Network Working Group S. Kini, Ed.
Internet-Draft Ericsson
Intended status: Informational K. Kompella
Expires: March 02, 2014 Juniper
S. Sivabalan
Cisco
August 29, 2013

Entropy labels for source routed stacked tunnels
draft-kini-mpls-entropy-label-src-stacked-tunnels-01

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.

Table of Contents

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. Entropy labels (EL) [RFC6790] is a technique used by the MPLS data plane to do load balancing. Applying ELs to stacked tunnels brings up some issues and these are documented 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.

```
                  +-----+             +-----+
                  | S1  |       +------| P3  |------+
                  +-----+       |      +-----+      |
                  L1| |L2       |                   |
      +-----+     +-----+     +-----+             +-----+
      |  S  |-----| P1  |-----| P2  |             | S2  |
      +-----+     +-----+     +-----+             +-----+
                                |                   |
                                |      +-----+      |
                                +------| P4  |------+
                                       +-----+
                                          |
                                       +-----+
                                       |  D  |
                                       +-----+
```

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. 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.

3.4. 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 TBD for their comments.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

7. References

7.1. 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., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-rtgwg-segment-routing-use-cases-01 (work in progress), July 2013.

[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-00 (work in progress), June 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.

7.2. Informative References

[I-D.previdi-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., and S. Litkowski, "IS-IS Extensions for Segment Routing", draft-previdi-isis-segment-routing-extensions-02 (work in progress), July 2013.

[I-D.psenak-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., and R. Shakir, "OSPF Extensions for Segment Routing", draft-psenak-ospf-segment-routing-extensions-02 (work in progress), July 2013.

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