Network Working Group   S. Kini, Ed.
Internet-Draft   Ericsson
Intended status: Standards Track   K. Kompella
Expires: July 29, 2016   Juniper
                                                        S. Sivabalan
                                                               Cisco
                                                        S. Litkowski
                                                              Orange
                                                           R. Shakir

                                                             X. Xu
                                                             Huawei
                                                        W. Hendrickx
                                                      Alcatel-Lucent
                                                         J. Tantsura
                                                            Ericsson
                                                    January 26, 2016

Entropy labels for source routed tunnels with label stacks
draft-ietf-mpls-spring-entropy-label-02

Abstract

   Source routed tunnels with label 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 and
   describes how ELs are to be applied to source routed tunnels with
   label stacks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The source routed tunnels with label stacking paradigm is leveraged
by techniques such as Segment Routing (SR)
[I-D.ietf-spring-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 the label stack depth.

Clarifying statements on label stack depth have been provided in
[RFC7325] but the RFC does not address the case of source routed
stacked MPLS tunnels as described in

   [I-D.ietf-spring-segment-routing] where deeper label stacks are more
   prevalent.

   Entropy label (EL) [RFC6790] is a technique used in the MPLS data
   plane to provide entropy for load balancing. When using LSP
   hierarchies there are implications on how [RFC6790] should be
   applied. The current document addresses the case where the hierarchy
   is created at a single LSR as required by source routed tunnels with
   label stacks.

   A use-case requiring load balancing with source routed tunnels with
   label stacks is given in Section 3. A recommended solution is
   described in Section 4 keeping in consideration the limitations of
   implementations when applying [RFC6790] to deeper label stacks.
   Options that were considered to arrive at the recommended solution
   are documented for historical purposes in Section 5.

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].

   Although this document is not a protocol specification, the use of
   this language clarifies the instructions to protocol designers
   producing solutions that satisfy the requirements set out in this
   document.

2. Abbreviations and Terminology

      EL - Entropy Label

      ELI - Entropy Label Identifier

      ELC - Entropy Label Capability

      SR - Segment Routing

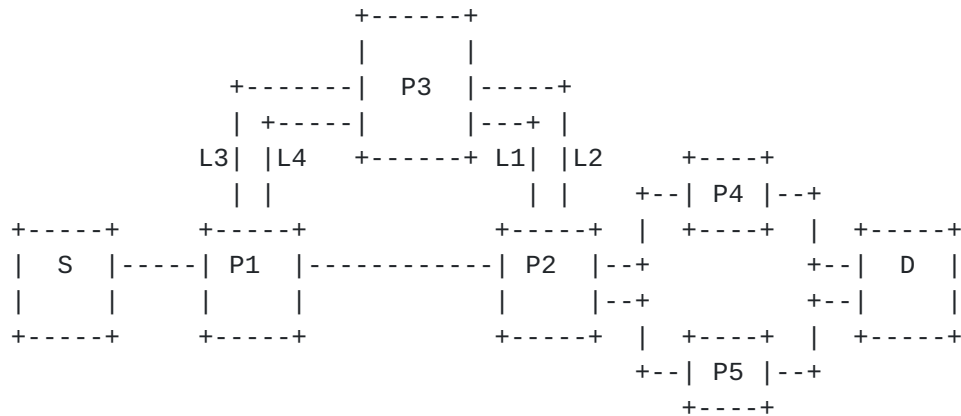   ECMP - Equal Cost Multi Paths

      MPLS - Multiprotocol Label Switching

      SID - Segment Identifier

      RLD - Readable Label Depth

      OAM - Operation, Administration and Maintenance

3. Use-case requiring multipath load balancing

```
                      +------+
                      |      |
              +-------|  P3  |-----+
              | +-----|      |---+ |
           L3| |L4   +------+ L1| |L2     +----+
             | |               | |    +--| P4 |--+
    +-----+     +-----+         +-----+  | +----+  |  +-----+
    |  S  |-----| P1  |------------| P2  |--+        +--|  D  |
    |     |     |     |            |     | |--+        +--|     |
    +-----+     +-----+         +-----+  |  +----+  |  +-----+
                                    +--| P5 |--+
                                       +----+
```

          S=Source LSR, D=Destination LSR, P1,P2,P3,P4,P5=Transit LSRs,
                            L1,L2,L3,L4=Links

                      Figure 1: Traffic engineering use-case

   Traffic-engineering (TE) is one of the applications of MPLS and is
   also a requirement for source routed tunnels with label stacks.
   Consider the topology shown in Figure 1. Lets say the LSR P1 has a
   limitation that it can only look four labels deep in the stack to do
   multipath decisions. All other transit LSRs in the figure can read
   deep label stacks and the LSR S can insert as many <ELI, EL> pairs as
   needed. The LSR S requires data to be sent to LSR D along a traffic-
   engineered path that goes over the link L1. Good load balancing is
   also required across equal cost paths (including parallel links). To
   engineer traffic along a path that takes link L1, the label stack
   that LSR S creates consists of a label to the node SID of LSR P3,
   stacked over the label for the adjacency SID of link L1 and that in
   turn is stacked over the label to the node SID of LSR D. For
   simplicity lets assume that all LSRs use the same label space for
   source routed label stacks. Lets L_N-P denote the label to be used
   to reach the node SID of LSR P. Let L_A-Ln denote the label used for
   the adjacency SID for link Ln. The LSR S must use the label stack
   <L_N-P3, L_A-L1, L_N-D> for traffic-engineering. However to achieve
   good load balancing over the equal cost paths P2-P4-D, P2-P5-D and
   the parallel links L3, L4, a mechanism such as Entropy labels
   [RFC6790] should be adapted for source routed label stacks. Multiple
   ways to apply entropy labels were considered and are documented in
   Section 5 along with their tradeoffs. A recommended solution is
   described in Section 4.

4. **Recommended EL solution for SPRING**

   The solution described in this section follows [RFC6790].

   An LSR may have a limitation in its ability to read and process the
   label stack in order to do multipath load balancing. This limitation
   expressed in terms of the number of label stack entries that the LSR
   can read is henceforth referred to as the Readable Label Depth (RLD)
   capability of that LSR. If an EL does not occur within the RLD of an
   LSR in the label stack of the MPLS packet that it receives, then it
   would lead to poor load balancing at that LSR. The RLD of an LSR is
   a characteristic of the forwarding plane of that LSR's implementation
   and determining it is outside the scope of this document.

   In order for the EL to occur within the RLD of LSRs along the path
   corresponding to a label stack, multiple <ELI, EL> pairs MAY be
   inserted in the label stack as long as the tunnel's label below which
   they are inserted are advertised with entropy label capability
   enabled. The LSR that inserts <ELI, EL> pairs MAY have limitations
   on the number of such pairs that it can insert and also the depth at
   which it can insert them. If due to any limitation, the inserted ELs
   are at positions such that an LSR along the path receives an MPLS
   packet without an EL in the label stack within that LSR's RLD, then
   the load balancing performed by that LSR would be poor. Special
   attention should be paid when a forwarding adjacency LSP (FA-LSP)
   [RFC4206] is used as a link along the path of a source routed LSP's
   label stack, since the labels of the FA-LSP would additionally count
   towards the depth of the label stack when calculating the appropriate
   positions to insert the ELs. The recommendations for inserting <ELI,
   EL> pairs are:

   o An LSR that is limited in the number of <ELI, EL> pairs that it
      can insert SHOULD insert such pairs deeper in the stack.

   o An LSR SHOULD try to insert <ELI, EL> pairs at positions so that
      for the maximum number of transit LSRs, the EL occurs within the
      RLD of the incoming packet to that LSR.

   o An LSR SHOULD try to insert the minimum number of such pairs while
      trying to satisfy the above criteria.

   A sample algorithm to insert ELs is shown below. Implementations can
   choose any algorithm as long as it follows the above recommendations.

```
   Initialize the current EL insertion point to the
     bottommost label in the stack that is EL-capable
   while (local-node can push more <ELI,EL> pairs OR
             insertion point is not above label stack) {
      insert an <ELI,EL> pair below current insertion point
      move new insertion point up from current insertion point until
         ((last inserted EL is below the RLD) AND (RLD > 2)
                         AND
         (new insertion point is EL-capable))
      set current insertion point to new insertion point
   }
```

       Figure 2: Algorithm to insert <ELI, EL> pairs in a label stack

   When this algorithm is applied to the example described in Section 3
   it will result in ELs being inserted in two positions, one below the
   label L_N-D and another below L_N-P3. Thus the resulting label stack
   would be <L_N-P3, ELI, EL, L_A-L1, L_N-D, ELI, EL>

   The RLD can be advertised via protocols and those extensions are
   described in separate documents [I-D.xu-isis-mpls-elc] and
   [I-D.xu-ospf-mpls-elc].

   The recommendations above are not expected to bring any additional
   OAM considerations beyond those described in section 6 of [RFC6790].
   However, the OAM requirements and solutions for source routed tunnels
   formed by label stacking are still under discussion and future
   revisions of this document will address those if needed.

5. Options considered

   Different options that were considered to arrive at the recommended
   solution are documented in this section.

5.1. 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) at the bottom of the
   label stack. In the example described in Section 3 it will result in
   the label stack at LSR S to look like <L_N-P3, L_A-L1, L_N-D, 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 goes correspondingly deeper in the label stack. Hence transit
   LSRs have to access a larger number of bytes in the packet header
   when making forwarding decisions. In the example described in
   Section 3 the LSR P1 would poorly load-balance traffic on the
   parallel links L3, L4 since the EL is below the RLD of the packet

   received by P1. A load balanced network design using this approach
   must ensure that all intermediate LSRs have the capability to
   traverse the maximum label stack depth as required for that
   application that uses source routed stacking.

   In the case where the hardware is capable of pushing a single <ELI,
   EL> pair at any depth, this option is the same as the recommended
   solution in Section 4.

   This option was discounted since there exist a number of hardware
   implementations which have a low maximum readable label depth.
   Choosing this option can lead to a loss of load-balancing using EL in
   a significant part of the network but that is a critical requirement
   in a service provider network.

5.2. 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. In the
   example described in Section 3, the source LSR S encoded label stack
   would be <L_N-P3, ELI, EL, L_A-L1, L_N-D, ELI, EL> where all the ELs
   can be the same. Accessing the EL at an intermediate LSR is
   independent of the depth of the label stack and hence independent of
   the specific application that uses source routed tunnels with label
   stacking in that network. 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.

   In the case where the RLD is the minimum value (3) for all LSRs, all
   LSRs are EL capable and the LSR that is inserting <ELI, EL> pairs has
   no limit on how many it can insert then this option is the same as
   the recommended solution in Section 4.

   This option was discounted due to the existence of hardware
   implementations that can push a limited number of labels on the label
   stack. Choosing this option would result in a hardware requirement
   to push two additional labels per tunnel label. Hence it would
   restrict the number of tunnels that can form a LSP and constrain the
   types of LSPs that can be created. This was considered unacceptable.

5.3. 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.

   In Section 3 above, the source LSR S encoded label stack would be
   <L_N-P3, ELI, EL, L_A-L1, L_N-D>. At P1 the outgoing label stack
   would be <L_N-P3, ELI, EL, L_A-L1, L_N-D> after it has load balanced
   to one of the links L3 or L4. At P3 the outgoing label stack would
   be <L_N-D, ELI, EL>. At P2 the outgoing label stack would be <L_N-D,
   ELI, EL> and it would load balance to one of the nexthop LSRs P4 or
   P5. Accessing the EL at an intermediate LSR (e.g. P1) is
   independent of the depth of the label stack and hence independent of
   the specific use-case to which the label stack is applied.

   This option was discounted due to the significant change in label
   swap operations that would be required for existing hardware.

5.3.1. 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 was discounted due to the significant change in label
   swap operations that would be required for existing hardware.

5.4. 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. For the same Section 3 above, if LSR P1
   needs to have the EL within a depth of 4, then the source LSR S
   encoded label stack would be <L_N-P3, ELI, EL, L_A-L1, L_N-D, ELI,
   EL> where all the ELs would typically have the same value.

   In the case where the RLD has different values along the path and the
   LSR that is inserting <ELI, EL> pairs has no limit on how many pairs
   it can insert, and it knows the appropriate positions in the stack
   where they should be inserted, then this option is the same as the
   recommended solution in Section 4.

   A variant of this solution was selected which balances the number of
   labels that need to be pushed against the requirement for entropy.

6. Acknowledgements

   The authors would like to thank John Drake, Loa Andersson, Curtis
   Villamizar, Greg Mirsky, Markus Jork, Kamran Raza and Nobo Akiya for
   their review comments and suggestions.

7. IANA Considerations

   This memo includes no request to IANA. Note to RFC Editor: Remove
   this section before publication.

8. Security Considerations

   This document does not introduce any new security considerations
   beyond those already listed in [RFC6790].

9. References

9.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and
              L. Yong, "The Use of Entropy Labels in MPLS Forwarding",
              RFC 6790, DOI 10.17487/RFC6790, November 2012,
              <http://www.rfc-editor.org/info/rfc6790>.

9.2. Informative References

   [I-D.ietf-spring-segment-routing]
             Filsfils, C., Previdi, S., Decraene, B., Litkowski, S.,
             and r. rjs@rob.sh, "Segment Routing Architecture", draft-
             ietf-spring-segment-routing-07 (work in progress),
             December 2015.

   [I-D.xu-isis-mpls-elc]
             Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S.
             Litkowski, "Signaling Entropy Label Capability Using IS-
             IS", draft-xu-isis-mpls-elc-02 (work in progress), April
             2015.

   [I-D.xu-ospf-mpls-elc]
             Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S.
             Litkowski, "Signaling Entropy Label Capability Using
             OSPF", draft-xu-ospf-mpls-elc-01 (work in progress),
             October 2014.

   [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP)
             Hierarchy with Generalized Multi-Protocol Label Switching
             (GMPLS) Traffic Engineering (TE)", RFC 4206,
             DOI 10.17487/RFC4206, October 2005,
             <http://www.rfc-editor.org/info/rfc4206>.

   [RFC7325] Villamizar, C., Ed., Kompella, K., Amante, S., Malis, A.,
             and C. Pignataro, "MPLS Forwarding Compliance and
             Performance Requirements", RFC 7325, DOI 10.17487/RFC7325,
             August 2014, <http://www.rfc-editor.org/info/rfc7325>.

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

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Rob Shakir

Email: rjs@rob.sh

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Wim Hendrickx
Alcatel-Lucent

Email: wim.henderickx@alcatel-lucent.com

Jeff Tantsura
Ericsson

Email: jeff.tantsura@ericsson.com