Network Working Group
Internet-Draft
Updates: 3031 (if approved)
Intended status: Standards Track
Expires: September 7, 2011

K. Kompella
J. Drake
Juniper Networks
S. Amante
Level 3 Communications, LLC
W. Henderickx
Alcatel-Lucent
L. Yong
Huawei USA

March 6, 2011

The Use of Entropy Labels in MPLS Forwarding
draft-kompella-mpls-entropy-label-02

Abstract

Load balancing is a powerful tool for engineering traffic across a network. This memo suggests ways of improving load balancing across MPLS networks using the concept of "entropy labels". It defines the concept, describes why entropy labels are useful, enumerates properties of entropy labels that allow maximal benefit, and shows how they can be signaled and used for various applications.

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 September 7, 2011.

Table of Contents

1. Introduction

Load balancing, or multi-pathing, is an attempt to balance traffic across a network by allowing the traffic to use multiple paths. Load balancing has several benefits: it eases capacity planning; it can help absorb traffic surges by spreading them across multiple paths; it allows better resilience by offering alternate paths in the event of a link or node failure.

As providers scale their networks, they use several techniques to achieve greater bandwidth between nodes. Two widely used techniques are: Link Aggregation Group (LAG) and Equal-Cost Multi-Path (ECMP). LAG is used to bond together several physical circuits between two adjacent nodes so they appear to higher-layer protocols as a single, higher bandwidth 'virtual' pipe. ECMP is used between two nodes separated by one or more hops, to allow load balancing over several shortest paths in the network. This is typically obtained by arranging IGP metrics such that there are several equal cost paths between source-destination pairs. Both of these techniques may, and often do, co-exist in various parts of a given provider's network, depending on various choices made by the provider.

A very important requirement when load balancing is that packets belonging to a given 'flow' must be mapped to the same path, i.e., the same exact sequence of links across the network. This is to avoid jitter, latency and re-ordering issues for the flow. What constitutes a flow varies considerably. A common example of a flow is a TCP session. Other examples are an L2TP session corresponding to a given broadband user, or traffic within an ATM virtual circuit.

To meet this requirement, a node uses certain fields, termed 'keys', within a packet's header as input to a load balancing function (typically a hash function) that selects the path for all packets in a given flow. The keys chosen for the load balancing function depend on the packet type; a typical set (for IP packets) is the IP source and destination addresses, the protocol type, and (for TCP and UDP traffic) the source and destination port numbers. An overly conservative choice of fields may lead to many flows mapping to the same hash value (and consequently poorer load balancing); an overly aggressive choice may map a flow to multiple values, potentially violating the above requirement.

For MPLS networks, most of the same principles (and benefits) apply. However, finding useful keys in a packet for the purpose of load balancing can be more of a challenge. In many cases, MPLS encapsulation may require fairly deep inspection of packets to find these keys at transit LSRs.

One way to eliminate the need for this deep inspection is to have the ingress LSR of an MPLS Label Switched Path extract the appropriate keys from a given packet, input them to its load balancing function, and place the result in an additional label, termed the 'entropy label', as part of the MPLS label stack it pushes onto that packet.

The packet's MPLS entire label stack can then be used by transit LSRs to perform load balancing, as the entropy label introduces the right level of "entropy" into the label stack.

There are four key reasons why this is beneficial:

1. at the ingress LSR, MPLS encapsulation hasn't yet occurred, so deep inspection is not necessary;

2. the ingress LSR has more context and information about incoming packets than transit LSRs;

3. ingress LSRs usually operate at lower bandwidths than transit LSRs, allowing them to do more work per packet, and

4. transit LSRs do not need to perform deep packet inspection and can load balance effectively using only a packet's MPLS label stack.

This memo describes why entropy labels are needed and defines the properties of entropy labels; in particular how they are generated and received, and the expected behavior of transit LSRs. Finally, it describes in general how signaling works and what needs to be signaled, as well as specifics for the signaling of entropy labels for LDP ([RFC5036]), BGP ([RFC3107], [RFC4364]), and RSVP-TE ([RFC3209]).

1.1. Conventions used

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 [RFC2119].

The following acronyms are used:

LSR: Label Switching Router;

LER: Label Edge Router;

PE: Provider Edge router;

CE: Customer Edge device; and

FEC: Forwarding Equivalence Class.

The term ingress (or egress) LSR is used interchangeably with ingress (or egress) LER. The term application throughout the text refers to an MPLS application (such as a VPN or VPLS).

A label stack (say of three labels) is denoted by <L1, L2, L3>, where L1 is the "outermost" label and L3 the innermost (closest to the payload). Packet flows are depicted left to right, and signaling is shown right to left (unless otherwise indicated).

The term 'label' is used both for the entire 32-bit label and the 20-bit label field within a label. It should be clear from the context which is meant.

1.2. Motivation

MPLS is very successful generic forwarding substrate that transports several dozen types of protocols, most notably: IP, PWE3, VPLS and IP VPNs. Within each type of protocol, there typically exist several variants, each with a different set of load balancing keys, e.g., for IP: IPv4, IPv6, IPv6 in IPv4, etc.; for PWE3: Ethernet, ATM, Frame-Relay, etc. There are also several different types of Ethernet over PW encapsulation, ATM over PW encapsulation, etc. as well. Finally, given the popularity of MPLS, it is likely that it will continue to be extended to transport new protocols.

Currently, each transit LSR along the path of a given LSP has to try to infer the underlying protocol within an MPLS packet in order to extract appropriate keys for load balancing. Unfortunately, if the transit LSR is unable to infer the MPLS packet's protocol (as is often the case), it will typically use the topmost (or all) MPLS labels in the label stack as keys for the load balancing function. The result may be an extremely inequitable distribution of traffic across equal-cost paths exiting that LSR. This is because MPLS labels are generally fairly coarse-grained forwarding labels that typically describe a next-hop, or provide some of demultiplexing and/or forwarding function, and do not describe the packet's underlying protocol.

On the other hand, an ingress LSR (e.g., a PE router) has detailed knowledge of an packet's contents, typically through a priori configuration of the encapsulation(s) that are expected at a given PE-CE interface, (e.g., IPv4, IPv6, VPLS, etc.). They also have more flexible forwarding hardware. PE routers need this information and these capabilities to:

a) apply the required services for the CE;

b) discern the packet's CoS forwarding treatment;

c) apply filters to forward or block traffic to/from the CE;

d) to forward routing/control traffic to an onboard management processor; and,

e) load-balance the traffic on its uplinks to transit LSRs (e.g., P routers).

By knowing the expected encapsulation types, an ingress LSR router can apply a more specific set of payload parsing routines to extract the keys appropriate for a given protocol. This allows for significantly improved accuracy in determining the appropriate load balancing behavior for each protocol.

If the ingress LSR were to capture the flow information so gathered in a convenient form for downstream transit LSRs, transit LSRs could remain completely oblivious to the contents of each MPLS packet, and use only the captured flow information to perform load balancing. In particular, there will be no reason to duplicate an ingress LSR's complex packet/payload parsing functionality in a transit LSR. This will result in less complex transit LSRs, enabling them to more easily scale to higher forwarding rates, larger port density, lower power consumption, etc. The idea in this memo is to capture this flow information as a label, the so-called entropy label.

Ingress LSRs can also adapt more readily to new protocols and extract the appropriate keys to use for load balancing packets of those protocols. This means that deploying new protocols or services in edge devices requires fewer concommitant changes in the core, resulting in higher edge service velocity and at the same time more stable core networks.

2. Approaches

There are two main approaches to encoding load balancing information in the label stack. The first allocates multiple labels for a particular Forwarding Equivalance Class (FEC). These labels are equivalent in terms of forwarding semantics, but having multiple labels allows flexibility in assigning labels to flows belonging to the same FEC. This approach has the advantage that the label stack has the same depth whether or not one uses label-based load balancing; and so, consequently, there is no change to forwarding operations on transit and egress LSRs. However, it has a major

drawback in that there is a significant increase in both signaling and forwarding state.

The other approach encodes the load balancing information as an additional label in the label stack, thus increasing the depth of the label stack by one. With this approach, there is minimal change to signaling state for a FEC; also, there is no change in forwarding operations in transit LSRs, and no increase of forwarding state in any LSR. The only purpose of the additional label is to increase the entropy in the label stack, so this is called an "entropy label". This memo focuses solely on this approach.

3. Entropy Labels

An entropy label (as used here) is a label:

1. that is not used for forwarding;

2. that is not signaled; and

3. whose only purpose in the label stack is to provide 'entropy' to improve load balancing.

Entropy labels are generated by an ingress LSR, based entirely on load balancing information. However, they MUST NOT have values in the reserved label space (0-15). Entropy labels MUST be at the bottom of the label stack, and thus the 'Bottom of Stack' (S) bit ([RFC3032]) in the label should be set. To ensure that they are not used inadvertently for forwarding, entropy labels SHOULD have a TTL of 0.

Since entropy labels are generated by an ingress LSR, an egress LSR MUST be able to tell unambiguously that a given label is an entropy label. If any ambiguity is possible, the label above the entropy label MUST be an 'entropy label indicator' (ELI), which indicates that the following Label is an entropy label. An ELI is typically signaled by an egress LSR and is added to the MPLS label stack along with an entropy label by an ingress LSR. For many applications, the use of entropy labels is unambiguous, and an ELI is not needed. If used, an ELI MUST have S = 0 and SHOULD have a TTL of 0.

Applications for MPLS entropy labels include pseudowires ([RFC4447]), Layer 3 VPNs ([RFC4364]), VPLS ([RFC4761], [RFC4762]) and Tunnel LSPs carrying, say, IP traffic. [I-D.ietf-pwe3-fat-pw] explains how entropy labels can be used for RFC 4447-style pseudowires, and thus is complementary to this memo, which focuses on several other applications of entropy labels.

4. Data Plane Processing of Entropy Labels

4.1. Ingress LSR

Suppose that for a particular application (or service or FEC), an ingress LSR X is to push label stack <TL, AL>, where TL is the 'tunnel label' and AL is the 'application label'. (Note the use of the convention for label stacks described in Section 1.1. The use of a two-label stack is just for illustrative purposes.) Suppose furthermore that the egress LSR Y has told X that it is capable of processing entropy labels for this application. If X can insert entropy labels, it may use a label stack of <TL, AL, EL> for this application, where EL is the entropy label.

When a packet for this application arrives at X, X does the following:

1. X identifies the application to which the packet belongs, identifies the egress LSR as Y, and thereby picks the outgoing label stack <TL, AL> to push onto the packet to send to Y;

2. X determines which keys that it will use for load balancing;

3. X, having kept state that Y can process entropy labels for this application, generates an entropy label EL (based on the output of the load balancing function), and

4. X pushes <TL, AL, EL> on to the packet before forwarding it to the next LSR on its way to Y.

EL is a 'regular' 32-bit label whose S bit MUST be 1 and whose TTL field SHOULD be 0. The load balancing information is encoded in the 20-bit label field. If X is told (via signaling) that it must use an entropy label indicator with label value E, then X instead pushes <TL, AL, ELI, EL> onto the packet, where ELI is a label whose S bit MUST be 0, whose TTL SHOULD be 0, and whose 20-bit label field MUST be E. The CoS fields for EL and ELI can be set to any values.

Note that ingress LSR X MUST NOT include an entropy label unless the egress LSR Y for this application has indicated that it is ready to receive entropy labels. Furthermore, if Y has signaled that an ELI is needed, then X MUST include the ELI before the entropy label.

Note that the signaling and use of entropy labels in one direction (signaling from Y to X, and data path from X to Y) has no bearing on the behavior in the opposite direction (signaling from X to Y, and data path from Y to X).

4.2. Transit LSR

Transit LSRs have virtually no change in forwarding behavior. For load balancing, transit LSRs SHOULD use the whole label stack as keys for the load balancing function. Transit LSRs MAY choose to look beyond the label stack for further keys; however, if entropy labels are being used, this may not be very useful. Looking beyond the label stack may be the simplest approach in an environment where some ingress LSRs use entropy labels and others don't, or for backward compatibility. Thus, other than using the full label stack as input to the load balancing function, transit LSRs are almost unaffected by the use of entropy labels.

4.3. Egress LSR

If egresss LSR Y signals that it is capable of processing entropy labels without an ELI for an application, then when Y receives a packet with the application label, then Y looks to see if the S bit is set. If so, Y applies its usual processing rules to the packet, including popping the application label. If the S bit is not set, Y assumes that the label below the application label is an entropy label and pops both the application label and the entropy label. Y SHOULD ensure that the entropy label has its S bit set. Y then processes the packet as usual. Implementations may choose the order in which they apply these operations, but the net result should be as specified.

If Y signals that it is capable of processing entropy labels but that an ELI is necessary for a given application, then when Y receives a packet with the application label, Y processes the application label as usual, then pops it. Y then checks whether the S bit on the application label is set. If not, Y looks to see if the label below the application label is the ELI. If so, Y further pops both the ELI and the label below (which should be the entropy label). Y SHOULD ensure that the ELI has its S bit unset, and that the entropy label has its S bit set. If the S bit of the application label is set, or the label below is not the ELI, Y processes the packet as usual (there is no entropy label).

5. Signaling for Entropy Labels

An egress LSR Y may signal to ingress LSR(s) its ability to process entropy labels on a per-application (or per-FEC) basis. As part of this signaling, Y also signals the ELI to use, if any.

In cases where an application label is used and must be the bottommost label in the label stack, Y MAY signal that no ELI is

needed for that application.

In cases where no application label exists, or where the application label may not be the bottommost label in the label stack, Y MUST signal a valid ELI to be used in conjunction with the entropy label for this FEC. In this case, an ingress LSR will either not add an entropy label, or push the ELI before the entropy label. This makes the use or non-use of an entropy label by the ingress LSR unambiguous. Valid ELI label values are strictly greater than 15.

It should be noted that egress LSR Y may use the same ELI value for all applications for which an ELI is needed. The ELI MUST be a label that does not conflict with any other labels that Y has advertised to other LSRs for other applications. Furthermore, it should be noted that the ability to process entropy labels (and the corresponding ELI) may be asymmetric: an LSR X may be willing to process entropy labels, whereas LSR Y may not be willing to process entropy labels. The signaling extensions below allow for this asymmetry.

For an illustration of signaling and forwarding with entropy labels, see Figure 9.

5.1. LDP Signaling

When using LDP for signaling tunnel labels ([RFC5036]), a Label Mapping Message sub-TLV (Entropy Label sub-TLV) is used to signal an egress LSR's ability to process entropy labels.

The presence of the Entropy Label sub-TLV in the Label Mapping Message indicates to ingress LSRs that the egress LSR can process an entropy label. In addition, the Entropy Label sub-TLV contains a label value for the ELI. If the ELI is zero, this indicates the egress doesn't need an ELI for the signaled application; if not, the egress requires the given ELI with entropy labels. An example where an ELI is needed is when the signaled application is an LSP that can carry IP traffic.

The structure of the Entropy Label sub-TLV is shown below.

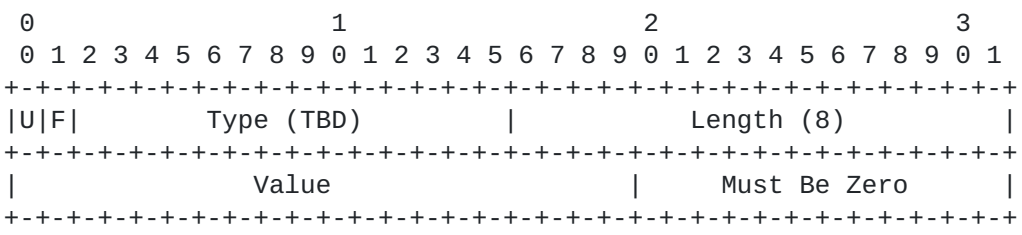


Figure 1: Entropy Label sub-TLV

where:

U: Unknown bit. This bit MUST be set to 1. If the Entropy Label sub-TLV is not understood, then the TLV is not known to the receiver and MUST be ignored.

F: Forward bit. This bit MUST be set be set to 1. Since this sub-TLV is going to be propagated hop-by-hop, the sub-TLV should be forwarded even by nodes that may not understand it.

Type: sub-TLV Type field, as specified by IANA.

Length: sub-TLV Length field. This field specifies the total length in octets of the Entropy Label sub-TLV.

Value: value of the Entropy Label Indicator Label.

5.2. BGP Signaling

When BGP [RFC4271] is used for distributing Network Layer Reachability Information (NLRI) as described in, for example, [RFC3107], [RFC4364] and [RFC4761], the BGP UPDATE message may include the Entropy Label attribute. This is an optional, transitive BGP attribute of type TBD. The inclusion of this attribute with an NLRI indicates that the advertising BGP router can process entropy labels as an egress LSR for that NLRI. If the attribute length is less than three octets, this indicates that the egress doesn't need an ELI for the signaled application. If the attribute length is at least three octets, the first three octets encode an ELI label value as the high order 20 bits; the egress requires this ELI with entropy labels. An example where an ELI is needed is when the NLRI contains unlabeled IP prefixes.

A BGP speaker S that originates an UPDATE should only include the Entropy Label attribute if both of the following are true:

A1: S sets the BGP NEXT_HOP attribute to itself; AND

A2: S can process entropy labels for the given application.

If both A1 and A2 are true, and S needs an ELI to recognize entropy labels, then S MUST include the ELI label value as part of the Entropy Label attribute. An UPDATE SHOULD contain at most one Entropy Label attribute.

Suppose a BGP speaker T receives an UPDATE U with the Entropy Label attribute ELA. T has two choices. T can simply re-advertise U with the same ELA if either of the following is true:

B1: T does not change the NEXT_HOP attribute; OR

B2: T simply swaps labels without popping the entire label stack and processing the payload below.

An example of the use of B1 is Route Reflectors; an example of the use of B2 is illustrated in Section 9.3.1.2.

However, if T changes the NEXT_HOP attribute for U and in the data plane pops the entire label stack to process the payload, T MUST remove ELA. T MAY include a new Entropy Label attribute ELA' for UPDATE U' if both of the following are true:

C1: T sets the NEXT_HOP attribute of U' to itself; AND

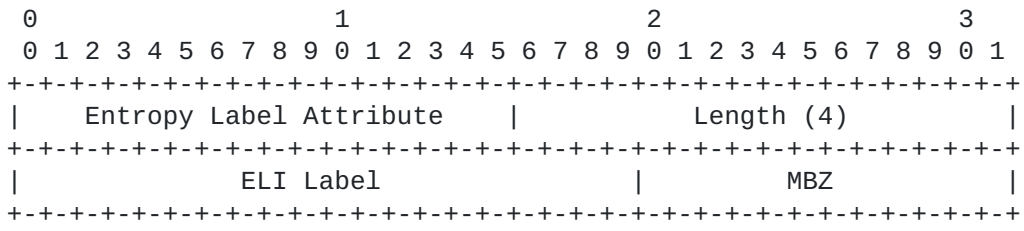
C2: T can process entropy labels for the given application.

Again, if both C1 and C2 are true, and T needs an ELI to recognize entropy labels, then T MUST include the ELI label value as part of the Entropy Label attribute.

5.3. RSVP-TE Signaling

Entropy Label support is signaled in RSVP-TE [RFC3209] using an Entropy Label Attribute TLV (Type TBD) of the LSP_ATTRIBUTES object [RFC5420]. The presence of this attribute indicates that the signaler (the egress in the downstream direction using Resv messages; the ingress in the upstream direction using Path messages) can process entropy labels. The Entropy Label Attribute contains a value for the ELI. If the ELI is zero, this indicates that the signaler doesn't need an ELI for this application; if not, then the signaler requires the given ELI with entropy labels. An example where an ELI is needed is when the signaled LSP can carry IP traffic.

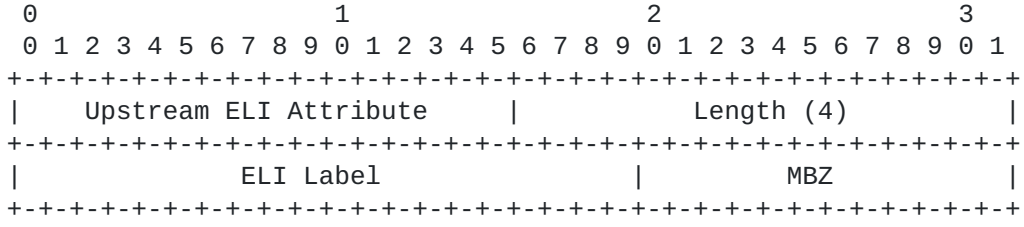
The format of the Entropy Label Attribute is as follows:



An egress LSR includes the Entropy Label Attribute in a Resv message to indicate that it can process entropy labels in the downstream direction of the signaled LSP.

An ingress LSR includes the Entropy Label Attribute in a Path message for a bi-directional LSP to indicate that it can process entropy labels in the upstream direction of the signaled LSP. If the signaled LSP is not bidirectional, the Entropy Label Attribute SHOULD NOT be included in the Path message, and egress LSR(s) SHOULD ignore the attribute, if any.

As described in Section 8, there is also the need to distribute an ELI from the ingress (upstream label allocation). In the case of RSVP-TE, this is accomplished using the Upstream ELI Attribute TLV of the LSP_ATTRIBUTES object, as shown below:



6. Operations, Administration, and Maintenance (OAM) and Entropy Labels

Generally OAM comprises a set of functions operating in the data plane to allow a network operator to monitor its network infrastructure and to implement mechanisms in order to enhance the general behavior and the level of performance of its network, e.g., the efficient and automatic detection, localization, diagnosis and handling of defects.

Currently defined OAM mechanisms for MPLS include LSP Ping/Traceroute [RFC4379] and Bidirectional Failure Detection (BFD) for MPLS [RFC5884]. The latter provides connectivity verification between the endpoints of an LSP, and recommends establishing a separate BFD session for every path between the endpoints.

The LSP traceroute procedures of [RFC4379] allow an ingress LSR to obtain label ranges that can be used to send packets on every path to the egress LSR. It works by having ingress LSR sequentially ask the transit LSRs along a particular path to a given egress LSR to return a label range such that the inclusion of a label in that range in a packet will cause the replying transit LSR to send that packet out the egress interface for that path. The ingress provides the label range returned by transit LSR N to transit LSR N + 1, which returns a label range which is less than or equal in span to the range provided to it. This process iterates until the penultimate transit LSR replies to the ingress LSR with a label range that is acceptable to it and to all LSRs along path preceding it for forwarding a packet along the path.

However, the LSP traceroute procedures do not specify where in the label stack the value from the label range is to be placed, whether deep packet inspection is allowed and if so, which keys and key values are to be used.

This memo updates LSP traceroute by specifying that the value from the label range is to be placed in the entropy label. Deep packet inspection is thus not necessary, although an LSR may use it, provided it do so consistently, i.e., if the label range to go to a given downstream LSR is computed with deep packet inspection, then the data path should use the same approach and the same keys.

In order to have a BFD session on a given path, a value from the label range for that path should be used as the EL value for BFD packets sent on that path.

As part of the MPLS-TP work, an in-band OAM channel is defined in [RFC5586]. Packets sent in this channel are identified with a reserved label, the Generic Associated Channel Label (GAL) placed at the bottom of the MPLS label stack. In order to use the inband OAM channel with entropy labels, this memo relaxes the restriction that the GAL must be at the bottom of the MPLS label stack. Rather, the GAL is placed in the MPLS label stack above the entropy label so that it effectively functions as an application label.

7. MPLS-TP and Entropy Labels

Since MPLS-TP does not use ECMP, entropy labels are not applicable to an MPLS-TP deployment.

8. Point-to-Multipoint LSPs and Entropy Labels

Point-to-Multipoint (P2MP) LSPs [RFC4875] typically do not use ECMP for load balancing, as the combination of replication and multipathing can lead to duplicate traffic delivery. However, P2MP LSPs can traverse Bundled Links [RFC4201] and LAGs. In both these cases, load balancing is useful, and hence entropy labels can be of some value for P2MP LSPs.

There are two potential complications with the use of entropy labels in the context of P2MP LSPs, both a consequence of the fact that the entire label stack below the P2MP label must be the same for all egress LSRs. First, all egress LSRs must be willing to receive entropy labels; if even one egress LSR is not willing, then entropy labels MUST NOT be used for this P2MP LSP. Second, if an ELI is required, all egress LSRs must agree to the same value of ELI. This can be achieved by upstream allocation of the ELI; in particular, for RSVP-TE P2MP LSPs, the ingress LSR distributes the ELI value using the Upstream ELI Attribute TLV of the LSP_ATTRIBUTES object, defined in Section 5.3.

With regard to the first issue, the ingress LSR MUST keep track of the ability of each egress LSR to process entropy labels, especially since the set of egress LSRs of a given P2MP LSP may change over time. Whenever an existing egress LSR leaves, or a new egress LSR joins the P2MP LSP, the ingress MUST re-evaluate whether or not to include entropy labels for the P2MP LSP.

In some cases, it may be feasible to deploy two P2MP LSPs, one to entropy label capable egress LSRs, and the other to the remaining egress LSRs. However, this requires more state in the network, more bandwidth, and more operational overhead (tracking EL-capable LSRs, and provisioning P2MP LSPs accordingly). Furthermore, this approach may not work for some applications (such mVPNs and VPLS) which automatically create and/or use P2MP LSPs for their multicast requirements.

9. Entropy Labels and Applications

This section describes the usage of entropy labels in various scenarios with different applications.

9.1. Tunnels

Tunnel LSPs, signaled with either LDP or RSVP-TE, typically carry other MPLS applications such as VPNs or pseudowires. This being the case, if the egress LSR of a tunnel LSP is willing to process entropy

labels, it would signal the need for an Entropy Label Indicator to distinguish between entropy labels and other application labels.

In the figures below, the following convention is used to depict information signaled between X and Y:

```
                        X ---------- ... ---------- Y
                  app:   <--- [label L, ELI value]
```

This means Y signals to X label L for application app. The ELI value can be one of:

-: meaning entropy labels are NOT accepted;

0: meaning entropy labels are accepted, no ELI is needed; or

E: entropy labels are accepted, ELI label E is required.

The following illustrates a simple intra-AS tunnel LSP.

```
                  X -------- A --- ... --- B -------- Y
      tunnel LSP L:   [TL,  E] <---  ...  <--- [TL0, E]

      IP pkt:         push <TL, E, EL> --------------->
```

Figure 2: Tunnel LSPs and Entropy Labels

Tunnel LSPs may cross Autonomous System (AS) boundaries, usually using BGP ([RFC3107]). In this case, the AS Border Routers (ASBRs) MAY simply propagate the egress LSR's ability to process entropy labels, or they MAY declare that entropy labels may not be used. If an ASBR (say A2 below) chooses to propagate the egress LSR Y's ability to process entropy labels, A2 MUST also propagate Y's choice of ELI.

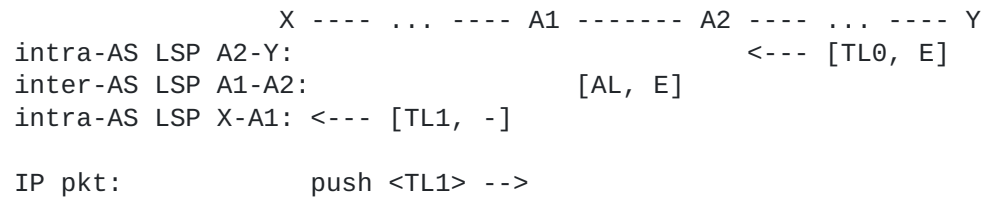
```
                 X ---- ... ---- A1 ------- A2 ---- ... ---- Y
  intra-AS LSP A2-Y:                          <--- [TL0, E]
  inter-AS LSP A1-A2:                [AL, E]
  intra-AS LSP X-A1: <--- [TL1, E]

  IP pkt:           push <TL1, E, EL>
```

Here, ASBR A2 chooses to propagate Y's ability to process entropy labels, by "translating" Y's signaling of entropy label capability (say using LDP) to BGP; and A1 translate A2's BGP signaling to (say) RSVP-TE. The end-to-end tunnel (X to Y) will have entropy labels if

X chooses to insert them.

Figure 3: Inter-AS Tunnel LSP with Entropy Labels



Here, ASBR A1 decided that entropy labels are not to be used; thus, the end-to-end tunnel cannot have entropy labels, even though both X and Y may be capable of inserting and processing entropy labels.

Figure 4: Inter-AS Tunnel LSP with no Entropy Labels

9.2. LDP Pseudowires

[I-D.ietf-pwe3-fat-pw] describes the signaling and use of entropy labels in the context of RFC 4447 pseudowires, so this will not be described further here.

[RFC4762] specifies the use of LDP for signaling VPLS pseudowires. An egress VPLS PE that can process entropy labels can indicate this by adding the Entropy Label sub-TLV in the LDP message it sends to other PEs. An ELI is not required. An ingress PE must maintain state per egress PE as to whether it can process entropy labels.

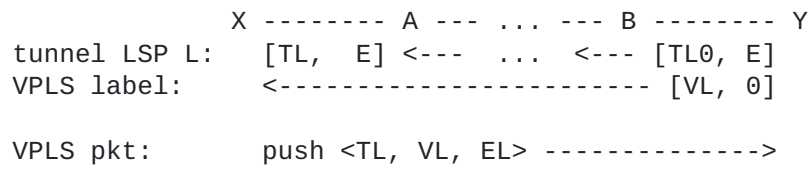


Figure 5: Entropy Labels with LDP VPLS

Note that although the underlying tunnel LSP signaling indicated the need for an ELI, VPLS packets don't need an ELI, and thus the label stack pushed by X do not have one.

[RFC4762] also describes the notion of "hierarchical VPLS" (H-VPLS). In H-VPLS, 'hub PEs' remove the label stack and process VPLS packets; thus, they must make their own decisions on the use of entropy labels, independent of other hub PEs or spoke PEs with which they exchange signaling. In the example below, spoke PEs X and Y and hub

PE B can process entropy labels, but hub PE A cannot.

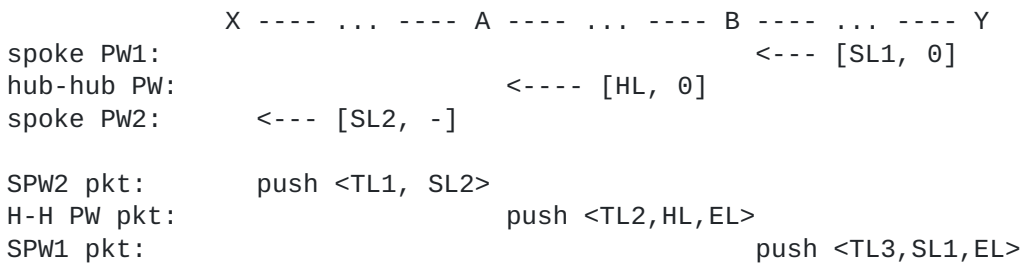


Figure 6: Entropy Labels with H-VPLS

9.3. BGP Applications

Section 9.1 described a BGP application for the creation of inter-AS tunnel LSPs. This section describes two other BGP applications, IP VPNs ([RFC4364]) and BGP VPLS ([RFC4761]). An egress PE for either of these applications indicates its ability to process entropy labels by adding the Entropy Label attribute to its BGP UPDATE message. Again, ingress PEs must maintain per-egress PE state regarding its ability to process entropy labels. In this section, both of these applications will be referred to as VPNs.

In the intra-AS case, PEs signal application labels and entropy label capability to each other, either directly, or via Route Reflectors (RRs). If RRs are used, they must not change the BGP NEXT_HOP attribute in the UPDATE messages; furthermore, they can simply pass on the Entropy Label attribute as is.

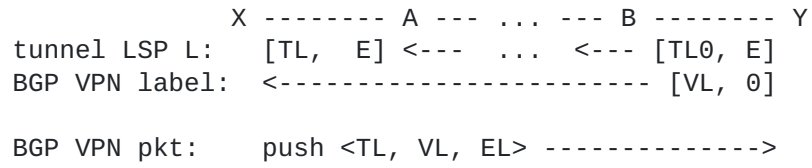


Figure 7: Entropy Labels with Intra-AS BGP apps

For BGP VPLS, the application label is at the bottom of stack, so no ELI is needed. For BGP IP VPNs, the application label is usually at the bottom of stack, so again no ELI is needed. However, in the case of Carrier's Carrier (CsC) VPNs, the BGP VPN label may not be at the bottom of stack. In this case, an ELI is necessary for CsC VPN packets with entropy labels to distinguish them from nested VPN packets. In the example below, the nested VPN signaling is not shown; the egress PE for the nested VPN (not shown) must signal

whether or not it can process egress labels, and the ingress nested VPN PE may insert an entropy label if so.

Three cases are shown: a plain BGP VPN packet, a CsC VPN packet originating from X, and a transit nested VPN packet originating from a nested VPN ingress PE (conceptually to the left of X). It is assumed that the nested VPN packet arrives at X with label stack <ZL, CVL> where ZL is the tunnel label (to be swapped with <TL, CL>) and CVL is the nested VPN label. Note that Y can use the same ELI for the tunnel LSP and the CsC VPN (and any other application that needs an ELI).

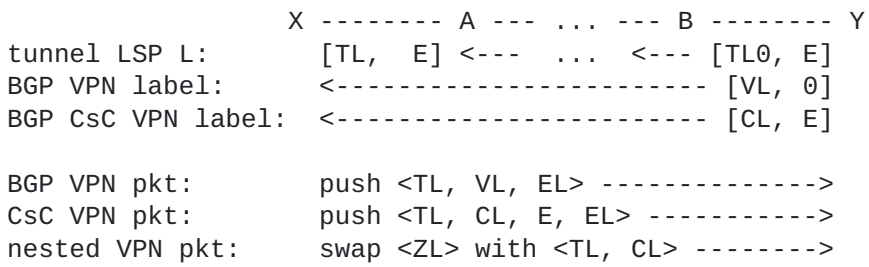


Figure 8: Entropy Labels with CoC VPN

9.3.1. Inter-AS BGP VPNs

There are three commonly used options for inter-AS IP VPNs and BGP VPLS, known informally as "Option A", "Option B" and "Option C". This section describes how entropy labels can be used in these options.

9.3.1.1. Option A Inter-AS VPNs

In option A, an ASBR pops the full label stack of a VPN packet exiting an AS, processes the payload header (IP or Ethernet), and forwards the packet natively (i.e., as IP or Ethernet, but not as MPLS) to the peer ASBR. Thus, entropy label signaling and insertion are completely local to each AS. The inter-AS paths do not use entropy labels, as they do not use a label stack.

9.3.1.2. Option B Inter-AS VPNs

The ASBRs in option B inter-AS VPNs have a choice (usually determined by configuration) of whether to just swap labels (from within the AS to the neighbor AS or vice versa), or to pop the full label stack and process the packet natively. This choice occurs at each ASBR in each direction. In the case of native packet processing at an ASBR, entropy label signaling and insertion is local to each AS and to the

inter-AS paths (which, unlike option A, do have labeled packets).

In the case of simple label swapping at an ASBR, the ASBR can propagate received entropy label signaling onward. That is, if a PE signals to its ASBR that it can process entropy labels (via an Entropy Label attribute), the ASBR can propagate that attribute to its peer ASBR; if a peer ASBR signals that it can process entropy labels, the ASBR can propagate that to all PEs within its AS). Note that this is the case even though ASBRs change the BGP NEXT_HOP attribute to "self", because of clause B2 in Section 5.2.

9.3.1.3. Option C Inter-AS VPNs

In Option C inter-AS VPNs, the ASBRs are not involved in signaling; they do not have VPN state; they simply swap labels of inter-AS tunnels. Signaling is PE to PE, usually via Route Reflectors; however, if RRs are used, the RRs do not change the BGP NEXT_HOP attribute. Thus, entropy label signaling and insertion are on a PE-pair basis, and the intermediate routers, ASBRs and RRs do not play a role.

9.4. Multiple Applications

It has been mentioned earlier that an ingress PE must keep state per egress PE with regard to its ability to process entropy labels. An ingress PE must also keep state per application, as entropy label processing must be based on the application context in which a packet is received (and of course, the corresponding entropy label signaling).

In the example below, an egress LSR Y signals a tunnel LSP L, and is prepared to receive entropy labels on L, but requires an ELI. Furthermore, Y signals two pseudowires PW1 and PW2 with labels PL1 and PL2, respectively, and indicates that it can receive entropy labels for both pseudowires without the need of an ELI; and finally, Y signals a L3 VPN with label VL, but Y does not indicate that it can receive entropy labels for the L3 VPN. Ingress LSR X chooses to send native IP packets to Y over L with entropy labels, thus X must include the given ELI (yielding a label stack of <TL, ELI, EL>). X chooses to add entropy labels on PW1 packets to Y, with a label stack of <TL, PL1, EL>, but chooses not to do so for PW2 packets. X must not send entropy labels on L3 VPN packets to Y, i.e., the label stack must be <TL, VL>.

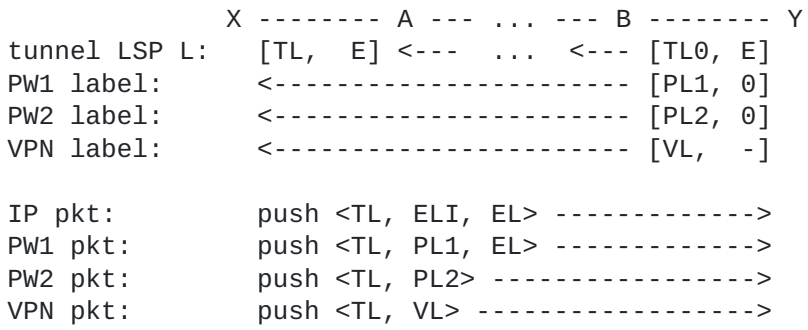


Figure 9: Entropy Labels for Multiple Applications

10. Security Considerations

This document describes advertisement of the capability to support receipt of entropy-labels and an Entropy Label Indicator that an ingress LSR may apply to MPLS packets in order to allow transit LSRs to attain better load-balancing across LAG and/or ECMP paths in the network.

This document does not introduce new security vulnerabilities to LDP. Please refer to the Security Considerations section of LDP ([RFC5036]) for security mechanisms applicable to LDP.

Given that there is no end-user control over the values used for entropy labels, there is little risk of Entropy Label forgery which could cause uneven load-balancing in the network.

If Entropy Label Capability is not signaled from an egress PE to an ingress PE, due to, for example, malicious configuration activity on the egress PE, then the PE's will fall back to not using entropy labels for load-balancing traffic over LAG or ECMP paths which, in some cases, in no worse than the behavior observed in current production networks. That said, operators are recommended to monitor changes to PE configurations and, more importantly, the fairness of load distribution over equal-cost LAG or ECMP paths. If the fairness of load distribution over a set of paths changes that could indicate a misconfiguration, bug or other non-optimal behavior on their PE's and they should take corrective action.

Given that most applications already signal an Application Label, e.g.: IPVPNs, LDP VPLS, BGP VPLS, whose Bottom of Stack bit is being re-used to signal entropy label capability, there is little to no additional risk that traffic could be misdirected into an inappropriate IPVPN VRF or VPLS VSI at the egress PE.

In the context of downstream-signaled entropy labels that require the use of an Entropy Label Indicator (ELI), there should be little to no additional risk because the egress PE is solely responsible for allocating an ELI value and ensuring that ELI label value DOES NOT conflict with other MPLS labels it has previously allocated. On the other hand, for upstream-signaled entropy labels, e.g.: RSVP-TE point-to-point or point-to-multipoint LSP's or Multicast LDP (mLDP) point-to-multipoint or multipoint-to-multipoint LSP's, there is a risk that the head-end MPLS LER may choose an ELI value that is already in use by a downstream LSR or LER. In this case, it is the responsibility of the downstream LSR or LER to ensure that it MUST NOT accept signaling for an ELI value that conflicts with MPLS label(s) that are already in use.

11. IANA Considerations

11.1. LDP Entropy Label TLV

IANA is requested to allocate the next available value from the IETF Consensus range in the LDP TLV Type Name Space Registry as the "Entropy Label TLV".

11.2. BGP Entropy Label Attribute

IANA is requested to allocate the next available Path Attribute Type Code from the "BGP Path Attributes" registry as the "BGP Entropy Label Attribute".

11.3. Attribute Flags for LSP_Attributes Object

IANA is requested to allocate a new bit from the "Attribute Flags" sub-registry of the "RSVP TE Parameters" registry.

Bit No	Name	Attribute Flags Path	Attribute Flags Resv	RRO
TBD	Entropy Label LSP	Yes	Yes	No

11.4. Attributes TLV for LSP_Attributes Object

IANA is requested to allocate the next available value from the "Attributes TLV" sub-registry of the "RSVP TE Parameters" registry.

12. Acknowledgments

We wish to thank Ulrich Drafz for his contributions, as well as the entire 'hash label' team for their valuable comments and discussion.

13. References

13.1. Normative References

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

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, May 2001.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC5420] Farrel, A., Papadimitriou, D., Vasseur, JP., and A. Ayyangarps, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, February 2009.

13.2. Informative References

[I-D.ietf-pwe3-fat-pw] Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow Aware Transport of Pseudowires over an MPLS PSN", draft-ietf-pwe3-fat-pw-05 (work in progress), October 2010.

[RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling in MPLS Traffic Engineering (TE)", RFC 4201, October 2005.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379,

February 2006.

[RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.

[RFC4762] Lasserre, M. and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, January 2007.

[RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

Appendix A. Applicability of LDP Entropy Label sub-TLV

In the case of unlabeled IPv4 (Internet) traffic, the Best Current Practice is for an egress LSR to propagate eBGP learned routes within a SP's Autonomous System after resetting the BGP next-hop attribute to one of its Loopback IP addresses. That Loopback IP address is injected into the Service Provider's IGP and, concurrently, a label assigned to it via LDP. Thus, when an ingress LSR is performing a forwarding lookup for a BGP destination it recursively resolves the associated next-hop to a Loopback IP address and associated LDP label of the egress LSR.

Thus, in the context of unlabeled IPv4 traffic, the LDP Entropy Label sub-TLV will typically be applied only to the FEC for the Loopback IP address of the egress LSR and the egress LSR will not announce an entropy label capability for the eBGP learned route.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kireeti@juniper.net

John Drake
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: jdrake@juniper.net

Shane Amante
Level 3 Communications, LLC
1025 Eldorado Blvd
Broomfield, CO 80021
US

Email: shane@level3.net

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
2018 Antwerp
Belgium

Email: wim.henderickx@alcatel-lucent.com

Lucy Yong
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075
US

Email: lucyyong@huawei.com