The HbH IOAM data fields carry the Option-Type(s) that require processing at the intermediate and/or encapsulating and decapsulating nodes. The IOAM Option-Type carried can be IOAM Pre-allocated Trace Option-Type, IOAM Incremental Trace Option-Type and IOAM Proof of

Transit (POT) Option-Type, as well as Edge-to-Edge Option-Type [I-D.ietf-ippm-ioam-data].

5.2. Procedure for Hop-by-Hop IOAM

The HbH IOAM procedure is summarized as following:

- The encapsulating node inserts the HbH IOAM Indicator Label and one or more IOAM data fields in the MPLS header.

- The intermediate node enabled with HbH IOAM functions processes the data packet including the IOAM data fields as defined in [I-D.ietf-ippm-ioam-data] when the node recognizes the HbH IOAM Indicator Label present in the MPLS header. The intermediate node may 'punt the timestamped copy' of the received data packet including the IOAM data fields as required by the IOAM data fields processing. The copy of the packet is punted with receive timestamp to the slow path for IOAM processing.

- The intermediate node forwards a copy of the processed data packet downstream.

- The decapsulating node "punts the timestamped copy" of the received data packet as is including the IOAM data fields when the node recognizes the IOAM Indicator Label. The copy of the packet is punted with receive timestamp to the slow path for IOAM data fields processing. The receive timestamp is required by the various E2E OAM use-cases, including streaming telemetry. Note that it is not necessarily punted to the control-plane.

- The decapsulating node processes the IOAM data fields using the procedures defined in [I-D.ietf-ippm-ioam-data]. An example of IOAM processing is to export the data fields, send data fields via streaming telemetry, etc.

- The decapsulating node also pops the IOAM Indicator Label and the IOAM data fields from the received packet. The decapsulated packet is forwarded downstream or terminated locally similar to the regular data packets.

5.3. Hop-by-Hop IOAM Indicator Label Allocation

The HbH IOAM Indicator Label is used to indicate the presence of the HbH IOAM data fields in the MPLS header. The HbH IOAM Indicator Label can be allocated using one of the following three methods:

- Label assigned by IANA with value TBA2 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].

- Label allocated by a Controller from the network-wide global table. The Controller provisions the labels on all nodes participating in IOAM functions along the data traffic path.

- Labels allocated by the intermediate and decapsulating nodes and signalled or advertised in the network. The signaling and/or advertisement extension for this is outside the scope of this document.

6. Considerations for IOAM Indicator Label

6.1. Considerations for ECMP

The encapsulating node needs to make sure the IOAM data fields do not start with a well-known IP Version Number (e.g. 0x4 for IPv4 and 0x6 for IPv6) as that can alter the hashing function for ECMP that uses the IP header. This is achieved by using the IOAM G-ACh with IP Version Number 0001b after the MPLS label stack [RFC5586].

Note that the hashing function for ECMP that uses the labels from the MPLS header may now include the IOAM Indicator Label.

When entropy label [RFC6790] is used for hashing function for ECMP, the procedure defined in this document does not alter the hashing function.

6.2. Node Capability

The decapsulating node that has to pop the IOAM Indicator Label, data fields, and perform the IOAM function may not be capable of supporting it. The encapsulating node needs to know if the decapsulating node can support the IOAM function. The signaling extension for this capability exchange is outside the scope of this document.

The intermediate node that is not capable of supporting the IOAM functions defined in this document, can simply skip the IOAM processing of the MPLS header.

6.3. MSD Considerations

The SR path computation needs to know the Maximum SID Depth (MSD) that can be imposed at each node/link of a given SR path [RFC8664]. This ensures that the SID stack depth of a computed path does not exceed the number of SIDs the node is capable of imposing. The MSD used for path computation MUST include the IOAM Indicator Label.

6.4. Nested MPLS Encapsulation

The data packets with IOAM data fields carry only one IOAM Indicator Label in the MPLS header. Any intermediate node that adds additional MPLS encapsulation in the MPLS header may further update the IOAM data fields in the header without inserting another IOAM Indicator Label. When a packet is received with a HbH IOAM Indicator Label, the nested MPLS encapsulating node can add a HbH and/or E2E IOAM Option-Type. However, when a packet is received with an E2E IOAM Indicator Label, the nested MPLS encapsulating node SHOULD NOT add a HbH IOAM Option-Type, as intermediate nodes will not process it.

7. MPLS Encapsulation with Control Word and Another G-ACh for IOAM Data Fields

The IOAM data fields, including IOAM G-ACh header are added in the MPLS encapsulation immediately after the MPLS header. Any Control Word [RFC4385] or another G-ACh [RFC5586] MUST be added after the IOAM data fields in the packet as shown in the Figure 4 and Figure 5, respectively. This allows the intermediate nodes to easily access the HbH IOAM data fields located immediately after the MPLS header. The decapsulating node can remove the MPLS encapsulation including the IOAM data fields and then process the Control Word or another G-ACh following it.

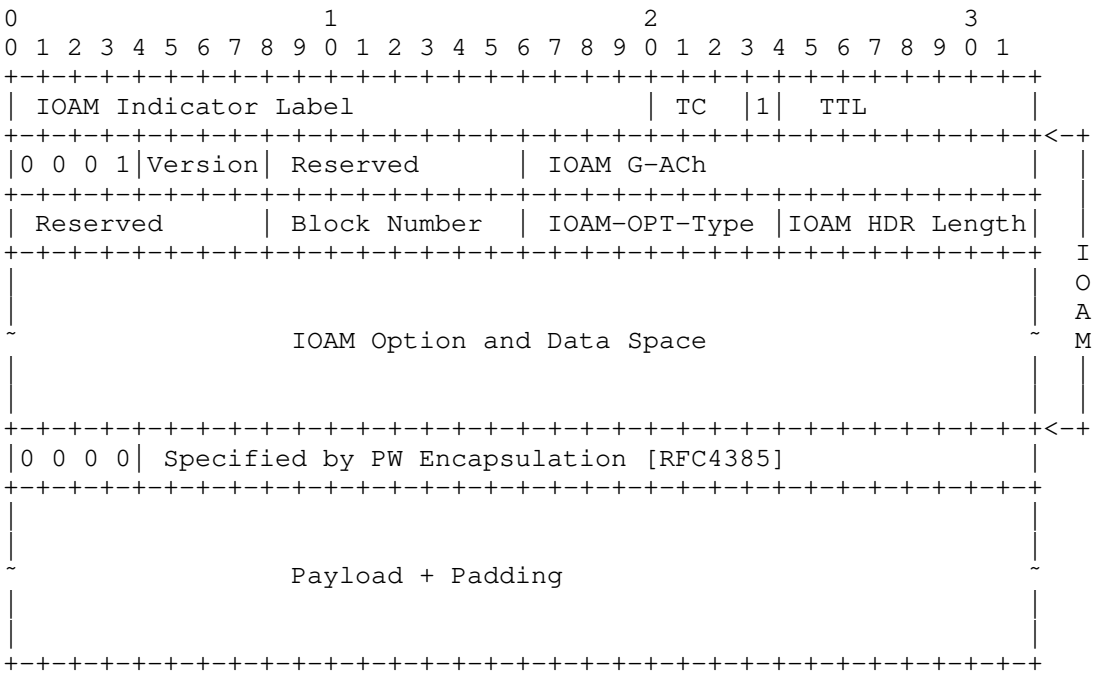

Figure 4: Example MPLS Encapsulation with Generic PW Control Word with IOAM

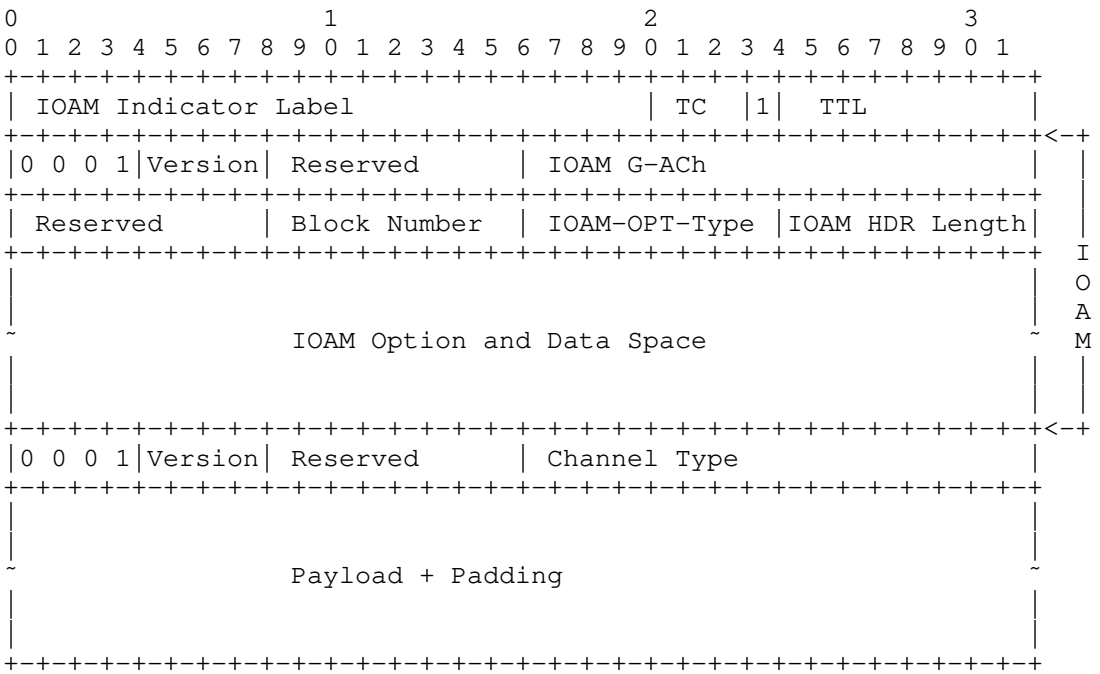

Figure 5: Example MPLS Encapsulation with Another G-ACh with IOAM

8. Example MPLS Encapsulations

8.1. Example SR-MPLS Encapsulation with IOAM

Segment Routing (SR) technology leverages the source routing paradigm [RFC8660]. A node steers a packet through a controlled set of instructions, called segments, by pre-pending the packet with an SR header. In the SR with MPLS data plane (SR-MPLS), the SR header is instantiated through a label stack.

An example of data packet with SR-MPLS encapsulation containing Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] and E2E IOAM data fields is shown in Figure 6. The PSID allows to identify the path associated with the data traffic being monitored for IOAM on the decapsulating node.

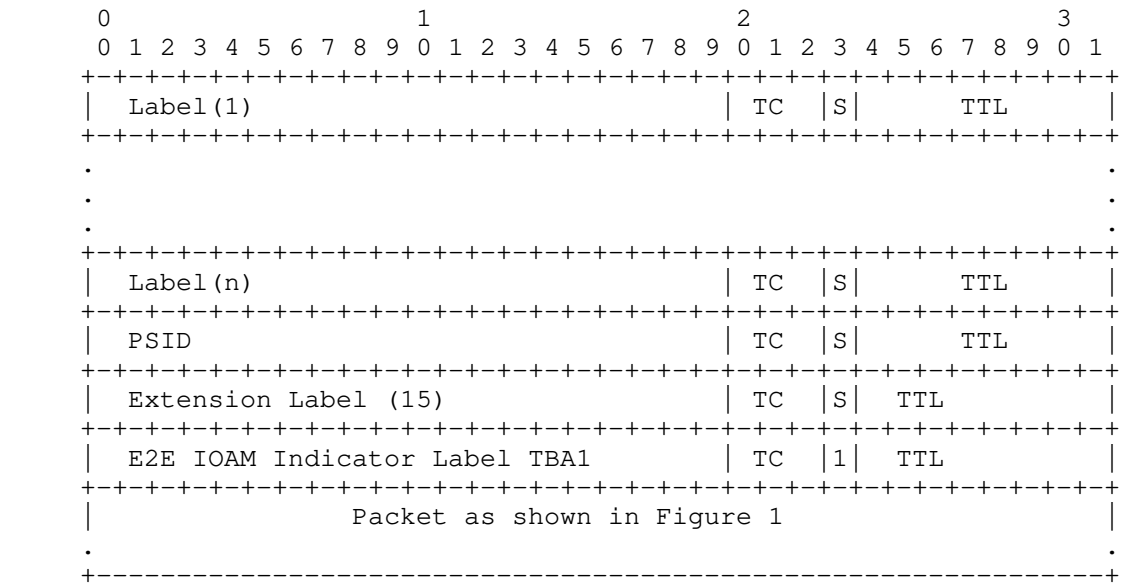

Figure 6: Example SR-MPLS Encapsulation with E2E IOAM Data Fields

9. Security Considerations

The security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

Routers that support G-ACh are subject to the same security considerations as defined in [RFC4385] and [RFC5586].

10. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>). IANA is requested to allocate IOAM Indicator Label value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	E2E IOAM Indicator Label	This document
TBA2	HbH IOAM Indicator Label	This document

Table 1: IOAM Indicator Label Values

IANA maintains G-ACh Type Registry (see <https://www.iana.org/assignments/g-ach-parameters/g-ach-parameters.xhtml>). IANA is requested to allocate a value for IOAM G-ACh Type from "MPLS Generalized Associated Channel (G-ACh) Types (including Pseudowire Associated Channel Types)" registry.

Value	Description	Reference
TBA3	IOAM G-ACh Type	This document

Table 2: IOAM G-ACh Type

11. References

11.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-ietf-ippm-ioam-flags-03 (work in progress), October 2020.

[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>.

[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>.

[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>.

[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>.

11.2. Informative References

[I-D.ietf-mpls-spl-terminology] Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-06 (work in progress), January 2021.

[I-D.ietf-spring-mpls-path-segment] Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-03 (work in progress), September 2020.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

Acknowledgements

The authors would like to thank Patrick Khordoc, Shwetha Bhandari and Vengada Prasad Govindan for the discussions on IOAM. The authors would also like to thank Tarek Saad, Loa Andersson, Greg Mirsky, Stewart Bryant, Xiao Min, and Cheng Li for providing many useful comments. The authors would also like to thank Mach Chen, Andrew Malis, Matthew Bocci, and Nick Delregno for the MPLS-RT reviews.

Contributors

Sagar Soni
Cisco Systems, Inc.

Email: sagsoni@cisco.com

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Voitek Kozak
Comcast

Email: Voitek_Kozak@comcast.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 18 July 2021

K. Kompella
R. Balaji
R. Thomas
Juniper Networks

14 January 2021

Label Distribution Using ARP
draft-kompella-mpls-larp-09

Abstract

This document describes extensions to the Address Resolution Protocol to distribute MPLS labels for IPv4 and IPv6 host addresses. Distribution of labels via ARP enables simple plug-and-play operation of MPLS, which is key to deploying MPLS in data centers and enterprises.

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 18 July 2021.

Table of Contents

1. Introduction

This document describes extensions to the Address Resolution Protocol (ARP) [RFC0826] to advertise label bindings for IP host addresses. While there are well-established protocols, such as LDP, RSVP and BGP, that provide robust mechanisms for label distribution, these protocols tend to be relatively complex, and often require detailed configuration for proper operation. There are situations where a simpler protocol may be more suitable from an operational standpoint. An example is the case where an MPLS Fabric is the underlay technology in a Data Center; here, MPLS tunnels originate from host machines. The host thus needs a mechanism to acquire label bindings to participate in the MPLS Fabric, but in a simple, plug-and-play manner. Existing signaling/routing protocols do not always meet this need. Labeled ARP (L-ARP) is a proposal to fill that gap.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The term "server" will be used in this document to refer to an ARP/L-ARP server; the term "host" will be used to refer to a compute server or other device acting as an ARP/L-ARP client.

1.2. Approach

ARP is a nearly ubiquitous protocol; every device with an Ethernet interface, from hand-helds to hosts, have an implementation of ARP. ARP is plug-and-play; ARP clients do not need configuration to use ARP. That suggests that ARP may be a good fit for devices that want to source and sink MPLS tunnels, but do so in a zero-config, plug-and-play manner, with minimal impact to their code.

The approach taken here is to create a minor variant of the ARP protocol, labeled ARP (L-ARP), which is distinguished by a new hardware type, MPLS-over-Ethernet. Regular (Ethernet) ARP (E-ARP) and L-ARP can coexist; a device, as an ARP client, can choose to send out an E-ARP or an L-ARP request, depending on whether it needs Ethernet or MPLS connectivity. Another device may choose to function as an E-ARP server and/or an L-ARP server, depending on its ability to provide an IP-to-Ethernet and/or IP-to-MPLS mapping.

2. Overview of Ethernet ARP

In the most straightforward mode of operation [RFC0826], ARP queries are sent to resolve "directly connected" IP addresses. The ARP request is broadcast, with the Target Protocol Address field (see Section 5 for a description of the fields in an ARP message) carrying the IP address of another node in the same subnet. All the nodes in the LAN receive this ARP request. All the nodes, except the node that owns the IP address, ignore the ARP request. The IP address owner learns the MAC address of the sender from the Source Hardware Address field in the ARP request, and unicasts an ARP reply to the sender. The ARP reply carries the replying node’s MAC address in the Source Hardware Address field, thus enabling two-way communication between the two nodes.

A variation of this scheme, known as "proxy ARP" [RFC2002], allows a node to respond to an ARP request with its own MAC address, even when the responding node does not own the requested IP address. Generally, the proxy ARP response is generated by routers to attract traffic for prefixes they can forward packets to. This scheme

requires the host to send ARP queries for the IP address the host is trying to reach, rather than the IP address of the router. When there is more than one router connected to a network, proxy ARP enables a host to automatically select an exit router without running any routing protocol to determine IP reachability. Unlike regular ARP, a proxy ARP request can elicit multiple responses, e.g., when more than one router has connectivity to the address being resolved. The sender must be prepared to select one of the responding routers.

Yet another variation of the ARP protocol, called 'Gratuitous ARP' [RFC2002], allows a node to update the ARP cache of other nodes in an unsolicited fashion. Gratuitous ARP is sent as either an ARP request or an ARP reply. In either case, the Source Protocol Address and Target Protocol Address contain the sender's address, and the Source Hardware Address is set to the sender's hardware address. In case of a gratuitous ARP reply, the Target Hardware Address is also set to the sender's address.

3. L-ARP Protocol Operation

The L-ARP protocol builds on the proxy ARP model, and also leverages gratuitous ARP model for asynchronous updates.

In this memo, we will refer to L-ARP clients (that make L-ARP requests) and L-ARP servers (that send L-ARP responses). In Figure 1, H1, H2 and H3 are L-ARP clients, and T1, T2 and T3 are L-ARP servers. T4 is a member of the MPLS Fabric that may not be an L-ARP server. Within the MPLS Fabric, the usual MPLS protocols (IGP, LDP, RSVP-TE) are run. Say H1, H2 and H3 want to establish MPLS tunnels to each other (for example, they are using BGP MPLS VPNs as the overlay virtual network technology). H1 might also want to talk to a member of the MPLS Fabric, say T. Also, the "protocol" addresses in L-ARP requests are either IPv4 or IPv6 addresses; note that while it is common to use Neighbor Discovery (ND) [RFC4861] for "regular" ARP requests when dealing with IPv6 (i.e., to obtain Ethernet MAC addresses corresponding to an IPv6 address), ND is not used when the ARP request is for an MPLS label.

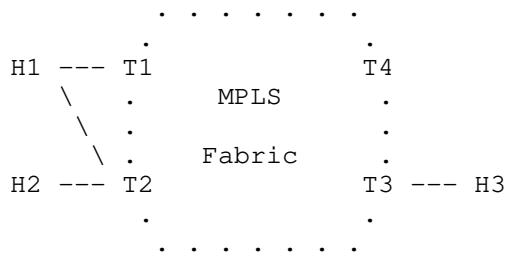

Figure 1: MPLS Fabric

3.1. Setup

In Figure 1, the nodes T1-T4, and those in between making up the "MPLS Fabric" are assumed to be running some protocol whereby they can signal MPLS reachability to themselves and to other nodes (like hosts H1-H3). T1-T3 are L-ARP servers; T4 need not be, since it doesn't have an attached L-ARP client. H1-H3 are L-ARP clients.

3.2. Egress Operation

A node (say T3) that wants an attached node (say H3) to have MPLS reachability allocates a label L3 to reach H3 and advertises this label into the MPLS Fabric. This can be triggered by configuration on T3, or when T3 first receives an L-ARP request from H3 (indicating that H3 wants MPLS connectivity), or via some other protocol. T3 then advertises (H3, L3) to its peers in the MPLS Fabric so that all members of the Fabric have connectivity to H3. This advertisement can be one of the following:

* a "proxy" LDP message (sent on behalf of H3) with prefix H3 and label L3; or

* a node SID advertised on behalf of H3; or

* a labeled BGP advertisement, with prefix H3, label L3 and next hop self.

  On receiving a packet with label L3, T3 pops the label and send the packet to H3. (In the case of labeled BGP, there would be a two-label stack, with outer label to reach T3 and inner label of L3.) This is the usual operation of an MPLS Fabric, with the addition of advertising labels for nodes outside the fabric.

3.3. Ingress Operation

A node (say H1, an L-ARP client) that needs an MPLS tunnel to another node (say H3) identified by a host address (either IPv4 or IPv6) broadcasts over all its interfaces an L-ARP request with the Target Protocol Address set to H3 and Hardware Type set to "MPLS-over-Ethernet". A node receiving the L-ARP request (say T1, an L-ARP server) does the following:

1. checks if it has reachability to H3. If not, it ignores the L-ARP request.

2. if it does, T1 allocates a label TL3 to reach H3 (if it doesn't already have such a label) and installs an L-FIB entry to swap L1 with the label (stack) to reach H3.

3. sends a (proxy) L-ARP reply to H1 with the Source Hardware Address (SHA) set to (L, M), where M is T1's metric to H3. T1 may also set some attribute bits in the SHA.

3.4. Data Plane

To send a packet to H3 over an MPLS tunnel, H1 pushes L1 onto the packet, sets the destination MAC address to M1 and sends it to T1. On receiving this packet, T1 swaps the top label with the label(s) for its MPLS tunnel to H3. If T1's reachability to H3 is via a SPRING label stack, the label L1 acts as an implicit binding SID.

If H1 and H3 have an overlay connection (say an IPVPN [RFC4364] VPN-foo) whereby VM1 on H1 wishes to talk to VM3 on H3 over VPN-foo, H1 does the following:

1. H1 learns information about VPN-foo via BGP (or an SDN controller), including the VPN label VL3 to use to talk to VM3;

2. H1 installs a VRF for VPN-foo, with prefix VM3, label VL3 and next hop H3;

3. H1 binds the local "veth" interface to VM1 to this VRF.

4. When VM1 sends a packet to dest IP address VM3 over its veth interface, H1 looks up VM3 in the corresponding VRF, gets label VL3. It then sends an L-ARP request for next hop H3, and gets TL3.

5. Finally, H1 pushes the label pair (TL3, VL3) onto the packet from VM1 and sends this to T1. This packet will then end up at VM3 on H3.

Note that H1 broadcasts its L-ARP request over its attached interfaces. H1 may receive several L-ARP replies; in that case, H1 can select any subset of these to send MPLS packets destined to H3. As described later, the L-ARP response may contain certain parameters that enable the client to make an informed choice. If the target H3 belongs to one of the subnets that H1 participates in, and H3 is capable of sending L-ARP replies, H1 can use H3's response to send MPLS packets to H3.

4. Attributes

In addition to carrying a label stack to be used in the data plane, an L-ARP reply carries some attributes that are typically used in the control plane. One of these is a metric. The metric is the distance from the L-ARP server to the destination. This allows an L-ARP client that receives multiple responses to decide which ones to use, and whether to load-balance across some of them. The metric typically will be the IGP shortest path distance from server to the destination; this makes comparing metrics from different servers meaningful.

Another attribute is Entropy Label (EL) Capability. This attribute says whether the destination is EL capable (ELC). In Figure 1, if T3 advertises a label to reach H3 and T3 is ELC, T3 can include in its signaling to T1 that it is ELC. In that case, T1's L-ARP reply to H1 can have ELC bit set. This tells H1 that it may include (below the outermost label) an Entropy Label Indicator followed by an Entropy Label. This will help improve load balancing across the MPLS Fabric, and possibly on the last hop to H3.

4.1. Secondary Attributes

Beyond the basic attributes that are carried with every L-ARP request, there more optional attributes, for example, to ask for certain characteristics of the path traffic takes to the destination. These attributes are carried in TLVs that are carried in L-ARP requests and replies.

One such TLV is the "CT" TLV. This TLV allows the L-ARP client to request a label to a destination over a tunnel in the Transport Class given by CT [I-D.kaliraj-idr-bgp-classful-transport-planes]. To satisfy this request, the L-ARP server creates (or finds) a tunnel to the destination that is routed over the CT Transport Plane, allocates a label L, inserts an entry in the LFIB to swap L to the tunnel, and sends L to the L-ARP client in its reply.

5. L-ARP Message Format

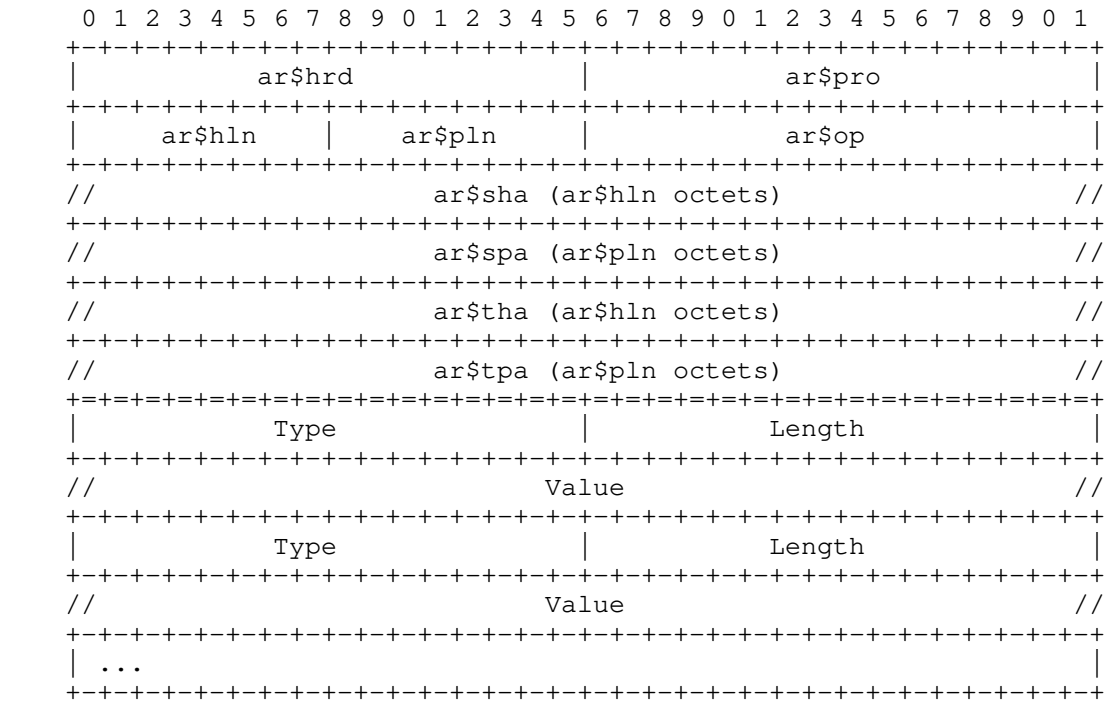

Figure 2: L-ARP Packet Format

ar$hrd: Hardware Type: MPLS-over-Ethernet. The value of the field used here is HTYPE-MPLS. To start with, we will use the experimental value HW_EXP2 (256).

ar$pro: Protocol Type: IPv4/IPv6. The value of the field used here is 0x0800 to resolve an IPv4 address and 0x86DD to resolve an IPv6 address.

ar$hln: Hardware Address Length: 6

ar$pln: Protocol Address Length: for an IPv4 address, the length is 4 octets; for an IPv6 address, it is 16.

ar$op: Operation Code: set to 1 for request, 2 for reply, and 10 for ARP-NAK. Other op codes may be used as needed.

ar$sha: Source Hardware Address: In an L-ARP request, this is

usually all zeros. In an L-ARP reply, Source Hardware Address is the label to reach ar$spa, as specified in Figure 3 below.

ar$spa: Source Protocol Address: In an L-ARP request, this field

carries the sender's IP address. In an L-ARP reply, this field carries the requested IP address (which may not be the sender's IP address).

ar$tha: Target Hardware Address: In an L-ARP message, this is all

zeros.

ar$tpa: Target Protocol Address: In an L-ARP request, this field

carries the IP address for which the client is seeking an MPLS label.

Type: a 2-octet field defining the Type of the TVL

Length: a 2-octet field defining the Length L of the TVL

Value: an L-octet field with the Value of the TLV

5.1. Hardware Address Format

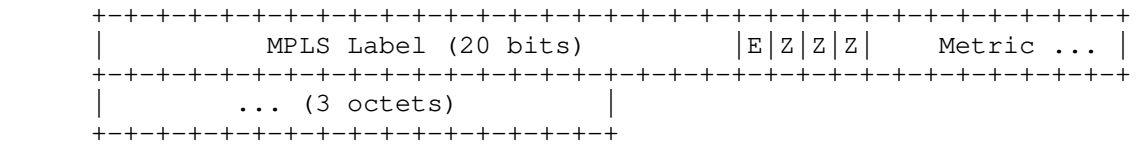

Figure 3: Label Format in L-ARP

MPLS Label:
The 20-bit label

E-bit:
Entropy Label Capable: this flag indicates whether the

corresponding label in the label stack can be followd by an Entropy Label. If this flag is set, the client has the option of inserting ELI and EL as specified in [RFC6790]. The client can choose not to insert ELI/EL pair. If this flag is clear, the client must not insert ELI/EL after the corresponding label.

Z: These bits are not used, and SHOULD be set to zero on sending and
ignored on receipt.

Metric:
The IGP metric to ar$tha from the point of view of the L-ARP

replier.

5.2. CT TLV

The CT TLV has Type (TBD) and Length 4 octets; the Value field consists of the CT attribute.

6. L-ARP Client Server Synchronisation

The information that is communicated in the L-ARP reply can change and hence it is necessary to have both the client and server synchronised with each other. Loss of synchronisation between client and server can have undesirable side effects such as packet drops or packets getting diverted to wrong or suboptimal path. To keep the client cache synchronised with server L-ARP protocol employs two methods

1. Periodic refresh from L-ARP client side: To prevent stale information remaining in the cache indefinitely, L-ARP client should periodically send unicast L-ARP requests and refresh its cache. In the absence of any replies for a configured no of times, L-ARP client should purge the entry from its cache. The server SHOULD send an explicit L-ARP NAK message in reply for such unicast L-ARP requests received if it has no mapping for the requested IP or when the entry is withdrawn.

2. Explicit notifications from server side: On advertised parameter changes, the L-ARP server should broadcast an unsolicited L-ARP reply with the updated parameters. L-ARP client on receipt of the unsolicited reply should update the cache if the entry already exists in its cache. If the entry does not exist client should drop the unsolicited L-ARP packet without any processing.

6.1. L-ARP NAK

On events like label withdrawal the L-ARP server SHOULD notify the clients to invalidate the cache entry by broadcasting an L-ARP NAK message for that label. On receiving the NAK message a node SHOULD delete the cache entry associated with the corresponding label.

6.2. Bulk withdrawal

In cases where the server doesn't store advertised label bindings in a persistent storage, it would be necessary to withdraw all the advertised labels in case of server events like reboot. To handle such scenarios L-ARP provides a bulk withdrawal request of its advertised labels. Bulk withdrawal L-ARP request is made by broadcasting an L-ARP NAK packet with all-zero address in the source protocol address field.

6.3. Garbage Collection Requirements

To limit the storage needed on both server and client side for the L-ARP caches, a node may need to periodically garbage-collect old entries. The client can follow a LRU-based policy to reclaim the entries. On server side the liveness of the entry can be determined by the periodic refreshes from the client and in the absence of any refreshes for a configurable time interval the labels advertised can be reclaimed.

7. Security Considerations

There are many possible attacks on ARP: ARP spoofing, ARP cache poisoning and ARP poison routing, to name a few. These attacks use gratuitous ARP as the underlying mechanism, a mechanism used by L-ARP. Thus, these types of attacks are applicable to L-ARP. Furthermore, ARP does not have built-in security mechanisms; defenses rely on means external to the protocol.

It is well outside the scope of this document to present a general solution to the ARP security problem. One simple answer is to add a TLV that contains a digital signature of the contents of the ARP message. This TLV would be defined for use only in L-ARP messages, although in principle, other ARP messages could use it as well. Such an approach would, of course, need a review and approval by the Security Directorate. If approved, the type of this TLV and its procedures would be defined in this document. If some other technique is suggested, the authors would be happy to include the relevant text in this document, and refer to some other document for the full solution.

8. IANA Considerations

IANA is requested to allocate a new ARP hardware type (from registry hrd) for HTYPE-MPLS.

9. Acknowledgments

Many thanks to Shane Amante for his detailed comments and suggestions. Many thanks to the team in Juniper prototyping this work for their suggestions on making this variant workable in the context of existing ARP implementations. Thanks too to Luyuan Fang, Alex Semenyaka and Dmitry Afanasiev for their comments and encouragement.

10. References

10.1. Normative References

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC2002] Perkins, C., Ed., "IP Mobility Support", RFC 2002, DOI 10.17487/RFC2002, October 1996, <https://www.rfc-editor.org/info/rfc2002>.

[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>.

[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>.

10.2. Informative References

[I-D.kaliraj-idr-bgp-classful-transport-planes] Vairavakkalai, K., Venkataraman, N., and B. Rajagopalan, "BGP Classful Transport Planes", Work in Progress, Internet-Draft, draft-kaliraj-idr-bgp-classful-transport-planes-00, 8 May 2020, <http://www.ietf.org/internet-drafts/draft-kaliraj-idr-bgp-classful-transport-planes-00.txt>.

[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>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale 94089
USA

Phone: +1-408-745-2000
Email: kireeti.kompella@gmail.com

R. Balaji
Juniper Networks
Survey No.111/1 to 115/4, Wing A & B
Bangalore 560103
India

Email: balajir@juniper.net

Reji Thomas
Juniper Networks
Survey No.111/1 to 115/4, Wing A & B
Bangalore 560103
India

Email: rejithomas@juniper.net

MPLS Working Group
Internet-Draft
Updates: 8595 (if approved)
Intended status: Standards Track
Expires: August 25, 2021

Y. Liu
G. Mirsky
ZTE Corporation

February 21, 2021

MPLS-based Service Function Path(SFP) Consistency Verification
draft-lm-mpls-sfc-path-verification-02

Abstract

This document describes extensions to MPLS LSP ping mechanisms to support verification between the control/management plane and the data plane state for SR-MPLS service programming and MPLS-based NSH SFC.

This document defines the signaling of the Generic Associated Channel (G-ACh) over a Service Function Path (SFP) with an MPLS forwarding plane using the basic unit defined in RFC 8595. The document updates RFC 8595 in respect to SFF's handiling TTL expiration. The document also describes the processing of the G-ACh by the elements of the SFP.

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 August 25, 2021.

Copyright Notice

Table of Contents

1. Introduction

Service Function Chain (SFC) defined in [RFC7665] as an ordered set of service functions (SFs) to be applied to packets and/or frames, and/or flows selected as a result of classification.

SFC can be achieved through a variety of encapsulation methods, such as NSH [RFC8300], SR service programming [I-D.ietf-spring-sr-service-programming] and MPLS-based NSH SFC [RFC8595].

This document describes extensions to MPLS LSP ping [RFC8029] mechanisms to support verification between the control/management plane and the data plane state for both SR-MPLS service programming and MPLS-based NSH SFC.

An MPLS LSP ping is a component of the MPLS Operation, Administration, and Maintenance (OAM) toolset. OAM packets used to monitor a specific Service Function Path (SFP) can be transported

over a Generic Associated Channel (G-ACh). This document defines the signaling of the G-ACh over an SFP with an MPLS forwarding plane using the basic unit defined in [RFC8595]. The document updates [RFC8595] in respect to SFF's handiling TTL expiration. The document also describes the processing of the G-ACh by the elements of the SFP.

2. Conventions used in this document

2.1. 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.

2.2. Terminology and Acronyms

SFC: Service Function Chain

SFF: Service Function Forwarder

SF: Service Function

SFI: Instance of an SF

SFP: Service Function Path

RSP: Rendered Service Path

SFC-MPLS: SFC over an MPLS forwarding plane introduced in [RFC8595]

SR-SFC: SFC achieved by SR service programming [I-D.ietf-spring-sr-service-programming]

NSH-SR: SFC based on the integration of Network Service Header (NSH) and SR for SFC [I-D.ietf-spring-nsh-sr]

SPL: Special-Purpose Label

bSPL: Base SPL

eSPL: Extended SPL

GAL: Generic Associated Channel Label

ELI: Entropy Label Indicator

OAM: Operation, Administration, and Maintenance

G-ACh: Generic Associated Channel

GAL: Generic Associated Channel Label

3. MPLS-based SFP Consistency Verification

MPLS echo request and reply messages [RFC8029] can be extended to support the verification of the consistency of an MPLS-based Service Function Path (SFP).

SR-MPLS/MPLS can be used to realize an SFP. Two methods have been defined:

- [I-D.ietf-spring-sr-service-programming] describes how to achieve service function chaining in SR-enabled MPLS and IPv6 networks. In an SR-MPLS network, each SF is associated with an MPLS label. As a result, an SFP can be encoded as a stack of MPLS labels and pushed on top of the packet.

- [RFC8595] provides another method to realize SFC in an MPLS network by means of using a logical representation of the Network Service Header (NSH) in an MPLS label stack. This method, throughout this document, is referred to as SFC over an MPLS data plane (SFC-MPLS). When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used, which can be present one or more times in the label stack. This unit comprises two MPLS labels, one carries a label to provide a context within the SFC scope (the SFC Context Label), and the other carries a label to show which SF is to be enacted (the SF Label). SFC forwarding can be achieved by label swapping, label stacking, or the mix of both. When an SFF receives a packet containing an MPLS label stack, it examines the top basic unit of the MPLS label stack for SFC, {SPI, SI} or {context label, SFI index}, to determine where to send the packet next.

In MPLS Label Switched Paths (LSPs), MPLS LSP ping [RFC8029] is used to check the correctness of the data plane functioning and to verify the data plane against the control plane.

The proposed extension of MPLS LSP ping allows verification of the correlation between the control/management (if data model-based central controller used) plane and the data plane state in SR-MPLS/ MPLS-based SFC.

As for NSH-SR, OAM defined for NSH in [draft-ietf-sfc-multi-layer-oam] can be re-used and it is out of the scope of this document.

4. LSP Ping in SFC-MPLS

In SFC-MPLS, SFFs are responsible for MPLS echo request processing. there're two reasons:

- In SFC-MPLS, the packet forwarding decision is made by SFFs based on the basic unit. SFs are not aware of the FEC of the basic unit.

- Generally, except for the designed specific functions, the packet processing functions supported by SFs are limited. SFs may not support control and/or management protocols operated over the G-ACh defined in [RFC5586], e.g., MPLS OAM protocols like LSP ping. Such packets may be mishandled.

To support that processing, the basic unit can use the mechanism described in Section 4.1.

4.1. Special-purpose Label in SFC-MPLS Environment

When an SFC-MPLS is used, an SFF needs to identify an OAM packet with the SFP scope. To achieve that, this specification first defines the use of a base special-purpose label (bSPL) [RFC3032] or an extended special-purpose label (eSPL) [RFC7274] (referred to in this document as SPL Unit) with the basic unit defined in [RFC8595]. And based on that, the use of Generic Associated Channel Label (GAL) [RFC5586] with the basic unit in the SFC-MPLS environment.

Special-purpose label (SPL), whether bSPL or eSPL, has special significance in the data and control planes. An ability to use an SPL in the basic unit allows for a closer functional match between the NSH-based SFC and SFC-MPLS. For example, Entropy Label Indicator (ELI) [RFC6790] with the basic unit can be used as the Flow ID TLV [I-D.ietf-sfc-nsh-tlv] to allow an SFF to balance SFC flows among SFs of the same type. An SPL MAY be used with the basic unit in SFC-MPLS, as displayed in Figure 1. Note that an SPL unit MAY be present in one or more basic units when MPLS label stacking is used to carry the SFC information.

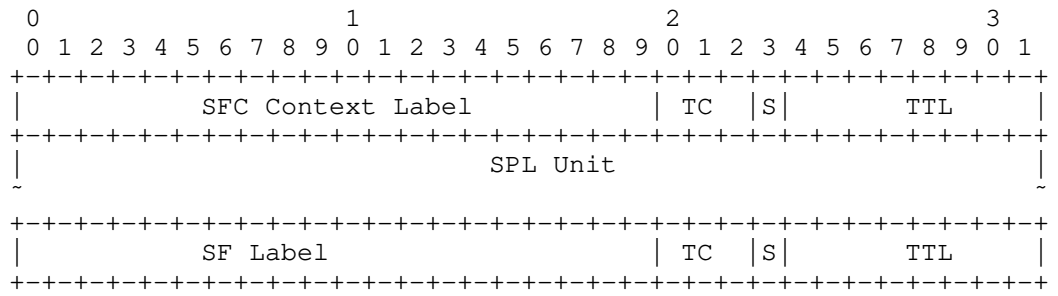

Figure 1: Special-purpose Label Unit with the Basic Unit of MPLS Label Stack for SFC

4.1.1. G-ACh over SFC-MPLS

SFC-MPLS environment could include instances of an SF (SFI) or SFC proxies that cannot properly process control and/or management protocol messages that are exchanged between nodes over the G-ACh associated with the particular SFP. To support OAM over G-ACh, it is beneficial to avoid handing over a test packet to the SFI or SFC proxy. Hence, this specification defines that if the Generic Associated Channel Label (GAL) immediately follows the SFC Context label [RFC8595], then the packet is recognized as an SFP OAM packet.

Below are the processing rules of an SFP OAM packet by an SFF:

- An SFF MUST NOT pass the packet to a local SFI or SFC proxy.
- The SFF MUST decrement SF Label entry's TTL value. If the resulting value equals zero, the SFF MUST pass the SFP OAM packet to the control plane for processing. An implementation that supports this specification MUST provide control to limit the rate of SFP OAM packets passed to the control plane for processing.
- If the TTL value is not zero, the SFP OAM packet is processed as defined in Section 6, Section 7, and Section 8 [RFC8595], according to the type of MPLS forwarding used in the SFP.

4.2. SFC Basic Unit FEC Sub-TLV

Unlike standard MPLS forwarding, based on a single label, packet forwarding defined in [RFC8595] is based on the basic unit of MPLS label stack for SFC(SFC Context Label+SF Label). A new SFC Basic Unit FEC sub-TLV with Type value (TBA1) is defined in this document. The SFC Basic Unit FEC sub-TLV MAY be used to carry the corresponding FEC of the basic unit.

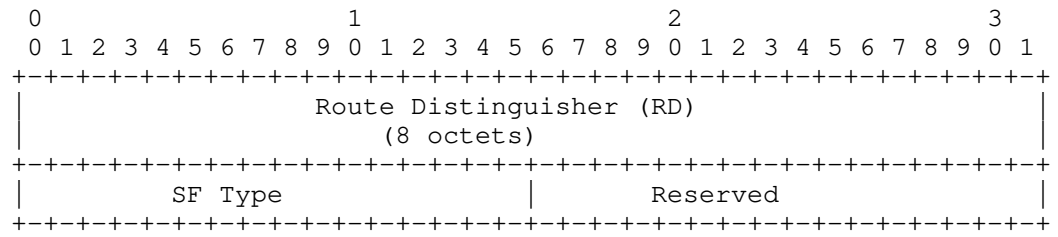

Figure 2: SFC Basic Unit sub-TLV

The format of the basic unit sub-TLV is shown in Figure 2 and includes the following fields:

Route Distinguisher (RD): 8 octets field in SFIR Route Type specific NLRI [I-D.ietf-bess-nsh-bgp-control-plane].

SF Type: 2 octets. It is defined in [I-D.ietf-bess-nsh-bgp-control-plane] and indicates the type of SF, such as DPI, firewall, etc.

Note: [I-D.ietf-bess-nsh-bgp-control-plane] covers the BGP control plane of MPLS-SFC as well.

A node that receives an LSP ping with the Target FEC Stack TLV and the SFC Basic Unit FEC Sub-TLV included will check if it is its Route Distinguisher and whether it advertised that Service Function Type. If the validation is not passed, the SFF will generate an MPLS echo reply with an error code as defined in [RFC8029].

4.3. SFC Basic Unit Nil FEC Sub-TLV

[RFC8029] is based on the premise that one label corresponds to one FEC sub-TLV. For example, in [RFC8029] section 4.4 step 4, before the FEC validation process of an intermediate node first the node should determine FEC-stack-depth from the Downstream Detailed Mapping TLV, and then if the number of FECs in the FEC stack is greater than or equal to FEC-stack-depth, FEC validation is triggered.

In SFC-MPLS OAM, since one basic unit is related to only one FEC sub-TLV, there may be situations that the label stack in Downstream Detailed Mapping TLV contains two labels, but there is only one FEC in the FEC stack.

The SFC Basic Unit Nil Sub-TLV(TBA2) is introduced in this document to ensure that the proper validation can still be performed.

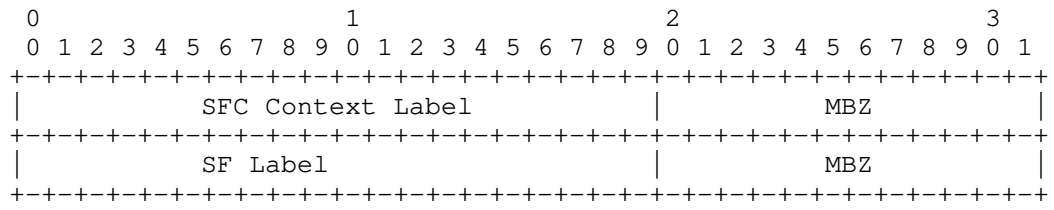

Figure 3: SFC Basic Unit Nil sub-TLV

SFC Context Label and SF Label are the actual label values inserted in the label stack; the MBZ fields MUST be zero when sent and ignored on receipt.

The SFC Basic Unit Nil sub-TLV, when present, MUST be immediately followed by an SFC Basic Unit sub-TLV. During FEC validation, an SFF should skip the SFC Basic Unit Nil sub-TLV and use the following SFC Basic Unit sub-TLV to validate the FEC of the basic unit.

4.4. Theory of Operation

An MPLS SFC validation request is an MPLS echo request with an SFC validation TLV, and the echo request is sent with a label stack corresponding to the SFP being tested. To trace SFC-MPLS, the Generic Associated Channel Label (GAL), which immediately follows the SFC Context label is also included.

If FEC validation is required, the SFC Basic Unit sub-TLV SHOULD be carried in the FEC stack of the request packet, and the SFC Basic Unit Nil sub-TLV MAY also be carried. A Downstream Detailed Mapping TLV MAY be included in the MPLS echo request of the SFP.

Sending an SFC echo request to the control plane is triggered by one of the following packet processing exceptions: IP TTL expiration, MPLS TTL expiration, or the receiver is SFP's egress SFF.

As described in Section 4.1.1, the packet with GAL is recognized by the SFF as an SFP OAM packet. The SFF then decrements the SF Label entry's TTL value. If the resulting value equals zero, the SFF passes the SFP OAM packet to the control plane for processing. The system that supports this specification then generates a reply message.

In "traceroute" mode the TTL of the SF Label is set successively to 1, 2, and so on. After all SFFs on the SFP send back MPLS echo reply, the sender collects information about all traversed SFFs and

SFs on the rendered service path (RSP). But the TTL processing in SR-MPLS is defined in Section 6 of [RFC8595], as follows:

> If an SFF decrements the TTL to zero, it MUST NOT send the packet and MUST discard the packet

and it excludes TTL expiration as the exception mechanism. As a result, tracing a path of an SFC-MPLS-based service chain is problematic. To support the tracing of an SFC, it must be changed to allow punting an OAM packet to the control plane though under throttling control. Hence, this document updates Section 6 of [RFC8595] to state that:

> If an SFF decrements the TTL to zero, an OAM packet MAY be sent to the control plane given it does not exceed the configured rate intended to protect the system from the possible denial-of-service attack.

5. LSP Ping in SR-SFC

In SR service programming, the packet forwarding decision is made based on every single SID/label. The SR proxy SHOULD process the OAM packet for the SF when the SF is not capable of doing so.

If only the SFP connectivity check is required, the current LSP Ping for SR-MPLS [RFC8287] is sufficient.

If operators want to check more information about the SFP(service segment related SF type, SR proxy type, etc.), new FEC sub-TLVs for the service segment should be defined. Details of the new FEC sub-TLVs will be added in the further version.

6. Security Considerations

This specification defines the processing of an SFP OAM packet. Such packets could be used as an attack vector. A system that supports this specification MUST provide control to limit the rate of SFP OAM packets sent to the control plane for processing.

This document defines additional MPLS LSP Ping sub-TLVs and follows the mechanisms defined in [RFC8029]. All the security considerations defined in [RFC8029] will be applicable for this document.

7. IANA Considerations

This document requests assigning two new sub-TLVs from the "sub-TLVs for TLV Types 1, 16, and 21" sub-registry of the "Multi-Protocol

Label Switching(MPLS) Label Switched Paths (LSPs) Ping Parameters" registry according to Table 1

Value	Description	Reference
TBA1	SFC Basic Unit	This document
TBA2	SFC Basic Unit Nil	This document

Table 1: Sub-TLV Values

8. References

8.1. Normative References

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", draft-ietf-bess-nsh-bgp-control-plane-18 (work in progress), August 2020.

[I-D.ietf-spring-nsh-sr]
Guichard, J. and J. Tantsura, "Integration of Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", draft-ietf-spring-nsh-sr-04 (work in progress), December 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[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>.

[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>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[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>.

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

[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>.

[RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <https://www.rfc-editor.org/info/rfc8595>.

8.2. Informative References

[I-D.ietf-sfc-nsh-tlv] Wei, Y., Elzur, U., Majee, S., and C. Pignataro, "Network Service Header Metadata Type 2 Variable-Length Context Headers", draft-ietf-sfc-nsh-tlv-04 (work in progress), January 2021.

[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>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Liu Yao
ZTE Corporation
Nanjing
China

Email: liu.yao71@zte.com.cn

Greg Mirsky
ZTE Corporation

Email: gregory.mirsky@ztetx.com