

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

S. Kini, Ed.
Ericsson
K. Kompella
Juniper
S. Sivabalan
Cisco
February 14, 2014

Entropy labels for source routed stacked tunnels
draft-kini-mpls-spring-entropy-label-00

Abstract

Source routed tunnel stacking is a technique that can be leveraged to provide a method to steer a packet through a controlled set of segments. This can be applied to the Multi Protocol Label Switching (MPLS) data plane. Entropy label (EL) is a technique used in MPLS to improve load balancing. This document examines how ELs are to be applied to source routed stacked tunnels.

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 <http://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 18, 2014.

Copyright Notice

Copyright (c) 2014 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 (<http://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

Internet-Draft EL for source routed stacked tunnels February 2014

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.	Requirements Language	3
2.	Abbreviations and Terminology	3
3.	Entropy Labels for source routed stacked tunnels	3
3.1.	Single EL at the bottom of the stack of tunnels	4
3.2.	An EL per tunnel in the stack	4
3.3.	A re-usable EL for a stack of tunnels	5
3.3.1.	EL at top of stack	5
3.4.	ELs at readable label stack depths	5
4.	Acknowledgements	6
5.	Contributors	6
6.	Change Log	6
7.	IANA Considerations	6
8.	Security Considerations	6
9.	References	6
9.1.	Normative References	6
9.2.	Informative References	7
9.3.	URIs	8
	Authors' Addresses	8

[1.](#) Introduction

The source routed stacked tunnels paradigm is leveraged by techniques such as Segment Routing (SR) [[I-D.filsfils-rtgwg-segment-routing](#)] to steer a packet through a set of segments. This can be directly applied to the MPLS data plane, but it has implications on label stack depth.

Clarifying statements on label stack depth have been provided in [[I-D.ietf-mpls-forwarding](#)] but they do not address the case of source routed stacked tunnels as described in [[I-D.gredler-spring-mpls](#)] or [[I-D.filsfils-rtgwg-segment-routing](#)] where deeper label stacks are more prevalent.

Entropy label (EL) [[RFC6790](#)] is a technique used by the MPLS data plane to do load balancing. When using LSP hierarchies there are

implications on how [\[RFC6790\]](#) should be applied. One such issue is addressed by [\[I-D.ravisingh-mpls-el-for-seamless-mpls\]](#) but that is when different levels of the hierarchy are at different routers. This draft addresses the case where the hierarchy is created at a

single LSR as required by source stacked tunnels. The different options and their issues are discussed in [Section 3](#).

[1.1](#). Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

[2](#). Abbreviations and Terminology

EL - Entropy Label

ELI - Entropy Label Identifier

SR - Segment Routing

ECMP - Equal Cost Multi Paths

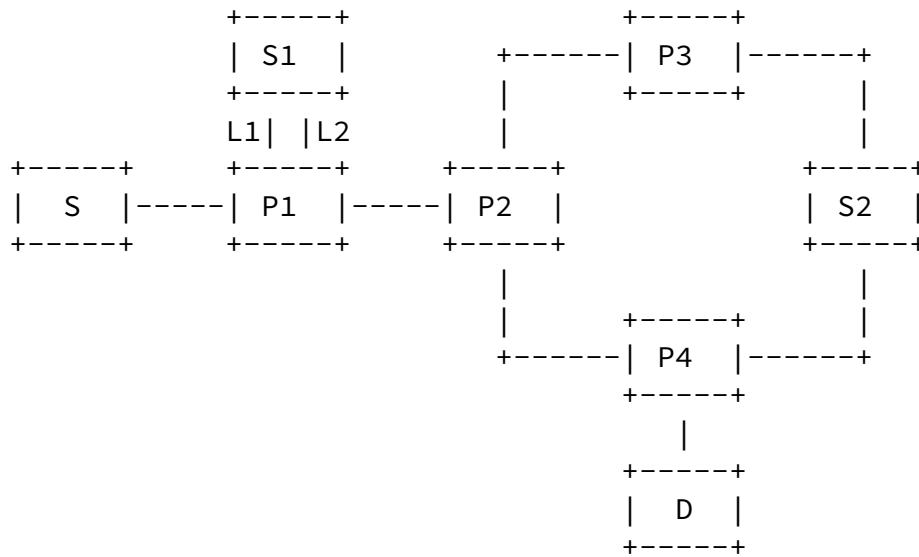
MPLS - Multi Protocol Label Switching

SID - Segment Identifier

[3](#). Entropy Labels for source routed stacked tunnels

Stacked tunnels have several use-cases, one of which is service chaining [\[I-D.filsfils-rtgwg-segment-routing-use-cases\]](#). Consider a service-chaining network in Figure 1. The source LSR S wants to send traffic to destination LSR D. This traffic is required to go through service nodes S1 and S2 to produce the service chain S-S1-S2-D. Segment Routing can be used to achieve this. Load balancing is required across the parallel links between P1 and S1. Load balancing is also required between the ECMP paths from S1 to S2, S1-P1-P2-P3-S2 and S1-P1-P2-P4-S2. The source LSR wants the intermediate LSRs P1 and P2 to take local load balancing decisions and does not specify the Segment Identifiers (SIDs) of specific interfaces. Entropy

labels should be used to achieve the desired load balancing. Two possible ways to use the entropy labels and their associated tradeoffs are discussed below. We denote SN to be the node segment identifier (SID) of LSR N and SN{L1,L2,...} to denote the SID of the adjacency set for links {L1,L2,...} of LSR N and S-N to denote the SID for a service at service node N. The label stack that the source LSR S uses for the service chain can be <SS1, S-S1, SS2, S-S2, SD> or <SP1, SP1{L1,L2}, S-S1, SS2, S-S2, SD>. The issues discussed in this document are equally applicable to both of these options.



S=Source LSR, D=Destination LSR, S1,S2=service-nodes, L1,L2=links,
P1,P2,P3,P4=Transit LSRs

Figure 1: Service chaining use-case

3.1. Single EL at the bottom of the stack of tunnels

In this option a single EL is used for the entire label stack. The source LSR S encodes the entropy label (EL) below the labels of all the stacked tunnels. In Figure 1 label stack at LSR S would look like <SP1, SP1{L1,L2}, SS1, S-S1, SS2, S-S2, SD, ELI, EL> <remaining packet header>. Note that the notation in [RFC6790] is used to

describe the label stack. An issue with this approach is that as the label stack grows due an increase in the number of SIDs, the EL correspondingly goes deeper in the label stack. As a result, intermediate LSRs (such as P1) that have to walk the label stack at least until the EL to perform load balancing decisions have to access a larger number of bytes in the packet header when making forwarding decisions. A network design using this approach, should ensure that all intermediate LSRs have the capability to traverse the maximum label stack depth in order to do effective load balancing. The use-case for which the tunnel stacking is applied would determine the maximum label stack depth.

[3.2.](#) An EL per tunnel in the stack

In this option each tunnel in the stack can be given its own EL. The source LSR pushes an <ELI, EL> before pushing a tunnel label when load balancing is required to direct traffic on that tunnel. For the same Figure 1 above, the source LSR S encoded label stack would be <SP1, SP1{L1,L2}, ELI, EL1, SS1, S-S1, SS2, ELI, EL2, SD> where all

the ELs would typically have the same value. Accessing the EL at an intermediate LSR is independent of the depth of the label stack and hence independent of the specific use-case to which the stacked tunnels are applied. A drawback is that the depth of the label stack grows significantly, almost 3 times as the number of labels in the label stack. The network design should ensure that source LSRs should have the capability to push such a deep label stack. Also, the bandwidth overhead and potential MTU issues of deep label stacks should be accounted for in the network design.

[3.3.](#) A re-usable EL for a stack of tunnels

In this option an LSR that terminates a tunnel re-uses the EL of the terminated tunnel for the next inner tunnel. It does this by storing the EL from the outer tunnel when that tunnel is terminated and re-inserting it below the next inner tunnel label during the label swap operation. The LSR that stacks tunnels SHOULD insert an EL below the outermost tunnel. It SHOULD NOT insert ELs for any inner tunnels. Also, the penultimate hop LSR of a segment MUST NOT pop the ELI and EL even though they are exposed as the top labels since the terminating LSR of that segment would re-use the EL for the next segment.

For the same Figure 1 above, the source LSR S encoded label stack would be <SP1, ELI, EL, SP1{L1,L2}, SS1, S-S1, SS2, SD>. At P1 the outgoing label stack would be <SS1, ELI, EL, S-S1, SS2, SD> after it has load balanced to one of the links L1 or L2. At S1 the outgoing label stack would be <SS2, ELI, EL, SD>. At P2 the outgoing label stack would be <SS2, ELI, EL, SD> and it would load balance to one of the nexthop LSRs P3 or P4. Accessing the EL at an intermediate LSR is independent of the depth of the label stack and hence independent of the specific use-case to which the stacked tunnels are applied.

[3.3.1.](#) EL at top of stack

A slight variant of the re-usable EL option is to keep the EL at the top of the stack rather than below the tunnel label. In this case each LSR that is not terminating a segment should continue to keep the received EL at the top of the stack when forwarding the packet along the segment. An LSR that terminates a segment should use the EL from the terminated segment at the top of the stack when forwarding onto the next segment. This option is under discussion.

[3.4.](#) ELs at readable label stack depths

In this option the source LSR inserts ELs for tunnels in the label stack at depths such that each LSR along the path that must load balance is able to access at least one EL. Note that the source LSR

may have to insert multiple ELs in the label stack at different depths for this to work since intermediate LSRs may have differing capabilities in accessing the depth of a label stack. The label stack depth access value of intermediate LSRs must be known to create such a label stack. How this value is determined is outside the scope of this document. This value can be advertised using a protocol such as an IGP. Details of this will follow in subsequent versions if this option is found to be worth pursuing. For the same Figure 1 above, if LSR P1 needs to have the EL within a depth of 4, then the source LSR S encoded label stack would be <SP1, SP1{L1,L2}, ELI, EL1, SS1, S-S1, SS2, ELI, EL2, SD> where all the ELs would typically have the same value.

[4.](#) Acknowledgements

The authors would like to thank Rob Shakir and John Drake for their comments.

[5.](#) Contributors

Stephane Litkowski

Email: stephane.litkowski@orange.com

[6.](#) Change Log

This draft is a renamed update of [\[I-D.kini-mpls-entropy-label-src-stacked-tunnels\]](#) that was presented at the IETF88 meeting in Vancouver [\[1\]](#).

[7.](#) IANA Considerations

This memo includes no request to IANA.

[8.](#) Security Considerations

[9.](#) References

[9.1.](#) Normative References

[I-D.filsfils-rtgwg-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", [draft-filsfils-rtgwg-segment-routing-use-cases-02](#) (work in progress), October 2013.

Kini, et al.

Expires August 18, 2014

[Page 6]

Internet-Draft

EL for source routed stacked tunnels

February 2014

[I-D.filsfils-rtgwg-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", [draft-filsfils-rtgwg-segment-routing-01](#) (work in progress), October 2013.

[I-D.gredler-spring-mpls]

Gredler, H., Rekhter, Y., Jalil, L., and S. Kini, "Supporting Source/Explicitly Routed Tunnels via Stacked LSPs", [draft-gredler-spring-mpls-02](#) (work in progress), October 2013.

[I-D.ietf-mpls-forwarding]

Villamizar, C., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", [draft-ietf-mpls-forwarding-07](#) (work in progress), February 2014.

[I-D.kini-mpls-entropy-label-src-stacked-tunnels]

Kini, S., Kompella, K., and S. Sivabalan, "Entropy labels for source routed stacked tunnels", [draft-kini-mpls-entropy-label-src-stacked-tunnels-01](#) (work in progress), August 2013.

[I-D.ravisingh-mpls-el-for-seamless-mpls]

Singh, R., Shen, Y., and J. Drake, "Entropy label for seamless MPLS", [draft-ravisingh-mpls-el-for-seamless-mpls-01](#) (work in progress), October 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", [RFC 6790](#), November 2012.

[9.2](#). Informative References

[I-D.previdi-isis-segment-routing-extensions]

Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., and J. Tantsura, "IS-IS Extensions for Segment Routing", [draft-previdi-isis-segment-routing-extensions-05](#) (work in progress), February 2014.

[I-D.psenak-ospf-segment-routing-extensions]

Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., and W. Henderickx, "OSPF Extensions for Segment Routing", [draft-psenak-ospf-segment-routing-extensions-04](#) (work in progress), February 2014.

9.3. URIs

[1] <http://www.ietf.org/proceedings/88/slides/slides-88-spring-0.pdf>

Authors' Addresses

Sriganesh Kini (editor)
Ericsson

Email: sriganesh.kini@ericsson.com

Kireeti Kompella
Juniper

Email: kireeti@juniper.net

Siva Sivabalan
Cisco

Email: msiva@cisco.com