Network Working Group                          Hannes Gredler
Internet Draft                                 Juniper Networks
Intended status: Standards Track
Expires: November 2014                          Yakov Rekhter
                                               Juniper Networks

                                                  Luay Jalil
                                                     Verizon

                                               Sriganesh Kini
                                                    Ericsson

                                                  Xiaohu Xu
                                                     Huawei

                                                  May 21 2014


        **Supporting Source/Explicitly Routed Tunnels via Stacked LSPs**


                    draft-gredler-spring-mpls-06.txt

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that other
   groups may also distribute working documents as Internet-Drafts.

   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."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

This document describes how source/explicitly routed tunnels could be
realized using stacked Label Switched Paths (LSPs).

This document also describes how use of IS-IS/OSPF as a label
distribution protocol fits into the MPLS architecture.

Table of Contents

## 1. Specification of Requirements

   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. Terminology

   We use the term "explicitly routed tunnels" as a synonym for such
   terms as "source routed tunnels" and "source-initiated routed
   tunnels".

   Note that the term "source routed tunnel", or "source-initiated
   routed tunnel" does not imply that intermediate nodes of such a
   tunnel forward packets traversing the tunnel based upon source
   addresses of these packets. In the context of "source routed tunnels"
   and "source-initiated routed tunnels" the term "source" refers to the

tunnels' ingress.

This document assumes that a reader is familiar with the MPLS
architecture [RFC3031] terminology.

For a given Label Switched Path (LSP) of level m, as defined in
section 3.15 of [RFC3031]:

  + the ingress node/router is the LSR that pushes the level m label,

  + intermediate nodes/routers are the LSRs making their forwarding
    decision on a level m label

## 3. Introduction

MPLS architecture [RFC3031] defines the concept of explicitly routed
tunnel as follows:

    If a Tunneled Packet travels from Ru to Rd over a path other
    than the Hop-by-hop path, we say that it is in an "Explicitly
    Routed Tunnel"

where Ru and Rd are Label Switch Routers (LSRs).

To realize explicitly routed tunnels [RFC3031] proposes to use
explicitly routed Label Switched Paths (LSPs):

    An "Explicitly Routed LSP Tunnel" is a LSP Tunnel that is also an
    Explicitly Routed LSP

Up until now there have been two possible protocols to instantiate/signal
such explicitly routed LSPs - RSVP-TE ([RFC3209]) and CR-LDP
([RFC3212]).

MPLS architecture ([RFC3031]) defines the notion of LSP hierarchy,
as LSP tunnels within LSPs. Use of MPLS label stack mechanism allows
LSP hierarchy to nest to any depth.

In this document we specify the procedures to realize explicitly
routed point-to-point tunnels by using LSP hierarchy, thus defining
yet another possible mechanism to realize such tunnels. (Note though
that the idea of using LSP hierarchy to realize explicitly routed
tunnels is not new - e.g., Remote LFA [R-LFA] uses explicitly routed
tunnels constructed by LSP hierarchy.)

An essential part of MPLS is the notion of label distribution
protocol. On the subject of whether it should be one, or more then
one label distribution protocol, MPLS architecture ([RFC3031]) said
the following:

   THE ARCHITECTURE DOES NOT ASSUME THAT THERE IS ONLY A SINGLE
   LABEL DISTRIBUTION PROTOCOL. In fact, a number of different
   label distribution protocols are being standardized.

Up until now IETF standardized the following label distribution
protocols for unicast: LDP ([RFC5036]), CR-LDP ([RFC3212]), RSVP-TE
[RFC3209] and BGP ([RFC3107], [RFC4364], [RFC4761]).

Recently there have been proposals ([gredler-isis], [gredler-ospf],
[previdi-isis], [psenak-ospf]) to extend IS-IS [RFC1142] and OSPF
[RFC1583] to make them yet another label distribution protocols.

This document describes how use of IS-IS or OSPF as label
distribution protocols fits into the MPLS architecture. This document
also describes the benefits of using IS-IS/OSPF as label distribution
protocols for the purpose of constructing explicitly routed tunnels
with stacked LSPs.


## 4. Constructing Explicitly Routed Tunnels by using Stacked LSPs

Instead of explicitly routed LSPs, one can use LSP hierarchy (stack
of LSPs) to construct explicitly routed point-to-point tunnels as
follows.

Consider an explicitly routed point-to-point tunnel with an explicit
route <R(0), R(1), R(2), ... R(n)>, where R(0) is the ingress of the
tunnel and R(n) is the egress of the tunnel. Denote the LSPs needed
to realize such a tunnel via an LSP stack as <LSP(1), LSP(2), ...
LSP(n)>, where LSP(1) is the topmost and LSP(n) is the bottommost LSP
in the stack. These LSPs are constructed as follows:

  + All the LSPs in the stack are constructed with the same ingress -
    R(0).  (See further down on why this is needed.)

  + LSP(i) is constructed with R(i) as its egress (e.g., LSP(1) is
    constructed with R(1) as its egress, LSP(2) with R(2) as its
    egress, etc...  LSP(n) with R(n) as its egress).

  + For every 0 < i < n, the first intermediate router of LSP(i+1).
    is constructed to be the same as the egress router of LSP(i).

  + The first intermediate router of LSP(1) is constructed to be one
    hop away from R(0). If R(1) is one hop away from R(0), then this
    intermediate router is also the egress of LSP(1) (in which case
    LSP(1) is a one-hop LSP). If R(1) is more than one hop away from
    R(0), then this intermediate router is some router other than
    R(1), and R(1) is still the egress of that LSP.

  + The first intermediate router of any LSP in the stack, could be
    either single or multi-hop away from the egress of that LSP.

  + All the LSPs in the stack are constructed with the penultimate
    hop popping. That is, for each LSP in the stack the penultimate
    router of that LSP pops the label corresponding to that LSP off
    the label stack before sending data to the egress router of that
    LSP.


When R(i) and R(i+1) are single hop away from each other, the first
intermediate router of LSP(i+1) is one hop away from the egress of
that LSP. When R(i) and R(i+1) are multi-hop away from each other,
the first intermediate router of LSP(i+1) is multi-hop away from the
egress of that LSP.

Following the above procedures, the LSPs in the stack satisfy the
following properties:

  + All LSPs in the stack have the same ingress.

  + The egress of a given LSP in the stack is the first intermediate
    router of the next LSP in the stack.

  + The first intermediate router of the LSP at the top of the stack
    is one hop away from the ingress.

  + The first intermediate router of any LSP in the stack could be
    either single or multi-hop away from the egress of that LSP.
    (Thus the egress of a given LSP in the stack could be either
    single or multi-hop away from the egress of the next LSP in the
    stack.)


Such stack of LSPs provides the functionality to forward a packet
through a sequence of egresses of the LSPs on the stack - the
sequence of these egresses represents the explicit route of the
explicitly routed point-to-point tunnel constructed by using these
stacked LSPs. The ingress of all these LSPs is the ingress of the
tunnel.

When the first intermediate router of a given LSP in the stack is
multi-hop away from the egress of that LSP, the existing label
distribution protocols (LDP, RSVP-TE, etc. ) can be used to establish
a multi-hop LSP fragment for this LSP. When IS-IS or OSPF, in
addition to being a routing protocol, is also used as a label
distribution protocol (see section "IS-IS or OSPF as Label
Distribution Protocol"), it can also be used to establish such multi-
hop LSP fragment.

To construct the label stack associated with the stack of LSPs the
ingress of all these LSPs, R0, uses the following procedures:

  + For (n > i > 0) R(0) obtains from R(i) label binding for LSP(i+1)
    and places the label onto the stack, starting from the bottommost
    label (the label that corresponds to LSP(n).

    In section "IS-IS or OSPF as Label Distribution Protocol" we
    describe how IS-IS or OSPF with appropriate extensions could be
    used as a label distribution protocol to obtain such label
    bindings.

    If the first intermediate router of LSP(i+1) is either (a) a
    single hop away from the egress of that LSP, or (b) multi-hop
    away, and LDP is used as a label distribution protocol to
    establish a multi-hop LSP fragment between the first intermediate
    router and the egress of that LSP, then R(0) can use targeted LDP
    session with R(i) to obtain such label bindings.

  + For LSP(1) if R(1) is one hop away from R(0), then no label is
    needed (as LSP(1), just like all other LSPs in the stack, is
    constructed with the penultimate hop popping), and the label
    stack construction terminates with the topmost label that R(0)
    obtains from R(1) for LSP(2).

  + Otherwise, if R(1) is more than one hop away from R(0), then R(0)
    obtains label binding for LSP(1) from the first intermediate
    router of LSP(1), and places this label at the top of the stack.


Note that the above procedures require all the LSPs in the stack to
have the same ingress - R(0). This requirement comes from the
observation that (a) R(0) is the router that constructs the whole
label stack needed to realize the explicitly routed tunnel, and (b)
according to MPLS architecture [RFC3031] when a router wants to
create a label stack, the router has to be the head-end of all the
LSPs corresponding to the labels in the stack.

Since the MPLS label stack mechanism allows stack of LSPs to nest to
any depth, use of LSP hierarchy for explicitly routed tunnels does
not place any protocol restrictions on the number of entries in the
explicit route of an explicitly routed tunnel. Note though that there
may be some other restrictions (e.g., due to MTU, or hardware) that
would place an upper bound on the depth of the label stack, and thus
on the number of entries in the explicit route.  Also, the depth of
the label stack may have implications on ECMP, and specifically on
the use of the Entropy label (see [kini] for more).

**4.1**. **Examples of Constructing Explicitly Routed Tunnels by Stacked LSPs**

In this section we illustrate how to construct an explicitly routed
tunnel by using stacked LSPs. The first example illustrates this for
an explicitly routed tunnel where consecutive hops that define the
tunnel are one hop away from each other. The second example
illustrates this when these hops are more than one hop away from each
other.

**4.1.1**. **Explicitly Routed Tunnel with Single Hops**

Consider a network topology shown below:

```
    R0-----R4
    |      /|
    |     / |
    |    /  |
    |   /   |
    |  /    |
    | /     |
    R1     R3
     \     /
      \   /
       R2
```
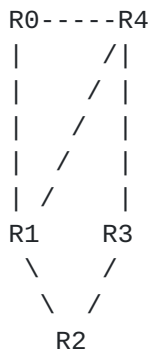
Assume that R0 wants to construct an explicitly routed tunnel with
(R0, R1, R2, R3, R4). The consecutive hops that define the tunnel are
one hop away from each other. That is, R0 is one hop away from R1, R2
is one hop away from R1, R3 is one hop away from R2, and R4 is one
hop away from R3.

R0 constructs this tunnel using the following stack of LSPs:

```
    LSP1: (R0, R1) - top of the stack
    LSP2: (R0, R1, R2)
```

        LSP3: (R0, R2, R3)
        LSP4: (R0, R3, R4) - bottom of the stack

   Note that this stack of LSPs meets the requirements specified in
   section "Constructing Explicitly Routed Tunnels by using Stacked
   LSPs". Specifically,

     + All four LSPs in the stack have the same ingress - R0, which is
       also the ingress of the explicitly routed tunnel.

     + The egress of LSP1, R1, is the first intermediate router of the
       next LSP in the stack, LSP2. The egress of LSP2, R2, is the first
       intermediate router of the next LSP in the stack, LSP3. Likewise,
       the egress of LSP3, R3, is the first intermediate router of the
       next (and the last) LSP in the stack, LSP4.

     + The LSP at the top of the stack, LSP1, has its first intermediate
       router, R1, one hop away from its ingress, R0. Because of that,
       this intermediate router is also the egress of that LSP, and that
       LSP is a one-hop LSP.

     + In that particular example the first intermediate router of every
       LSP in the stack is one hop away from the egress of that LSP.
       That is, the first intermediate router of LSP2, R1, is one hop
       away from the egress of that LSP, R2; the first intermediate
       router of LSP3, R2, is one hop away from the egress of that LSP,
       R3; and the first intermediate router of LSP4, R3, is one hop
       away from the egress of that LSP, R4.  As a result, the egress of
       a given LSP in the stack is one hop away from the egress of the
       next LSP in the stack.


   The first intermediate router of each of these LSPs creates label
   bindings for these LSPs as follows. R3 creates label binding for LSP4
   by binding a particular label, L1, to the address of R4, creating a
   Next Hop Label Forwarding Entry (NHLFE) whose next hop is the link
   from R3 to R4, and setting the Incoming Label Map (ILM) so that L1
   maps to that NHLFE. Likewise, R2 creates label binding for LSP3 by
   binding a particular label, L2, to the address of R3, creating an
   NHLFE whose next hop is the link from R2 to R3, and setting the ILM
   so that L2 maps to that NHLFE.  Finally, R1 creates label binding for
   LSP2 by binding a particular label, L3, to the address of R2,
   creating an NHLFE whose next hop is the link from R1 to R2, and
   setting the ILM so that L3 maps to that NHLFE.

   To get from the first hop of LSP4, R0, to the second hop of LSP4, R3,
   the packet has to go through the LSP tunnel provided by LSP3.  To get
   from the first hop of LSP3, R0, to the second hop of LSP3, R2, a

packet has to go through the LSP tunnel provided by LSP2. To get from
the first hop of LSP2, R0, to the second hop of LSP2, R1, a packet
has to go through the LSP tunnel provided by LSP1.

In order to accomplish this R0 constructs the label stack for the
explicitly routed tunnel as follows:

  + Step 1: R0 obtains label binding L1 created by R3 for LSP4 (R0,
    R3, R4), and starts building the label stack by pushing L1 onto
    the label stack.

  + Step 2: R0 obtains label binding L2 created by R2 for LSP3 (R0,
    R2, R3), and  pushes L2 into the stack. At this point the stack
    contains (L2, L1).

  + Step 3: R0 obtains label binding L3 created by R1 for LSP2 (R0,
    R1, R2), and pushes L3 into the stack. At this point the stack
    contains (L3, L2, L1).

  + Step 4: Since R0 and R1 are one hop away from each other the
    label stack construction is completed (R0 does not need a label
    for one-hop LSP1, as all the LSPs use penultimate hop popping).


So far we did not say anything about how R0 obtains from R3 label
binding for LSP4, from R2 label binding for LSP3, and from R1 label
binding for LSP2.

At least in principle, these label bindings could be obtain by such
already defined label distribution protocols as LDP (to be more
precise, targeted LDP if the two routers are more than one hop away
from each other). E.g., if one uses targeted LDP, then R0 would need
to dynamically establish and maintain a targeted LDP session with R3
and another targeted LDP session with R2 (R0 would maintain a
"vanilla" LDP session with R1). Using these LDP sessions R0 would
obtain from R3 label binding for LSP4, from R2 label binding for
LSP3, and from R1 label binding for LSP2. Note that obtaining such
labels bindings with targeted LDP may also require defining a new FEC
to be used by targeted LDP.

In section "IS-IS or OSPF as Label Distribution Protocol" we describe
how IS-IS or OSPF with appropriate extensions could be used as a
label distribution protocol to obtain such label bindings.

When R0 wants to forward a packet along the explicitly constructed
tunnel (R0, R1, R2, R3, R4), R0 pushes (L3, L2, L1) onto the label
stack of the packet, and forwards the packet to R1. R1 performs the
lookup on the topmost label, L3, and based on this lookup forwards

the packet to R2. Prior to forwarding the packet to R2, R1 (acting as
a penultimate hop for LSP2) pops the topmost label, L3.  When R2
receives the packet, R2 performs the lookup on the topmost label, L2,
and based on this lookup forwards the packet to R3. Prior to
forwarding the packet to R3, R2 (acting as a penultimate hop for
LSP3) pops the topmost label, L2. When R3 receives the packet, R3
performs the lookup on the topmost label, L1, and based on this
lookup forwards the packet to R4. Prior to forwarding the packet to
R4, R3 (acting as a penultimate hop for LSP4) pops the topmost label,
L1.


4.1.2. **Explicitly Routed Tunnel with Multi-Hops**

   Consider a network topology shown below:

```
     R0---R1
    /      \
   R2       R5---R6
    \      /
     R3---R4
```
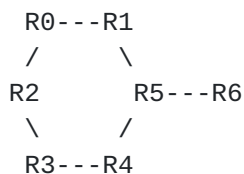
   Assume that R0 wants to construct an explicitly routed tunnel with
   (R0, R4, R6) as hops. Note that R4 is multi-hop away from R0, and R6
   is multi-hop away from R4.

   R0 constructs this tunnel using the following stack of LSPs:

      LSP1: (R0, R2, R3, R4) - top of the stack
      LSP2: (R0, R4, R5, R6) - bottom of the stack

   Note that this stack of LSPs meets the requirements specified in
   section "Constructing Explicitly Routed Tunnels by using Stacked
   LSPs". Specifically,

     + Both LSPs in the stack have the same ingress - R0, which is also
       the ingress of the explicitly routed tunnel.

     + The egress of LSP1, R4, is the first intermediate router of the
       next LSP in the stack, LSP2.

     + The LSP at the top of the stack, LSP1, has its first intermediate
       router, R2, one hop away from its ingress, R0. However, this
       intermediate router is not the egress of that LSP, and therefore
       this LSP is a multi-hop LSP.

  + In that particular example the first intermediate router of every
    LSP in the stack is multi-hop away from the egress of that LSP.
    That is, the first intermediate router of LSP1, R2, is multi-hop
    away from the egress of that LSP, R4; the first intermediate
    router of LSP2, R4, is multi-hop away from the egress of that
    LSP, R6.  As a result, the egress of a given LSP in the stack is
    multi-hop away from the egress of the next LSP in the stack.


  In this example we assume that LDP is used as a label distribution
  protocol for both LSP1 and LSP2. Since R0 and R4 are not IGP
  neighbors, they are remote label distribution peers. Thus R0 and R4
  use targeted LDP for label distribution. All other routers use
  "vanilla" LDP procedures.

  To get from the first hop of LSP2, R0, to its second hop, R4, the
  packet has to go through the LSP tunnel provided by LSP1.

  In order to accomplish this R0 constructs the label stack for the
  explicitly routed tunnel as follows:

   + Step 1: R0 (using targeted LDP) obtains label binding L1 created
     by R4 for LSP2 (R0, R4, R5, R6), and starts building the label
     stack by pushing L1 onto the label stack.

   + Step 2: R0 (using "vanilla" LDP procedures) obtains label binding
     L2 created by R2 for LSP1 (R0, R2, R3, R4), and  pushes L2 into
     the stack. At this point the stack contains (L2, L1).

   + Step 3: Since R0 and R1 are one hop away from each other the
     label stack construction is completed (R0 does not need a label
     for one-hop LSP1).


  A reader familiar with Remote LFA FRR [R-LFA] should be able to
  notice that the example described in this section is nothing more
  than an instance of Remote LFA FRR, where Remote LFA FRR provides
  fast reroute to the traffic going from R0 to R6 in the presence of
  the (R0, R2) link failure, with R0 being the Point of Local Repair
  (PLR), R4 being the PQ-node, and R6 being the ultimate destination.
  The explicitly routed tunnel (R0, R4, R6) consists of the PLR as the
  head-node, the PQ-node as the next hop, and the ultimate destination
  as yet another hop.

**5**. **IS-IS or OSPF as Label Distribution Protocol**

   When OSPF or IS-IS, in addition to being a routing protocol, is also
   used as a label distribution protocol (as proposed in [gredler-isis],
   [gredler-ospf], [previdi-isis], [psenak-ospf), the OSPF/IS-IS Link
   State Advertisements originated by a router carry label bindings for
   LSPs that either transit or originated by the router. Doing this
   allows to extend such LSPs. The criteria for selecting among all
   these LSPs a subset for which the router would originate label
   binding advertisements in IS-IS/OSPF are purely local to the router.
   The router could be either single or multi-hop away from the egresses
   of the LSPs in the subset. Existing label distribution protocols
   (LDP, RSVP-TE, etc.) can be used to establish multi-hop LSP fragments
   if the router is multi-hop away from the egress of a particular LSP
   in the subset. When IS-IS or OSPF, in addition to being a routing
   protocol, is also used as a label distribution protocol, it can also
   be used to establish such multi-hop LSP fragments.

   Use of IS-IS or OSPF as a label distribution protocol supports
   advertisements of label mappings for such FECs as:

     + IPv4/IPv6 prefix FEC via a hop-by-hop LSP established using IS-
       IS/OSPF as a label distribution protocol.

     + IPv4/IPv6 prefix FEC via a hop-by-hop LSP established using LDP
       as a label distribution protocol.

     + IPv4/IPv6 address FEC where the address identifies the remote end
       of one of the advertising router's links via an LSP that
       traverses the link and terminates on the remote end of the link.

     + IPv4/IPv6 address FEC where the address identifies the remote end
       of one of the advertising router's point-to-point links via an
       LSP that traverses the link and terminates on the remote end of
       the link.

     + IPv4/IPv6 address FEC where the address identifies a (remote)
       router connected to the advertising router by a broadcast link
       via an LSP that traverses the link and terminates on the remote
       router identified by its node-id.

     + IPv4/IPv6 prefix FEC via an explicitly routed LSP established
       using RSVP-TE, where path computation for such LSP is done by
       either distributed CSPF, or by PCE.


   When a router obtains label binding for a given FEC from more than
   one label distribution protocol (e.g., one binding from targeted LDP

and another from IS-IS/OSPF), deciding which label binding to use is
a matter of policy local to the router. In the scenario where a
router obtains label binding for a given FEC from both (targeted) LDP
and IS-IS/OSPF, the default behavior RECOMMENDED in this document is
to prefer the one obtained from (targeted) LDP.  An implementation
MUST support the ability to override the default behavior via
configuration.

MPLS architecture [RFC3031] defines the notion of local/remote label
distribution peers as follows:

   When two LSRs are IGP neighbors, we will refer to them as "local
   label distribution peers".  When two LSRs may be label distribution
   peers, but are not IGP neighbors, we will refer to them as "remote
   label distribution peers."

Following OSPF/IS-IS procedures each router passes Link State
Advertisements originated by other routers unmodified. When these
advertisements carry label binding information, this information is
also passed unmodified. Therefore, the router that originates label
bindings advertisements in IS-IS/OSPF can be either single or multi-
hop away from the routers that receive and use these bindings.  In
the former case the IGP neighbors of the router that originates the
advertisements will be the local label distribution peers of the
router. In the latter case other routers in the same IGP domain will
be the remote label distribution peers of the router.

Use of OSPF or IS-IS as a label distribution protocol provides
scalable support for remote label distribution peering in terms of
the number of label distribution peers a given router has to
maintain.  This is because label distribution protocol messages (Link
State Advertisements) are exchanged only between IGP neighbors,
without requiring control plane peering between a router that
originates Link State Advertisements and each of its remote label
distribution peers.

It is important to note that the existing MPLS control plane already
has mechanisms/protocols to support remote label distribution peering
(using BGP or targeted LDP [RFC5036]). Thus the practical relevance
of the ability to provide scalable support for remote label
distribution peering with IS-IS or OSPF as a label distribution
protocol depends on a particular use case.

If for a given subset of routers within an MPLS network each router
within the subset is assigned a distinct index, then one could
compress announcements of labels bound to the LSPs whose FECs are the
IP addresses of these routers by (a) advertising these indices in IS-
IS/OSPF, and (b) making each router advertise a label block in IS-

IS/OSPF as well. A router R1 that advertises a given label block
algorithmically binds a FEC associated with an IP address of some
other router R2 to the label from that block that is identified by
the index that R2 advertises in IGP. A router R1 that receives label
block originated by some other router R2 can determine the label
bound to a FEC associated with an IP address of some other router R3
by using the index advertised by R3 as an offset into the label block
advertised by R2. Note that to avoid wasting labels this scheme
requires a fairly dense assignment of indices. Also note that to
expand the number of labels that a router advertises using label
blocks, the router may advertise more than one label block.

Note though, that the benefits of scaling improvements in terms of
label distribution peering come at a cost, as every router in the
domain ends up keeping all the labels assigned/bounded by every other
router in the domain, whether it really needs to know them or not.
Whether this cost is of practical significance depends on (a) the
number of label bindings being advertised, and (b) the encoding of
label bindings (e.g., use of label blocks vs enumerating each label
binding).

## 5.1. Example of IS-IS/OSPF as Label Distribution Protocols

In this section we illustrate how IS-IS/OSPF with extensions, as
defined in [gredler-isis], [gredler-ospf], [previdi-isis], [psenak-
ospf] could be used as a label distribution protocol to support
explicitly routed tunnels realized by stacked LSPs. For the purpose
of this illustration we assume the scenario described in section
"Example of Constructing Explicitly Routed Tunnels by Stacked LSPs".
In that example one of the key issues is the ability of R0 to obtain
from R3 label binding for LSP4, from R2 label binding for LSP3, and
from R1 label binding for LSP2.

To obtains such label bindings, the Link State Advertisement
originated by R3 carries label L1 (this is the label that R3 binds to
LSP4). Using IS-IS/OSPF procedures this Link State Advertisement is
propagated by R2 and R1 (as well as by R4) to R0. This is how R0
obtains from R3 label binding for LSP4. In a similar fashion, the
Link State Advertisement originated by R2 carries label L2 (which is
the label that R2 binds to LSP3). Using IS-IS/OSPF procedures this
Link State Advertisement is propagated by R1 (as well as by R3 and
R4) to R0. This is how R0 obtains from R2 label binding for LSP3.
Likewise, the Link State Advertisement originated by R1 carries label
L3 (which is the label that R1 binds to LSP2).  Using IS-IS/OSPF
procedures this Link State Advertisement is delivered to R0. This is
how R0 obtains from R1 label binding for LSP2.

Note that while from R0's perspective both R2 and R3 are remote label
distribution peers, R0 does not maintain any control plane peering
(e.g., targeted LDP) with either R2 or R3.

## 6. IANA Considerations

This document introduces no new IANA Considerations.

## 7. Security Considerations

TBD

## 8. Acknowledgements

We would like to thank John Drake (Juniper Networks) and John Scudder
(Juniper Networks) for their review and comments.

We would also like to thank Bruno Decraene (Orange) and Robert Raszuk
for their review and comments.

## 9. Normative References

[RFC3031] Rosen, E., et. al., "Multiprotocol Label Switching
Architecture", RFC3031, January 2001

## 10. Informative References

[kini] Kini, S., et. al., "Entropy labels for source routed stacked
tunnels", draft-kini-mpls-entropy-label-src-stacked-tunnels, work in
progress

[gredler] Gredler, H., et. al., "Advertising MPLS labels in IGPs"
draft-gredler-rtgwg-igp-label-advertisement, work in progress

[gredler-isis] Gredler, H., et. al., "Advertising MPLS labels in IS-
IS" draft-gredler-isis-label-advertisement, work in progress

[gredler-ospf] Gredler, H., et. al., "Advertising MPLS labels in
OSPF" draft-gredler-ospf-label-advertisement, work in progress

[previdi-isis] Previdi, S., et. al., "IS-IS Extensions for Segment
Routing" draft-previdi-isis-segment-routing-extensions, work in
progress

[psenak-ospf] Psenak, P., et. al., "OSPF Extensions for Segment
Routing" draft-psenak-ospf-segment-routing-extensions, work in
progress

[R-LFA] Bryant, S., et. al., "Remote LFA FRR", draft-ietf-rtgwg-
remote-lfa, work in progress

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", RFC2119, March 1997

[RFC3107] Rekhter, Y., et. al., "Carrying Label Information in
BGP-4", RFC3107, May 2001

[RFC3209] Awduche, D., et. al., "RSVP-TE: Extensions to RSVP for LSP
Tunnels", RFC3209, December 2001

[RFC3212] Jamoussi, B., et. al., "Constraint-Based LSP Setup using
LDP", RFC3212, January 2002

[RFC4364] Rosen E., et. al., "BGP/MPLS IP Virtual Private Networks
(VPNs)", RFC4364, February 2006

[RFC4761] Kompella, K., et. al., "Virtual Private LAN Service (VPLS)
Using BGP for Auto-Discovery and Signaling", RFC4761, January 2007

[RFC5036] L. Andersson, et. al., "LDP Specification", RFC5036,
October 2007

## 11. Authors' Addresses

Hannes Gredler
Juniper Networks
e-mail: hannes@juniper.net

Yakov Rekhter
Juniper Networks
e-mail: yakov@juniper.net

Luay Jalil
Verizon
e-mail: luay.jalil@verizon.com

Sriganesh Kini
Ericsson
Email: sriganesh.kini@ericsson.com

Xiaohu Xu

Huawei Technologies,
Email: xuxiaohu@huawei.com