

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

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions . . . . .	3
2.1. Requirement Language . . . . .	3
2.2. Abbreviations . . . . .	3
3. MPLS Extensions for IOAM Data Fields . . . . .	4
3.1. IOAM Generic Associated Channel . . . . .	4
3.2. IOAM Indicator Labels . . . . .	5
4. Edge-to-Edge IOAM . . . . .	5
4.1. Edge-to-Edge IOAM Indicator Label . . . . .	5
4.2. Procedure for Edge-to-Edge IOAM . . . . .	6
4.3. Edge-to-Edge IOAM Indicator Label Allocation . . . . .	7
5. Hop-by-Hop IOAM . . . . .	7
5.1. Hop-by-Hop IOAM Indicator Label . . . . .	7
5.2. Procedure for Hop-by-Hop IOAM . . . . .	8
5.3. Hop-by-Hop IOAM Indicator Label Allocation . . . . .	8
6. Considerations for IOAM Indicator Label . . . . .	9
6.1. Considerations for ECMP . . . . .	9
6.2. Node Capability . . . . .	9
6.3. MSD Considerations . . . . .	9
6.4. Nested MPLS Encapsulation . . . . .	10
7. MPLS Encapsulation with Control Word and Another G-ACh for IOAM Data Fields . . . . .	10
8. Example MPLS Encapsulations . . . . .	12
8.1. Example SR-MPLS Encapsulation with IOAM . . . . .	12
9. Security Considerations . . . . .	13
10. IANA Considerations . . . . .	13
11. References . . . . .	14
11.1. Normative References . . . . .	14
11.2. Informative References . . . . .	15
Acknowledgements . . . . .	16
Contributors . . . . .	16
Authors' Addresses . . . . .	16

## 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.

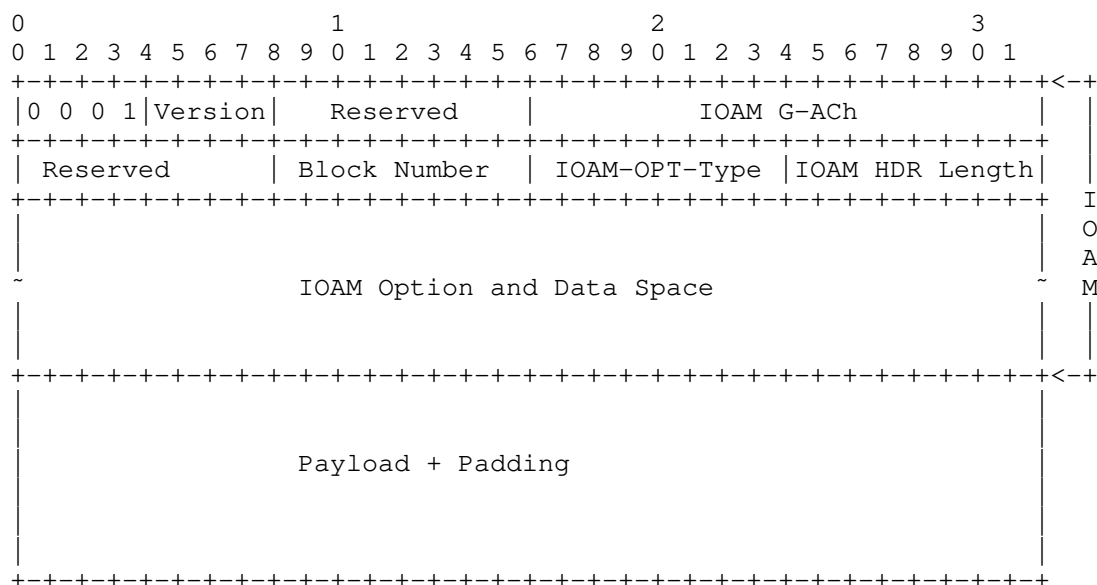


Figure 1: IOAM Generic Associated Channel with IOAM Data Fields

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.

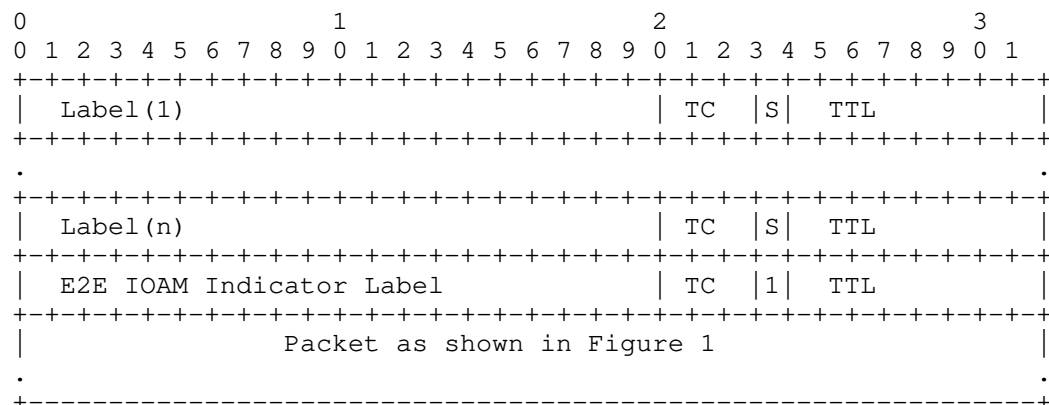


Figure 2: MPLS Encapsulation for E2E IOAM

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:

- o The encapsulating node inserts the E2E IOAM Indicator Label and one or more IOAM data fields in the MPLS header.
- o The intermediate nodes do not process IOAM data fields.
- o 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.
- o 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.
- o 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:

- o Label assigned by IANA with value TBA1 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].
- o Label allocated by a Controller from the global table of the decapsulating node. The Controller provisions the label on both encapsulating and decapsulating nodes.
- o 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.

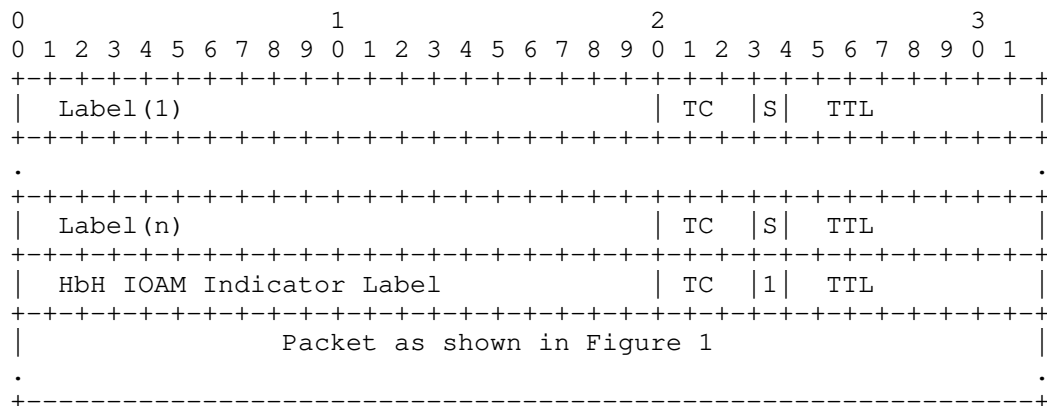


Figure 3: MPLS Encapsulation for HbH IOAM

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:

- o The encapsulating node inserts the HbH IOAM Indicator Label and one or more IOAM data fields in the MPLS header.
- o 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.
- o The intermediate node forwards a copy of the processed data packet downstream.
- o 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.
- o 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.
- o 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:

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



- o 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.
- o 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.

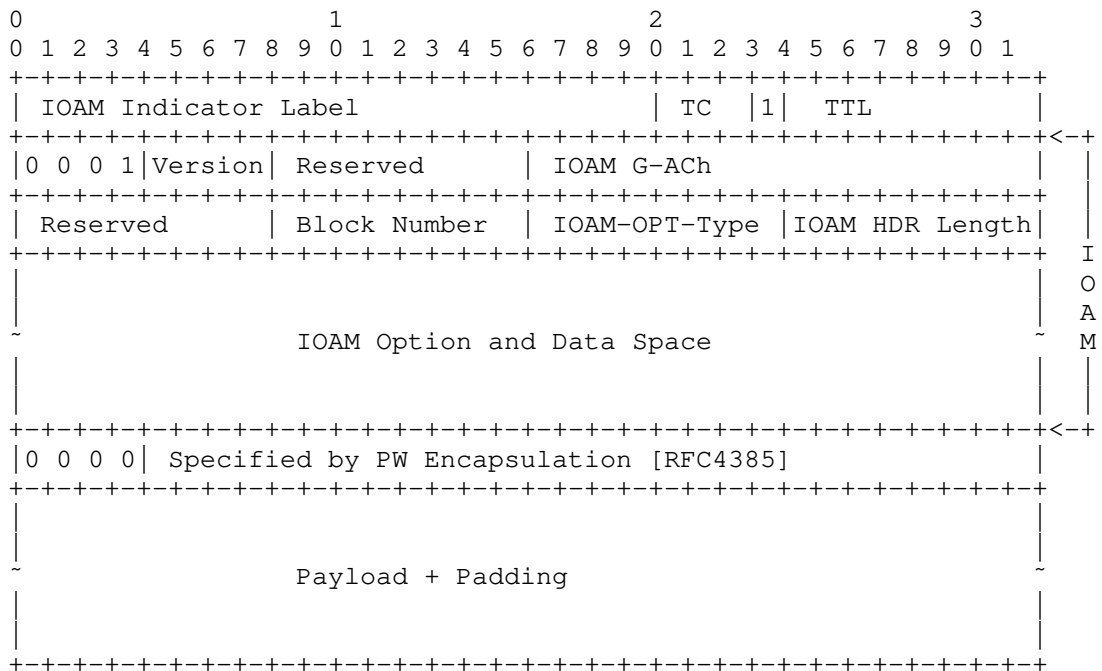


Figure 4: Example MPLS Encapsulation with Generic PW Control Word with IOAM

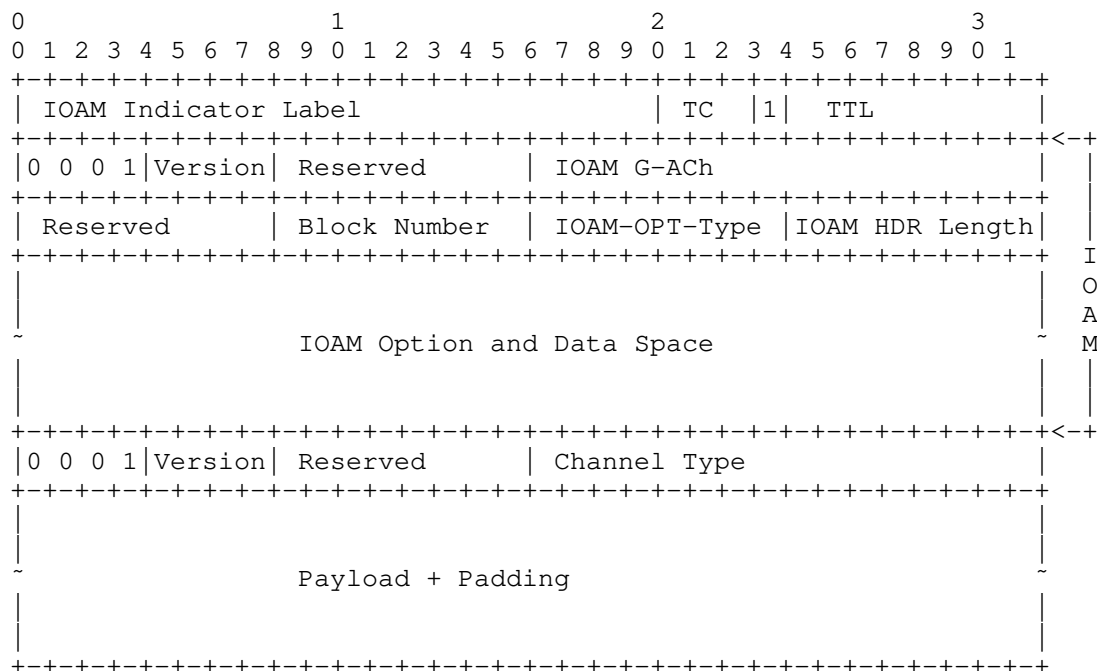


Figure 5: Example MPLS Encapsulation with Another G-ACh with IOAM

## 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.

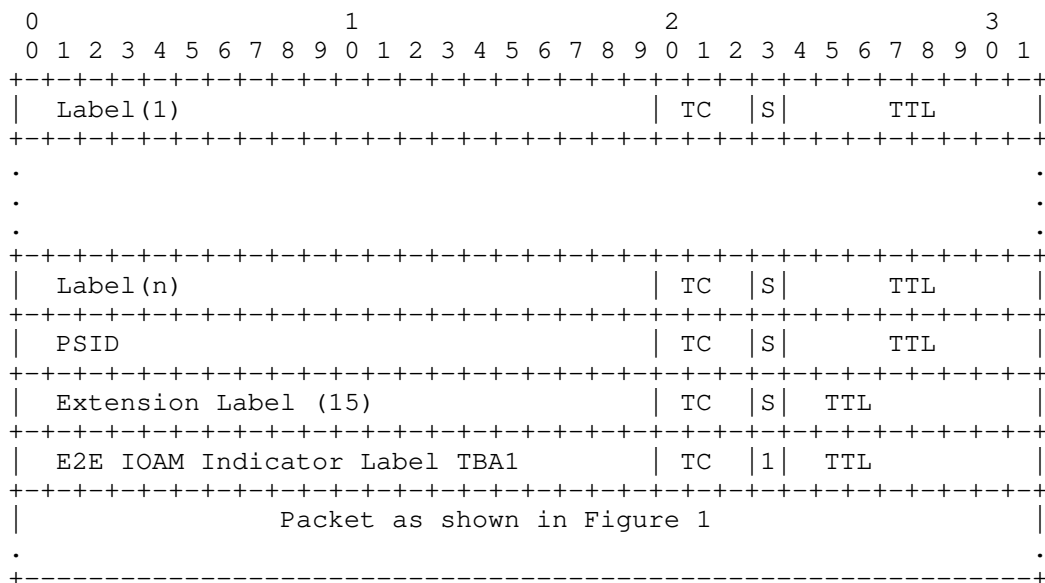


Figure 6: Example SR-MPLS Encapsulation with E2E IOAM Data Fields

## 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 WG  
Internet-Draft  
Intended status: Standards Track  
Expires: August 26, 2021

K. Kompella  
V. Beeram  
T. Saad  
Juniper Networks  
I. Meilik  
Broadcom  
February 22, 2021

Multi-purpose Special Purpose Label for Forwarding Actions  
draft-kompella-mpls-mspl4fa-00

Abstract

A Slice Selector is packet metadata that dictates the packet's forwarding handling in order to conform to its slice requirements. There are multiple proposals for carrying slice selectors in MPLS networks. One of the more practical proposals is the "Global Identifier for Slice Selector" (GISS). Global uniqueness requires the GISS label be identified as such, via a special purpose label (ideally a base special purpose label (bSPL)). However, bSPLs are a precious commodity, and there are many requests for them. This document serves two purposes: to define a bSPL for carrying a GISS, and to show how this bSPL can consolidate many current requests for special purpose labels while carrying associated data compactly and efficiently.

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 26, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	3
1.2. Slice Selector . . . . .	3
2. Multi-purpose bSPL: the Forwarding Actions Indicator . . . . .	3
2.1. The FAI bSPL . . . . .	4
3. Issues to be Resolved . . . . .	7
3.1. Preventing FAI From Reaching Top of Stack . . . . .	7
3.2. Repeating the FAI at "Readable Stack Depth" . . . . .	8
3.3. First Nibble Issues . . . . .	8
4. Contributors . . . . .	8
5. Acknowledgments . . . . .	8
6. IANA Considerations . . . . .	8
7. Security Considerations . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

Network slicing is an important ongoing effort both for network design, as well as for standardization, in particular at the IETF [I-D.nsd-t-teas-ns-framework]. A key issue is identifying which slice a packet belongs to, by means of a "slice selector" carried in the packet header. [I-D.bestbar-teas-ns-packet] describes several such methods for MPLS networks, of which the Global Identifier for Slice Selector (GISS) is one of the more practical solutions. This document shows how to realize the GISS using a base special purpose label (bSPL).

Base Special Purpose Labels are a precious commodity; there are only 16 such values, of which 8 have already been allocated. There are currently five requests for bSPLs that the authors are aware of; this document proposes another use case for a bSPL, in all consuming nearly all the remaining values. Therefore, this document also suggests a method whereby a single bSPL can be used for all the purposes currently documented. This leads to perhaps the more valuable long-term contribution of this document: an approach to the definition and use of bSPLs (and SPLs in general) whereby a single value can be used for multiple purposes, and provide a flexible and efficient means of carrying associated data.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 1.2. Slice Selector

In MPLS networks, a GISS is a data plane construct identifying packets belonging to a slice aggregate (the set of packets that belong to the slice). The GISS dictates forwarding actions for the slice aggregate: QoS behavior and next hop selection. The purpose of the GISS is detailed in [I-D.bestbar-teas-ns-packet]. To embed a GISS in a label stack, one must preface it with a bSPL identifying it as such. For reasons that will become apparent, this bSPL is called the Forwarding Actions Indicator (FAI).

## 2. Multi-purpose bSPL: the Forwarding Actions Indicator

This document proposes the use of a single bSPL to tell routers one or more forwarding actions they should take on a packet, e.g.:

- o to treat a packet according to its slice, given its GISS;
- o to load balance a packet, given its entropy;
- o whether or not to perform fast reroute on a failure [I-D.kompella-mpls-nffrr];
- o whether or not the packet has a Flow ID;
- o to update statistics based on the path identifier [I-D.hegde-spring-traffic-accounting-for-sr-paths];

- o to view/update OAM metadata;  
[I-D.cheng-mpls-inband-pm-encapsulation],  
[I-D.gandhi-mpls-ioam-sr], other approaches.
- o to reassemble a fragmented packet  
[I-D.zzhang-tsvwg-generic-transport-functions];
- o and perhaps other functions in the future.

This bSPL is called the "Forwarding Actions Indicator" (FAI). The FAI uses the label's TC bits and TTL field to inform the forwarding plane of the required actions. Each of these actions may have associated data, the Forwarding Actions Data (FAD). The FAD may be carried in the Label Stack (LS FAD) or in the payload (PL FAD).

#### 2.1. The FAI bSPL

The design of the bSPL hinges on a key insight: for labels not at the top of the label stack, the only bits that a forwarding engine looks at are the label value field (to compute entropy and identify SPLs) and the End of Stack (S) bit (to know when the label stack ends). [If you know of a forwarding engine that looks at other bits of labels below the top of stack, please contact the authors.] This means that for a bSPL that will never appear at the top of stack, the TC bits and the TTL bits can be used to carry additional information. Furthermore, for the LS FAD, the entire 4-octet label word, the S bit excepted, can be used to carry data. We use this technique to make the FAI bSPL multipurpose, and to make the FAD words compact and efficient.

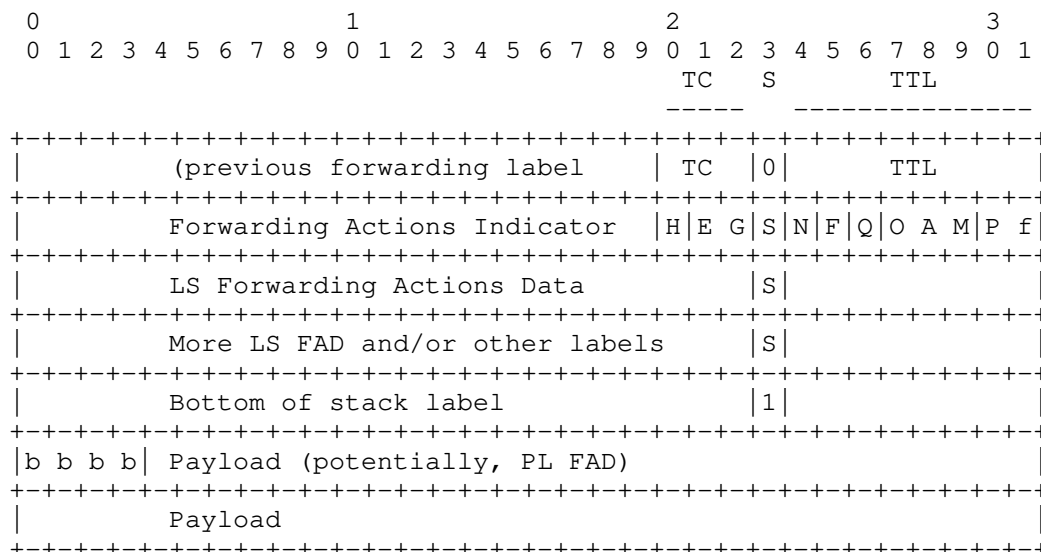


Figure 1: Format for FAI, LS FAD and PL FAD

The FAI's label value MUST be the IANA allocated value. The S bit MUST be reflect whether the label stack ends at this label or not.

The TC and TTL bits are used as flags, defined as follows:

H: if set, the FAI is followed by a Forwarding Actions Header (FAH).

EG: this is a 2-bit flag indicating whether the LS FAD carries Entropy and/or GISS information.

S: MUST be set if the FAI is the end of stack, and clear otherwise.

N: If set, do not do fast reroute (NFFRR).

F: If set, the LS has a Flow ID.

Q: if set, the payload contains Opaque data.

OAM: a 3-bit field that specifies what type(s) of OAM is carried in the in the label stack and/or payload.

P: If set, the PL FAD contains a Path Identifier.

F: If set, the payload contains a Fragmentation Header.

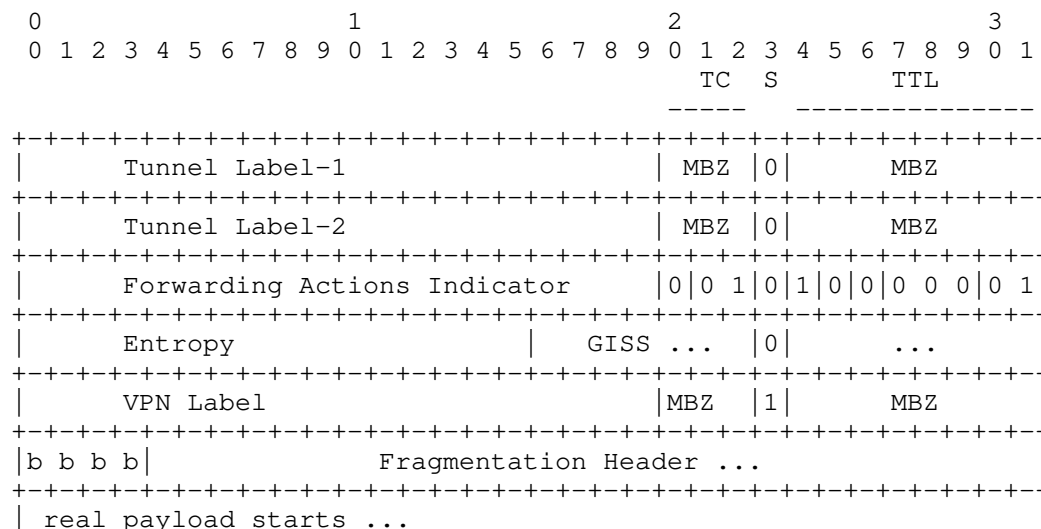
The EG field is used as follows:

- 00: No Entropy or GISS present
- 01: LS FAD 0 contains 16 bits of Entropy in the high order 16 bits and 15 bits of GISS in the low order 16 bits (S bit excepted).
- 10: LS FAD 0 contains 20 bits of Entropy in the high order 20 bits and 11 bits of GISS in the low order 12 bits (S bit excepted).
- 11: LS FAD 0 contains the 31-bit Entropy; LS FAD 1 contains the 31-bit GISS. In LS FAD 0, the S bit MUST be 0; the packet forwarding engine may choose to use this as part of the Entropy, as it doesn't affect the outcome. In LS FAD 1, the S bit may be 0 or 1.

Here's how the LS FAD is parsed. One must keep track of the S bit to know when the stack ends. It is an error if the label stack ends while there are more PL FAD words to process.

1. Set NL ("next label") to the first 4-octet word of the LS FAD. Set PL ("payload") to the first 4-octet word of the payload.
2. Process H: if set, (TBD); otherwise, NL is unchanged.
3. Process EG:
  1. If EG is 00, NL is unchanged.
  2. If EG is 01 or 10, NL contains both GISS and Entropy. Increment NL.
  3. If EG is 11, NL contains GISS; NL+1 contains Entropy. Increment NL by 2.
4. Process N. NL is unchanged.
5. Process F:
  1. If F is set, NL contains the Flow ID; increment NL.
6. Process Q:
  1. If set, (TBD); otherwise, NL is unchanged.
7. NL now points at next label in the stack.

A similar procedure applies to parsing the PL; details will be forthcoming when the OAM field is better defined.



H = 0; ignore.  
 EG = 01: LS FAD 0 contains Entropy + GISS.  
 N = 1: NFFRR is set.  
 F = 0: No Flow ID.  
 Q = 0: ignore.  
 OAM = 0; ignore.  
 P = 0: no Path Identifier in payload.  
 F = 1: Fragmentation Header is present.

Figure 2: Example of FAI + LS FAD + PL FAD

The real payload starts after the Fragmentation Header.

### 3. Issues to be Resolved

#### 3.1. Preventing FAI From Reaching Top of Stack

As was said earlier, the FAI MUST NOT be at the top of stack, since its TC and TTL bits have been repurposed. There are two ways to prevent this. If an LSR X pops a label and encounters an FAI, X can pop the FAI and all LS FAD words. To do that, it must be able to parse the FAI to determine how many LS FAD words there exist. This can be used in conjunction with Section 3.2. However, there are cases when it is desired to preserve the FAI+FAD until the egress. In this case, X should push an explicit NULL (label value 0 or 2) onto the stack above the FAI, with the correct TC and TTL values.

Other options will be pursued.



### 3.2. Repeating the FAI at "Readable Stack Depth"

For LSRs which cannot parse the entire label stack, or would prefer not to unless needed, it is possible to repeat the FAI at "readable stack depth", say every 4 labels. In the above case, the FAI+LS FAD can be repeated every 4 labels; or a truncated version, just the FAD with GE set to 00 can be used. Only the last FAI would contain the full information, reducing the burden on the label stack. However, in this case, LSRs that don't process the whole stack may not load balance less effectively, and potentially not adhere to the slice service level objectives.

Other options will be described in future versions of this document.

### 3.3. First Nibble Issues

The first nibble of the first word of the payload SHOULD NOT be 0x4 or 0x6, as legacy LSRs may use the heuristic that this indicates a payload of IPv4/IPv6. iOAM data has a first nibble of 0x1. However, if there is no iOAM data, the first nibble of the Path Identifier, if any, else that of the Fragmentation Header, MUST NOT be 0x4 or 0x6. However, it is inefficient to have to address this issue for every type of PL FAD, as it may be the first word in the payload. A future version of this document will propose an alternative solution.

It is unclear when a Control Word may be present as the first word of the payload; this is sometimes signaled and sometimes configured. When it is present, the above issue is moot.

## 4. Contributors

Many thanks to Colby Barth, Chandra Ramachandran and Srihari Sangli for their contributions to this draft.

## 5. Acknowledgments

We'd like to acknowledge the helpful discussions with Swamy SRK.

## 6. IANA Considerations

If this draft is deemed useful and adopted as a WG document, the authors request the allocation of a bSPL for the FAI. We suggest the early allocation of label 8 for this.

## 7. Security Considerations

A malicious or compromised LSR can insert the FAI and associated data into a label stack, preventing (for example) FRR from occurring. If so, protection will not kick in for failures that could have been protected, and there will be unnecessary packet loss. Similarly, inserting or removing a Fragmentation Header means that a packet's contents cannot be accurately reconstructed. Inserting or changing a GISS means that the packet will be misclassified, perhaps leaving or entering a high-value slice and causing damage.

## 8. References

### 8.1. Normative References

[I-D.bestbar-teas-ns-packet]

Saad, T., Beeram, V., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., and X. Liu, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-01 (work in progress), December 2020.

[I-D.cheng-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-cheng-mpls-inband-pm-encapsulation-04 (work in progress), September 2020.

[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-05 (work in progress), January 2021.

[I-D.hegde-spring-traffic-accounting-for-sr-paths]

Hegde, S., "Traffic Accounting for MPLS Segment Routing Paths", draft-hegde-spring-traffic-accounting-for-sr-paths-02 (work in progress), October 2018.

[I-D.kompella-mpls-nffrr]

Kompella, K. and W. Lin, "No Further Fast Reroute", draft-kompella-mpls-nffrr-01 (work in progress), November 2020.

[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.

- [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>>.
- [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>>.

## 8.2. Informative References

- [I-D.nsdtd-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

### Authors' Addresses

Kireeti Kompella  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [kireeti.ietf@gmail.com](mailto:kireeti.ietf@gmail.com)

Vishnu Pavan Beeram  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Tarek Saad  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Israel Meilik  
Broadcom

Email: [israel.meilik@broadcom.com](mailto:israel.meilik@broadcom.com)

MPLS WG  
Internet-Draft  
Intended status: Standards Track  
Expires: 12 August 2022

K. Kompella  
V.P. Beeram  
T. Saad  
Juniper Networks  
I. Meilik  
Broadcom  
8 February 2022

Multi-purpose Special Purpose Label for Forwarding Actions  
draft-kompella-mpls-mspl4fa-02

Abstract

The MPLS architecture introduced Special Purpose Labels (SPLs) to indicate special forwarding actions and offered a few simple examples, such as Router Alert. In the two decades since the original architecture was crafted, the range, complexity and sheer number of such actions has grown; in addition, there now is need for "associated data" for some of the forwarding actions. Likewise, the capabilities and scale of forwarding engines has also improved vastly over the same time period. There is a pressing need to match the needs with the capabilities to deliver the next generation of MPLS architecture.

In this memo, we propose an alternate mechanism whereby a single SPL can encode multiple forwarding actions and carry associated data, some in the label stack and some after the label stack. This proposal also solves the problem of scarcity of base SPLs.

This approach can immediately address several use cases:

- \* to carry a Slice Selector for IETF network slicing;
- \* to signal that further fast reroute may have harmful consequences;
- \* to indicate that there is relevant data after the label stack;
- \* among others.

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 12 August 2022.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

1. Introduction . . . . .	3
1.1. Conventions and Definitions . . . . .	3
1.2. Revision History . . . . .	3
1.2.1. Changes from -00 to -01 . . . . .	4
1.2.2. Changes from -01 to -02 . . . . .	4
1.3. Slice Selector . . . . .	5
2. Multi-purpose bSPL: the Forwarding Actions Indicator . . . . .	5
2.1. The FAI bSPL . . . . .	6
2.1.1. ISD vs PSD . . . . .	6
2.2. Format of the FAI bSPL . . . . .	6
2.2.1. Definitions of the FAI Flag Bits . . . . .	7
2.2.2. Processing the FAI Flags and the ISD . . . . .	9
2.2.3. Example of the FAI . . . . .	9
3. Issues to be Resolved . . . . .	10
3.1. Preventing FAI From Reaching Top of Stack . . . . .	10
3.2. Repeating the FAI at "Readable Stack Depth" . . . . .	11
3.3. PSD . . . . .	11
4. Contributors . . . . .	11
5. Acknowledgments . . . . .	12
6. IANA Considerations . . . . .	12

7. Security Considerations . . . . .	12
8. References . . . . .	12
8.1. Normative References . . . . .	12
8.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

Base Special Purpose Labels (bSPLs) are a precious commodity; there are only 16 such values, of which 8 have already been allocated. There are currently five requests for bSPLs that the authors are aware of; this document proposes another use case for a bSPL, in all consuming nearly all the remaining values. This document suggests a method whereby a single bSPL can be used for all the purposes currently requested. This leads to perhaps the more valuable long-term contribution of this document: an approach to the definition and use of bSPLs (and SPLs in general) whereby a single value can be used for multiple purposes, and provide a flexible yet efficient means of carrying associated data.

### 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

FAI: Forwarding Actions Indicator

FFB: Forwarding Flags Block

ISD: In-Stack Data

sISD: Standard ISD

uISD: User-Defined ISD

PSD: Post-Stack Data

SPL: Special-purpose label

bSPL: Base special-purpose label

### 1.2. Revision History

This section (to be removed before publication) offers highlights from the draft's revision history.

## 1.2.1. Changes from -00 to -01

1. This section added.
2. Added a section discussing when data should be put in the LS FAD vs in the PL FAD.
3. Tweaked the bits in the FAI. Added a field "edist".
4. Elaborated on the use of the H bit and the FAH data.
5. Updated the processing of the LS FAD.
6. Added processing of edist.
7. Updated the FAI example.
8. Updated the Issues section.

## 1.2.2. Changes from -01 to -02

1. Updated Abstract and Introduction to focus on FAI; moved description of use cases to separate section.
2. Added terminology.
3. Changed terminology: LS FAD and PL FAD to ISD and PSD, respectively.
4. Updated text on criteria for putting associated data in ISD.
5. Introduced the terms FAI Block, FFB Block, sISD Block and uISD Block. Introduced an "end of block" bit, s. Updated flag bits; updated processing of ISD.
6. Removed field edist.
7. Updated the section on preventing the FAI from reaching the Top of Stack.
8. Updated the section on Readable Stack Depth



### 1.3. Slice Selector

Network slicing is an important ongoing effort both for network design, as well as for standardization, in particular at the IETF [I-D.nsdtd-teas-ns-framework]. A key issue is identifying which slice a packet belongs to, by means of a "slice selector" carried in the packet header. [I-D.bestbar-teas-ns-packet] describes several such methods for MPLS networks, of which the Global Identifier for Slice Selector (GISS) is one of the more practical solutions. This document shows how to realize the GISS using a base special purpose label (bSPL).

In MPLS networks, a GISS is a data plane construct identifying packets belonging to a slice aggregate (the set of packets that belong to the slice). The GISS dictates forwarding actions for the slice aggregate: QoS behavior and next hop selection. The purpose of the GISS is detailed in [I-D.bestbar-teas-ns-packet]. To embed a GISS in a label stack, one must preface it with a bSPL identifying it as such. For reasons that will become apparent, this bSPL is called the Forwarding Actions Indicator (FAI).

## 2. Multi-purpose bSPL: the Forwarding Actions Indicator

This document proposes the use of a single bSPL to tell routers one or more forwarding actions they should take on a packet, e.g.:

- \* to treat a packet according to its slice, given its GISS;
- \* to load balance a packet, given its entropy;
- \* whether or not to perform fast reroute on a failure [I-D.kompella-mpls-nffrr];
- \* whether or not a packet has metadata relevant to intermediate hops along the path;
- \* and perhaps other functions in the future.

This bSPL is called the "Forwarding Actions Indicator" (FAI). There are other suggestions for this name, including "Network Functions Indicator" and "Network Actions Indicator". We'll let WG consensus determine the final choice of name, but for now, we'll continue to use FAI.

The FAI uses the label's TC bits and TTL field to inform the forwarding plane of the required actions. Each of these actions may have associated data. This data may be carried in the label stack as "In-Stack Data" (ISD) or after the label stack as "Post-Stack Data" (PSD).

## 2.1. The FAI bSPL

The design of the bSPL hinges on two key insights: forwarding engines do not interpret the TC bits or the TTL field for labels that are not at the top of the label stack (ToS); nor do they do so for SPLs. For non-ToS labels, the important bit fields are the label value field (to compute entropy and identify SPLs) and the End of Stack (S) bit (to know when the label stack ends). [If you know of a forwarding engine that looks at other bit fields of labels below the ToS, please contact the authors.] This means that for a bSPL that will never appear at the ToS, the TC bits and the TTL bits can be used to carry additional information. Furthermore, for the ISD, the entire 4-octet label word, the S bit excepted, can be used to carry data. We use this technique to make the FAI bSPL multipurpose, and to make the ISD words compact and efficient.

### 2.1.1. ISD vs PSD

A pertinent question is when one should put data in the ISD versus in the PSD. One alternative is to put all such data in the PSD. However, this would mean that accessing such information would require finding the End of Stack, and parsing the PSD. For certain types of data, this would be a severe burden on the packet forwarding engine. Examples of such data are the Entropy label (needed for efficient load balancing) and the GISS (needed for accurate packet forwarding). Having any of this data in the PSD would hurt forwarding performance.

This memo suggests that data that is required for accurate and optimal forwarding should be put in the ISD, and data that is optional from a forwarding point of view should be put in the PSD. Furthermore, each flag bit should have no more than one word of associated ISD. The EG flag can thus have up to 2 words of associated data.

By the above criteria, this memo suggests that in-situ OAM data and the Flow ID be carried in the PSD.

## 2.2. Format of the FAI bSPL

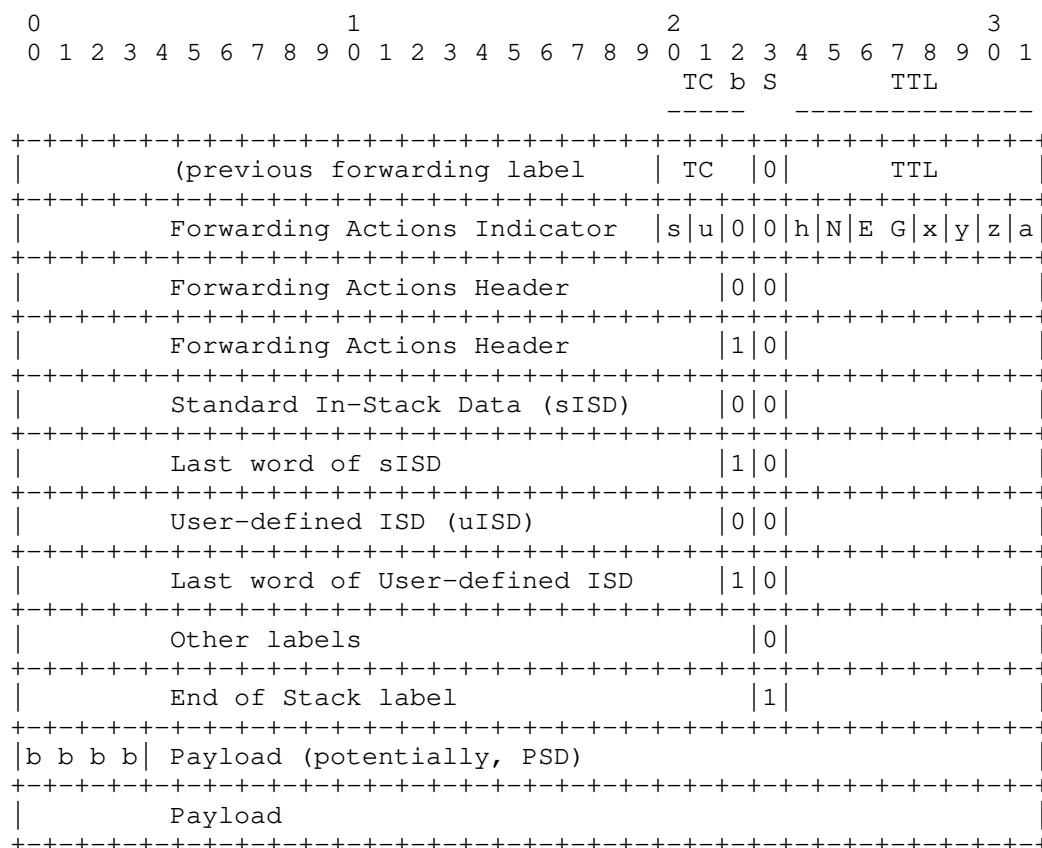


Figure 1: Format for FAI, ISD and PSD

The FAI's label value MUST be the IANA allocated value. The S bit MUST be reflect whether the label stack ends at this label or not.

#### 2.2.1. Definitions of the FAI Flag Bits

The TC and TTL bits are used as flags, defined as follows:

s: sISD is present (1) or not (0).

u: uISD is present (1) or not (0).

b: this is the "end of block" bit that indicates the end of the Forwarding Flags Block and the end of the ISD Block.

S: MUST be set if the FAI is the end of stack, and clear otherwise.

h: If set, the PSD contains hop-by-hop information. Every node in the path SHOULD attempt to process the hop-by-hop information, but not at the expense of exceeding the processing time budget, which could cause this (or other) packets to be dropped. If clear, no hop-by-hop data exists in the PSD: either the PSD is empty, or it contains only end-to-end data (to be processed by the egress).

N: If set, do not do fast reroute (NFFRR).

EG: this is a 2-bit flag indicating whether the ISD carries Entropy and/or GISS information.

The FAI Block consists of a Forwarding Flags Block, an sISD Block and a uISD Block. The two ISD Blocks are optional; their presence is indicated by the s and u bits. Each of these three blocks end when the b bit is set.

The Forwarding Flags Block extends from the FAI bSPL up to (and including) the first label that has the b bit set. If the FFB consists of just the bSPL, then its b bit must be set.

The sISD Block extends from the label after the FFB up to (and including) the label with the b bit set. If there is no sISD, the s bit in the FFB MUST be clear.

The uISD Block extends from the label after the sISD Block up to (and including) the label with the b bit set. If there is no uISD, the u bit in the FFB MUST be clear.

The EG field is used as follows:

00: No Entropy or GISS present

01: ISD 0 contains 16 bits of Entropy in the high order 16 bits and 14 bits of GISS in the low order 16 bits (S and b bits excepted).

10: ISD 0 contains 20 bits of Entropy in the high order 20 bits and 10 bits of GISS in the low order 12 bits (S and b bits excepted).

11: ISD 0 contains the 30-bit Entropy; ISD 1 contains the 30-bit GISS. In ISD 0, the S and b bits MUST be 0; the packet forwarding engine may choose to use the S and b bits as part of the Entropy, as it doesn't affect the outcome. In ISD 1, the S bit may be 0 or 1.

### 2.2.2. Processing the FAI Flags and the ISD

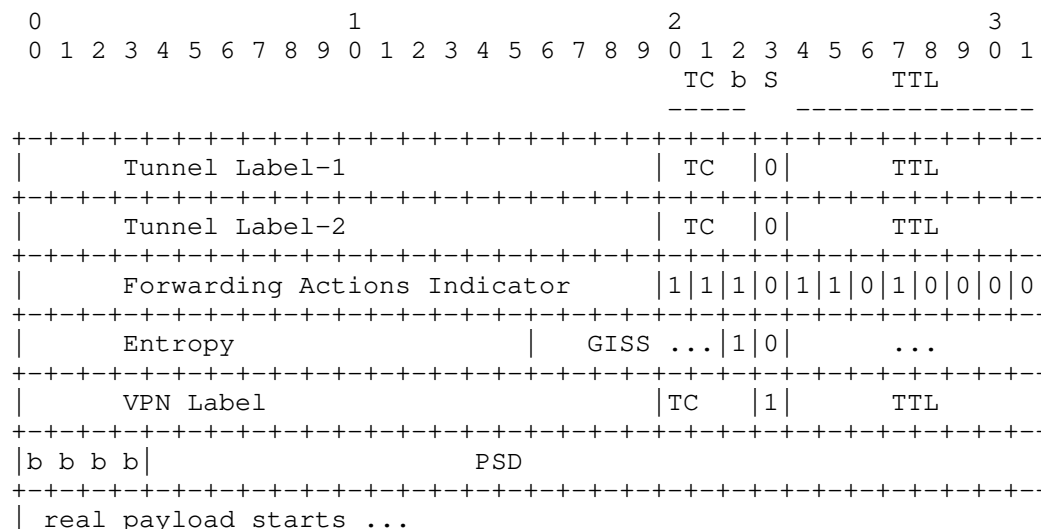
Here's how the Standard ISD is parsed. One must keep track of the s bit to know when the Standard ISD Block ends, and the S bit to know when the stack ends. The Standard ISD data appears in the order of the corresponding flags.

It is an error if the label stack ends while there are more ISD words to process. In particular, it is an error if the FAI's S bit is set, but the b bit is clear.

1. If s and u are both 0, done: there is no associated ISD.
2. Set CL ("current label") to the FAI label. LL is the last label (End of Stack); PL ("payload") is the first 4-octet word of the payload.
3. While b is clear:
  1. increment CL
4. Process N. CL is unchanged.
5. If s is set, Standard ISD is present: process standard flags.
  1. Process EG:
  2. If EG is 00, CL is unchanged.
  3. If EG is 01 or 10, increment CL. CL now contains both GISS and Entropy.
  4. If EG is 11, CL+1 contains Entropy; CL+2 contains GISS. Increment CL by 2.
  5. Process other standard data-bearing flags; increment CL by 1 for each.
6. If u is set, uISD is present.
  1. Process uISD until b is set.

Note that how the uISD is used is not defined here; this is up to the user. All that is included here is how a forwarding engine can tell where the uISD block ends.

### 2.2.3. Example of the FAI



s = 1: there is standard ISD.  
 u = 0: there is no user-defined ISD.  
 N = 1: NFFRR is set.  
 EG = 01: ISD 0 contains Entropy + GISS.  
 h = 1: There is hop-by-hop PSD.

Figure 2: Example of FAI + ISD + hop-by-hop PSD

The real payload starts after the PSD.

### 3. Issues to be Resolved

This section captures issues to be resolved, in this memo and others. As the issues are fixed, they should be removed from here; ideally, this section should be empty before publication.

#### 3.1. Preventing FAI From Reaching Top of Stack

As was said earlier, the FAI MUST NOT be at the top of stack, since its TC and TTL bits have been repurposed. There are two ways to prevent this. If an LSR X pops a label and the next label is the FAI, X can pop the FAI and all ISD words. This version of the memo introduces the "end-of-block" (s) bit, whereby a forwarding engine that knows the FAI can detect the entire FAI block, even if it doesn't know some of the flags. This can be used in conjunction with Section 3.2.

In case it is desired to preserve the FAI+FAD until the egress, X should push an explicit NULL (label value 0 or 2) onto the stack above the FAI, with the correct TC and TTL values.

Other options may be pursued; however, we believe this is an adequate resolution.

### 3.2. Repeating the FAI at "Readable Stack Depth"

For LSRs which cannot parse the entire label stack, or would prefer not to unless needed, it is possible to repeat the FAI at "readable stack depth" (rsd). Say the rsd is 10 labels, and the FAI block is 3 labels. Then, the FAI block can be repeated every 7 labels, allowing all forwarding engines in the path to process it. When a forwarding label is popped and the FAI block exposed, it is deleted in its entirety, since the same (or potentially different) FAI block is again within the rsd.

Note that the s or u bits set to 0 can be used to indicate that the corresponding ISD is absent. Only the last FAI would contain the full information, reducing the size of the label stack. However, in this case, LSRs that don't process the whole stack may not load balance less effectively, and potentially not adhere to the slice service level objectives.

Other options will be described in future versions of this document.

### 3.3. PSD

The format of the PSD, whether or not a Control Word is present, and handling of the first nibble, is outside the scope of this document. The FAI will not contain details about the contents of the PSD, besides the single flag on whether or not the PSD contains information relevant to (most) intermediate hops. It is assumed that another memo will document the format of the PSD, and that that memo will provide a means of parsing the PSD (e.g., a TLV structure) and thus determining its contents.

The PSD memo should also comment on the impact of processing the PSD on forwarding performance, especially in the case of hop-by-hop info.

## 4. Contributors

Many thanks to Colby Barth, Chandra Ramachandran and Srihari Sangli for their contributions to this draft.

## 5. Acknowledgments

We'd like to acknowledge the helpful discussions with Swamy SRK and folks from the Broadcom team on the impacts to existing and future forwarding engines.

The edist field was added thanks to Haoyu Song, who suggested the optimization to find End of Stack.

## 6. IANA Considerations

If this draft is deemed useful and adopted as a WG document, the authors request the allocation of a bSPL for the FAI. We suggest the early allocation of label 8 for this.

## 7. Security Considerations

A malicious or compromised LSR can insert the FAI and associated data into a label stack, preventing (for example) FRR from occurring. If so, protection will not kick in for failures that could have been protected, and there will be unnecessary packet loss. Similarly, inserting or removing a Fragmentation Header means that a packet's contents cannot be accurately reconstructed. Inserting or changing a GISS means that the packet will be misclassified, perhaps leaving or entering a high-value slice and causing damage.

## 8. References

### 8.1. Normative References

[I-D.bestbar-teas-ns-packet]

Saad, T., Beeram, V. P., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., Liu, X., Contreras, L. M., Rokui, R., and L. Jalil, "Realizing Network Slices in IP/MPLS Networks", Work in Progress, Internet-Draft, draft-bestbar-teas-ns-packet-07, 11 January 2022, <<https://www.ietf.org/archive/id/draft-bestbar-teas-ns-packet-07.txt>>.

[I-D.kompella-mpls-nffrr]

Kompella, K. and W. Lin, "No Further Fast Reroute", Work in Progress, Internet-Draft, draft-kompella-mpls-nffrr-02, 12 July 2021, <<https://www.ietf.org/archive/id/draft-kompella-mpls-nffrr-02.txt>>.



- [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>>.
- [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>>.

## 8.2. Informative References

- [I-D.nsd-t-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-nsd-t-teas-ns-framework-05, 2 February 2021, <<https://www.ietf.org/archive/id/draft-nsd-t-teas-ns-framework-05.txt>>.

## Authors' Addresses

Kireeti Kompella  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [kireeti.ietf@gmail.com](mailto:kireeti.ietf@gmail.com)

Vishnu Pavan Beeram  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Tarek Saad  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
United States

Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Israel Meilik  
Broadcom

Email: [israel.meilik@broadcom.com](mailto:israel.meilik@broadcom.com)

intarea  
Internet-Draft  
Intended status: Standards Track  
Expires: July 16, 2021

Z. Zhang  
R. Bonica  
K. Kompella  
Juniper Networks  
January 12, 2021

Generic Delivery Functions  
draft-zzhang-intarea-generic-delivery-functions-00

Abstract

Some functionalities (e.g. fragmentation/reassembly and Encapsulating Security Payload) provided by IPv6 can be viewed as delivery functions independent of IPv6 or even IP entirely. This document proposes to provide those functionalities at different layers (e.g., MPLS, BIER or even Ethernet) independent of IP.

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 July 16, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Specifications . . . . .	4
2.1. Generic Delivery Function Header . . . . .	4
2.2. Generic Fragmentation Header . . . . .	5
2.3. Payload Type Header . . . . .	6
2.4. Generic ESP/Authentication Header . . . . .	6
2.5. MPLS Signaling . . . . .	6
2.5.1. BGP Signaling . . . . .	6
2.5.2. IGP Signaling . . . . .	7
3. Security Considerations . . . . .	8
4. IANA Considerations . . . . .	8
5. Acknowledgements . . . . .	9
6. References . . . . .	9
6.1. Normative References . . . . .	9
6.2. Informative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

Consider an operator providing Ethernet services such as pseudowires, VPLS or EVPN. The Ethernet frames that a Provider Edge (PE) device receives from a Customer Edge (CE) device may have a larger size than the PE-PE path MTU (pMTU) in the provider network. This could be because

1. the provider network is built upon virtual connections (e.g. pseudowires) provided by another infrastructure provider, or
2. the customer network uses jumbo frames while the provider network does not, or
3. the provider-side overhead for transporting customers packets across the network pushes past the pMTU.

In any case, the provider cannot simply require its customers to change their MTU.

To get those large frames across the provider network, currently the only workaround is to encapsulate the frames in IP (with or without GRE) and then fragment the IP packets. Even if MPLS is used for service delimiting, IP is used for transportation (MPLS over IP/GRE). This may not be desirable in certain deployment scenarios, where MPLS

is the preferred transport or IP encapsulation overhead is deemed excessive.

IPv6 fragmentation and reassembly are based on the IPv6 Fragmentation header below [RFC8200]:

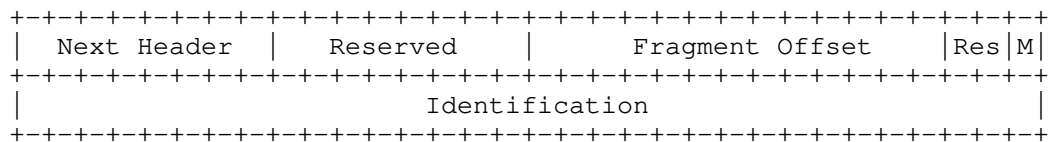


Figure 1: IPv6 Fragmentation Header

This document proposes adapting this header for use in non-IP contexts, since the fragmentation/reassembly function is actually independent of IPv6 except the following aspects:

- o The fragment header is identified as such by the "previous" header.
- o The "Next Header" value is from the "Internet Protocol Numbers" registry.
- o The "Identification" value is unique in the (source, destination) context provided by the IPv6 header

The "Identification" field, in conjunction with the IPv6 source and destination identifies fragments of the original packet, for the purpose of reassembly.

Therefore, the fragmentation/reassembly function can be applied at other layers as long as a) the fragment header is identified as such; and b) the context for packet identification is provided. Examples of such layers include MPLS, BIER, and Ethernet (if IEEE determines it is so desired).

For the same consideration, the IP Encapsulating Security Payload (ESP) [RFC4303] could also be applied at other layers if ESP is desired there. For example, if for whatever reason the Ethernet service provider wants to provide ESP between its PEs, it could do so without requiring IP encapsulation if ESP is applied at non-IP layers.

We refer to these as Generic Delivery Functions (GDFs), which could be achieved at a shim layer between a source and destination delivery points, for example:

- o Source and destination IP/Ethernet nodes
- o Ingress and egress nodes of MPLS Label Switch Paths (LSPs)
- o BIER Forwarding Ingress Routers (BFIRs) and BIER Forwarding Egress Routers (BFERs)

It is not the intention to apply the GDFs hop by hop between the source and destination delivery points.

The possibility of applying some other IP functions (e.g. Authentication Header [RFC4302]) is for further study.

## 2. Specifications

### 2.1. Generic Delivery Function Header

The following Generic Delivery Function Header (GDFH) is defined:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 0 0 0 | Rsved |   This Header   | Header Length | Next Header |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                               Variable field per "This header"                               ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 2: Generic Delivery Function Header

0000 nibble: Prevents the GDFH from being mistaken as an IP header by a router doing deep packet inspection for ECMP hashing purpose.

This Header: The type of this GDFH header. For example, TBD1 for generic fragmentation, TBD2 for generic ESP. The values are from a space independent of the "Next Header".

Header Length: The number of octets of the entire header.

Next Header: The type of next header. For functions that IETF is concerned with, the "Next Header" values are from the "Internet Protocol Numbers" registry. A next header could be another GDFH, so a value is to be assigned for GDFH from the registry.

The outer header MUST identify that a GDFH follows. Encoding "This Header" in the GDFH allows that a single outer header encoding can be used for different GDFHs.

If the outer header is BIER, a TBD value for the "proto" field in the BIER header identifies that a GDFH follows.

If the outer header is MPLS, the label preceding the GDFH indicates that a GDFH follows (see Section 2.5).

If the outer header is Ethernet (if IEEE would decide to provide the generic delivery functions on top of Ethernet directly), then a new Ethertype would be assigned by IEEE.

## 2.2. Generic Fragmentation Header

For generic fragmentation/reassembly functionality, the GDFH takes the following Generic Fragmentation Header (GFH) format:

```

+-----+
| 0 0 0 0 | Rsvd | Header Length |      TBD1      | Next Header |
+-----+
|      Fragment Offset      | R | S | M | Identification (variable) | ~
+-----+
|                                     Identification                                     |
|                                     (continues)                                     |
+-----+

```

Figure 3: Generic Fragmentation Header

The "Fragment Offset" and "M" flag bit fields are as in the IPv6 Fragmentation Header.

R: The "R" flag bit is reserved. It MUST be 0 on transmitting and ignored on receiving.

Identification: at least 4-octet long. // would 2-octet be ok as minimum?

S: If the "S" flag bit is clear, the context for the Identification field is provided by the outer header, and only the source-identifying information in the outer header is used.

If the "S" flag bit is set, the variable Identification field encodes both source-identifying information (e.g. the IP address of the node adding the GFH) and an identification number unique within that source.

When a GFH is used together with other GDFHs, the GFH SHOULD be the first GDFH.

If the outer header is BIER and the "S" flag bit is clear, the "BFIR-id" field in the BIER header provides the context for the "Identification" field.

If the outer header is MPLS, the "S" flag bit MAY be clear if the the label preceeding the GFH identifies the sending node in addition to indicating that a GFH follows (see Section 2.5).

### 2.3. Payload Type Header

While originally it is not the intention to provide a way to identify the payload type after an MPLS label stack, it has been pointed out that the GFH now provides the payload-identifying functionality as a by product - even when fragmentation is not needed, a GFH can be inserted, with the Fragmentation Offset, the M-bit and Identification fields set to 0, and the Next Header set appropriately.

If the payload-identifying functionality is deemed as desired, a dedicated header type could be assigned for this purpose, with a smaller header compared to GFH.

```

+-----+
| 0 0 0 0 | Rsved | Header Length:4 |      TBD3      | Next Header |
+-----+
```

Figure 4: Generic Payload Type Header

### 2.4. Generic ESP/Authentication Header

To be specified in future revisions.

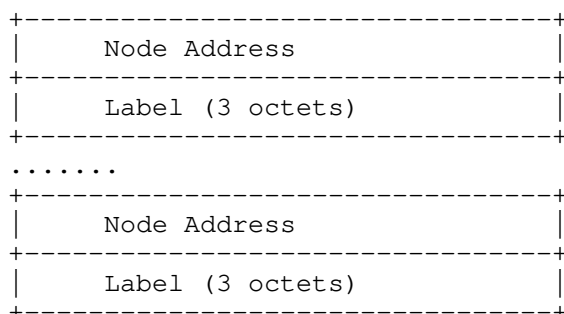
### 2.5. MPLS Signaling

When GDFH is used with MPLS, the preceeding label needs to indicate that a GDFH follows, and optionally identify the node that does the fragmentation. The label can be signaled via BGP or IGP as sepcified below.

#### 2.5.1. BGP Signaling

This document defines a new transitive BGP "GDFH Labels" attribute, very similar to the "PE Distinguisher Labels" attribute defined in [RFC6514] (and the text below is adapted from Section 8 of [RFC6514]):





The Label field contains an MPLS label encoded as 3 octets, where the high-order 20 bits contain the label value. The Node Address MAY be 0, meaning that the following label only indicates a GDFH follows when the label is used in the label stack of a data packet.

The Node Address MAY also be a unicast address, indicating that the following label when used in the label stack of a data packet will both indicate that a GDFH follows and identify the sending node.

If a node supports GDFH with MPLS, it attaches the attribute in the BGP routes for its local addresses. A border router SHOULD remove the attribute if no node beyond the border will use GDFH with MPLS to send traffic to the corresponding addresses.

A router that supports the attribute considers this attribute to be malformed if the Node Address field does not contain a unicast address or 0. The attribute is also considered to be malformed if: (a) the Node Address field is expected to be an IPv4 address, and the length of the attribute is not a multiple of 7 or (b) the Node Address field is expected to be an IPv6 address, and the length of the attribute is not a multiple of 19. The Address Family Indicator (AFI) of the BGP route that the attribute is attached to provides the information on whether the Node Address field contains an IPv4 or IPv6 address. Each of the Node Addresses in the attribute MUST be of the same address family as the route that is carrying the attribute.

#### 2.5.2. IGP Signaling

This document defines an OSPFv2 "GDFH Labels" sub-TLV of OSPFv2 Extended Prefix TLV [RFC7684], with the value part being the same as BGP "GDFH Labels" attribute above. If an OSPFv2 router supports GDFH with MPLS, it includes the GDFH Labels sub-TLV in the Extended Prefix TLV that is attached to its local addresses advertised in its OSPFv2 Extended Prefix Opaque LSA.

Similary, This document defines an OSPFv3 "GDFH Labels" sub-TLV of OSPFv3 Intra/Inter-Area-Prefix TLVs [RFC8362], with the value part being the same as BGP "GDFH Labels" attribute above. If an OSPFv3 router supports GDFH with MPLS, it includes the GDFH Labels sub-TLV in the Intra-Area-Prefix TLV for its local addresses.

This document also defines an ISIS "GDFH Labels" sub-TLV of ISIS prefix-reachability TLV [RFC5120] [RFC5305] [RFC5308], with the value part being the same as BGP "GDFH Labels" attribute above. If an ISIS router supports GDFH with MPLS, it includes the sub-TLV to the prefix-reachability TLV for its local addresses.

For both OSPF and ISIS, when advertising a prefix from one area/level to another, if there is a "GDFH Labels TLV" attached in the source area/level, the TLV SHOULD be attached in the target area/level and the prefix SHOULD NOT be summarized.

### 3. Security Considerations

To be provided.

### 4. IANA Considerations

This document makes the following IANA requests:

- o A new BGP Attribute type for "GDFH Labels" from the BGP Path Attributes registry
- o A new OSPFv2 sub-TLV type for "GDFH Labels" from the OSPFv2 Extended Prefix TLV Sub-TLVs registry
- o A new OSPFv3 sub-TLV type for "GDFH Labels" from the OSPFv3 Extended-LSA sub-TLV registry
- o A new BIER Next Protocol Identifier value for GDFH from BIER Next Protocol Identifiers registry
- o A new Internet Protocol Number for GDFH from the Internet Protocol Numbers registry

This document requests IANA to set up a "Generic Deliver Function Header Types" registry with the following initial assignments:

0: Reserved

1: Generic Fragmentation

2: Generic ESP

## 5. Acknowledgements

The authors thank Stewart Bryant and Tony Przygienda for their valuable comments and suggestions.

## 6. References

### 6.1. Normative References

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <<https://www.rfc-editor.org/info/rfc8362>>.

### 6.2. Informative References

- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.

#### Authors' Addresses

Zhaohui Zhang  
Juniper Networks  
1133 Innovation Way  
Sunnyvale 94089  
USA

Phone: +1 408 745 2000  
Email: [zzhang@juniper.net](mailto:zzhang@juniper.net)

Ron Bonica  
Juniper Networks  
1133 Innovation Way  
Sunnyvale 94089  
USA

Phone: +1 408 745 2000  
Email: [rbonica@juniper.net](mailto:rbonica@juniper.net)

Kireeti Kompella  
Juniper Networks  
1133 Innovation Way  
Sunnyvale 94089  
USA

Phone: +1 408 745 2000  
Email: [kireeti@juniper.net](mailto:kireeti@juniper.net)

intarea  
Internet-Draft  
Intended status: Standards Track  
Expires: February 26, 2022

Z. Zhang  
R. Bonica  
K. Kompella  
Juniper Networks  
G. Mirsky  
ZTE  
August 25, 2021

Generic Delivery Functions  
draft-zzhang-intarea-generic-delivery-functions-02

Abstract

Some functionalities (e.g., fragmentation/reassembly and Encapsulating Security Payload) provided by IPv6 can be viewed as delivery functions independent of IPv6 or even IP entirely. This document proposes to provide those functionalities at different layers (e.g., MPLS, BIER or even Ethernet) independent of IP.

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 February 10, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Specifications . . . . .	4
2.1. MPLS layer . . . . .	4
2.2. BIER layer . . . . .	5
2.3. Other layers . . . . .	5
2.4. Generic Fragmentation Header (GFH) {#gfh} . . . . .	6
3. Security Considerations . . . . .	6
4. Acknowledgements . . . . .	6
5. References . . . . .	7
5.1. Normative References . . . . .	7
5.2. Informative References . . . . .	7
Authors' Addresses . . . . .	8

## 1. Introduction

Consider an operator providing Ethernet services such as EVPN. The Ethernet frames that a Provider Edge (PE) device receives from a Customer Edge (CE) device may have a larger size than the PE-PE path MTU (PMTU) in the provider network. This could be because

1. the provider network is built upon virtual connections (e.g., pseudowires) provided by another infrastructure provider, or
2. the customer network uses jumbo frames while the provider network does not, or
3. the provider-side overhead for transporting customer packets across the network pushes past the PMTU.

In any case, the provider cannot simply require its customers to change their MTU.

To get those large frames across the provider network, currently, the only workaround is to encapsulate the frames in IP (with or without GRE) and then fragment the IP packets. Even if MPLS is used for service delimiting, IP is used for transportation (MPLS over IP/GRE). This may not be desirable in certain deployment scenarios, where MPLS is the preferred transport or IP encapsulation overhead is deemed excessive.

IPv6 fragmentation and reassembly are based on the IPv6 Fragmentation header below [RFC8200]:

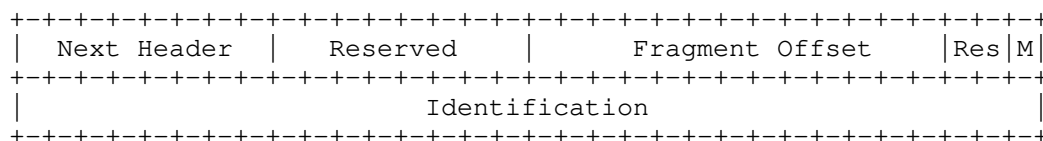


Figure 1: IPv6 Fragmentation Header

This document proposes adapting this header for use in non-IP contexts since the fragmentation/reassembly function is actually independent of IPv6 except for the following aspects:

- o The fragment header is identified as such by the "previous" header.
- o The "Next Header" value is from the "Internet Protocol Numbers" registry.
- o The "Identification" value is unique in the (source, destination) context provided by the IPv6 header.

The "Identification" field, in conjunction with the IPv6 source and destination addresses identifies fragments of the original packet for the purpose of reassembly.

Therefore, the fragmentation/reassembly function can be applied at other layers as long as a) the fragment header is identified as such; and b) the context for packet identification is provided. Examples of such layers include MPLS, BIER, and Ethernet (if IEEE determines it is so desired).

For the same consideration, the IP Encapsulating Security Payload (ESP) [RFC4303] could also be applied at other layers if ESP is desired there. For example, if for whatever reason the Ethernet service provider wants to provide ESP between its PEs, it could do so without requiring IP encapsulation if ESP is applied at non-IP layers.

Similarly, In-Situ OAM (IOAM) functions [I-D.ietf-ippm-ioam-data] can also be applied to many layers.

We refer to these as Generic Delivery Functions (GDFs), which could be achieved at a shim layer between a source and destination delivery points, for example:

- o Source and destination IP/Ethernet nodes
- o Ingress and egress nodes of MPLS Label Switch Paths (LSPs)

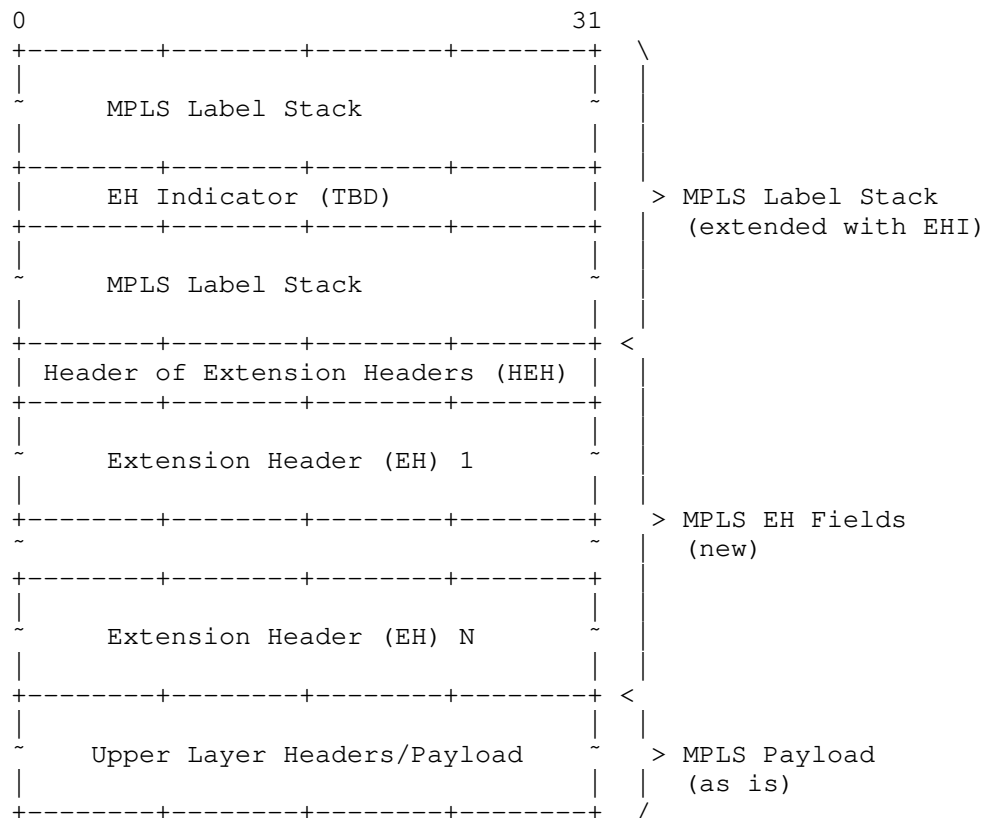
- o BIER Forwarding Ingress Routers (BFIRs) and BIER Forwarding Egress Routers (BFERs)

## 2. Specifications

A Generic Delivery Function, being generic, is likely applicable to IP as well. Therefore, IPv6 Extension Headers (for some GDFs) are directly used at other layers.

### 2.1. MPLS layer

[I-D.song-mpls-extension-header] specifies MPLS Extension Headers encoding. A label entry in the stack indicates the presence of extension headers after the label stack. It starts with a Header of Extension Headers, as depicted in the following excerpt from that specification:

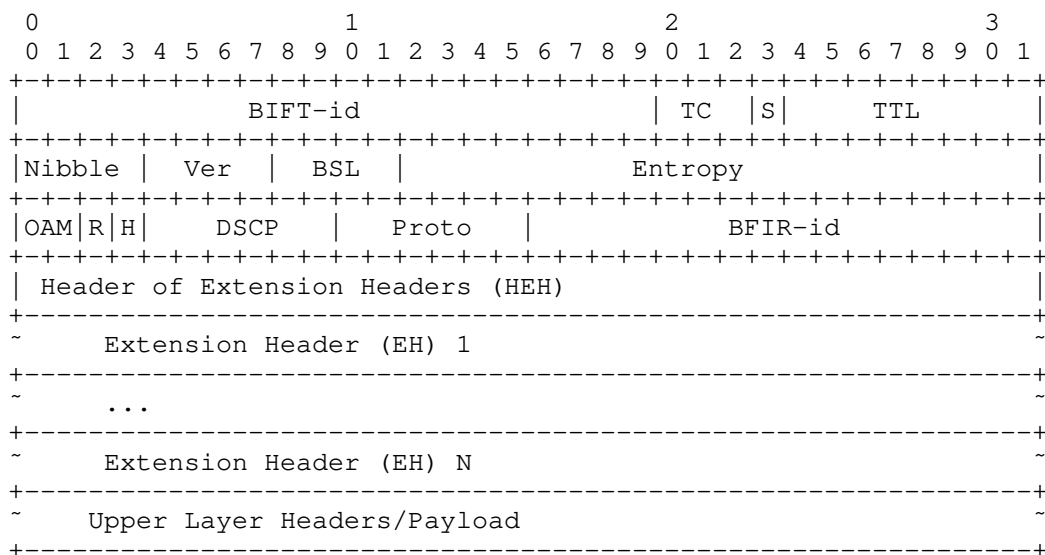


One or more of the EHs in the above can be an IPv6 Extension Header for a GDF.



## 2.2. BIER layer

For BIER layer, a TBD value for the "proto" field in the outer BIER header indicates that some BIER Extension Headers follow the BIER header, including some IPv6 Extension Headers for GDFs.



R: The "R" flag bit is reserved. It MUST be set to 0 on transmit and ignored on receive.

H: If the "H" flag bit, it indicates the presence of at least one extension header that needs to be processed hop by hop even before a BFER is reached. In this case, the Proto field must be set to the TBD value indicating the presence of extension headers.

## 2.3. Other layers

Similarly, any layer can have an indication in its packet header that some GDF extension headers follow, including some IPv6 Extension Headers for GDF purpose.

For example, if the outer header is Ethernet (if IEEE would decide to provide the generic delivery functions on top of Ethernet directly), then a new Ethertype would be assigned by IEEE to indicate the presence of GDF extension headers.

## 2.4. Generic Fragmentation Header (GFH){#gfh}

For generic fragmentation/reassembly functionality, the existing IPv6 Fragment Header needs to be enhanced for MPLS as following:

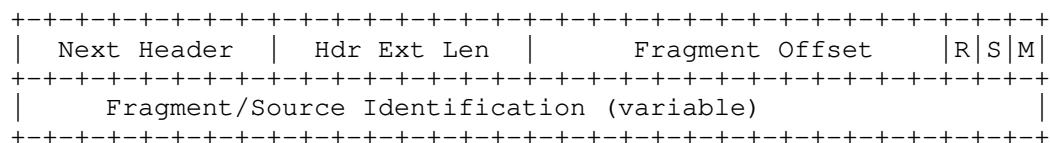


Figure 2: Generic Fragmentation Header

R: The "R" flag bit is reserved. It MUST be set to 0 on transmit and ignored on receive.

S: If the "S" flag bit is clear, the context for the Identification field is provided by the outer header, and only the source-identifying information in the outer header is used.

If the "S" flag bit is set, the variable Identification field encodes both source-identifying information (e.g., the IP address of the node adding the GFH) and an identification number unique within that source. The length of the Fragment header is encoded in the 8-bit "Hdr Ext Len" field (which is a Reserved field in the original IPv6 Fragment Header).

When a GFH is used together with other GDF Headers (GDFH), the GFH SHOULD be the first GDFH.

The above enhancement is not necessary but MAY be used for BIER as well. If the outer header is BIER and the "S" flag bit is clear, the "BFIR-id" field in the BIER header provides the context for the "Identification" field. If the bit is set, then the source information embedded in the source/fragment identification field is used.

## 3. Security Considerations

To be provided.

## 4. Acknowledgements

The authors thank Stewart Bryant and Tony Przygienda for their valuable comments and suggestions.

## 5. References

### 5.1. Normative References

- [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.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi  
Topology (MT) Routing in Intermediate System to  
Intermediate Systems (IS-IS)", RFC 5120,  
DOI 10.17487/RFC5120, February 2008,  
<<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic  
Engineering", RFC 5305, DOI 10.17487/RFC5305, October  
2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308,  
DOI 10.17487/RFC5308, October 2008,  
<<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W.,  
Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute  
Advertisement", RFC 7684, DOI 10.17487/RFC7684, November  
2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6  
(IPv6) Specification", STD 86, RFC 8200,  
DOI 10.17487/RFC8200, July 2017,  
<<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and  
F. Baker, "OSPFv3 Link State Advertisement (LSA)  
Extensibility", RFC 8362, DOI 10.17487/RFC8362, April  
2018, <<https://www.rfc-editor.org/info/rfc8362>>.

### 5.2. Informative 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-14 (work in  
progress), June 2021.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)",  
RFC 4303, DOI 10.17487/RFC4303, December 2005,  
<<https://www.rfc-editor.org/info/rfc4303>>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.

#### Authors' Addresses

Zhaohui Zhang  
Juniper Networks

Email: [zzhang@juniper.net](mailto:zzhang@juniper.net)

Ron Bonica  
Juniper Networks

Email: [rbonica@juniper.net](mailto:rbonica@juniper.net)

Kireeti Kompella  
Juniper Networks

Email: [kireeti@juniper.net](mailto:kireeti@juniper.net)

Gregory Mirsky  
ZTE

Email: [gregory.mirsky@ztetx.com](mailto:gregory.mirsky@ztetx.com)