

Workgroup: MPLS WG
Internet-Draft: draft-kompella-mpls-mspl4fa-02
Published: 8 February 2022
Intended Status: Standards Track
Expires: 12 August 2022
Authors: K. Kompella V.P. Beeram
 Juniper Networks Juniper Networks
 T. Saad I. Meilik
 Juniper Networks Broadcom

Multi-purpose Special Purpose Label for Forwarding Actions

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](#)
 - 1.1. [Conventions and Definitions](#)
 - 1.2. [Revision History](#)
 - 1.2.1. [Changes from -00 to -01](#)
 - 1.2.2. [Changes from -01 to -02](#)
 - 1.3. [Slice Selector](#)
- 2. [Multi-purpose bSPL: the Forwarding Actions Indicator](#)
 - 2.1. [The FAI bSPL](#)
 - 2.1.1. [ISD vs PSD](#)
 - 2.2. [Format of the FAI bSPL](#)
 - 2.2.1. [Definitions of the FAI Flag Bits](#)
 - 2.2.2. [Processing the FAI Flags and the ISD](#)
 - 2.2.3. [Example of the FAI](#)
- 3. [Issues to be Resolved](#)
 - 3.1. [Preventing FAI From Reaching Top of Stack](#)
 - 3.2. [Repeating the FAI at "Readable Stack Depth"](#)
 - 3.3. [PSD](#)
- 4. [Contributors](#)
- 5. [Acknowledgments](#)
- 6. [IANA Considerations](#)
- 7. [Security Considerations](#)
- 8. [References](#)
 - 8.1. [Normative References](#)
 - 8.2. [Informative References](#)
- [Authors' Addresses](#)

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.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).

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

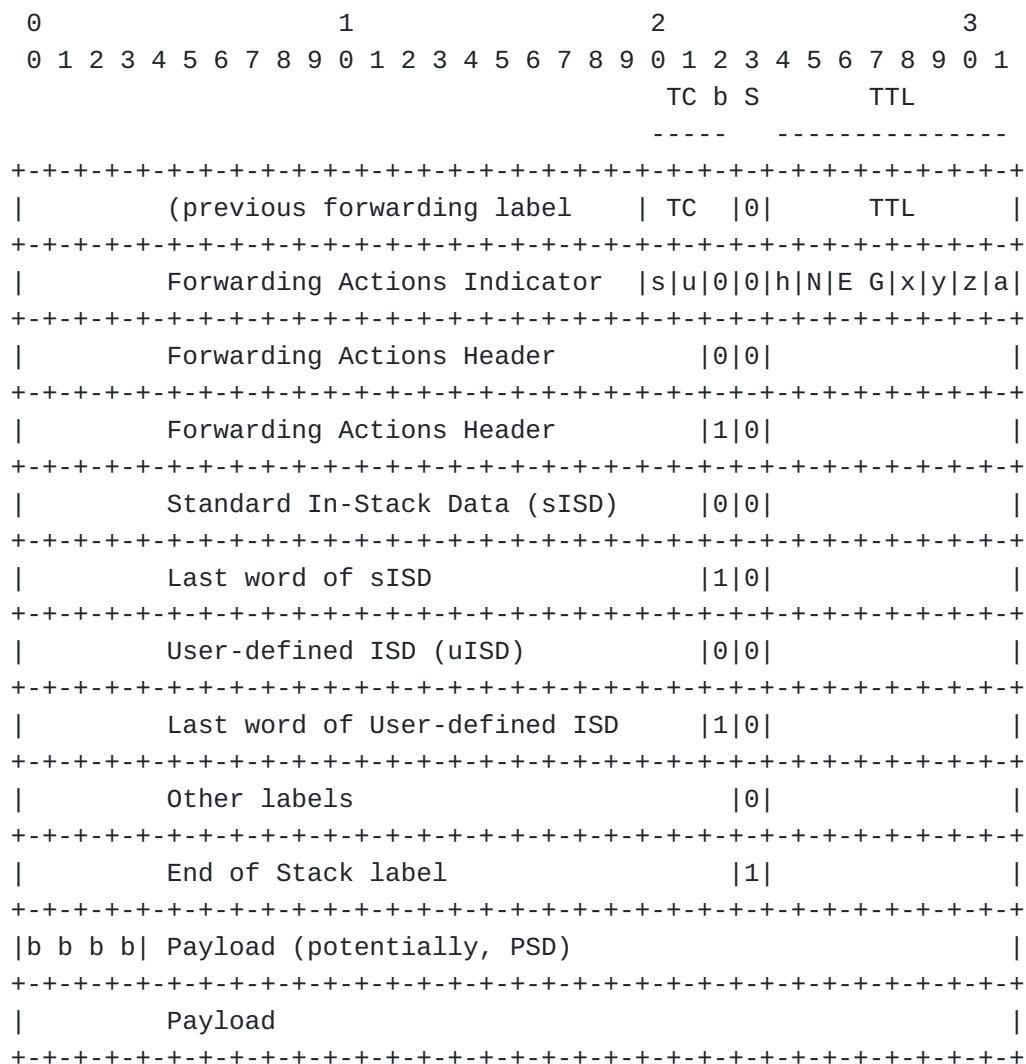


Figure 1: Format for FAI, ISD and PSD

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 end, 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

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                TC b S          TTL
                                -----
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Tunnel Label-1          | TC |0|      TTL      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Tunnel Label-2          | TC |0|      TTL      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Forwarding Actions Indicator  |1|1|1|0|1|1|0|1|0|0|0|0|
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Entropy          |      GISS ...|1|0|      ...      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      VPN Label          | TC |1|      TTL      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|b b b b|          PSD          |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| real payload starts ...

```

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-eas-n-s-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>>.

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