

Network Working Group
Internet-Draft
Updates: [3031](#), [5036](#) (if approved)
Intended status: Standards Track
Expires: January 11, 2013

K. Kompella
J. Drake
Juniper Networks
S. Amante
Level 3 Communications, LLC
W. Henderickx
Alcatel-Lucent
L. Yong
Huawei USA
July 10, 2012

The Use of Entropy Labels in MPLS Forwarding
draft-ietf-mpls-entropy-label-04

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 January 11, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Conventions used	4
1.2.	Motivation	6
2.	Approaches	7
3.	Entropy Labels and Their Structure	8
4.	Data Plane Processing of Entropy Labels	9
4.1.	Egress LSR	9
4.2.	Ingress LSR	10
4.3.	Transit LSR	11
4.4.	Penultimate Hop LSR	11
5.	Signaling for Entropy Labels	11
5.1.	LDP Signaling	12
5.1.1.	Processing the ELC TLV	12
5.2.	BGP Signaling	13
5.3.	RSVP-TE Signaling	14
5.4.	Multicast LSPs and Entropy Labels	14
6.	Operations, Administration, and Maintenance (OAM) and Entropy Labels	14
7.	MPLS-TP and Entropy Labels	15
8.	Entropy Labels in Various Scenarios	15
8.1.	LDP Tunnel	16
8.2.	LDP Over RSVP-TE	18
8.3.	MPLS Applications	18
9.	Security Considerations	19
10.	IANA Considerations	19
10.1.	Reserved Label for ELI	19
10.2.	LDP Entropy Label Capability TLV	19
10.3.	BGP Entropy Label Capability Attribute	20
10.4.	RSVP-TE Entropy Label Capability flag	20
11.	Acknowledgments	20
12.	References	20
12.1.	Normative References	20

12.2. Informative References	21
Appendix A. Applicability of LDP Entropy Label Capability TLV	22
Authors' Addresses	22

[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 five 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;
4. transit LSRs do not need to perform deep packet inspection and can load balance effectively using only a packet's MPLS label stack; and
5. transit LSRs, not having the full context that an ingress LSR does, have the hard choice between potentially misinterpreting fields in a packet as valid keys for load balancing (causing packet ordering problems) or adopting a conservative approach

(giving rise to sub-optimal load balancing). Entropy labels relieves them of making this choice.

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](#)]), 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:

BoS: Bottom of Stack

CE: Customer Edge device

ECMP: Equal Cost Multi-Path

EL: Entropy Label

ELC: Entropy Label Capability

ELI: Entropy Label Indicator

FEC: Forwarding Equivalence Class

LAG: Link Aggregation Group

LER: Label Edge Router

LSR: Label Switching Router

PE: Provider Edge Router

PHP: Penultimate Hop Popping

TC: Traffic Class

TTL: Time-to-Live

UHP: Ultimate Hop Popping

VPLS: Virtual Private LAN (Local Area Network) Service

VPN: Virtual Private Network

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 stack entry and the 20-bit label field within a label stack entry. 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 a 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 concomitant 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 Equivalence 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.

This latter approach uses upstream generated entropy labels, which may conflict with downstream allocated application labels. There are a few approaches to deal with this: 1) allocate a pair of labels for each FEC, one that must have an entropy label below it, and one that

must not; 2) use a label (the "Entropy Label Indicator") to indicate

that the next label is an entropy label; and 3) allow entropy labels only where there is no possible confusion. The first doubles control and data plane state in the network; the last is too restrictive. The approach taken here is the second. In making both the above choices, the trade-off is to increase label stack depth rather than control and data plane state in the network.

Finally, one may choose to associate ELs with MPLS tunnels (LSPs), or with MPLS applications (e.g., VPNs). (What this entails is described in later sections.) We take the former approach, for the following reasons:

1. There are a small number of tunneling protocols for MPLS, but a large and growing number of applications. Defining ELs on a tunnel basis means simpler standards, lower development, interoperability and testing efforts.
2. As a consequence, there will be much less churn in the network as new applications (services) are defined and deployed.
3. Processing application labels in the data plane is more complex than processing tunnel labels. Thus, it is preferable to burden the latter rather than the former with EL processing.
4. Associating ELs with tunnels makes it simpler to deal with hierarchy, be it LDP-over-RSVP-TE or Carrier's Carrier VPNs. Each layer in the hierarchy can choose independently whether or not they want ELs.

The cost of this approach is that ELIs will be mandatory; again, the trade-off is the size of the label stack. To summarize, the net increase in the label stack to use entropy labels is two: one reserved label for the ELI, and the entropy label itself.

3. Entropy Labels and Their Structure

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) [IANA MPLS Label Values]. To ensure that they are not used inadvertently for forwarding, entropy labels SHOULD have a TTL of 0. The CoS field of an entropy label can be set to any value deemed appropriate.

Since entropy labels are generated by an ingress LSR, an egress LSR MUST be able to distinguish unambiguously between entropy labels and application labels. This is accomplished by REQUIRING that the label immediately preceding an entropy label (EL) in the MPLS label stack be an 'entropy label indicator' (ELI), where preceding means closer to the top of the label stack (farther from bottom of stack indication). The ELI is a reserved label with value (TBD by IANA). An ELI MUST have 'Bottom of Stack' (BoS) bit = 0 ([RFC3032]). The TTL SHOULD be set to whatever value the label above it in the stack has. The CoS field can be set to any value deemed appropriate; typically, this will be the value in the label above the ELI in the label stack.

Entropy labels are useful for pseudowires ([RFC4447]). [RFC6391] explains how entropy labels can be used for RFC 4447-style pseudowires, and thus is complementary to this memo, which focuses on how entropy labels can be used for tunnels, and thus for all other MPLS applications.

[4.](#) Data Plane Processing of Entropy Labels

[4.1.](#) Egress LSR

Suppose egress LSR Y is capable of processing entropy labels for a tunnel. Y indicates this to all ingresses via signaling (see [Section 5](#)). Y MUST be prepared to deal both with packets with an imposed EL and those without; the ELI will distinguish these cases. If a particular ingress chooses not to impose an EL, Y's processing of the received label stack (which might be empty) is as if Y chose not to accept ELs.

If an ingress X chooses to impose an EL, then Y will receive a tunnel termination packet with label stack <TL, ELI, EL> <remaining packet header>. Y recognizes TL as the label it distributed to its upstreams for the tunnel, and pops it. (Note that TL may be the implicit null label, in which case it doesn't appear in the label stack.) Y then recognizes the ELI and pops two labels: the ELI and the EL. Y then processes the remaining packet header as normal; this may require further processing of tunnel termination, perhaps with

further ELI+EL pairs. When processing the final tunnel termination, Y MAY enqueue the packet based on that tunnel TL's or ELI's TC value,

and MAY use the tunnel TL's or ELI's TTL to compute the TTL of the remaining packet header. The EL's TTL MUST be ignored.

If any ELI processed by Y has BoS bit set, Y MUST discard the packet, and MAY log an error. The EL's BoS bit will indicate whether or not there are more labels in the stack.

[4.2.](#) Ingress LSR

If an egress LSR Y indicates via signaling that it can process ELs on a particular tunnel, an ingress LSR X can choose whether or not to insert ELs for packets going into that tunnel. Y MUST handle both cases.

The steps that X performs to insert ELs are as follows:

1. On an incoming packet, identify the application to which the packet belongs, and thereby pick the fields to input to the load balancing function; call the output LB.
2. Determine the application label AL (if any). Push <AL> onto the packet.
3. Based on the application, the load balancing output LB and other factors, determine the egress LSR Y, the tunnel to Y, the specific interface to the next hop, and thus the tunnel label TL. Use LB to generate the entropy label EL.
4. If, for the chosen tunnel, Y has not indicated that it can process ELs, push <TL> onto the packet. If Y has indicated that it can process ELs for the tunnel, push <TL, ELI, EL> onto the packet. X SHOULD put the same TTL and TC fields for the ELI as it does for TL. The TTL for the EL MUST be zero. The TC for the EL may be any value.
5. X then determines whether further tunnel hierarchy is needed; if so, X goes back to step 3, possibly with a new egress Y for the new tunnel. Otherwise, X is done, and sends out the packet.

Notes:

- a. X computes load balancing information and generates the EL based on the incoming application packet, even though the signaling of EL capability is associated with tunnels.
- b. X MAY insert several entropy labels in the stack (each, of course, preceded by an ELI), potentially one for each hierarchical tunnel, provided that the egress for that tunnel has

indicated that it can process ELs for that tunnel.

- c. X MUST NOT include an entropy label for a given tunnel unless the egress LSR Y has indicated that it can process entropy labels for that tunnel.
- d. The signaling and use of entropy labels in one direction (signaling from Y to X, and data path from X to Y) is completely independent of the signaling and use of entropy labels in the reverse direction (signaling from X to Y, and data path from Y to X).

[4.3.](#) Transit LSR

Transit LSRs MAY operate with no change in forwarding behavior. The following are suggestions for optimizations that improve load balancing, reduce the amount of packet data processed, and/or enhance backward compatibility.

If a transit LSR recognizes the ELI, it MAY choose to load balance solely on the following label (the EL); otherwise, it SHOULD use as much of the whole label stack as feasible as keys for the load balancing function, with the exception that reserved labels MUST NOT be used.

Some transit LSRs look beyond the label stack for better load balancing information. This is a simple, backward compatible approach in networks where some ingress LSRs impose ELs and others don't. However, this is of limited incremental value if an EL is indeed present, and requires more packet processing from the LSR. A transit LSR MAY choose to parse the label stack for the presence of the ELI, and look beyond the label stack only if it does not find it,

U: Unknown bit. This bit MUST be set to 1. If the ELC TLV is not understood by the receiver, then it MUST be ignored.

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

Type: Type field. To be assigned by IANA.

Length: Length field. This field specifies the total length in octets of the ELC TLV, and is currently defined to be 0.

5.1.1. Processing the ELC TLV

An LSR that receives a Label Mapping with the ELC TLV but does not understand it MUST propagate it intact to its neighbors and MUST NOT send a notification to the sender (following the meaning of the U- and F-bits).

An LSR X may receive multiple Label Mappings for a given FEC F from its neighbors. In its turn, X may advertise a Label Mapping for F to its neighbors. If X understands the ELC TLV, and if any of the

advertisements it received for FEC F does not include the ELC TLV, X MUST NOT include the ELC TLV in its own advertisements of F. If all the advertised Mappings for F include the ELC TLV, then X MUST advertise its Mapping for F with the ELC TLV. If any of X's neighbors resends its Mapping, sends a new Mapping or Withdraws a previously advertised Mapping for F, X MUST re-evaluate the status of ELC for FEC F, and, if there is a change, X MUST re-advertise its Mapping for F with the updated status of ELC.

5.2. BGP Signaling

When BGP [[RFC4271](#)] is used for distributing Network Layer Reachability Information (NLRI) as described in, for example, [[RFC3107](#)], the BGP UPDATE message may include the ELC attribute as part of the Path Attributes. This is an optional, transitive BGP attribute of type (to be assigned by IANA). The inclusion of this attribute with an NLRI indicates that the advertising BGP router can process entropy labels as an egress LSR for all routes in that NLRI.

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

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

A2: S can process entropy labels.

Suppose a BGP speaker T receives an UPDATE U with the ELC attribute. T has two choices. T can simply re-advertise U with the ELC attribute 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.

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 MAY include an ELC attribute 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.

Otherwise, T MUST remove the ELC attribute.

[5.3.](#) RSVP-TE Signaling

Entropy Label support is signaled in RSVP-TE [[RFC3209](#)] using the Entropy Label Capability (ELC) flag in the Attribute Flags TLV of the LSP_ATTRIBUTES object [[RFC5420](#)]. The presence of the ELC flag in a Path message indicates that the ingress can process entropy labels in the upstream direction; this only makes sense for a bidirectional LSP and MUST be ignored otherwise. The presence of the ELC flag in a Resv message indicates that the egress can process entropy labels in the downstream direction.

The bit number for the ELC flag is to be assigned by IANA.

5.4. Multicast LSPs and Entropy Labels

Multicast LSPs [[RFC4875](#)], [[RFC6388](#)] typically do not use ECMP for load balancing, as the combination of replication and multipathing can lead to duplicate traffic delivery. However, these LSPs can traverse bundled links [[RFC4201](#)] and LAGs. In both these cases, load balancing is useful, and hence entropy labels can be of value for multicast LSPs.

The methodology defined for entropy labels here will be used for multicast LSPs; however, the details of signaling and processing ELs for multicast LSPs will be specified in a companion document.

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.

[7.](#) MPLS-TP and Entropy Labels

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

[8.](#) Entropy Labels in Various Scenarios

This section describes the use of entropy labels in various scenarios.

In the figures below, the following conventions used to depict processing between X and Y. Note that control plane signaling goes right to left, whereas data plane processing goes left to right.

Protocols

Y:	<--- [L, E]	Y signals L to X
	X ----- Y	
LS:	<L, ELI, EL>	Label stack
X:	+<L, ELI, EL>	X pushes <L, ELI, EL>
Y:	-<L, ELI, EL>	Y pops <L, ELI, EL>

This means that Y signals to X label L for an LDP tunnel. E can be

one of:

0: meaning egress is NOT entropy label capable, or

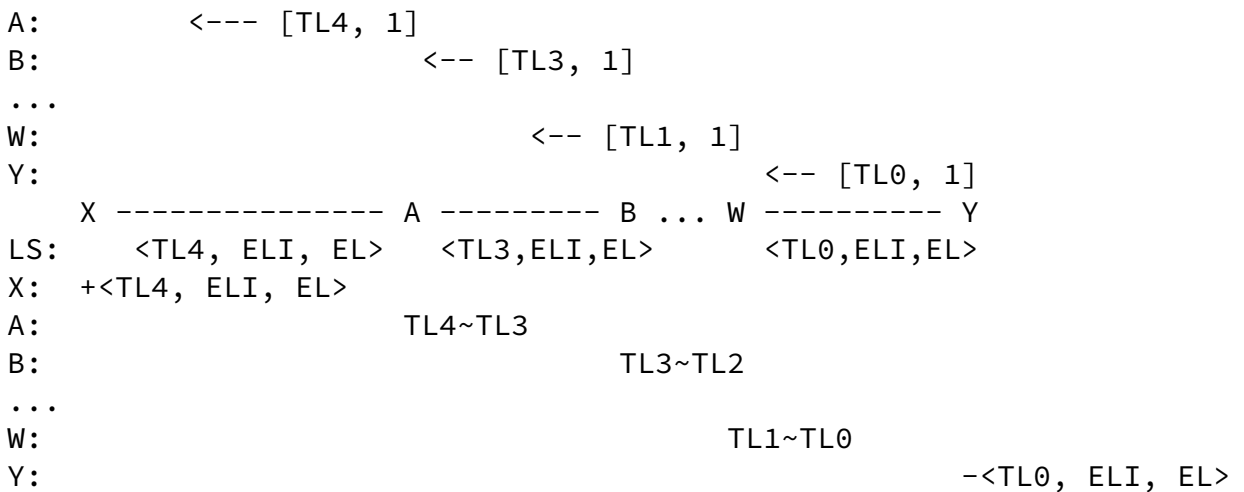
1: meaning egress is entropy label capable.

The line with LS: shows the label stack on the wire. Below that is the operation that each LSR does in the data plane, where + means push the following label stack, - means pop the following label stack, L~L' means swap L with L', and * means that the operation is not depicted.

8.1. LDP Tunnel

The following illustrates several simple intra-AS LDP tunnels. The first diagram shows ultimate hop popping (UHP) with ingress inserting an EL, the second UHP with no ELs, the third PHP with ELs, and finally, PHP with no ELs, but also with an application label AL (which could, for example, be a VPN label).

Note that, in all the cases below, the MPLS application does not matter; it may be that X pushes some more labels (perhaps for a VPN or VPLS) below the ones shown, and Y pops them.



LDP with UHP; ingress inserts ELs

Internet-Draft

MPLS Entropy Labels

July 2012

```

A:      <--- [TL4, 1]
B:      <-- [TL3, 1]
...
W:      <-- [TL1, 1]
Y:      <-- [TL0, 1]
X ----- A ----- B ... W ----- Y
LS:     <TL4>          <TL3>          <TL0>
X: +<TL4>
A:      TL4~TL3
B:      TL3~TL2
...
W:      TL1~TL0
Y:      -<TL0>

```

LDP with UHP; ingress does not insert ELs

```

A:      <--- [TL4, 1]
B:      <-- [TL3, 1]
...
W:      <-- [TL1, 1]
Y:      <-- [3, 1]
X ----- A ----- B ... W ----- Y
X: +<TL4, ELI, EL>
A:      TL4~TL3
B:      TL3~TL2
...
W:      -TL1
Y:      -<ELI, EL>

```

LDP with PHP; ingress inserts ELs

```

A:      <--- [TL4, 1]
B:      <-- [TL3, 1]
...
W:      <-- [TL1, 1]
Y:      <-- [3, 1]
VPN: <----- [AL]
X ----- A ----- B ... W ----- Y

```

```

LS:      <TL4, AL>      <TL3, AL>      <AL>
X:  +<TL4, AL>
A:              TL4~TL3
B:              TL3~TL2
...
W:              -TL1
Y:              -<AL>

```

LDP with PHP + VPN; ingress does not insert ELs

```

A:      <--- [TL4, 1]
B:              <-- [TL3, 1]
...
W:              <-- [TL1, 1]
Y:              <-- [3, 1]
VPN:  <----- [AL]
      X ----- A ----- B ... W ----- Y
LS:  <TL4,ELI,EL,AL> <TL3,ELI,EL,AL> <ELI,EL,AL>
X:  +<TL4,ELI,EL,AL>
A:              TL4~TL3
B:              TL3~TL2
...
W:              -TL1
Y:              -<ELI,EL,AL>

```

LDP with PHP + VPN; ingress inserts ELs

8.2. LDP Over RSVP-TE

The following illustrates "LDP over RSVP-TE" tunnels. X and Y are the ingress and egress (respectively) of the LDP tunnel; A and W are the ingress and egress of the RSVP-TE tunnel. It is assumed that both the LDP and RSVP-TE tunnels have PHP.

LDP with ELs, RSVP-TE without ELs

```

LDP:      <--- [L4, 1] <----- [L3, 1] <--- [3, 1]
RSVP-TE:      <-- [Rn, 0]
              <-- [3, 0]
      X ----- A ----- B ... W ----- Y
LS:  <L4, ELI, EL> <Rn,L3,ELI,EL> ... <ELI, EL>

```

DP: +<L4, ELI, EL> L4~<Rn, L3> * -L1 -<ELI, EL>

Figure 2: LDP over RSVP-TE Tunnels

8.3. MPLS Applications

An ingress LSR X must keep state per unicast tunnel as to whether the egress for that tunnel can process entropy labels. X does not have to keep state per application running over that tunnel. However, an ingress PE can choose on a per-application basis whether or not to insert ELs. For example, X may have an application for which it does not wish to use ECMP (e.g., circuit emulation), or for which it does not know which keys to use for load balancing (e.g., Appletalk over a pseudowire). In either of those cases, X may choose not to insert entropy labels, but may choose to insert entropy labels for an IP VPN

Kompella, et al. Expires January 11, 2013 [Page 18]

Internet-Draft MPLS Entropy Labels July 2012

over the same tunnel.

9. Security Considerations

This document describes advertisement of the capability to support receipt of entropy labels which an ingress LSR may insert in 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, BGP or RSVP-TE. Please refer to the Security Considerations section of these protocols ([\[RFC5036\]](#), [\[RFC4271\]](#) and [\[RFC3209\]](#)) for security mechanisms applicable to each.

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 will fall back to not using entropy labels for load-balancing traffic over LAG or ECMP paths which is in general no worse than the behavior observed in current production networks. That said, it is recommended that operators monitor changes to PE configurations and, more importantly, the fairness of load

distribution over 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 PEs and they should take corrective action.

[10.](#) IANA Considerations

[10.1.](#) Reserved Label for ELI

IANA is requested to allocate a reserved label for the Entropy Label Indicator (ELI) from the "Multiprotocol Label Switching Architecture (MPLS) Label Values" Registry.

[10.2.](#) LDP Entropy Label Capability 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 Capability TLV".

Kompella, et al. Expires January 11, 2013 [Page 19]

Internet-Draft MPLS Entropy Labels July 2012

[10.3.](#) BGP Entropy Label Capability Attribute

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

[10.4.](#) RSVP-TE Entropy Label Capability flag

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

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

[11.](#) Acknowledgments

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

Sincere thanks to Nischal Sheth for his many suggestions and comments, and his careful reading of the document, especially with regard to data plane processing of entropy labels.

12. References

12.1. 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.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", [RFC 5036](#), October 2007.

Kompella, et al. Expires January 11, 2013 [Page 20]

Internet-Draft MPLS Entropy Labels July 2012

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

12.2. Informative References

- [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.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", [RFC 5884](#), June 2010.
- [RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", [RFC 6388](#), November 2011.
- [RFC6391] Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires

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 Capability TLV will typically be applied only to the FEC for the Loopback IP address of the egress LSR and the egress LSR need 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
5340 Legacy Dr.
Plano, TX 75024
US

Email: lucy.yong@huawei.com

