MPLS WG
Internet-Draft
Intended status: Standards Track
Expires: 9 January 2023

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

8 July 2022

Multi-purpose Special Purpose Label for Forwarding Actions
draft-kompella-mpls-mspl4fa-03

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 data (if any) associated with the actions, some in the label stack and some after the label stack. This proposal also solves the problem of scarcity of base SPLs.

As proof of its utility and flexibility, this approach can immediately address several use cases:

* to carry an Entropy Label for better load balancing;

* to carry a Flow-Aggregate 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.

This Internet-Draft will expire on 9 January 2023.

Copyright Notice

Table of Contents

1. Introduction

Base Special Purpose Labels (bSPLs) [RFC3031], [RFC7274], [RFC9017] 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

ISD: In-Stack Data

sISD: Standard ISD

uISD: User-Defined ISD

ISDH: In-Stack Data Header

LSE: Label stack entry

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

1.2.3. Changes from -02 to -03

1. Separated the discussion of the LSE format from the flag definitions.

2. Replaced continuation bits with explicit length fields for easier parsing.

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 flow-aggregate, given its G-FAS;

* 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 stack entry, 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

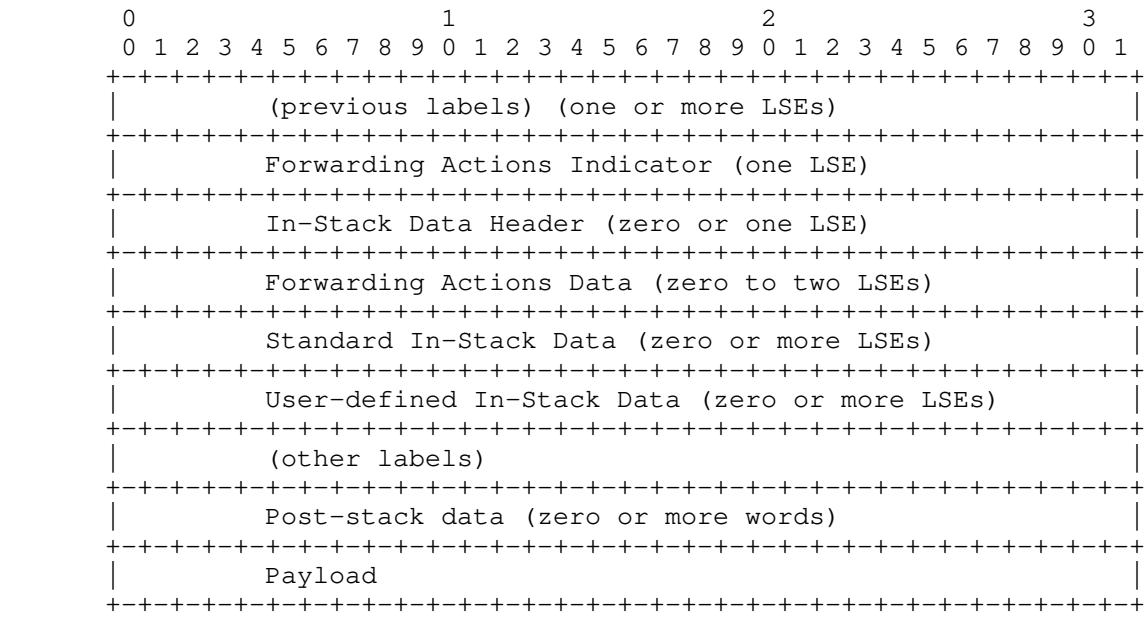
A pertinent question is when one should put data in the ISD versus in the PSD. One alternative is to put all such data in the PSD. However, this would mean that accessing such information would require finding the End of Stack, and parsing the PSD. For certain types of data, this would be a severe burden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

The format of a label stack that includes an FAI has the structure:



The flags and data associated with the FAI come in three forms:

1. Forwarding Actions Flags and Data; these flags are in the FAI LSE. These are defined in this document.

2. Standard In-Stack Flags and Data; these flags are in the In-Stack Data Header. These MUST be defined in a Standards-track document.

3. User-defined In-Stack Flags and Data; these flags are also in the In-Stack Data Header. These are perforce user-defined and not subject to standardization.

The format of the PSD is out of scope for this document.

2.2.1. The FAI Label Stack Entry (LSE)

The format of the FAI LSE is:

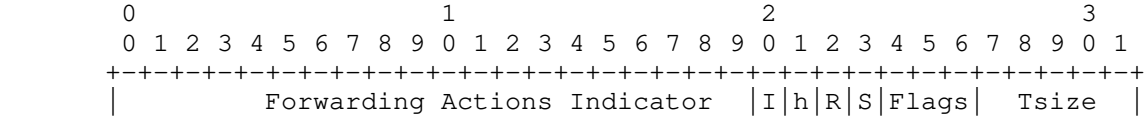


Figure 1: Format for the FAI label stack entry

The FAI LSE's label value MUST be the IANA allocated value. The remaining fields are defined as follows:

I: In-Stack Data Header (ISDH) is present (1) or not (0).

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) packet(s) 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).

R: Reserved for future use (SHOULD be ignored by receivers).

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

Flags: Each bit in the Flags field represents a forwarding action. Except for the flags defined herein, the semantics of a forwarding action are out of scope for this document. Each flag is allocated from the Forwarding Actions Flag Registry (Section 11.1) and is assigned an index. Flag indices 0 to 2 correspond to bits 24 to 26 within the FAI LSE. Flags continue into the In-Stack Data Header, if present.

Tsize: This is the total size of the FAI block, i.e., the FAI LSE and all associated data, in four-octet words.

2.2.2. The In-Stack Data Header

If the I flag is set, the label stack contains an In-Stack Data Header (ISDH) following a FAI LSE. The format of the ISDH is:

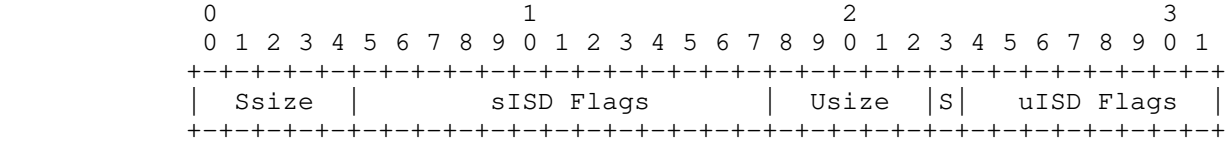


Figure 2: Format for the Forwarding Actions Header

The fields in the Forwarding Actions Header are:

Ssize: This is the size of the Standard ISD (sISD) in four-octet words.

sISD Flags: This is a continuation of the Flags field from the

preceeding FAI LSE, i.e., flag indices 3 to 15 correspond to bits 5 to 17 in the ISDH.

Usize: This is the size of the user-defined ISD (uISD) in four-octet words.

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

uISD Flags: These correspond to user-defined flags, and are not subject to standardization.

3. Forwarding Action Flags

This document defines the following Forwarding Action Flags.

3.1. No Further Fast Re-Route (NFFRR)

Index: 0

ISD: None

If set, do not perform fast re-route on this traffic.

3.2. Entropy and Slice information (EG)

These are two flags that have joint semantics:

Indices: 1, 2

ISD: 0, 1, or 2 ISD LSEs, depending on the values of the flags:

00: No Entropy or G-FAS present

01: ISD 0 contains 16 bits of Entropy in the high order 16 bits and 15 bits of G-FAS in the low order 16 bits (S bit excepted).

10: ISD 0 contains 20 bits of Entropy in the high order 20 bits and 11 bits of G-FAS in the low order 12 bits (S bit excepted).

11: ISD 0 contains the 31-bit Entropy; ISD 1 contains the 31-bit G-FAS. In ISD 0, the S bit MUST be 0; the packet forwarding engine may choose to use the S bit as part of the Entropy, as it doesn't affect the outcome. In ISD 1, the S bit may be 0 or 1.

4. Rules for Processing

4.1. Processing the FAI Flags and the ISD

Here's how the Standard ISD is parsed. One must keep track of Ssize to know when the Standard ISD data 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.

1. 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.

2. If I is 0, there is no associated ISDH; set Ssize = Usize = 0. Otherwise, note the values of Ssize and Usize; increment CL.

3. Process Forwarding Actions Flags and Data:

    1. Process N. CL is unchanged.

    2. Process EG:

        1. If EG is 00, CL is unchanged.

        2. If EG is 01 or 10, increment CL. CL now contains both G-FAS and Entropy.

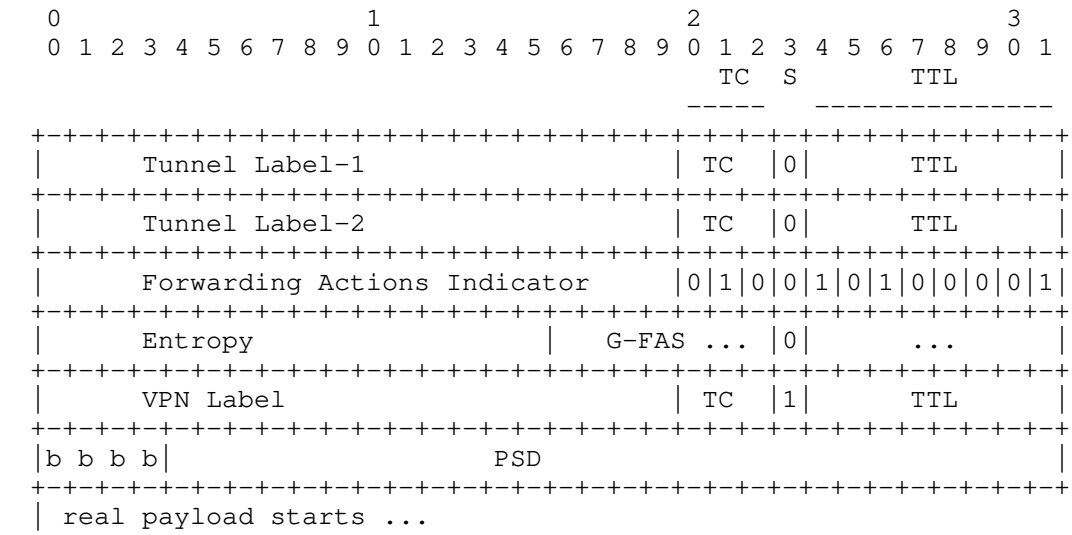
        3. If EG is 11, CL+1 contains Entropy; CL+2 contains G-FAS. Increment CL by 2.

4. If I == 1, process sISD Flags; increment CL by 1 for each (upto Ssize; skip any remaining sISD Flags once Ssize LSEs have been processed).

5. If I == 1, process uISD Flags; increment CL by 1 for each (upto Usize).

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 the size of uISD.

4.2. Example of the FAI



I = 0: there is no ISDH.
h = 1: There is hop-by-hop PSD.
R = 0: Ignore.
N = 1: NFFRR is set.
EG = 01: ISD 0 contains a 16-bit Entropy label and a 15-bit G-FAS.
Tsize = 1: The FAI block consists of 1 LSE beyond the FAI LSE.

Figure 3: Example of FAI + ISD + hop-by-hop PSD

The real payload starts after the PSD.

5. 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.

5.1. 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 MUST pop Tsize LSEs to remove the FAI LSE and associated data. This implies that the LSR MUST be able to recognize the FAI LSE and at least parse the Tsize field. This can be used in conjunction with Section 5.2.

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.

5.2. 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 LSEs, and the FAI block contains 3 LSEs. 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.

Other options were considered in [RFC8662], Section 10, and are discussed briefly in Section 8.

6. PSD

Currently, CW, ...

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.

7. Appendix 1 (normative) Use Cases

7.1. Flow-Aggregate Selector

Network slicing is an important ongoing effort both for network design, as well as for standardization, in particular at the IETF [I-D.nsdt-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.ietf-teas-ns-ip-mpls] describes several such methods for MPLS networks, of which the Global Identifier for Flow-Aggregate Selector (G-FAS) is one of the more practical solutions.

This document shows how to realize the G-FAS using a base special purpose label (bSPL).

In MPLS networks, a G-FAS is a data plane construct identifying packets belonging to a slice aggregate (the set of packets that belong to the slice). The G-FAS dictates forwarding actions for the slice aggregate: QoS behavior and next hop selection. The purpose of the G-FAS is detailed in [I-D.ietf-teas-ns-ip-mpls]. To embed a G-FAS 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).

7.2. Entropy Label

7.3. No Further Fast Re-Route

8. Appendix 2 (non-normative): Positioning of the FAI Block

9. Contributors

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

10. 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.

11. 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.

11.1. The Forwarding Actions Flag Registry

A new registry is needed for the allocation of the Forwarding Action flags. This registry should be called the "Forwarding Action Flags Registry" within the "Multiprotocol Label Switching Architecture (MPLS)" protocol registry. The policy for allocating new flags is Standards Action. Each flag MUST have a name, a brief description and the length of the associated ISD. IANA should allocate an index for each flag, starting with zero.

The initial contents of the registry should contain:

Name	Indices	Reference
No Further Fast Re-Route	0	This document
Entropy and Slice information	1, 2	This document

Table 1

12. 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 G-FAS means that the packet will be misclassified, perhaps leaving or entering a high-value slice and causing damage.

13. References

13.1. Normative References

[I-D.ietf-teas-ns-ip-mpls]
Saad, T., Beeram, V. P., Dong, J., 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-ietf-teas-ns-ip-mpls-00, 16 June 2022, <https://www.ietf.org/archive/id/draft-ietf-teas-ns-ip-mpls-00.txt>.

[I-D.kompella-mpls-nffrr]
Kompella, K. and W. Lin, "No Further Fast Reroute", Work in Progress, Internet-Draft, draft-kompella-mpls-nffrr-03, 8 July 2022, <https://www.ietf.org/archive/id/draft-kompella-mpls-nffrr-03.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>.

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

[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>.

[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>.

[RFC9017] Andersson, L., Kompella, K., and A. Farrel, "Special-Purpose Label Terminology", RFC 9017, DOI 10.17487/RFC9017, April 2021, <https://www.rfc-editor.org/info/rfc9017>.

13.2. Informative References

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

[RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <https://www.rfc-editor.org/info/rfc8662>.

Authors' Addresses

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

Vishnu Pavan Beeram
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States
Email: vbeeram@juniper.net

Tarek Saad
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States
Email: tsaad@juniper.net

Israel Meilik
Broadcom
Email: israel.meilik@broadcom.com