

MPLS
Internet-Draft
Intended status: Standards Track
Expires: November 19, 2021

S. Bryant
A. Clemm
T. Eckert
Futurewei Technologies, Inc.
May 18, 2021

Use of an MPLS LSE as an Ancillary Data Pointer
draft-bryant-mpls-aux-data-pointer-00

Abstract

The purpose of this memo is to describe how Label Stack Entries (LSEs) can be used to point to ancillary or meta-data carried below the MPLS label stack.

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 November 19, 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.

Internet-Draft

MPLS-Aux-Ptr

May 2021

Table of Contents

1.	Introduction	2
2.	Background Documents	2
3.	Use of SPLs as Pointers	3
4.	Label Operations: Popping and Swapping	5
5.	Use of Multiple Pointers	6
6.	Disposition of the Ancillary Data	9
7.	Structure of Ancillary Data	9
8.	Structure of Pointer Label	9
9.	Backward Compatibility	10
10.	Security Considerations	10
11.	IANA Considerations	11
12.	Appendix 1: CONTROVERSY ALERT - Use Of Ordinary Labels	11
13.	Appendix 2: Other Issues for Discussion	12
14.	Appendix 3: Ancillary vs Auxiliary vs Metadata	12
15.	References	13
15.1.	Normative References	13
15.2.	Informative References	13
	Authors' Addresses	14

[1.](#) Introduction

There has been significant recent interest in developing the MPLS data plane to address new needs, and in particular to carry ancillary or meta-data below the stack. In this document we consider that this ancillary data is further subdivided into a sequence of blocks. This draft does not prescribe the information or its structure of the ancillary data. For the sake of examples, it could range from a single ancillary data unit to a structured set of ancillary data blocks similar to an IPv6 extension header. There has also been recent interest in carrying additional flags or other indicators to qualify the forwarding operations.

This memo proposes the use of "spare" bits in a Special Purpose Label (SPL) [[I-D.kompella-mpls-mspl4fa](#)] be used as a pointer to items of ancillary data carried below the bottom of stack (BoS). Finally we speculate that in certain network scopes we may usefully be able to create pseudo-SPLs from the ordinary label pool.

[2.](#) Background Documents

[[I-D.kompella-mpls-mspl4fa](#)] notes that the forwarder does not need to

use the TC, or TTL fields in an LSE [[RFC3032](#)] that does not become top of stack (ToS). It proposes to exploit these fields as indicators of forwarding actions, by modifying the semantics of these fields.

There are a number of key proposals in that draft:

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

This draft proposes that these "spare" bits in an SPL or pseudo-SPL be used as a pointer to ancillary data below the stack.

This proposal can be used in conjunction with the other indicator proposals, for example by using different SPLs for different options, such as one SPL indicating the presence of a pointer vs one or more other SPLs for the other proposals.

[3.](#) Use of SPLs as Pointers

Previously it had been proposed to use the "spare" bits in an SPL that is not ToS as a bit field or as an enumerator of a slice. However, it would appear to be an advantage to take things a bit further and use them as a pointer to ancillary data below the BoS. This ancillary data can then be accessed and processed as needed whenever the SPL is being processed.

The advantages of doing this are:

- o The ability to find the ancillary data without scanning the whole stack. Speculatively scanning the label stack can be expensive in Network Processor Unit (NPU) processing time, particularly if the

stack is deep.

- o Ability to specify which ancillary data is applicable at the hop being processed.
- o The use of a pointer or set of pointers allows for a simple packet parser.
- o The approach is inherently general and extensible.

This concept is illustrated in Figure 1.

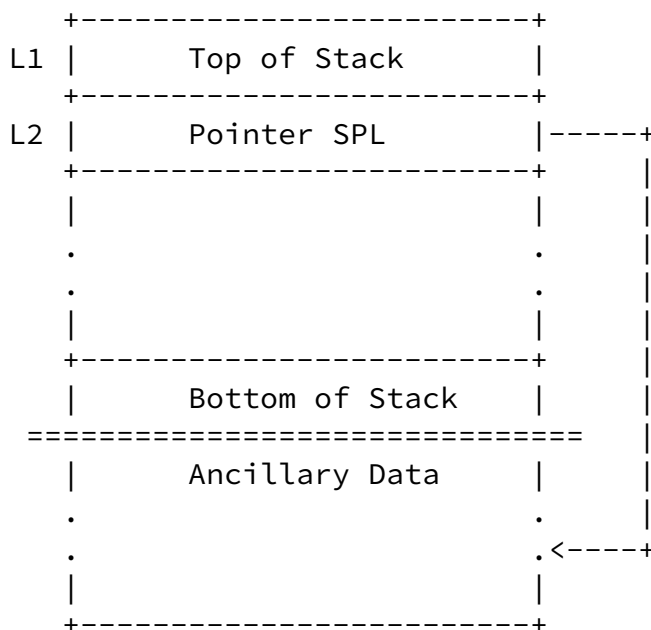


Figure 1: Use of In-stack MPLS pointer

The ToS label (L1) and Pointer SPL (L2) form a tuple with the semantic "process the action that the Forwarding Equivalence Class (FEC) of the ToS label specifies with the assistance of the information pointed to by the following SPL". Ideally L2 is SPL requiring a single LSE rather than an Extended SPL (ESPL) requiring two LSEs. Whilst the additional LSE required for an ESPL may not initially seem significant, the authors imagine that there may be cases where multiple pointer labels will be required.

Let us consider the case when the ToS is not an SPL of any kind. In this case, the forwarder looks at the following LSE (i.e., the LSE that immediately succeeds the ToS). If that LSE is not a pointer SPL, forwarding is performed as normal. If, on the other hand, the following label is a pointer SPL, the forwarder uses the information pointed to by the pointer as assistance in the forwarding operation.

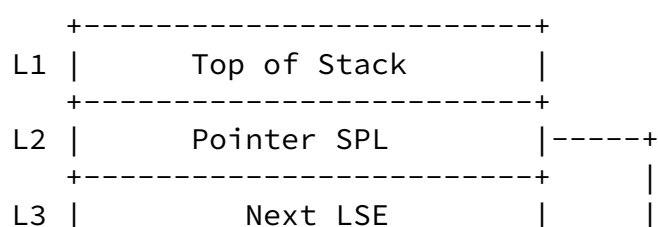
Note that whilst the pointer can be simply point to the end of the stack, which aligns with the other MPLS proposals being made, the ideas discussed here can actually point to a specific item within the MPLS payload i.e. to a specific item of ancillary data. This in turn also means that different LSEs can point to different ancillary data components. This allows the MPLS application or packet designer to express sophisticated behavior in which it is possibly to apply different ancillary data to different LSEs, i.e. different network segments.

[4.](#) Label Operations: Popping and Swapping

When the ToS is popped, consideration needs to be given to any Pointer SPL immediately following it.

In the basic case, a Pointer SPL will simply be popped along with the ToS.

There will be cases in in which the same Pointer SPL applies to multiple labels. In those cases, requiring the forwarder to pop the Pointer SPL along with the ToS results in the need to carry multiple instances of the same Pointer SPL, one for each label to which it applies. As an optimization, it will make sense to offer a second behavioral option in which, upon popping the ToS, any subsequent Pointer SPL will be swapped with the next FEC label. This case is depicted in Figure 2 and Figure 3.



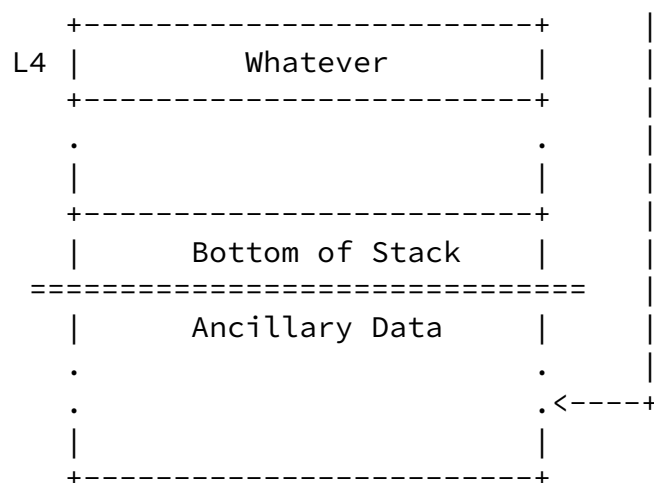
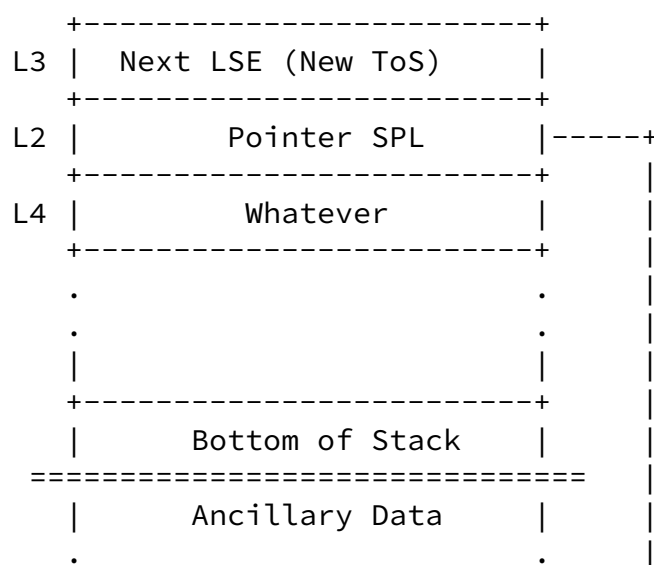


Figure 2: Before Pop - Swap operation:



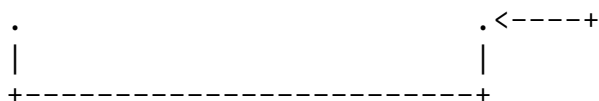


Figure 3: After Pop - Swap operation:

When this optimization is applied, there needs to be a distinction that allows a forwarder to determine whether a Pointer SPL should be popped along with its ToS, or whether it should be swapped with the next FEC label below.

One possibility is to indicate this in the FEC of the LSE that precedes the Pointer SPL, or perhaps it can be indicated by using one of its bits as a corresponding flag. Alternatively a perhaps some method can be found whereby a "TTL" can be associated with the pointer label. How to best indicate this end-of-use distinction is for further study.

5. Use of Multiple Pointers

One problem to be solved is how to support multiple independent (sets of) ancillary data in the MPLS header in support of different (forwarding, OAM etc) operations associated with the ancillary data. In IP/IPv6, ancillary data is encoded in the packet header through a sequence of Extension Headers (EH). For IPv6, [\[RFC8200\]](#) defines several EH types, each of which implies a specific set of nodes that have to process the EH and their order. This approach results in a complex parsing requirement for IPv6 packets when multiple extension headers are used and very rigid encoding and EH semantic difficult to extend.

Instead of limiting processing to a single pointer, it is possible to generalize the above concepts to allow for multiple pointers. This increases flexibility by allowing the packet designer to include

pointers to multiple sets of ancillary data, each of which can be potentially used for a different purpose. Therefore, the use of multiple adjacent Pointer SPLs is allowed. This means that ToS processing takes into considerations all Pointer SPLs that immediately follow.

A pointer mechanism in the MPLS label stack provides a method of using multiple pointers to express two (or more) sets of ancillary

data, for example a latency object and an iOAM object. An example is shown in Figure 4.

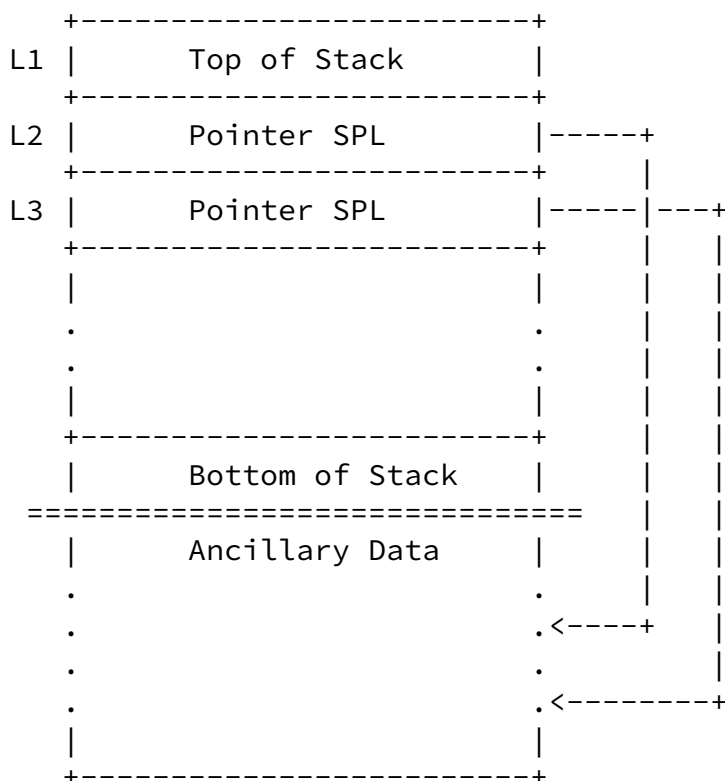


Figure 4: Use of Multiple In-stack MPLS Pointers

As in Figure 1 the top three labels are a tuple that in this case have the semantics "process the action that the FEC of the ToS label specifies with the assistance of the information pointed to by the following SPLs", in this case the label set L1, L2, L3. The tuple terminates when either an "ordinary" label, an SPL that is not a pointer SPL, or a label with the S bit set is encountered.

The Figure 5 further illustrates this capability. Here in this example two LSEs, Li and Lj, are each associated with two pointers. The FEC of Li therefore indicates that execution of Li includes the use of Ancillary Data 1 and 2, and execution of Lj includes the use of ancillary data 2 and 3.

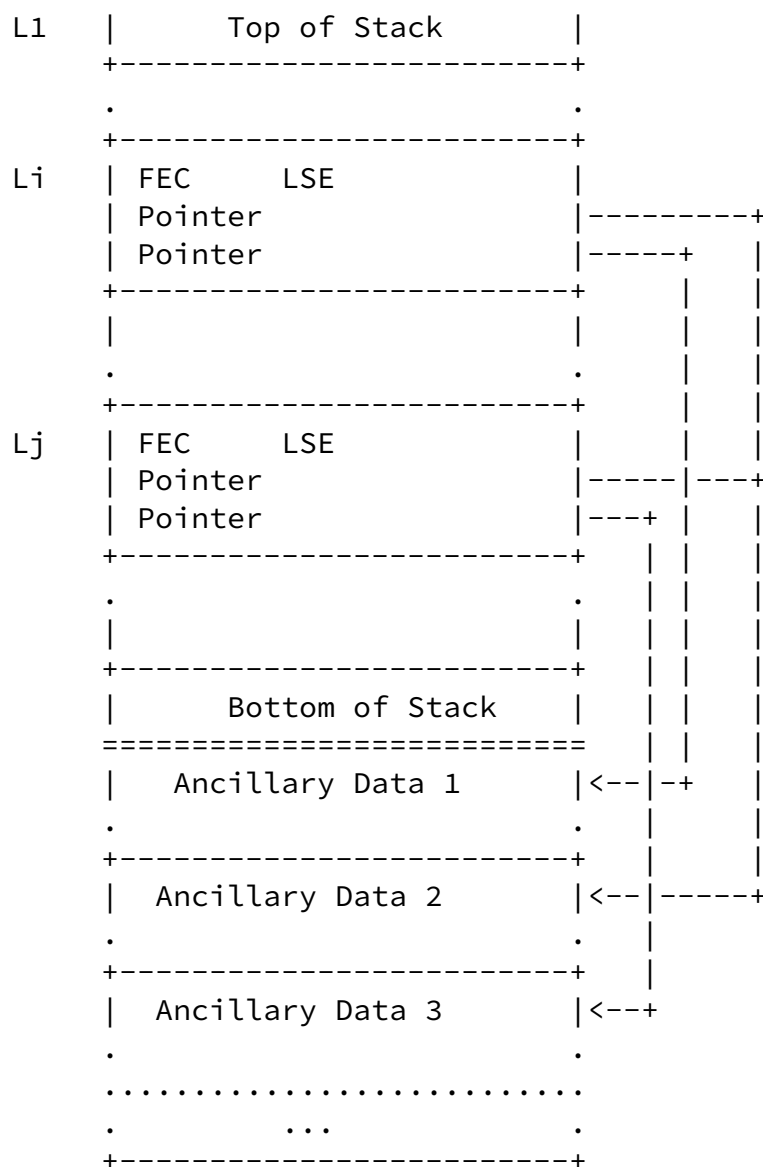


Figure 5: Further Example of Multiple In-stack MPLS Pointers

To support multiple Pointer SPLs, the following additional considerations apply:

The pop operation for the ToS needs to be extended to apply to the entire tuple of Pointer SPLs that are in its scope, i.e. that immediately succeed it. The default pop behavior will be to pop the entire tuple of Pointer SPLs along with the ToS.

As described earlier, as an optimization to reduce the size of the label stack, Pointer SPLs can be designated to not be popped but instead swapped with other LSEs in the stack. This will allow the same Pointer SPL set to be applied to multiple LSEs.

For an in stack swap operation where multiple pointers form a pointer set, the entire tuple, or group, would be swapped with the next FEC label below.

In addition, mixed cases are conceivable in which some Pointer SPLs are popped whereas others are swapped. Whether to pop or swap a Pointer SPL needs to be specified as part of the associated LSE's disposition behavior.

6. Disposition of the Ancillary Data

The ancillary data must be removed before the payload is passed out of the MPLS domain. There are three methods whereby the egress PE can know of the presence of the ancillary data:

- o The FEC of the BoS LSE can indicate the need to do this in a manner similar to pseudowires or MPLS VPN.
- o The BoS LSE can be a special purpose label indicating the presence of the ancillary data.
- o The BOS LSE can point to an item of ancillary data that describes the disposition of the ancillary data.

The removal of the ancillary data may be relatively complex depending on its purpose, i.e. it may be more complex than removing some number of bytes, for example, if it is carrying latency or iOAM information.

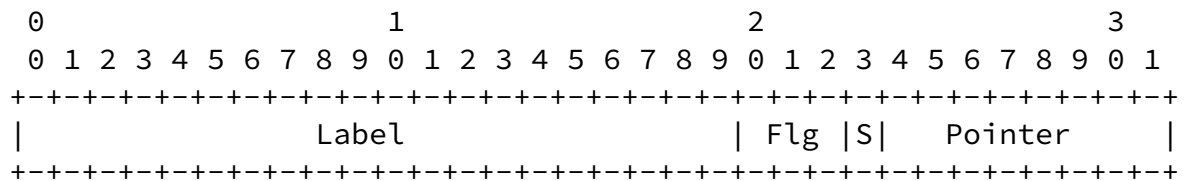
The structure and quantity of ancillary data including any methods whereby the ancillary data points to other ancillary, or whether there are pointer to the payload itself is out of scope for this document. Such information will need to be included in the ancillary data design so that it can be safely processed and/or removed.

7. Structure of Ancillary Data

The structure of the ancillary data is outside the scope of this memo.

8. Structure of Pointer Label

A possible structure for a pointer LSE is shown in Figure 6.



Label : contains the label that triggers the pointer behavior

Flg (Flags): Contains a number of flags that clarify the pointer

Bit 20: Size of pointer units, Bit 20 = 0 units are octets,
Bit = 1 units are 16 bit quantities

S : BoS as per {{RFC3032}}

Pointer : Pointer to the start of the specific ancillary data block.

Figure 6: MPLS Pointer LSE

The label is recognized by the forwarder as being the trigger for the pointer behavior. The pointer is the offset from the pointer LSE to the start of the auxiliary data that is to be used at this hop to process the ToS label. The pointer may be in units of octets or 16 bit words (or 32 bit words TBD) as specified by the flag. The S bit had its normal meaning in an MPLS LSE.

9. Backward Compatibility

If the LSP includes a legacy node that does not understand the pointer SPL it will forward based on the FEC of the ToS alone omitting the feature. If that feature absence results in a service shortfall for traffic on the LSP or MPLS-SR path then obviously the LSP or path has to be constructed to avoid any node that is unable to execute the feature. The various methods of constructing LSPs via LSRs with certain capabilities is well known routing technology and will not be further discussed in this memo.

10. Security Considerations

This proposal does not change the security of the MPLS data plane.

Normal operational practice is to prohibit the ingress of an MPLS packet from other than a trusted source. An attacker that breaches the physical security of an MPLS domain has many methods of attack by manipulating the label stack, and this mechanism does not significantly increase that risk.

[11.](#) IANA Considerations

This document has no IANA requests.

[12.](#) Appendix 1: CONTROVERSY ALERT - Use Of Ordinary Labels

Given the restricted number of Base SPLs [[RFC9017](#)] it is interesting to consider whether we might use an ordinary label for this purpose. At the time of writing there are eight out of 16 base SPLS still available. This is a dilemma since there is a protocol need for single label SPLs to support MPLS stack efficiency, but those that we have available must last the development lifetime of MPLS. On the other hand using a non-SPL has potential run-time/hardware issues if we need lots of them. However there probably exists a compromise number where we can safely allocate Base SPLs but not significantly impact the forwarder performance with this approach.

The label becomes a run-time constant that the forwarder needs to check during the parsing of the label stack. This is a 20 bit compare of a run-time constant. This is simple for a software or microcoded forwarder but needs a programmable register in a fully hardware based forwarder. Clearly from a protocol design perspective it is necessary to check the restrictions on the deployed hardware, but this certainly seems feasible. In deployment it will of course be necessary to verify that the routers along the LSP can support this feature before the LSP can be constructed.

If an ordinary label were assigned to this purpose from the 16-104857516 label set, there are two cases to consider: LSRs that have the capability of associating a FEC with a label of this value and LSRs that do not.

If an LSR has the capability to allocate a FEC with the chosen value it is necessary to preallocate this label before any MPLS application takes that value. This may impact a number of MPLS applications, but it seems feasible.

An LSR that does not have the capability to allocate a FEC with this value simply has the issue of adapting the forwarding behavior.

The matter of choosing a suitable value and distributing it is outside the scope of this memo, but is something that is routinely done in the routing system so is not a factor in assessing feasibility.

Clearly the LSP needs to be constructed so as to avoid any LSR that is unable to process a packet with one of these sequestered ordinary labels, but that is no different to the case where an SPL is used.

The final consideration is what happens if the label every becomes ToS. For this to happen the packet must have been incorrectly processed, and that is no different from any other case of a incorrectly processed MPLS packet.

[13.](#) Appendix 2: Other Issues for Discussion

This appendix briefly describes a number of issue that require further consideration.

1) Pointer labels as described earlier in this document are defined using an offset that is calculated from the pointer label. An alternative approach, given that we may not get rid of scanning for the BoS in MPLS header parsing, is to consider a design in which offsets were relative to the BoS instead. We could use this as a standard method, or we could specify this via a flag in the LSE carrying the offset.

The relative to BoS relative approach can be more efficient in some circumstances, e.g.: When the ancillary data is applicable to multiple hops of a label stack that is indicating a steering path, such as in SR-MPLS, the FEC of every steering hop label could indicate to "reuse" the ancillary data for every hop. The MPLS operation would then consist of a pop of the ToS label followed by a swap of the two top labels with each other, so that the following

steering label effectively gets pulled up as ToS.

2) To support pointers being valid across multiple hops, the pointer either needs to be indicated either as an offset relative to BoS, or the value of the pointer in the pointer-SPL needs to be adjusted by a swap operation.

3) The LSE pointer could include a lifetime indicating the number of times it is to be propagated. This raises the following issue.

4) The pointer mechanism has to be able to deal with multiple instances of ancillary data applicable to specific elements within the LSP. So it is important to know when to stop propagating any pointer if that approach is adopted (instead of, for example adopting an approach of one pointer LSE per ToS label.

5) We need to decide of correct name for pointer SPL.

[14.](#) Appendix 3: Ancillary vs Auxiliary vs Metadata

From the Oxford English Dictionary:

Bryant, et al.

Expires November 19, 2021

[Page 12]

Internet-Draft

MPLS-Aux-Ptr

May 2021

- o Ancillary: Designating activities and services that provide essential support to the functioning of a central service or industry;
- o Auxiliary: Helpful, assistant, affording aid, rendering assistance, giving support or succour(sic).
- o Metadata(sic): data whose purpose is to describe and give information about other data.

The two terms ancillary and auxiliary are similar but the additional qualifier that ancillary is `_essential_` support make it, in the author's view, the preferred term.

Metadata, the term often used in the technical discussions does appear to be sufficiently descriptive of the purpose of this information that is included in the packet.

15. References

15.1. Normative References

- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [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>>.

15.2. Informative References

- [I-D.kompella-mpls-mspl4fa]
Kompella, K., Beeram, V. P., Saad, T., and I. Meilik, "Multi-purpose Special Purpose Label for Forwarding Actions", [draft-kompella-mpls-mspl4fa-00](#) (work in progress), February 2021.
- [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>>.

Bryant, et al.

Expires November 19, 2021

[Page 13]

Internet-Draft

MPLS-Aux-Ptr

May 2021

Authors' Addresses

Stewart Bryant
Futurewei Technologies, Inc.

Email: sb@stewartbryant.com

Alexander Clemm
Futurewei Technologies, Inc.

Email: ludwig@clemm.org

Toerless Eckert
Futurewei Technologies, Inc.

Email: tte@cs.fau.de