

Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2014

A. Saxena
M. Jethanandani
Ciena Corporation
October 11, 2013

Upstream mapping in Echo Request
draft-ankur-mpls-upstream-mapping-00.txt

Abstract

This document describes an enhancement to the Echo Request and Echo Response message to carry upstream mapping information for co-routed bidirectional MPLS-TP tunnels.

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 April 14, 2014.

Copyright Notice

Copyright (c) 2013 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	2
1.1. Conventions Used in This Document	2
1.2. Abbreviations	2
2. Motivation	3
3. Packet format	3
3.1. Return Codes	3
3.2. Upstream TLV	3
4. Theory of Operations	5
4.1. Usefulness of Upstream TLV in a Bidirectional LSP sharing the same path	5
5. Security Considerations	6
6. IANA Considerations	7
6.1. New TLV	7
6.2. New Returns Codes	7
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	8

1. Introduction

Detecting MPLS Data Plane Failures [RFC4379] defines mechanisms for collecting downstream mapping information using Downstream Mapping (DSMAP) TLV. However, it does not describe a method by which similar information can be captured for the upstream mapping. An operator would generally be interested in the path taken by a packet in both the downstream and the upstream direction. Currently the only way the operator would be able to get that information would be by running the same command from the other end point. This document describes a method by which both Downstream Mapping (DSMAP) and Upstream Mapping (UPMAP) information can be collected by the same device.

1.1. Conventions Used in This Document

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] [RFC2119].

1.2. Abbreviations

Abbreviation	Meaning
DSMAP	Downstream Mapping

LSP	Label Switched Path
UPMAP	Upstream Mapping

2. Motivation

Detecting MPLS Data Plane Failures [RFC4379], describes the method by which an operator can find a fault in a bidirectional LSP. The operator starts by issuing a traceroute command from a node in the network to a node that is beyond the failed node. The operator then has to issue the same command from the node that was targeted in the first command. In many cases, the operator does not have access to the other node in the network. The operator is however interested in both the upstream and downstream LSP. This draft suggests a method by which the operator can issue a single traceroute command from one of the nodes in the network and mpls echo request and response packet will carry information to validate both the DSMAP and UPMAP information. The UPMAP can only be used in case of a bidirectional LSP, where the Forward LSP and the Reverse LSP share their path. When used in a non-bidirectional LSP, the UPMAP information will be filled with zeros and SHOULD be ignored on reception. A router that does not support the UPMAP TLV will silently ignore the TLV.

3. Packet format

The packet format is similar to the packet format described in Section 3 of RFC4379. [RFC4379]

This draft proposes to add two new return codes as outlined in section and a new TLV as specified in section .

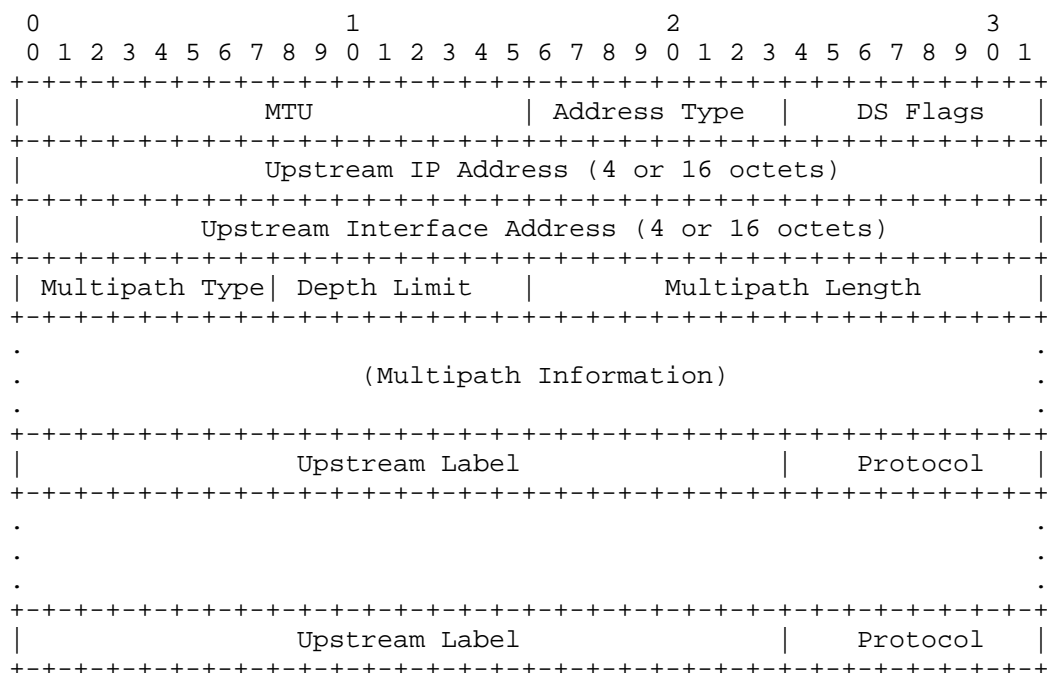
3.1. Return Codes

Value	Meaning
TBD	Upstream Mapping Mismatch
TBD	Downstream and Upstream Mapping Mismatch

3.2. Upstream TLV

The upstream mapping TLV is an object that MUST be included for all reply modes in the MPLS Echo packet when the operator has requested a traceroute on a bidirectional LSP, where the Forward LSP and Reverse LSP share the same path. The presence of an upstream TLV by the requester means that the replying router SHOULD validate the upstream TLV and if correct, fill the upstream TLV with upstream FEC of the replying router. If incorrect, it should fill the return code with one of the values specified in section to indicate "Upstream Mapping Mismatch" and leave the upstream TLV as is. If the node is an LER router and the upstream TLV is included in the MPLS echo request packet, it SHOULD fill the upstream TLV with the appropriate information and MUST include it in the MPLS echo reply.

As defined in RFC 4379, the length of this TLV is $K + M + 4*N$ octets, where M is the Multipath Length, and N is the number of Downstream Labels. Values for K are found in the description of Address Type below. The Value field of a Upstream TLV has the following format:



Upstream IP Address and Upstream Interface Address

IPv4 addresses and interface indices are encoded in 4 octets; IPv6 addresses are encoded in 16 octets. If the interface to the upstream node is numbered, then the Address Type MUST be set to IPv4 or IPv6,

the Upstream IP Address MUST be set to either the Upstream node's Router ID or the interface address of the Upstream node, and the Upstream Interface Address MUST be set to the upstream node's interface address. If the interface to the upstream node is unnumbered, the Address Type MUST be IPv4 Unnumbered or IPv6 Unnumbered, the Upstream IP Address MUST be the upstream node's Router ID, and the Upstream Interface Address MUST be set to the index assigned by the node to the interface.

If a node does not know the IP address of its neighbor, then it MUST set the Address Type to either IPv4 Unnumbered or IPv6 Unnumbered. For IPv4, it must set the Upstream IP Address to 127.0.0.1; for IPv6 the address is set to 0::1. In both cases, the interface index MUST be set to 0.

Upstream Label(s)

The set of labels in the label stack should appear as if this router were forwarding the packet through this interface. Any Implicit Null labels are explicitly included. Labels are treated as numbers, i.e., they are right justified in the field.

A Upstream Label is 24 bits, in the same format as an MPLS label minus the TTL field, i.e., the MSBit of the label is bit 0, the LSBit is bit 19, the EXP bits are bits 20-22, and bit 23 is the S bit. The replying router SHOULD fill in the EXP and S bits; the LSR receiving the echo reply MAY choose to ignore these bits.

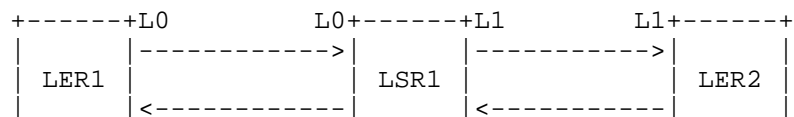
For explanation of rest of the fields in the Upstream TLV please refer section 3.3 of Detecting MPLS Data Plane Failures [RFC4379].

4. Theory of Operations

4.1. Usefulness of Upstream TLV in a Bidirectional LSP sharing the same path

The Upstream TLV MUST only be used in case of a bidirectional LSP where Forward and Reverse Paths are same, for example, MPLS-TP Co-routed tunnels or Multisegment Pseudo wire. In which case, the transit nodes will know all the information required to fill both the Downstream Mapping TLV and Upstream TLV.

Consider the following example:



+-----+L3 L3+-----+L2 L2+-----+

In the above fig, LER1 is the ingress node with forward out going label L0 and reverse in coming label of L3. LSR1 is the transit router with forward incoming and outgoing labels as L0 and L1 respectively and reverse incoming and outgoing labels of L2 and L3 respectively. LER2 is the egress router with forward incoming label of L1 and reverse outgoing label of L2.

The ingress node SHOULD fill its Downstream TLV for label L0 and Upstream TLV for label L3. When this MPLS Echo request packet (containing the Upstream TLV and the DownStream TLV) reaches the transit node, then the node validates both Upstream TLV for label L3 and Downstream TLV for Label L0. If the Downstream TLV for label L0 specified in the packet does not match the information the transit node has, then the transit node sends a return code specifying Downstream TLV mismatch. Similarly, if the Upstream TLV specified in the packet does not match the Upstream information the transit node has, then the transit node SHOULD send a return code of Upstream TLV mismatch. If both, the Upstream TLV and Downstream TLV does not match then the transit node should send a return code of Upstream and Downstream TLV mismatch. And if both the TLVs match then the transit node populates it's Downstream Mapping for label L1 and the Upstream Mapping for label L2 and sends the reply back to the ingress node. The ingress node uses this new Downstream TLV and Upstream TLV in it's next Echo Request packet. The egress node on receiving the Echo Request packet validates Upstream TLV and Downstream TLV. If both the TLVs match then the egress node SHOULD send a return code of Replying router is egress, else it SHOULD send the return code depending on which TLV did not match.

In case a bidirectional LSP does not share the Forward and Reverse path, for example, MPLS-TP Associated LSPs, traceroute SHOULD NOT add Upstream TLV as part of the MPLS Echo Request. If the Forward and Reverse LSPs are not on the same node then the transit node of the Forward LSP won't have any information to fill the Upstream TLV.

5. Security Considerations

Security considerations, as discussed in Detecting MPLS Data Plane Failures [RFC4379], are applicable to this document.

6. IANA Considerations

6.1. New TLV

IANA would have to assign a new TLV value to the following TLV from the "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry, "TLVs and sub-TLVs" sub-registry.

Upstream Detailed Mapping TLV (see Section).

6.2. New Returns Codes

IANA needs to assign a new Return Code values from the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry, "Return Codes" sub-registry, as follows using a Standards Action value.

Value	Meaning
TBD	Upstream mapping mismatch
TBD	Downstream and Upstream mapping mismatch

7. Acknowledgements

We would like to thank Ashesh Mishra and Vijay D'Souza for their feedback on this draft.

8. References

8.1. Normative References

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

8.2. Informative References

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

[RFC6424] Bahadur, N., Kompella, K., and G. Swallow, "Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels", RFC 6424, November 2011.

[RFC6426] Gray, E., Bahadur, N., Boutros, S., and R. Aggarwal, "MPLS On-Demand Connectivity Verification and Route Tracing", RFC 6426, November 2011.

Authors' Addresses

Ankur Saxena
Ciena Corporation
3939 N. First Street
San Jose, CA 95134
USA

Phone: +1 (408) 904-2109
Email: ankurpsaxena@gmail.com

Mahesh Jethanandani
Ciena Corporation
3939 N. First Street
San Jose, CA 95134
USA

Phone: +1 (408) 904-2160
Email: mjethanandani@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 16, 2014

M. Chen
X. Xu
Z. Li
Huawei
L. Fang
Microsoft
G. Mirsky
Ericsson
February 12, 2014

MultiProtocol Label Switching (MPLS) Source Label
draft-chen-mpls-source-label-02

Abstract

An MultiProtocol Label Switching (MPLS) label is originally defined to identify a Forwarding Equivalence Class (FEC), a packet is assigned to a specific FEC based on its network layer destination address. It's difficult or even impossible to derive the source information from the label. For some applications, source identification is a critical requirement. For example, performance monitoring, traffic matrix measurement and collection, where the monitoring node needs to identify where a packet was sent from.

This document introduces the concept of Source Label (SL) that is carried in the label stack and used to identify the ingress Label Switching Router (LSR) of an Label Switched Path (LSP).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 August 16, 2014.

Copyright Notice

Copyright (c) 2014 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. Problem Statement and Introduction	3
2. Source Label	4
3. Use Cases	4
3.1. Performance Measurement	4
3.2. Traffic Matrix Measurement and Steering	5
3.3. Source Filtering	7
4. Data Plane Processing	7
4.1. Ingress LSR	7
4.2. Transit LSR	8
4.3. Egress LSR	8
4.4. Penultimate Hop LSR	8
5. Source Label Signaling	8
5.1. Source Label Capability Signaling	8
5.1.1. LDP Extensions	8
5.1.2. BGP Extensions	9
5.1.3. RSVP-TE Extensions	10
5.2. Source Label Distribution	10
6. IANA Considerations	11
6.1. Source Label Indication	11
6.2. LDP Source Label Capability TLV	11
6.3. BGP Source Label Capability Attribute	11
6.4. RSVP-TE Source Label Capability	11
7. Security Considerations	11
8. Acknowledgements	12
9. References	12
9.1. Normative References	12

9.2. Informative References	12
Authors' Addresses	13

1. Problem Statement and Introduction

An MultiProtocol Label Switching (MPLS) label [RFC3031] is originally defined for packet forwarding and assumes the forwarding/destination address semantics. As no source address information is carried in the label stack, there is no way to directly derive the source address information from the label or label stack.

MPLS LSPs can be categorized into four different types:

Point-to-Point (P2P)

Point-to-Multipoint (P2MP)

Multipoint-to-Point (MP2P)

Multipoint-to-Multipoint (MP2MP)

For Resource Reservation Protocol Traffic Engineering (RSVP-TE) [RFC3209] based P2P and P2MP LSPs, the source address information may be implicitly derived from the label when Penultimate Hop Popping (PHP) is disabled. Note that such LSP may be characterized as MPLS-TP LSP [RFC5960]. But it requires that some further information is used (e.g., control plane information).

For Label Distribution Protocol (LDP) based LSPs [RFC5036] [RFC6388], Layer 3 Private Network (L3VPN) and Virtual Local Area Network (VPLS) LSPs that normally belong to P2MP, MP2P and MP2MP LSPs, ingress LSR that sent particular MPLS frame over P2MP, MP2P or MP2MP LSP cannot be identified by egress LSR.

Comparing to the pure IP forwarding where both source and destination addresses are encoded in the IP packet header, the essential issue of the MPLS encoding is that the label stack does not explicitly include any source address information, i.e., a Source Label (SL). For some applications, source identification is a critical requirement. For example, performance monitoring, the monitoring nodes need to identify where packets were sent from and then can count the packets according to some constraints. In addition, traffic matrix measurement and collection is the precondition of traffic steering, and capable of traffic steering is an important requirement of Software Defined Network (SDN). To measure and collect traffic matrix information, the source address information is necessary.

In addition, Segment Routing [I-D.filsfils-rtgwg-segment-routing] also explicitly points out that there are requirements to preserve the ingress information to fulfill the accounting and billing purposes.

This document introduces the concept of Source Label. An SL uniquely identifies a node within an administrative domain, it is carried in the label stack and used to identify one of the ingress LSR(s) of an LSP.

2. Source Label

A Source Label is defined to uniquely identify a node that is (one of) the ingress LSR(s) to a specific LSP. In its function as a Source Label, it MUST be unique within a domain. In cases where a Source Label is used across domains it MUST be unique within the scope it is used.

Source Labels SHOULD NOT be used for forwarding. The Source Labels are allocated from a dedicated label space that is completely different from the space of the normal Forwarding Labels. Configuration system (e.g., static configuration) is one way to make sure the uniqueness of each SL assigned to specific LSR. There may be some other potential dynamic solutions that can be used for SL allocation and distribution. This is out of the scope of this document.

In order to indicate whether a label is a source label, a Source Label Indicator (SLI) is introduced. The SLI is a (extended) special purpose label that is placed immediately before the source label in the label stack, which is used to indicate that the next label in the label stack is a source label. The value of SLI is TBD1.

3. Use Cases

This section outlines a number of use cases where solutions built on Source Label.

3.1. Performance Measurement

There are two typical types of performance measurement: one is active performance measurement, and the other is passive performance measurement.

In active performance measurement the receiver measures the injected packets to evaluate the performance of a path. The active measurement measures the performance of the extra injected packets. The IP Performance Metrics (IPPM) working group has defined

specifications [RFC4656][RFC5357] for the active performance measurement.

In passive performance measurement, no artificial traffic is injected into the flow and measurements are taken to record the performance metrics of the real traffic. The Multiprotocol Label Switching (MPLS) PM protocol [RFC6374] for packet loss is an example of passive performance measurement, but it can only apply to MPLS-TE LSPs. For a specific receiver, in order to count the received packets of a flow, it has to know whether a received packet belongs to which target flow under test and the source identification is a critical condition.

As discussed in the previous section, the existing MPLS label or label stack do not carry the source information. So, for an LSP, the ingress LSR can put a source label in the label stack, and then the egress LSR can use the source label for packets identifying and counting.

3.2. Traffic Matrix Measurement and Steering

A Traffic Matrix (TM) provides, for every ingress node (i) into the network and every egress node (j) out of the network, the volume of traffic $T(i,j)$ from i to j over a given time interval.

Since the ingress node knows the source and destination of the traffic, it's normal to measure the traffic matrix at every ingress node. But in some scenarios, it may need to measure the traffic at the egress or intermediate nodes. Taking Figure 1 as an example, from the west to east point of view, there are three ingress nodes (I1, I2 and I3) and three egress nodes (E1, E2 and E3), A, B and C are intermediate nodes. It is not necessary to measure the traffic matrix of the whole network all the time, it sometimes just wants to know the received traffic matrix of a specific egress node (e.g., E2). So, to measure received traffic matrix at node E2 would be then a better choice.

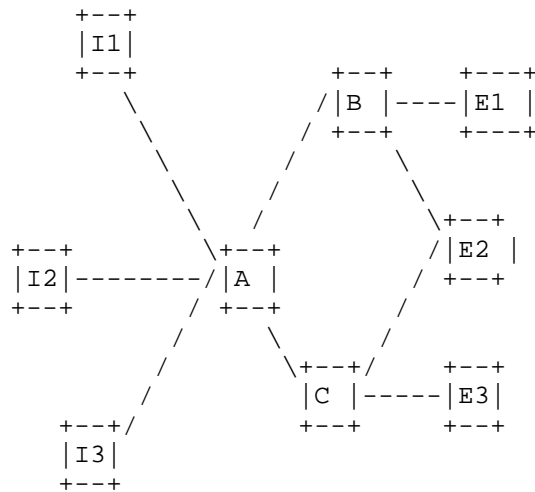


Figure 1: Traffic Matrix Measurement and Steering

In addition, for an intermediate node (e.g., node A), to steer traffic from congested path (e.g., path A-C) to idle path (e.g., path A-B), it needs to identify which flows contribute to the congestion and then determine which flows (e.g., the flows from specific ingress node) should be moved to the idle path.

Another scenario is domain exit traffic steering. Taking figure 2 as an example, node D is the domain gateway and has multiple exit links. Sometime, it may need to perform ingress/source node based traffic steering. It means that traffic from specific ingress node is required to be forwarded through specific exit link. For example, traffic from node A is required to be sent along with link 1, traffic from node B is required to be sent along with link 2, and traffic from node C is required to be sent along with link 3. To achieve this, node D needs to identify from which a flow is sent.

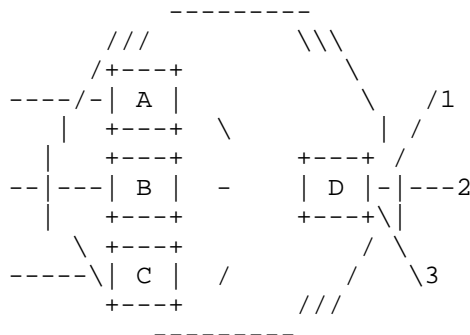


Figure 2: Domain Exit Traffic Steering

According above, wherever at egress or intermediate node, source identification is necessary. It should be possible to configure the ingress LSR to put the source label into the label stack to enable the egress and intermediate LSR to identify, measure and steer the traffic.

3.3. Source Filtering

Network Ingress Filtering [RFC2827] is an important tool to defeat DoS attacks and is widely deployed. In the past, since there is no source information carried in the stack, it's impossible to perform source filtering. With the Source Label, it enables to filter the packets with specific Source Label.

4. Data Plane Processing

4.1. Ingress LSR

For an LSP, the ingress LSR MUST make sure that the egress LSR is able to process the Source Label before inserting an SL and SLI into the label stack. Therefore, an egress LSR SHOULD signal (see Section 5.1) to the ingress LSR whether it is able to process the Source Label. Once the ingress LSR knows that the egress LSR can process Source Label, it can choose whether or not to insert the SL and SLI into the label stack.

When an SL to be included in a label stack, the steps are as follows:

1. Push the SL label, the BoS bit for the SL depends on whether the SL is the bottom label;
2. Push the SLI, the TTL and TC field for the SLI SHOULD be set to the same values as for the LSP Label (L);

3. Push the LSP Label (L) .

Then the label stack looks like: <...L, SLI, SL...>. There may be multiple pairs of SLI and SL inserted into the label stack, each pair is related to an LSP. For the given LSP, only one pair of SLI and SL SHOULD be inserted.

4.2. Transit LSR

There is no change in forwarding behavior for transit LSRs. But if a transit LSR can recognize the SLI, it can use the SL to collect traffic throughput and/or measure the performance of the LSP.

4.3. Egress LSR

When an egress LSR receives a packet with a SLI/SL pair, if the egress LSR is able to process the SL; it pops the LSP label (if any), SLI and SL; then processes remaining packet header as normal. If the egress LSR is not able to process the SL, the packet SHOULD be dropped as specified for the handling of any unknown label according to [RFC3031].

4.4. Penultimate Hop LSR

There is no change in forwarding behavior for the penultimate hop LSR.

5. Source Label Signaling

Source label signaling includes two aspects: one is source label capability signaling, the other is source label distribution.

5.1. Source Label Capability Signaling

Before inserting a source label in the label stack, an ingress LSR MUST know whether the egress LSR is able to process the source label. Therefore, an egress LSR should signal to the ingress LSRs its ability to process the Source Label. This is called Source Label Capability (SLC), it is very similar to the "Entropy Label Capability (ELC)" [RFC6790].

5.1.1. LDP Extensions

A new LDP TLV [RFC5036], SLC TLV, is defined to signal an egress's ability to process source label. The SLC TLV may appear as an Optional Parameter of the Label Mapping Message. The presence of the SLC TLV in a Label Mapping Message indicates to ingress LSRs that the egress LSR can process source labels for the associated LSP.

The structure of the SLC TLV is shown below.

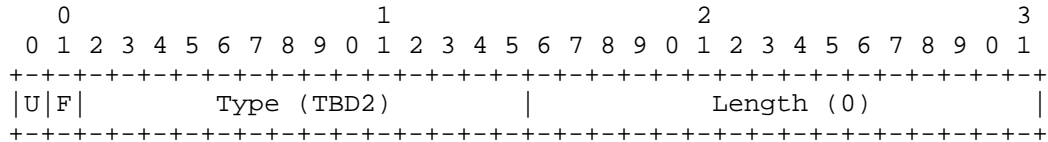


Figure 1: Source Label Capability TLV

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

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

Type: TBD2.

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

An LSR that receives a Label Mapping with the SLC 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). If the LSR has no other neighbors and does not understand the SLC TLV, means it is the ingress LSR, it could just ignore it. 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 SLC TLV, and if any of the advertisements it received for FEC F does not include the SLC TLV, X MUST NOT include the SLC TLV in its own advertisements of F. If all the advertised Mappings for F include the SLC TLV, then X MUST advertise its Mapping for F with the SLC TLV. If any of X's neighbors resends its Mapping, sends a new Mapping or sends a Label Withdraw for a previously advertised Mapping for F, X MUST re-evaluate the status of SLC for FEC F, and, if there is a change, X MUST re-advertise its Mapping for F with the updated status of SLC.

5.1.2. BGP Extensions

When Border Gateway Protocol (BGP) [RFC4271] is used for distributing Network Layer Reachability Information (NLRI) as described in, for example, [RFC3107], [RFC4364], the BGP UPDATE message may include the SLC attribute as part of the Path Attributes. This is an optional, transitive BGP attribute of value TBD3. The inclusion of this attribute with an NLRI indicates that the advertising BGP router can process source labels as an egress LSR for all routes in that NLRI.

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

A1: S sets the BGP NEXT_HOP attribute to itself AND

A2: S can process source labels.

Suppose a BGP speaker T receives an UPDATE U with the SLC attribute. T has two choices. T can simply re-advertise U with the SLC 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 SLC 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 source labels. Otherwise, T MUST remove the SLC attribute.

5.1.3. RSVP-TE Extensions

[RFC5420] introduces the LSP_ATTRIBUTES object, it gives a perfect way to carry LSP attribute through the object. To signal the Source Label Capability in RSVP-TE [RFC3209], this document defines a flag in the Attribute Flags TLV of the the LSP_ATTRIBUTES object [RFC3209].

The presence of the SLC flag in a Path message indicates that the ingress can process source labels in the upstream direction; this only makes sense for a bidirectional LSP and MUST be ignored otherwise. The presence of the SLC flag in a Resv message indicates that the egress can process source labels in the downstream direction. The bit number for the SLC flag is TBD4.

5.2. Source Label Distribution

Based on the Source Label, an egress or intermediate LSR can identify from where an MPLS packet is sent. To achieve this, the egress and/or intermediate LSRs have to know which ingress LSR is related to which Source Label before using the Source Label to derive the source information. Therefore, there needs to be a mechanism to distribute

the mapping information between an ingress LSR and its Source Label. This can be done, for example, by defining extensions to LDP, BGP, RSVP-TE and/or Interior Gateway Protocol (IGP) to distribute to source label mapping. The source label distribution will be defined in another document(s).

6. IANA Considerations

6.1. Source Label Indication

IANA is required to allocate a special purpose label (TBD1) for the Source Label Indicator (SLI) from the "Multiprotocol Label Switching Architecture (MPLS) Label Values" Registry.

6.2. LDP Source Label Capability TLV

IANA is requested to allocate a value of TBD2 from the IETF Consensus range (0x0001-0x07FF) in the "TLV Type Name Space" registry as the "Source Label Capability TLV".

6.3. BGP Source Label Capability Attribute

IANA is requested to allocate a Path Attribute Type Code TBD3 from the "BGP Path Attributes" registry as the "BGP Source Label Capability Attribute".

6.4. RSVP-TE Source Label Capability

IANA is requested to allocate a new bit from the "Attribute Flags" sub-registry of the "Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters" registry.

Bit	Name	Attribute	Attribute	RRO
No		Flags Path	Flags Resv	
-----+-----+-----+-----+-----				
TBD4	Source Label Capability	Yes	Yes	No

7. Security Considerations

This document does not introduce extra security issues. On the contrary, with the Source Label carried in the stack, it may bring additional security enhancement that enables an LSR to perform source label based checking and/or filtering.

8. Acknowledgements

The process of "Source Label Capability Signaling" is largely referred to the process of "ELC signaling"[RFC6790].

The authors would like to thank Carlos Pignataro, Loa Andersson for their review, suggestion and comments to this document.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, 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.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [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.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

9.2. Informative References

- [I-D.filsfils-rtgwg-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-rtgwg-segment-routing-01 (work in progress), October 2013.

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5960] Frost, D., Bryant, S., and M. Bocci, "MPLS Transport Profile Data Plane Architecture", RFC 5960, August 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.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

Authors' Addresses

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Zhenbin Li
Huawei

Email: lizhenbin@huawei.com

Luyuan Fang
Microsoft

Email: lufang@microsoft.com

Greg Mirsky
Ericsson

Email: Gregory.mirsky@ericsson.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

Z. Cui
R. Winter
NEC
February 14, 2014

Use Cases and Requirements for MPLS-TP multi-failure protection
draft-cui-mpls-tp-mfp-use-case-and-requirements-01

Abstract

MPLS Transport Profile (MPLS-TP) linear protection is defined in [RFC6378]. That however is limited to 1+1 and 1:1 protection and is not able to care that the multiple failures are occurred on both working and protection paths.

This document describes why we need to consider the case for multiple failures, and lists some requirements for multi-failure protection (MFP) functionality.

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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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	2
1.1. Document scope	2
1.2. Requirements notation	3
2. Summary of the problem statement and use case	3
3. Architecture	4
4. Requirements	4
4.1. Configuration	5
4.2. Resource reservation	5
4.3. Protection switching time	5
5. Security Considerations	5
6. IANA Considerations	5
7. Normative References	5
Authors' Addresses	6

1. Introduction

Network survivability - the ability of the network to remain functioning in the face of failures - is an important property of a network built to provide service guarantees.

For MPLS-TP networks, the protocol for linear protection is defined in [RFC6378]. That protocol can restore user traffic when a single failure condition is detected.

If need take a long time to repair, some operators may have to find other protection resources to protect the user traffic since the user traffic is unprotected. However, common linear protection not allows an overlap between a protection group and a other different path.

This document describes the detail of the problem statements, and lists a number of requirements for new protection functionality.

1.1. Document scope

This document describes the use cases and requirements for multi-failure protection in MPLS-TP networks without the use of control plane protocols. Existing solutions based on control plane such as GMPLS may be able to restore user traffic when multiple failures occur. Some networks however do not use full control plane operation for reasons such as service provider preferences, certain limitations or the requirement for fast service restoration (faster than achievable with control plane mechanisms). These networks are the

focus of this document which defines a set of requirements for multi-failure protection not based on control plane support.

1.2. Requirements notation

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].

2. Summary of the problem statement and use case

The following Figure 1 shows the network topology of an operation scenario. As shown in the Figure 1, there are three independent paths i, j and k between LER-A and LER-B. We assume a protection domain between LER-A and LER-B, using path i (working path) and j (protection path). Additionally, path k is a sharing resource.

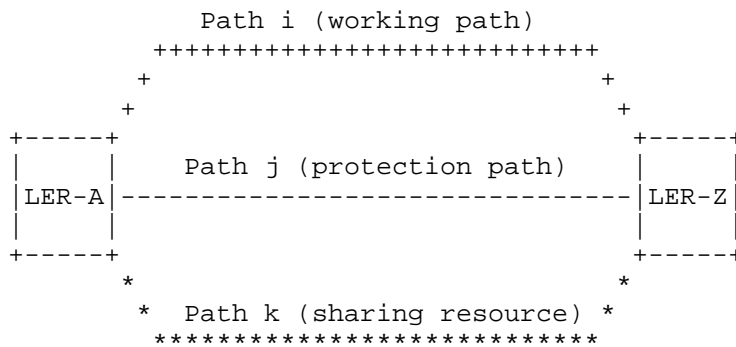


Figure 1: The network topology of an operation scenario

When a failure is detected in path i, we can restore user traffic to path j using existing protection schemes such as 1+1 protection and 1:1 protection.

However, since the user traffic is unprotected until the working path is repaired, some operators may take the following measures to protect the user service.

1) After a single failure condition is detected on the working path i,

1-1) Remove the protection group between path i and j.

1-2) Create a new protection group between path j and k to protect user traffic.

- 2) The failure condition of working path is repaired,
 - 2-1) In order to clear the sharing resources, remove the relationship of protection group between path j and k.
 - 2-2) Re-create a protection group between path i and j.

However, above progresses are very complex, may increase the risk for operation mistake and pressure. An automatic restoration mechanism such as GMPLS [RFC3945], are well-known. But some operators in particular in the transport sector that do not operate their MPLS networks under the control plane. Therefore, we suggest that define a non-control-plane based protection scheme that allows an overlap between a protection group and other paths.

3. Architecture

Figure 2 shows a new protection domain with a single working path and N protection paths. Each of the protection paths MAY be assigned a priority that could decide which protection path to use, i.e. protection path #1 > protection path #2, thus, the protection path #2 will not be selected to deliver user traffic when protection path #1 is available.

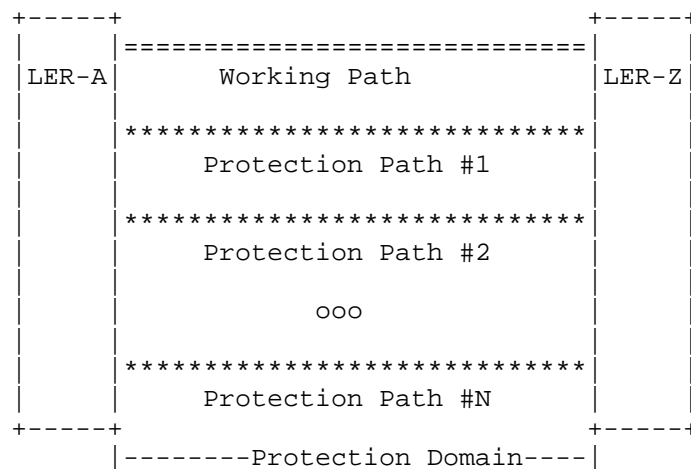


Figure 2: A basic example of multi-failure protection

4. Requirements

This section contains the requirements on the protection functionality derived from the problem statement and use case in section 2.

4.1. Configuration

Before failure detection and/or notification, one or more protection paths are instantiated between the same ingress-egress node pair as the working path shown in figure 2. The protection paths MAY be added or removed if necessary, but any performance degradation of user traffic should be avoided.

4.2. Resource reservation

The resource of the protection paths MAY be shared with other transport paths. In this case, the multiple failure protection SHOULD be supported by a shared mesh protection solution. The solution is out of scope of this document.

4.3. Protection switching time

Protection switching time refers to the transfer time (T_t) defined in [G.808.1] and recovery switching time defined in [RFC4427]. A multiple failure protection solution MUST support switching time within 50 ms from the moment of fault detection in a network.

5. Security Considerations

TBD

6. IANA Considerations

TBD

7. Normative References

- [I-D.ietf-mpls-smp-requirements]
Weingarten, Y., Aldrin, S., Pan, P., Ryoo, J., and G. Mirsky, "Requirements for MPLS Shared Mesh Protection", draft-ietf-mpls-smp-requirements-03 (work in progress), January 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3945] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.
- [RFC4427] Mannie, E. and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, March 2006.

[RFC6378] Weingarten, Y., Bryant, S., Osborne, E., Sprecher, N., and
A. Fulignoli, "MPLS Transport Profile (MPLS-TP) Linear
Protection", RFC 6378, October 2011.

Authors' Addresses

Zhenlong Cui
NEC

Email: c-sai@bx.jp.nec.com

Rolf Winter
NEC

Email: Rolf.Winter@neclab.eu

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 14, 2014

M. Bhatia
Alcatel-Lucent
D. Zhang
Huawei Technologies co., LTD.
M. Jethanandani
Ciena Corporation
September 10, 2013

Analysis of Bidirectional Forwarding Detection (BFD) Security According
to KARP Design Guide
draft-ietf-karp-bfd-analysis-01

Abstract

This document analyzes the Bidirectional Forwarding Detection protocol (BFD) according to the guidelines set forth in section 4.2 of KARP Design Guidelines [RFC6518].

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 March 14, 2014.

Copyright Notice

Copyright (c) 2013 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.

1. Introduction

This document performs a gap analysis of the current state of Bidirectional Forwarding Detection [RFC5880] according to the requirements of KARP Design Guidelines [RFC6518]. Previously, the OPSEC working group has provided an analysis of cryptographic issues with BFD in Issues with Existing Cryptographic Protection Methods for Routing Protocols [RFC6039].

The existing BFD specifications provide a basic security solution. Key ID is provided so that the key used in securing a packet can be changed on demand. Two cryptographic algorithms (MD5 and SHA-1) are supported for integrity protection of the control packets; the algorithms are both demonstrated to be subject to collision attacks. Routing protocols like RIPv2 Cryptographic Authentication [RFC4822], IS-IS Generic Cryptographic Authentication [RFC5310] and OSPFv2 HMAC-SHA Cryptographic Authentication [RFC5709] have started to use BFD for liveness check. Moving the routing protocols to a stronger algorithm while using weaker algorithm for BFD would require the attacker to bring down BFD in order to bring down the routing protocol. BFD therefore needs to match the routing protocols in its strength of algorithm.

While BFD uses a non-decreasing per-packet sequence number to protect itself from intra-connection replay attacks, it still leaves the protocol vulnerable to the inter-session replay attacks.

1.1. Conventions Used in This Document

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. Requirements to Meet

There are several requirements described in section 3 of The Threat Analysis and Requirements for Cryptographic Authentication of Routing Protocols' Transports [I-D.ietf-karp-threats-reqs] that BFD does not currently meet:

Replay Protection: BFD provides an incomplete intra-session and no inter-session replay attack protection; this creates significant denial-of-service opportunities.

Strong Algorithms: the cryptographic algorithms adopted for message authentication in BFD are MD5 or SHA-1 based. However, both algorithms are known to be vulnerable to collision attacks. BFD Generic Cryptographic Authentication [I-D.ietf-bfd-generic-crypto-auth] and Authenticating BFD using HMAC-SHA-2 procedures [I-D.ietf-bfd-hmac-sha] together propose a solution to support HMAC with the SHA-2 family of hash functions for BFD.

DoS Attacks: BFD packets can be sent at millisecond intervals (the protocol uses timers at microsecond intervals). When malicious packets are sent at short intervals, with the authentication bit set, it can cause a DoS attack.

The remainder of this document explains the details of how these requirements fail to be met and proposes mechanisms for addressing them.

3. Current State of Security Methods

BFD [RFC5880] describes five authentication mechanisms for the integrity protection of BFD control packets: Simple Password, Keyed MD5 The MD5 Message-Digest Algorithm [RFC1321], Meticulous Keyed MD5, Keyed SHA-1 and Meticulous SHA-1. In the simple password mechanism, every control packet is associated with a password transported in plain text; attacks eavesdropping the network traffic can easily learn the password and compromise the security of the corresponding BFD session. In the Keyed MD5 and the Meticulous Keyed MD5 mechanisms, BFD nodes use share secret keys to generate keyed MD5 digests for control packets. Similarly, in the Keyed SHA-1 and the Meticulous Keyed SHA-1 mechanisms, BFD nodes use shared secret keys to generate keyed SHA-1 digests for control packets. Note that in the keyed authentication mechanisms, every BFD control packet is associated with a non-decreasing 32-bit sequence number to resist replay attacks. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is only required to increase occasionally. However, in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms, the sequence member is required to monotonically increase with each successive packet.

Additionally, limited key updating functionality is provided. There is a Key ID in every authenticated BFD control packet, indicating the key used to hash the packet. However, there is no mechanism described to provide a smooth key rollover that the BFD routers can use when moving from one key to the other.

The BFD session timers are defined with the granularity of microseconds, and it is common in practice to send BFD packets at

millisecond intervals. Since the cryptographic sequence number space is only 32 bits, a sequence number used in a BFD session may reach its maximum value and roll over within limited period. For instance, if a sequence number is increased by one every 3.3 millisecond, then it will reach its maximum value in less than 24 weeks. This can result in potential inter-session replay attacks especially when BFD uses the non-meticulous authentication modes.

Note that when using authentication mechanisms, BFD requests the sequence of a received BFD packets drops with a limited range ($3 \times$ Detection time multiplier). Therefore, when meticulous authentication modes are used, a replayed BFD packet will be rejected if it cannot fit into a relatively short window (3 times of the detect interval of the session). This introduces some difficulties for replaying packets. However, in a non-meticulous authentication mode, such windows can be large as sequence numbers are only increased occasionally, thus making it easier to perform replay attacks .

In a BFD session, each node needs to select a 32-bit discriminator to identify itself. Therefore, a BFD session is identified by two discriminators. If a node will randomly select a new discriminator for a new session and use authentication mechanism to secure the control packets, inter-session replay attacks can be mitigated to some extent. However, in existing BFD demultiplexing mechanisms, the discriminators used in a new BFD session may be predictable. In some deployment scenarios, the discriminators of BFD routers may be decided by the destination and source addresses. So, if the sequence number of a BFD router rolls over for some reasons (e.g., reboot), the discriminators used to identify the new session will be identical to the ones used in the previous session. This makes performing a replay attack relatively simple.

BFD allows a mode called the echo mode. Echo packets are not defined in the BFD specification, though they can keep the BFD session up. The format of the echo packet is local to the sending side and there are no guidelines on the properties of these packets beyond the choice of the source and destination addresses. While the BFD specification recommends applying security mechanisms to prevent spoofing of these packets, there are no guidelines on what type of mechanisms are appropriate.

4. Impacts of BFD Replays

As discussed, BFD cannot meet the requirements of inter-session or intra-session replay protection. This section discusses the impacts of BFD replays.

When cryptographic authentication mechanisms are adopted for BFD, a non-decreasing 32-bit long sequence number is used. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is not required to increase for every packet. Therefore an attacker can keep replaying the packets with the latest sequence number until the sequence number is updated. This issue is eliminated in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms. However, note that a sequence number may reach its maximum and be rolled over in a session. In this case, without the support from an automatic key management mechanism, the BFD session will be vulnerable to replay attacks performed by sending the packets before the roll over of the sequence number. For instance, an attacker can replay a packet with a sequence number which is larger than the current one. If the replayed packet is accepted, the victim will reject the legal packets whose sequence members are less than the one in the replayed packet. Therefore, the attacker can get a good chance to bring down the BFD session.

Additionally, the BFD specification allows for the change of authentication state based on the state of a received packet. For instance, according to BFD [RFC5880], if the state of an accepted packet is down, the receiver of the packet needs to transfer its state to down as well. Therefore, an elaborately selected replayed packet can cause a serious denial-of-service attack.

BFD does not provide any solution to deal with inter-session replay attacks. If two subsequent BFD sessions adopt an identical discriminator pair and use the same cryptographic key to secure the control packets, it is intuitive to use a malicious authenticated packet (stored from the past session) to perform inter-connection replay attacks.

Any security issues in the BFD echo mode will directly affect the BFD protocol and session states, and hence the network stability. For instance, any replay attacks would be indistinguishable from normal forwarding of the tested router. An attack would still cause a faulty link to be believed to be up, but there is little that can be done about it. However, if the echo packets are guessable, it may be possible to spoof from an external source and cause BFD to believe that a one-way link is really bidirectional. As a result, it is important that the echo packets contain random material that is also checked upon reception.

5. Impact of New Authentication Requirements

BFD can be run in software or hardware. Hardware implementations run BFD at a much smaller timeout, typically in the order of few milliseconds. For instance with a timeout of 3.3 milliseconds, a BFD session is required to send or receive 3 packets every 10 milliseconds. Software implementations typically run with a timeout in hundreds of milliseconds.

Additionally, it is not common to find hardware support for computing the authentication data for the BFD session in hardware or software. In the keyed MD5 and Keyed SHA-1 implementation where the sequence number does not increase with every packet, software can be used to compute the authentication data. This is true if the time between increasing sequence number is long enough to compute the data in software. The ability to compute the hash in software is difficult with Meticulous Keyed MD5 and Meticulous Keyed SHA-1 if the time interval between transmits or between receives is small.

Implementors should assess the impact of authenticating BFD sessions on their platform.

6. Considerations for improvement

This section suggests changes that can be adopted to improve the protection of BFD.

As mentioned in section 3, a 32 bit sequence number space can wrap around in less than 24 weeks when set for the minimum time interval of 3.3 milliseconds. To prevent a replay attack the sequence number can be tied to notion of real time where part of the sequence number reflects say the UTC time. A replay attack therefore can easily be detected. However, it does require that the two stations exchanging BFD packets are synchornized with respect to time. Alternatively, the sequence number can be a nonce number generated using the shared key. But nonce numbers will also run out in 24 weeks.

Increasing the sequence number space to 64 bits makes the wrap around time be a little less than 2 million years. Combined with nonce or part of the number reflecting real time would make replay attacks difficult if not impossible.

The security risks brought by SHA-1 and MD5 have been well understood. However, when using stronger digest algorithm, e.g., SHA-2, the imposed computing overhead will seriously affect the performance of BFD implementation. In order to make the trade-off between the strong algorithm requirement and the imposed overhead, Galois Message Authentication Code (GMAC) can be a candidate option.

This algorithm is relative effective and has been supported by IPsec for data origin authentication. More detailed information can be found in [RFC4543].

7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

9. Acknowledgements

We would like to thank Alexander Vainshtein for his comments on this document.

10. References

10.1. Normative References

- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.

10.2. Informative References

- [I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., and D. Zhang, "BFD Generic Cryptographic Authentication", draft-ietf-bfd-generic-crypto-auth-04 (work in progress), April 2013.
- [I-D.ietf-bfd-hmac-sha]
Zhang, D., Bhatia, M., and V. Manral, "Authenticating BFD using HMAC-SHA-2 procedures", draft-ietf-bfd-hmac-sha-03 (work in progress), April 2013.
- [I-D.ietf-karp-threats-reqs]

Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", draft-ietf-karp-threats-reqs-07 (work in progress), December 2012.

[RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, May 2006.

[RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, February 2007.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

Authors' Addresses

Manav Bhatia
Alcatel-Lucent
Bangalore
India

Email: manav.bhatia@alcatel-lucent.com

Dacheng Zhang
Huawei Technologies co., LTD.
Beijing
China

Email: zhangdacheng@huawei.com

Mahesh Jethanandani
Ciena Corporation
1741 Technology Drive, #400
San Jose, CA 95110
USA

Phone: 408.436.3313
Fax: 408.436.5582
Email: mjethanandani@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August, 2014

Q. Zhao
Huawei Technology
K. Raza
C. Zhou
Cisco Systems
L. Fang
Microsoft
L. Li
China Mobile
D. King
Old Dog Consulting
February 14, 2014

LDP Extensions for Multi Topology
draft-ietf-mpls-ldp-multi-topology-11.txt

Abstract

Multi-Topology (MT) routing is supported in IP networks with the use of MT aware IGPs. In order to provide MT routing within Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) networks new extensions are required.

This document describes the LDP protocol extensions required to support MT routing in an MPLS environment.

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 August, 2014.

Copyright Notice

Copyright (c) 2014 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Signaling Extensions	4
3.1. Topology-Scoped Forwarding Equivalence Class (FEC)	4
3.2. New Address Families: MT IP	4
3.3. LDP FEC Elements with MT IP AF	5
3.4. IGP MT-ID Mapping and Translation	6
3.5. LDP MT Capability Advertisement	6
3.5.1. Protocol Extension	6
3.5.2. Procedures	7
3.6. LDP Sessions	8
3.7. Reserved MT ID Values	8
4. MT Applicability on FEC-based features	9
4.1. Typed Wildcard FEC Element	9
4.2. End-of-LIB	9
4.3. LSP Ping	9
4.3.1. New FEC Sub-Types	10
4.3.2. MT LDP IPv4 FEC Sub-TLV	10
4.3.3. MT LDP IPv6 FEC Sub-TLV	11
4.3.4. Operation Considerations	11
5. Error Handling	11
5.1. MT Error Notification for Invalid Topology ID	12
6. Backwards Compatibility	12
7. MPLS Forwarding in MT	12

Internet-Draft	LDP Multi Topology Extensions	February 2013
8.	Security Consideration12
9.	IANA Considerations13
10.	Manageability Considerations14
10.1.	Control of Function and Policy14
10.2.	Information and Data Models14
10.3.	Liveness Detection and Monitoring14
10.4.	Verify Correct Operations14
10.5.	Requirements On Other Protocols15
10.6.	Impact On Network Operations15
11.	Contributors15
12.	Acknowledgement16
13.	References16
13.1.	Normative References16
13.2.	Informative References17
	Authors' Addresses17

1. Introduction

Multi-Topology (MT) routing is supported in IP networks with the use of MT aware IGPs. It would be advantageous for communications Service Providers (CSP) to support Multiple Topologies (MT) within MPLS environments (MPLS-MT). The benefits of MPLS-MT enabled networks include:

- o A CSP may want to assign varying Quality of Service (QoS) profiles to traffic, based on a specific MT.
- o Separate routing and MPLS domains may be used to isolate multicast and IPv6 islands within the backbone network.
- o Specific IP address space could be routed across an MT based on security or operational isolation requirements.
- o Low latency links could be assigned to an MT for delay sensitive traffic.
- o Management traffic could be separated from customer traffic using multiple MTs, where the management traffic MT does not use links that carry customer traffic.

This document describes the Label Distribution Protocol (LDP) procedures and protocol extensions required to support MT routing in an MPLS environment.

This document also updates RFC4379 by defining two new FEC types for Label Switched Path (LSP) ping.

This document uses MPLS terminology defined in [RFC5036]. Additional terms are defined below:

- o MT-ID: A 16 bit value used to represent the Multi-Topology ID.
- o Default MT Topology: A topology that is built using the MT-ID default value of 0.
- o MT Topology: A topology that is built using the corresponding MT-ID.

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].

3. Signaling Extensions

3.1. Topology-Scoped Forwarding Equivalence Class (FEC)

LDP assigns and binds a label to a Forwarding Equivalence Class (FEC), where a FEC is a list of one or more FEC elements. To setup LSPs for unicast IP routing paths, LDP assigns local labels for IP prefixes, and advertises these labels to its peers so that an LSP is setup along the routing path. To setup MT LSPs for IP prefixes under a given topology scope, the LDP "prefix-related" FEC element must be extended to include topology information. This implies that MT-ID becomes an attribute of Prefix-related FEC element, and all FEC-Label binding operations are performed under the context of given topology (MT-ID).

The following Subsection 3.2(New Address Families (AF): MT IP) defines the extension required to bind "prefix-related" FEC to a topology.

3.2. New Address Families: MT IP

The LDP base specification [RFC5036] (Section 4.1) defines the "Prefix" FEC Element. The "Prefix" encoding is defined for a given "Address Family" (AF), and has length (in bits) specified by the "PreLen" field.

To extend IP address families for MT, two new Address Families named "MT IP" and "MT IPv6" are used to specify IPv4 and IPv6 prefixes within a topology scope.

The format of data associated with these new Address Families is described below:

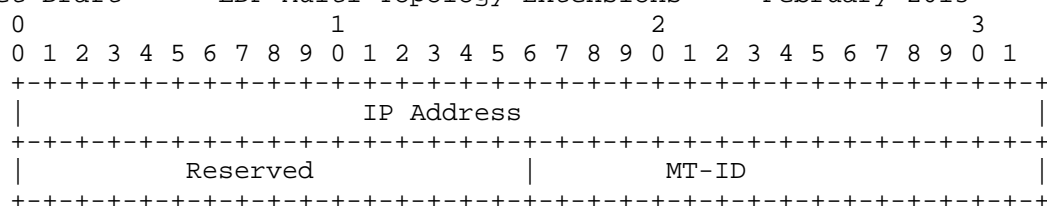


Figure 1: MT IP Address Family Format

Where "IP Address" is an IPv4 and IPv6 address/prefix for "MT IP" and "MT IPv6" AF respectively, and the field "MT-ID" corresponds to 16-bit Topology ID for given address.

The definition and usage of the rest fields in the FEC Elements are same as defined for IP/IPv6 AF. The value of MT-ID 0 corresponds to default topology and MUST be ignored on receipt so as to not cause any conflict/confusion with existing non-MT procedures.

The defined FEC Elements with "MT IP" Address Family can be used in any LDP message and procedures that currently specify and allow the use of FEC Elements with IP/IPv6 Address Family.

3.3. LDP FEC Elements with MT IP AF

The following section specifies the format extensions of the existing LDP FEC Elements to support MT. The "Address Family" of these FEC elements will be set to "MT IP" or "MT IPv6".

The MT Prefix FEC element encoding is as follows:

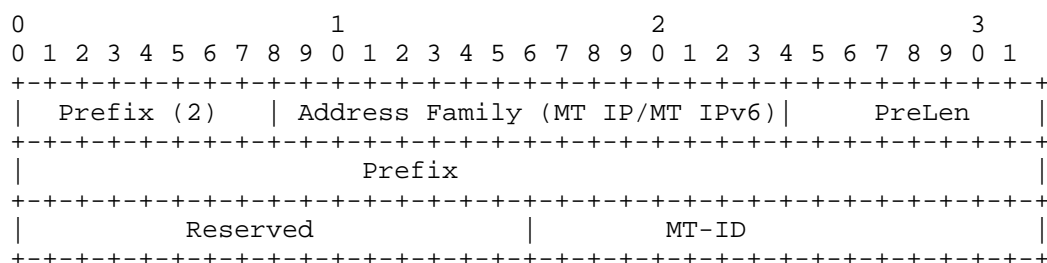


Figure 2: MT Prefix FEC Element Format

The MT Typed Wildcard FEC element encoding is as follows:


```

Internet-Draft      LDP Multi Topology Extensions      February 2013
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Typed Wcard (5) |   FEC Type   |   Len = 6   |   AF = MT IP .. |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... or MT IPv6 |           MT ID           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 3: MT Typed Wildcard FEC Element

The above format can be used for any LDP FEC Element that allows use of IP/IPv6 address family. In the scope of this document, the allowed "FEC Type" in a MT Typed Wildcard FEC Element is "Prefix" FEC element.

3.4. IGP MT-ID Mapping and Translation

The non-reserved non-special IGP MT-ID values can be used and carried in LDP without the need for translation. However, there is a need for translating reserved or special IGP MT-ID values to corresponding LDP MT-IDs. The assigned, unassigned and special LDP MT-ID values are requested In Section 9. (IANA Considerations).

How future LDP MT-ID values are allocated are out of of scope of this document. Instead a new Internet-Draft will be created to document the allocation policy and process for requesting new MT-ID values.

3.5. LDP MT Capability Advertisement

3.5.1. Protocol Extension

We specify a new LDP capability, named "Multi-Topology (MT)", which is defined in accordance with LDP Capability definition guidelines [RFC5561]. The LDP "MT" capability can be advertised by an LDP speaker to its peers either during the LDP session initialization or after the LDP session is setup to announce LSR capability to support MT for the given IP address family. An LDP speaker MUST NOT send messages containing MT FEC elements unless the peer has said it can handle it.

The MT capability is specified using "Multi-Topology Capability" TLV. The "Multi-Topology Capability" TLV format is in accordance with LDP capability guidelines as defined in [RFC5561]. To be able to specify IP address family, the capability specific data (i.e. "Capability Data" field of Capability TLV) is populated using "Typed Wildcard FEC Element" as defined in [RFC5918].

The format of "Multi-Topology Capability" TLV is as follows:

- o U- and F-bits: MUST be 1 and 0, respectively, as per Section 3. (Signaling Extensions) of LDP Capabilities [RFC5561].
- o Multi-Topology Capability: Capability TLV type (IANA assigned)
- o S-bit: MUST be 1 if used in LDP "Initialization" message. MAY be set to 0 or 1 in dynamic "Capability" message to advertise or withdraw the capability respectively.
- o Typed Wildcard FEC element(s): One or more elements specified as the "Capability data".
- o Length: length of Value field, starting from S bit, in octets.
- o The encoding of Typed Wildcard FEC element, as defined in [RFC5918], is defined in the section 3.3 (Typed Wildcard FEC Element) of this document. The MT-ID field of MT Typed Wildcard FEC Element MUST be set to "Wildcard Topology" when it is specified in MT Capability TLV.

To announce its MT capability for an IP address family, LDP FEC type, and Multi Topology, an LDP speaker sends an "MT Capability" including the exact Typed Wildcard FEC element with corresponding "AddressFamily" field (i.e., set to "MT IP" for IPv4 and set to "MT IPv6" for IPv6 address family), corresponding "FEC Type" field (i.e., set to "Prefix"), and corresponding "MT-ID". To announce its MT capability for both IPv4 and IPv6 address family, or for multiple FEC types, or for multiple Multi Topologies, an LDP speaker sends "MT Capability" with one or more MT Typed FEC elements in it.

- [Page 7]

advertised during LDP session initialization stage by including the LDP MT capability TLV in LDP Initialization message. After an LDP session is established, the MT capability can also be advertised or withdrawn using Capability message (only if "Dynamic Announcement" capability [RFC5561] has already been successfully negotiated).

- o If an LSR has not advertised MT capability, its peer MUST NOT send any LDP messages with FEC elements that include MT identifier to this LSR.
- o If an LSR is changed from non-MT capable to MT capable, it sets the S bit in MT capability TLV and advertises via the Capability message (if it supports Dynamic Announcement Capability). The existing LSP is treated as LSP for default MT (ID 0).
- o o If an LSR is changed from LDP-MT capable to non-MT capable, it initiates withdraw of all label mapping for existing LSPs of all non-default MTs. It also cleans up all the LSPs of all non-default MTs locally. Then it clears the S bit in MT capability TLV and advertises via the Capability message (if it supports Dynamic Announcement Capability). When an LSR knows the peer node is changed from LDP-MT capable to non-MT capable, it cleanup all the LSPs of all non-default MTs locally and initiate withdraw of all label mapping for existing LSPs of all non-default MTs. Both sides of the nodes send label release to its peer once they receive the label release messages even both sides have already cleaned up all the LSPs locally.
- o If an LSR does not support "Dynamic Announcement Capability", it MUST reset session with its peer whenever LSR changes its local capability with regards to supporting LDP MT.
- o If an LSR is changed from IGP-MT capable to non-MT capable, it may wait until the routes update to withdraw FEC and release the label mapping for existing LSPs of specific MT.

3.6. LDP Sessions

Since using different label spaces for different topologies would imply significant changes to the data plane, a single global label space is supported in this solution. There will be one session supported between a pair of peers, even if there are multiple topologies supported between these two peers.

3.7. Reserved MT ID Values

Certain MT topologies are assigned to serve predetermined purposes.

Internet-Draft LDP Multi Topology Extensions February 2013
In Section 9. (IANA Considerations), this document defines a new IANA registry "LDP Multi-Topology ID Name Space" under IANA "LDP Parameter" namespace to keep an LDP MT-ID reserved value.

If an LSR receives a FEC element with an "MT-ID" value that is "Reserved" for future use (and not IANA allocated yet), the LSR MUST abort the processing of the FEC element, and SHOULD send a notification message with status code "Invalid Topology ID" to the sender.

4. MT Applicability on FEC-based features

4.1. Typed Wildcard FEC Element

[RFC5918] extends base LDP and defines Typed Wildcard FEC Element framework. Typed Wildcard FEC element can be used in any LDP message to specify a wildcard operation/action for given type of FEC.

The MT extensions defined in document do not require any extension to procedures for Typed Wildcard FEC element, and these procedures apply as-is to MT wildcarding. The MT extensions, though, allow use of "MT IP" or "MT IPv6" in the Address Family field of the Typed Wildcard FEC element in order to use wildcard operations in the context of a given topology. The use of MT-scoped address family also allows us to specify MT-ID in these operations.

The defined format in Section 3.3 (Typed Wildcard FEC Element) allows an LSR to perform wildcard FEC operations under the scope of a topology. If an LSR wishes to perform wildcard operation that applies to all topologies, it can use a "Wildcard Topology" MT-ID. For example, upon local de-configuration of a topology "x", an LSR may send a typed wildcard label withdraw message with MT-ID "x" to withdraw all its labels from the peer that advertised under the scope of topology "x". Additionally, upon a global configuration change, an LSR may send a typed wildcard label withdraw message with the MT-ID set to "Wildcard Topology" to withdraw all its labels under all topologies from the peer.

4.2. End-of-LIB

[RFC5919] specifies extensions and procedures for an LDP speaker to signal its convergence for a given FEC type towards a peer. The procedures defined in [RFC5919] applies as-is to an MT FEC element. This allows an LDP speaker to signal its IP convergence using Typed Wildcard FEC element, and its MT IP convergence per topology using a MT Typed Wildcard FEC element.

4.3. LSP Ping

Internet-Draft LDP Multi Topology Extensions February 2013
[RFC4379] defines procedures to detect data-plane failures in MPLS LSPs via LSP ping. That specification defines a "Target FEC Stack" TLV that describes the FEC stack being tested. This TLV is sent in an MPLS echo request message towards LSPs egress LSR, and is forwarded along the same data path as other packets belonging to the FEC.

"Target FEC Stack" TLV contains one or more sub-TLVs pertaining to different FEC types. Section 3.2 of [RFC4379] defines Sub-Types and format for the FEC. To support LSP ping for MT LDP LSPs, this document defines following extensions to [RFC4379].

4.3.1. New FEC Sub-Types

We define two new FEC types for LSP ping:

- o MT LDP IPv4 FEC
- o MT LDP IPv6 FEC

We also define following new sub-types for sub-TLVs to specify these FECs in the "Target FEC Stack" TLV of [RFC4379]:

Sub-Type	Length	Value Field
-----	-----	-----
TBA5	8	MT LDP IPv4 prefix
TBA6	20	MT LDP IPv6 prefix

Figure 5: new sub-types for sub-TLVs

The rules and procedures of using these sub-TLVs in an MPLS echo request message are same as defined for LDP IPv4/IPv6 FEC sub-TLV types in [RFC4379].

4.3.2. MT LDP IPv4 FEC Sub-TLV

The format of "MT LDP IPv4 FEC" sub-TLV to be used in a "Target FEC Stack" [RFC4379] is:

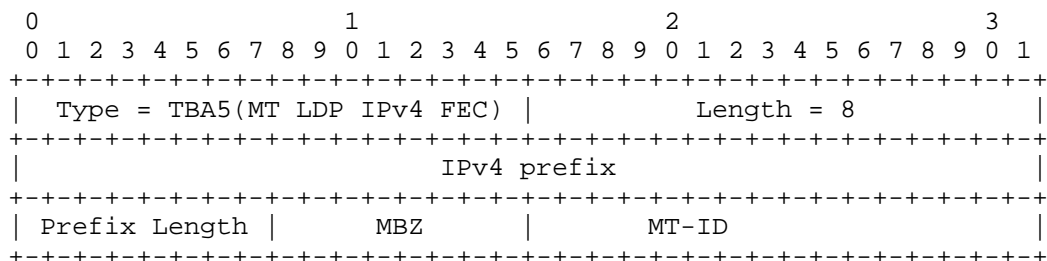


Figure 6: MT LDP IPv4 FEC sub-TLV

The format of this sub-TLV is similar to LDP IPv4 FEC sub-TLV as defined in [RFC4379]. In addition to "IPv4 prefix" and "Prefix Length" fields, this new sub-TLV also specifies MT-ID (Multi-Topology ID). The Length for this sub-TLV is 5.

4.3.3. MT LDP IPv6 FEC Sub-TLV

The format of "MT LDP IPv6 FEC" sub-TLV to be used in a "Target FEC Stack" [RFC4379] is:

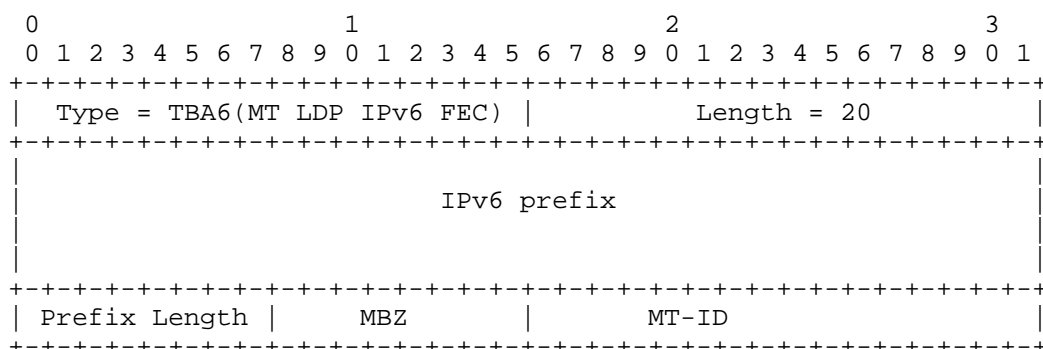


Figure 7: MT LDP IPv6 FEC sub-TLV

The format of this sub-TLV is similar to LDP IPv6 FEC sub-TLV as defined in [RFC4379]. In addition to "IPv6 prefix" and "Prefix Length" fields, this new sub-TLV also specifies MT-ID (Multi-Topology ID). The Length for this sub-TLV is 17.

4.3.4. Operation Considerations

To detect data plane failures using LSP Ping for a specific topology, the router will initiate an LSP Ping request with the target FEC stack TLV containing LDP MT IP Prefix Sub-TLV in the Echo Request packet. The Echo Request packet is sent with the label bound to the IP Prefix in the topology. Once the echo request packet reaches the target router, it will process the packet and perform checks for the LDP MT IP Prefix sub-TLV present in the Target FEC Stack as described in [RFC4379] and respond according to [RFC4379] processing rules. For the case that the LSP ping with return path is not specified, the reply packet must go through the default topology instead of the topology where the Echo Request goes through.

5. Error Handling

The extensions defined in this document utilize the existing LDP error handling defined in [RFC5036]. If an LSR receives an error notification from a peer for a session, it terminates the LDP session by closing the TCP transport connection for the session and discarding all multi-topology label mappings learned via the session.

5.1. MT Error Notification for Invalid Topology ID

An LSR should respond with an "Invalid Topology ID" status code in LDP Notification message when it receives an LDP message with a FEC element specifying an MT-ID which is not locally known or not supported. The LSR MUST also discard the entire message before sending the Notification.

6. Backwards Compatibility

The MPLS-MT solution is backwards compatible with existing LDP enhancements defined in [RFC5036], including message authenticity, integrity of message, and topology loop detection.

The legacy node which does not support MT should not receive any MT related LDP messages. In case the bad things happen, according to [RFC5036], processing of such messages should be aborted.

7. MPLS Forwarding in MT

Although forwarding is out of the scope of this draft, we include some forwarding consideration for informational purpose here.

The specified signaling mechanisms allow all the topologies to share the platform-specific label space, This feature allows the existing data plane techniques to be used. Also, there is no way for the data plane to associate a received packet with any one topology, meaning that topology-specific label spaces cannot be used.

8. Security Consideration

The use of MT over existing MPLS solutions does not offer any specific security benefit.

General LDP Communication security threats and how these may be mitigated are described in [RFC5036], these threats include:

- o Spoofing

- o Denial of Service

For further discussion regarding possible LDP communication threats and mitigation techniques see [RFC5920].

9. IANA Considerations

The document introduces following new protocol elements that require IANA consideration and assignments:

- o New LDP Capability TLV: "Multi-Topology Capability" TLV (requested code point: TBA1 from LDP registry "TLV Type Name Space").
- o New Status Code: "Invalid Topology ID" (requested code point: TBA2 from LDP registry "Status Code Name Space").

Registry:	
Range/Value	Description
-----	-----
TBA1	Invalid Topology ID

Figure 8: New Code Points for LDP Multi Topology Extensions

- o New address families under IANA registry "Address Family Numbers":
 - MT IP: Multi-Topology IP version 4 (requested codepoint:26)
 - MT IPv6: Multi-Topology IP version 6 (requested codepoint:27)

Figure 9: Address Family Numbers

- o New registry "MPLS Multi-Topology Identifiers". The allocation policies for this registry are:

Range/Value	Purpose	Reference
-----	-----	-----
0	Default/standard topology	[This.I-D]
1	IPv4 in-band management	[This.I-D]
2	IPv6 routing topology	[This.I-D]
3	IPv4 multicast topology	[This.I-D]
4	IPv6 multicast topology	[This.I-D]
5	IPv6 in-band management	[This.I-D]
6-3995	Unassigned for future IGP topologies	[This.I-D]
	Assigned by Standards Action	[This.I-D]
3996-4095	Experimental	[This.I-D]
4096-65534	Unassigned for MPLS topologies	[This.I-D]
	Assigned by Standards Action	[This.I-D]
65535	Wildcard Topology	[This.I-D]

- o New Sub-TLV Types for LSP ping: Following new sub-type values under TLV type 1 (Target FEC Stack) from "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry, and "TLVs and sub-TLVs" sub-registry.

Sub-Type	Value Field
-----	-----
TBA3	MT LDP IPv4 prefix
TBA4	MT LDP IPv6 prefix

Figure 11: New Sub-TLV Types for LSP ping

IANA should allocate the next available numbers for these TBAs.

As highlighted at the end of Section 3.4 (IGP MT-ID Mapping and Translation), a new Internet-Draft will be created to document the policy and process for allocating new MT-ID values.

10. Manageability Considerations

10.1. Control of Function and Policy

There are capabilities that should be configurable to enable good manageability. One such example is to allow enable or disable LDP Multi-Topology capability. It is assumed that the mapping of the LDP MT ID and IGP MT ID is manually configured on every router by default. If an automatic mapping between IGP MT IDs and LDP MT IDs is needed, there must be explicit configuration to do so.

10.2. Information and Data Models

Any extensions that may be required for existing MIBs are beyond the scope of this document.

10.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements.

10.4. Verify Correct Operations

If an operator is trying to debug LDP MT enabled network and wants to make the association between the LDP label advertisement and the IGP routing advertisement, then the user MUST understand the mapping mechanism to convert the IGP MT ID to the LDP MT ID. This type of mapping mechanisms is out of the scope of this document.

10.5. Requirements On Other Protocols

If the LDP MT ID has an implicit dependency on IGP MT ID, then the corresponding IGP MT features will need to be supported.

10.6. Impact On Network Operations

Mechanisms defined in this document do not have any impact on network operations.

11. Contributors

Ning So
Tata Communications
2613 Fairbourne Cir.
Plano, TX 75082
USA

Email: ning.so@tatacommunications.com

Raveendra Torvi
Juniper Networks
10, Technoogy Park Drive
Westford, MA 01886-3140
US

Email: rtorvi@juniper.net

Huaimo Chen
Huawei Technology
125 Nagog Technology Park
Acton, MA 01719
US

Email: huaimochen@huawei.com

Emily Chen
2717 Seville Blvd, Apt 1205,
Clearwater, FL 33764
US

Email: emily.chen220@gmail.com

Chen Li
China Mobile
53A, Xibianmennei Ave.
Xunwu District, Beijing 01719
China

Email: lichenyj@chinamobile.com

Lu Huang
China Mobile
53A, Xibianmennei Ave.
Xunwu District, Beijing 01719
China

Email: huanglu@chinamobile.com

Zhenbin Li
Huawei Technology
2330 Central Expressway
Santa Clara, CA 95050
US

Email: zhenbin.li@huawei.com

12. Acknowledgement

The authors would like to thank Dan Tappan, Nabil Bitar, Huang Xin, Eric Rosen, IJsbrand Wijnands, Dimitri Papadimitriou, Yiqun Chai, Pranjali Dutta, George Swallow, Curtis Villamizar, Adrian Farrel, Alia Atlas, Loa Anderson, Joel Halpern and Kathleen Moriarty for their valuable comments on this draft.

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.
- [RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009.

Internet-Draft LDP Multi Topology Extensions February 2013
[RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, August 2010.

[RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas, "Signaling LDP Label Advertisement Completion", RFC 5919, August 2010.

13.2. Informative References

[RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.

Authors' Addresses

Quintin Zhao
Huawei Technology
125 Nagog Technology Park
Acton, MA 01719
US

Email: quintin.zhao@huawei.com

Kamran Raza
Cisco Systems
2000 Innovation Drive
Kanata, ON K2K-3E8, MA
Canada

Email: E-mail: skraza@cisco.com

Chao Zhou
Cisco Systems
300 Beaver Brook Road
Boxborough, MA 01719
US

Email: czhou@cisco.com

Luyuan Fang
Microsoft

Email: lufang@microsoft.com

Internet-Draft LDP Multi Topology Extensions February 2013
Lianyuan Li
China Mobile
53A, Xibianmennei Ave.
Xunwu District, Beijing 01719
China

Email: lilianyuan@chinamobile.com

Daniel King
Old Dog Consulting

Email: lufang@microsoft.com

Network Working Group
INTERNET-DRAFT
Intended Status: Standards Track
Expires: June 21, 2015

M.Venkatesan
Dell Inc.
Kannan KV Sampath
Redeem
Sam K. Aldrin
Huawei Technologies
Thomas D. Nadeau
Brocade

December 18, 2014

MPLS-TP Traffic Engineering (TE) Management Information Base (MIB)
draft-ietf-mpls-tp-te-mib-11.txt

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes additional managed objects and textual conventions for Tunnels, Identifiers and Label Switching Router to support Multiprotocol Label Switching (MPLS) MIB modules for transport networks.

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/lid-abstracts.txt>.

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

This Internet-Draft will expire on June 21, 2015.

Copyright and License Notice

Copyright (c) 2014 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	4
2. The Internet-Standard Management Framework	5
3. Overview	5
3.1. Conventions used in this document	5
3.2. Terminology	5
3.3. Acronyms	7
4. Motivations	7
5. Feature List	7
6. Outline	9
6.1 MIB Module Extensions	9
6.1.1 Summary of MIB Module changes	9
6.2 MPLS-TE-EXT-STD-MIB	10
6.2.1 mplstunnelExtNodeConfigTable	10
6.2.2 mplstunnelExtNodeIpMapTable	11
6.2.3 mplstunnelExtNodeIccMapTable	11
6.2.4 mplstunnelExtTable	11
6.3 MPLS-TC-EXT-STD-MIB	11
6.4 MPLS-ID-STD-MIB	11
6.5 MPLS-LSR-EXT-STD-MIB	12
6.6 The Use of RowPointer	12
7. MIB Modules Interdependencies	13
8. Dependencies between MIB Module Tables	14
9. Example of MPLS-TP Tunnel Setup	15
9.1. Example of MPLS-TP static co-routed bidirectional tunnel setup	16
9.1.1. mplstunnelEntry	16
9.1.2. mplstunnelExtEntry	17
9.1.3. Forward direction mplsoutSegmentEntry	17
9.1.4. Reverse direction mplsinSegmentEntry	17
9.1.5. Forward direction mplsxCentry	18
9.1.6. Reverse direction mplsxCentry	18
9.1.7. Forward direction mplsxCextEntry	19

9.1.8. Reverse direction mplsXCExtEntry	19
9.2. Example of MPLS-TP static associated bidirectional tunnel setup	19
9.2.1. Forward direction mplsTunnelEntry	19
9.2.2. Forward direction mplsTunnelExtEntry	20
9.2.3. Forward direction mplsOutSegmentTable	20
9.2.4. Forward direction mplsXCEntry	21
9.2.5. Forward direction mplsXCExtEntry	21
9.2.6. Reverse direction mplsTunnelEntry	21
9.2.7. Reverse direction mplsTunnelExtEntry	22
9.2.8. Reverse direction mplsInSegmentEntry	22
9.2.9. Reverse direction mplsXCEntry	23
9.2.10. Reverse direction mplsXCExtEntry	23
9.3. Example of MPLS-TP signaled co-routed bidirectional tunnel setup	23
9.3.1. mplsTunnelEntry	24
9.3.2. mplsTunnelExtEntry	24
9.3.3. Forward direction mplsOutSegmentEntry	25
9.3.4. Reverse direction mplsInSegmentEntry	25
9.3.5. Forward direction mplsXCEntry	25
9.3.6. Reverse direction mplsXCEntry	25
9.3.7. Forward direction mplsXCExtEntry	25
9.3.8. Reverse direction mplsXCExtEntry	26
10. MPLS Textual Convention Extension MIB definitions	26
11. MPLS Identifier MIB definitions	29
12. MPLS LSR Extension MIB definitions	34
13. MPLS Tunnel Extension MIB definitions	39
14. Security Consideration	56
15. IANA Considerations	57
15.1. IANA Considerations for MPLS-TC-EXT-STD-MIB	58
15.2. IANA Considerations for MPLS-ID-STD-MIB	58
15.3. IANA Considerations for MPLS-LSR-EXT-STD-MIB	58
15.4. IANA Considerations for MPLS-TE-EXT-STD-MIB	58
16. References	58
16.1. Normative References	58
16.2. Informative References	59
17. Acknowledgments	60
18. Authors' Addresses	60

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes additional textual conventions and managed objects for Tunnels, Identifiers and Label Switching Router to support Multiprotocol Label Switching (MPLS) MIB modules for transport networks. MIB modules defined in this document extend the existing MPLS MIB objects in such a way that they support MPLS-TP but also other MPLS networks as well. Hence, the MPLS-TP name is not included in the MIB module name.

As described in the MPLS Traffic Engineering (TE) Management Information Base (MIB) definition [RFC3812], MPLS traffic engineering is concerned with the creation and management of MPLS tunnels. This term is a shorthand for a combination of one or more LSPs linking an ingress and an egress LSR. Several types of point-to-point MPLS tunnels may be constructed between a pair of LSRs A and B:

- Unidirectional with a single LSP (say) from A to B.
- Associated bidirectional consisting of two separately routed LSPs, one linking A to B and the other linking B to A. Together the pair provide a single logical bidirectional transport path.
- Co-routed bidirectional consisting of an associated bidirectional tunnel but with the second LSP from B to A following the reverse of the path of the LSP from A to B, in terms of both nodes and links.

Tunnels may be either statically configured by management action or dynamically created using a LSP management protocol.

The existing MPLS TE MIB [RFC3812] and the Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base [RFC4802] address only a subset of the combinations of statically and dynamically configured tunnel types, catering for statically configured unidirectional tunnels together with dynamically configured unidirectional and co-routed bidirectional tunnels. They are also restricted to two end point LSRs identified by IP addresses.

The MPLS-TP TE MIB defined in this document extends the MIB modules defined in [RFC3812] to cover all six combinations (that is adding support for statically configured associated and co-routed bidirectional plus dynamically configured associated bidirectional tunnels). It also extends support to end points that are identified other than with IP addresses.

This support is provided by a suite of four MIB modules that are to be used in conjunction with the MIB modules defined in [RFC3812] and the companion document [RFC3813] for MPLS Transport Profile (MPLS-TP) tunnel management.

At the time of writing, SNMP SET is no longer recommended as a way to configure MPLS networks as was described in [RFC3812]. However, since the MIB modules specified in this document extend and are intended to work in parallel with the MIB modules for MPLS specified in [RFC3812], certain objects defined here are specified with MAX-ACCESS of read-write or read-create so that specifications of the base tables in [RFC3812] and the extensions in this document are consistent. Although the examples described in Section 9 specify means to configure MPLS-TP tunnels in a similar way to the examples in [RFC3812], this should be seen as indicating how the MIB values would be returned in the specified circumstances having been configured by alternative means.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58 [RFC2578], STD 58 [RFC2579] and STD 58 [RFC2580].

3. Overview

3.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3.2. Terminology

This document uses terminology from the Multiprotocol Label Switching Architecture [RFC3031], Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) [RFC3812], Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB) [RFC3813] and MPLS Transport Profile

(MPLS-TP) Identifiers [RFC6370].

3.3. Acronyms

CC: Country Code
ICC: ITU Carrier Code
LSP: Label Switching Path
LSR: Label Switching Router
MPLS-TP: MPLS Transport Profile
TE: Traffic Engineering
TP: Transport Profile

4. Motivations

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) [RFC3812] provides support for Traffic Engineering tunnels. In MPLS, the actual transport of packets is provided by Label Switched Paths (LSPs). A transport service may be composed of multiple LSPs. In order to clearly identify the MPLS-TP service, as defined in [RFC6370], we use the term "MPLS-TP Tunnel" or simply "tunnel". However, with MPLS-TP, the characteristics of the tunnels were enhanced. For example, MPLS-TP tunnels, are bidirectional in nature and could be used with non-IP identifiers for the tunnel end points. As the existing MPLS-TE-STD-MIB and GMPLS-TE-STD-MIB were defined mainly to support unidirectional tunnels and signaled co-routed bidirectional tunnel definitions respectively, these existing MIB modules are not sufficient to capture all the characteristics of the tunnels. Hence, enhancing the MIB modules to support MPLS-TP tunnels is required. As most of the attributes of MPLS Traffic Engineering tunnels are also applicable to MPLS-TP tunnels, it is optimal to re-use and extend the existing MIB module definition instead of defining a new MIB module.

This document defines four additional MIB modules, namely MPLS-TE-EXT-STD-MIB, MPLS-TC-EXT-STD-MIB, MPLS-ID-STD-MIB and MPLS-LSR-EXT-STD-MIB. As these additional MIB modules are required for MPLS-TP functionality, these are all defined in this document, instead of being documented separately.

5. Feature List

The MIBs in this document satisfy the following requirements and constraints:

The MIB modules, taken together, support statically configured and dynamically signaled point-to-point, co-routed bidirectional and associated bidirectional tunnels.

- The MPLS tunnels need not be interfaces, but it is possible to configure an MPLS TP tunnel as an interface. Same ifType 150,

as defined in section 8 of [RFC3812], will be used for MPLS-TP tunnels as well.

- The `mplsTunnelTable` [RFC3812] is also to be used for MPLS-TP tunnels.
- New MPLS-TP specific textual conventions and identifiers are required.
- The `mplsTunnelTable` is sparsely extended to support MPLS-TP tunnel specific objects.
- A node configuration table (`mplsTunnelExtNodeConfigTable`), as detailed in the below section 6.1.2, is used to translate the `Global_ID::Node_ID` or `ICC_Operator_ID::Node_ID` to the local identifier in order to index `mplsTunnelTable`.
- The `mplsXCTable` is sparsely extended to support MPLS-TP XC(Cross Connect) specific objects.
- The MIB module supports persistent, as well as non-persistent tunnels.

6. Outline

Traffic Engineering support for the MPLS-TP tunnels requires the set up of the co-routed or associated bidirectional tunnel. The tables and MIB modules that are mentioned in the below subsections support the functionality described in documents [RFC5654] and [RFC6370]. These tables support both IP compatible and ITU Carrier Code (ICC) based tunnel configurations.

The below Figure 1 depicts how the table references are followed in this MIB.

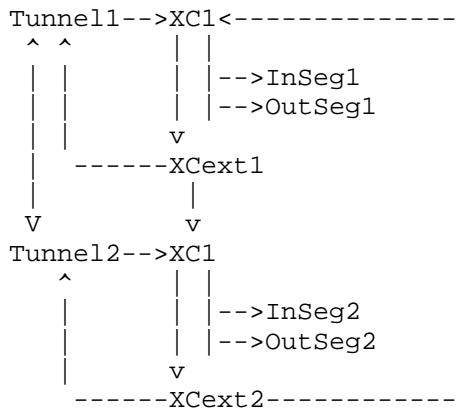


Figure 1: Table references of MIB modules

6.1 MIB Module Extensions

Four MIB modules are extended to support MPLS-TP tunnels, namely, MPLS-TE-EXT-STD-MIB, MPLS-TC-EXT-STD-MIB, MPLS-ID-STD-MIB and MPLS-LSR-EXT-STD-MIB. Following section provides the summary of changes.

6.1.1 Summary of MIB Module changes

- Node configuration table (mplsTunnelExtNodeConfigTable) for setting the local identifier for Tunnel Ingress and Egress identifiers.
- Node IP map table (mplsTunnelExtNodeIpMapTable) for querying the local identifier for a given Global_ID and Node_ID.
- Node ICC map table (mplsTunnelExtNodeIccMapTable) for querying the local identifier for a given ICC_Operator_ID and Node_ID.

- Tunnel extension table (mplsTunnelExtTable) for setting up MPLS-TP tunnels with sparse extension of mplsTunnelTable.
- Textual conventions and object definitions for MPLS-TP Tunnels
- Cross connect extension table (mplsXCExtTable) for setting up the MPLS-TP LSPs.

These tables are described in the subsequent sections.

6.2 MPLS-TE-EXT-STD-MIB

The TE MIB module extensions and details of the tables are described in the following sections.

6.2.1 mplsTunnelExtNodeConfigTable

The mplsTunnelExtNodeConfigTable is used to assign a local identifier for a given ICC_Operator_ID::Node_ID or Global_ID::Node_ID combination as defined in [RFC6923] and [RFC6370] respectively. The CC is a string of two characters, each being an uppercase Basic Latin alphabetic (i.e., A-Z). The ICC is a string of one to six characters, each an upper case Basic Latin alphabetic (i.e., A-Z) or numeric (i.e., 0-9). All of the characters are encoded using [T.50] as described in [RFC6370].

In the IP compatible mode, Global_ID::Node_ID, is used to uniquely identify a node. For each ICC_Operator_ID::Node_ID or Global_ID::Node_ID, there is a unique entry in the table representing a node. As the regular TE tunnels use IP address as LSR ID, the local identifier should be below the first valid IP address, which is 16777216[1.0.0.0]. Every node is assigned a local identifier within a range of 0 to 16777215. This local identifier is used for indexing into mplsTunnelTable as mplsTunnelIngressLSRId and mplsTunnelEgressLSRId.

For IP compatible environment, MPLS-TP tunnel is indexed by Tunnel Index, Tunnel Instance, Source Global_ID, Source Node_ID, Destination Global_ID and Destination Node_ID.

For ICC based environment, MPLS-TP tunnel is indexed by Tunnel Index, Tunnel Instance, Source CC, Source ICC, Source Node_ID, Destination CC, Destination ICC and Destination Node_ID.

As mplsTunnelTable is indexed by mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId, and mplsTunnelEgressLSRId, the MPLS-TP tunnel identifiers cannot be used directly.

The `mplsTunnelExtNodeConfigTable` will be used to store an entry for `ICC_Operator_ID::Node_ID` or `Global_ID::Node_ID` with a local identifier to be used as LSR ID in `mplsTunnelTable`.

6.2.2 `mplsTunnelExtNodeIpMapTable`

The read-only `mplsTunnelExtNodeIpMapTable` is used to query the local identifier assigned and stored in `mplsTunnelExtNodeConfigTable` for a given `Global_ID::Node_ID`. In order to query the local identifier, in the IP compatible mode, this table is indexed with `Global_ID::Node_ID`. In the IP compatible mode for a TP tunnel, `Global_ID::Node_ID` is used.

A separate query is made to get the local identifier of both Ingress and Egress `Global_ID::Node_ID` identifiers. These local identifiers are used as `mplsTunnelIngressLSRId` and `mplsTunnelEgressLSRId`, while indexing `mplsTunnelTable`.

6.2.3 `mplsTunnelExtNodeIccMapTable`

The read-only `mplsTunnelExtNodeIccMapTable` is used to query the local identifier assigned and stored in the `mplsTunnelExtNodeConfigTable` for a given `ICC_Operator_ID::Node_ID`.

A separate query is made to get the local identifier of both Ingress and Egress `ICC_Operator_ID::Node_ID`. These local identifiers are used as `mplsTunnelIngressLSRId` and `mplsTunnelEgressLSRId`, while indexing `mplsTunnelTable`.

6.2.4 `mplsTunnelExtTable`

This table sparsely extends the `mplsTunnelTable` in order to support MPLS-TP tunnels with additional objects. All the additional attributes specific to supporting TP tunnel are contained in this extended table and could be accessed with the `mplsTunnelTable` indices.

The `gmplsTunnelReversePerfTable` [RFC4802] should be used to provide per-tunnel packet performance information for the reverse direction of a bidirectional tunnel. It can be seen as supplementing the `mplsTunnelPerfTable`, which augments the `mplsTunnelTable`.

6.3 MPLS-TC-EXT-STD-MIB

This MIB module contains textual Conventions for LSPs of MPLS based transport networks.

6.4 MPLS-ID-STD-MIB

This MIB module contains generic object definitions for MPLS Traffic Engineering in transport networks.

6.5 MPLS-LSR-EXT-STD-MIB

This MIB module contains generic object definitions (Cross connect extension table - `mplsXCExtTable`, for setting up the MPLS-TP LSPs with sparse extension of `mplsXCTable`) for MPLS LSRs in transport networks.

6.6 The Use of RowPointer

This document follows the RowPointer usage as described in the section 10 of [RFC3812].

A new RowPointer object, `mplsTunnelExtOppositeDirPtr`, is added to `mplsTunnelExtTable` of MPLS-TE-EXT-STD-MIB MIB module. This RowPointer object points to the opposite direction tunnel entry.

Two additional RowPointers objects, `mplsXCExtTunnelPointer` and `mplsXCExtOppositeDirXCPtr` are added to `mplsXCExtTable` of MPLS-LSR-EXT-STD-MIB. The RowPointer `mplsXCExtTunnelPointer` is read-only object used to indicate the back pointer to the tunnel entry. The RowPointer `mplsXCExtOppositeDirXCPtr` object points to the opposite direction XC entry.

If these RowPointer returns `zeroDotZero`, it implies that there is no entry associated with the RowPointer object.

7. MIB Modules Interdependencies

This section provides an overview of the relationship between the MPLS-TP TE MIB module and other MPLS MIB modules.

The arrows in the following diagram show a 'depends on' relationship. A relationship "MIB module A depends on MIB module B" means that MIB module A uses an object, object identifier, or textual convention defined in MIB module B, or that MIB module A contains a pointer (index or RowPointer) to an object in MIB module B.

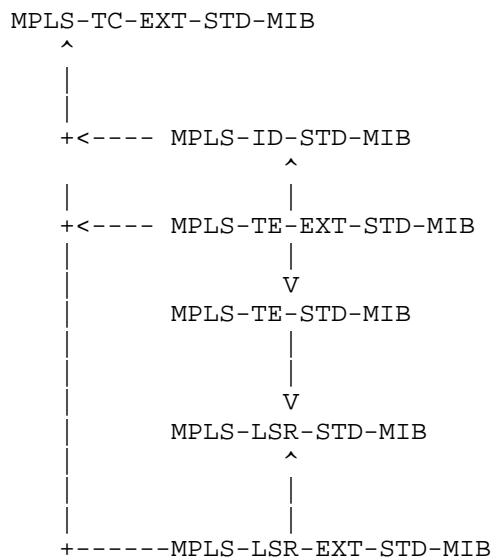


Figure 2: MIB modules interdependencies

Thus:

- All the new MPLS extension MIB modules depend on MPLS-TC-EXT-STD-MIB.
- MPLS-ID-STD-MIB contains references to objects in MPLS-TE-STD-MIB [RFC3812].
- MPLS-TE-EXT-STD-MIB contains references to objects in MPLS-TE-STD-MIB [RFC3812].
- MPLS-LSR-EXT-STD-MIB contains references to objects in MPLS-LSR-STD-MIB [RFC3813].

The `mplsTunnelExtTable` sparsely extends the `mplsTunnelTable` of MPLS-TE-STD-MIB [RFC3812]. This helps in associating the reverse direction tunnel information.

The `mplsXCExtTable` sparsely extends the `mplsXCTable` of MPLS-LSR-STD-MIB [RFC3813]. This helps in pointing back to the tunnel entry for easy tunnel access from XC entry.

Note that all of the MIB modules shown above in the figure also have a dependency on MPLS-TC-STD-MIB.

8. Dependencies between MIB Module Tables

The tables in MPLS-TE-EXT-STD-MIB are related as shown on the diagram below. The arrows indicate a reference from one table to another.

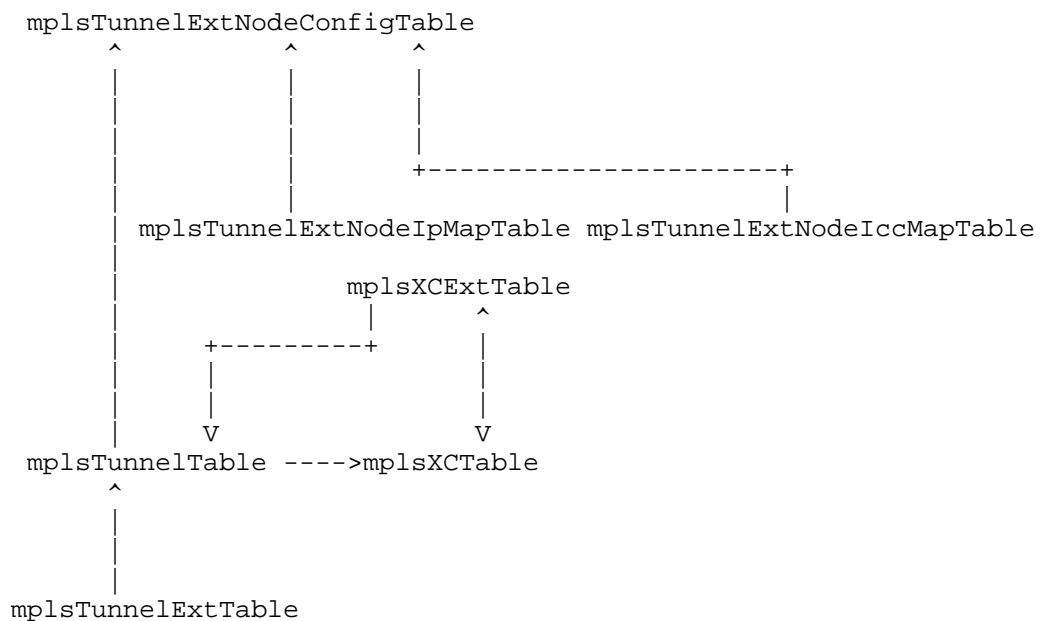


Figure 3: Dependencies between MIB module tables

An existing `mplsTunnelTable` uses the `mplsTunnelExtNodeConfigTable` table to map the `Global_ID::Node_ID` and/or `ICC_Operator_ID::Node_ID` with the local number in order to accommodate in the existing tunnel table's ingress/egress LSR-id.

New `mplsTunnelExtTable` table provides the reverse direction LSP information for the existing tunnel table in order to achieve bidirectional LSPs.

mplsXCExtTable sparsely extends the mplsLsrXCTable to provide backward reference to tunnel entry.

9. Example of MPLS-TP Tunnel Setup

In this section, we provide an example to configure MPLS-TP bidirectional tunnels with IP tunnel identifiers. This example provides the usage of MPLS-TP Tunnel MIB along with the extended MIB modules introduced in this document.

Do note that a MPLS-TP tunnel could be setup statically as well as signaled via control plane. This example considers accessing MIB objects on a head-end for a static and signaled MPLS-TP tunnels. This section shows the configuration of the forward and reverse direction MPLS-TP LSPs that run between East and West and vice-versa. Only objects relevant to MPLS-TP tunnels are illustrated here.

In mplsTunnelExtNodeConfigTable:

```
{
-- Non-IP Ingress LSR-Id (Index to the table)

    mplsTunnelExtNodeConfigLocalId          = 1,

    mplsTunnelExtNodeConfigGlobalId          = 1234,
    mplsTunnelExtNodeConfigNodeId            = 10,
-- Mandatory parameters needed to activate the row go here
    mplsTunnelExtNodeConfigRowStatus         = createAndGo (4)

-- Non-IP Egress LSR-Id (Index to the table)
    mplsTunnelExtNodeConfigLocalId          = 2,
    mplsTunnelExtNodeConfigGlobalId          = 1234,
    mplsTunnelExtNodeConfigNodeId            = 20,
-- Mandatory parameters needed to activate the row go here
    mplsTunnelExtNodeConfigRowStatus         = createAndGo (4)
}
```

This will create an entry in the mplsTunnelExtNodeConfigTable for a Global_ID::Node_ID. A separate entry is made for both Ingress LSR and Egress LSR.

The following read-only mplsTunnelExtNodeIpMapTable table is populated automatically upon creating an entry in mplsTunnelExtNodeConfigTable and this table is used to retrieve the local identifier for the given Global_ID::Node_ID.

In mplsTunnelExtNodeIpMapTable:


```

{
-- Global_ID (Index to the table)
  mplstTunnelExtNodeIpMapGlobalId          = 1234,
-- Node Identifier (Index to the table)
  mplstTunnelExtNodeIpMapNodeId            = 10,
  mplstTunnelExtNodeIpMapLocalId           = 1

-- Global_ID (Index to the table)
  mplstTunnelExtNodeIpMapGlobalId          = 1234,
-- Node Identifier (Index to the table)
  mplstTunnelExtNodeIpMapNodeId            = 20,
  mplstTunnelExtNodeIpMapLocalId           = 2
}

```

9.1. Example of MPLS-TP static co-routed bidirectional tunnel setup

The following denotes the co-routed bidirectional tunnel "head" entry.

9.1.1. mplstTunnelEntry

In mplstTunnelTable:

```

{
  mplstTunnelIndex          = 1,
  mplstTunnelInstance       = 1,
-- Local map number created in mplstTunnelExtNodeConfigTable for
-- Ingress LSR-Id
  mplstTunnelIngressLSRId   = 1,

-- Local map number created in mplstTunnelExtNodeConfigTable for
-- Egress LSR-Id
  mplstTunnelEgressLSRId    = 2,
  mplstTunnelName           = "TP co-routed bidirectional LSP",
  mplstTunnelDescr          = "East to West",
  mplstTunnelIsIf           = true (1),
-- RowPointer MUST point to the first accessible column
  mplstTunnelXCPointer      =
                                mplstXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
  mplstTunnelSignallingProto = none (1),
  mplstTunnelSetupPrio       = 0,
  mplstTunnelHoldingPrio     = 0,
  mplstTunnelSessionAttributes = 0,
  mplstTunnelLocalProtectInUse = false (0),
-- RowPointer MUST point to the first accessible column
  mplstTunnelResourcePointer = mplstTunnelResourceMaxRate.5,
  mplstTunnelInstancePriority = 1,
  mplstTunnelHopTableIndex   = 1,
}

```



```

mplsTunnelIncludeAnyAffinity = 0,
mplsTunnelIncludeAllAffinity = 0,
mplsTunnelExcludeAnyAffinity = 0,
mplsTunnelRole                = head (1),
-- Mandatory parameters needed to activate the row go here
mplsTunnelRowStatus            = createAndGo (4)
}

```

9.1.2. mplsTunnelExtEntry

```

-- An MPLS extension table
In mplsTunnelExtTable:
{
  -- This opposite direction tunnel pointer may point to 0.0
  -- if co-routed bidirectional tunnel is managed by single tunnel
  -- entry
  mplsTunnelExtOppositeDirTnlPtr      = 0.0
  -- Set both the Ingress and Egress LocalId objects to TRUE as
  -- this tunnel entry uses the local identifiers.
  mplsTunnelExtIngressLSRLocalIdValid = true,
  mplsTunnelExtEgressLSRLocalIdValid = true
}

```

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the mplsInSegmentTable and mplsOutSegmentTable.

9.1.3. Forward direction mplsOutSegmentEntry

For the forward direction,

```

In mplsOutSegmentTable:
{
  mplsOutSegmentIndex          = 0x00000001,
  mplsOutSegmentInterface      = 13, -- outgoing interface
  mplsOutSegmentPushTopLabel   = true(1),
  mplsOutSegmentTopLabel       = 22, -- outgoing label

  -- RowPointer MUST point to the first accessible column.
  mplsOutSegmentTrafficParamPtr = 0.0,
  mplsOutSegmentRowStatus       = createAndGo (4)
}

```

9.1.4. Reverse direction mplsInSegmentEntry

For the reverse direction,


```

In mplsInSegmentTable:
{
    mplsInSegmentIndex      = 0x00000001
    mplsInSegmentLabel      = 21, -- incoming label
    mplsInSegmentNPop       = 1,
    mplsInSegmentInterface  = 13, -- incoming interface

    -- RowPointer MUST point to the first accessible column.
    mplsInSegmentTrafficParamPtr = 0.0,
    mplsInSegmentRowStatus    = createAndGo (4)
}

```

Next, two cross-connect entries are created in the mplsXCTable of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.1.1.5. Forward direction mplsXCEntry

```

In mplsXCTable:
{
    mplsXCIndex              = 0x01,
    mplsXCInSegmentIndex     = 0x00000000,
    mplsXCOutSegmentIndex    = 0x00000001,
    mplsXCLspId              = 0x0102 -- unique ID

    -- only a single outgoing label
    mplsXCLabelStackIndex    = 0x00,
    mplsXCRowStatus          = createAndGo(4)
}

```

9.1.1.6. Reverse direction mplsXCEntry

```

In mplsXCTable:
{
    mplsXCIndex              = 0x01,
    mplsXCInSegmentIndex     = 0x00000001,
    mplsXCOutSegmentIndex    = 0x00000000,
    mplsXCLspId              = 0x0102 -- unique ID
    -- only a single outgoing label
    mplsXCLabelStackIndex    = 0x00,
    mplsXCRowStatus          = createAndGo(4)
}

```

This table entry is extended by entry in the mplsXCExtTable. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsXCExtTable has the same index values as the entry in

the mplsXCTable.

9.1.7. Forward direction mplsXCExtEntry

```
In mplsXCExtTable (0x01, 0x00000000, 0x00000001)
{
    -- Back pointer from XC table to Tunnel table
    mplsXCExtTunnelPointer      = mplsTunnelName.1.1.1.2
    mplsXCExtOppositeDirXCPtr   =
                                mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0
}
```

9.1.8. Reverse direction mplsXCExtEntry

Next for the reverse direction:

```
In mplsXCExtTable (0x01, 0x00000001, 0x00000000)
{
    -- Back pointer from XC table to Tunnel table
    mplsXCExtTunnelPointer      = mplsTunnelName.1.1.1.2
    mplsXCExtOppositeDirXCPtr   =
                                mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1
}
```

9.2. Example of MPLS-TP static associated bidirectional tunnel setup

The MPLS-TP associated bidirectional tunnel is implemented by two different unidirectional tunnels [Forward and Reverse LSPs] and these are associated together using mplsTunnelExtTable. Two different tunnel entries to provide the forward and reverse directions MAY be used for co-routed bidirectional tunnels as well.

The following denotes the associated bidirectional forward tunnel "head" entry:

9.2.1. Forward direction mplsTunnelEntry

```
In mplsTunnelTable:

{
    mplsTunnelIndex              = 1,
    mplsTunnelInstance           = 1,
    -- Local map number created in mplsTunnelExtNodeConfigTable for
    -- Ingress LSR-Id
    mplsTunnelIngressLSRId       = 1,

    -- Local map number created in mplsTunnelExtNodeConfigTable for
    -- Egress LSR-Id
```



```

mplsTunnelEgressLSRId      = 2,
mplsTunnelName             = "TP associated bidirectional
                           forward LSP",
mplsTunnelDescr            = "East to West",
mplsTunnelIsIf             = true (1),
-- RowPointer MUST point to the first accessible column
mplsTunnelXCPointer        =
                           mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
mplsTunnelSignallingProto  = none (1),
mplsTunnelSetupPrio        = 0,
mplsTunnelHoldingPrio      = 0,
mplsTunnelSessionAttributes = 0,
mplsTunnelLocalProtectInUse = false (0),
-- RowPointer MUST point to the first accessible column
mplsTunnelResourcePointer  = mplsTunnelResourceMaxRate.5,
mplsTunnelInstancePriority = 1,
mplsTunnelHopTableIndex    = 1,
mplsTunnelIncludeAnyAffinity = 0,

mplsTunnelIncludeAllAffinity = 0,
mplsTunnelExcludeAnyAffinity = 0,
mplsTunnelRole               = head (1),
-- Mandatory parameters needed to activate the row go here
mplsTunnelRowStatus          = createAndGo (4)
}

```

9.2.2. Forward direction mplsTunnelExtEntry

```

For Associated bidirectional forward LSP,
In mplsTunnelExtTable:
{
  mplsTunnelExtOppositeDirPtr      = mplsTunnelName.2.1.2.1
  -- Set both the Ingress and Egress LocalId objects to TRUE as
  -- this tunnel entry uses the local identifiers.
  mplsTunnelExtIngressLSRLocalIdValid = true,
  mplsTunnelExtEgressLSRLocalIdValid = true
}

```

9.2.3. Forward direction mplsOutSegmentTable

For the forward direction.

```

In mplsOutSegmentTable:
{
  mplsOutSegmentIndex      = 0x0000001,
  mplsOutSegmentInterface  = 13, -- outgoing interface
  mplsOutSegmentPushTopLabel = true(1),
  mplsOutSegmentTopLabel   = 22, -- outgoing label
}

```



```

    -- RowPointer MUST point to the first accessible column.
    mplsOutSegmentTrafficParamPtr = 0.0,
    mplsOutSegmentRowStatus      = createAndGo (4)
}

```

9.2.4. Forward direction mplsXCEntry

```

In mplsXCTable:
{
    mplsXCIndex          = 0x01,
    mplsXCInSegmentIndex = 0x00000000,
    mplsXCOutSegmentIndex = 0x00000001,
    mplsXCLspId          = 0x0102 -- unique ID
    -- only a single outgoing label
    mplsXCLabelStackIndex = 0x00,
    mplsXCRowStatus       = createAndGo(4)
}

```

9.2.5. Forward direction mplsXCExtEntry

```

In mplsXCExtTable (0x01, 0x00000000, 0x00000001)
{
    -- Back pointer from XC table to Tunnel table
    mplsXCExtTunnelPointer = mplsTunnelName.1.1.1.2
    mplsXCExtOppositeDirXCPtr =
        mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0
}

```

9.2.6. Reverse direction mplsTunnelEntry

The following denotes the configured associated bidirectional reverse tunnel "tail" entry:

```

In mplsTunnelTable:
{
    mplsTunnelIndex          = 2,
    mplsTunnelInstance       = 1,
    -- Local map number created in mplsTunnelExtNodeConfigTable for
    -- Ingress LSR-Id
    mplsTunnelIngressLSRId   = 2,
    -- Local map number created in mplsTunnelExtNodeConfigTable for
    -- Egress LSR-Id
    mplsTunnelEgressLSRId    = 1,
    mplsTunnelName           = "TP associated bidirectional
                             reverse LSP",
    mplsTunnelDescr          = "West to East",
}

```



```

mplsTunnelIsIf          = true (1),
-- RowPointer MUST point to the first accessible column
mplsTunnelXCPointer     =
                        mplsXCLspId.4.0.0.0.1.4.0.0.0.1.1.0,
mplsTunnelSignallingProto = none (1),
mplsTunnelSetupPrio     = 0,
mplsTunnelHoldingPrio   = 0,
mplsTunnelSessionAttributes = 0,
mplsTunnelLocalProtectInUse = false (0),

-- RowPointer MUST point to the first accessible column
mplsTunnelResourcePointer = mplsTunnelResourceMaxRate.5,
mplsTunnelInstancePriority = 1,
mplsTunnelHopTableIndex   = 1,
mplsTunnelIncludeAnyAffinity = 0,
mplsTunnelIncludeAllAffinity = 0,
mplsTunnelExcludeAnyAffinity = 0,
mplsTunnelRole            = head (1),
-- Mandatory parameters needed to activate the row go here

mplsTunnelRowStatus      = createAndGo (4)
}

```

9.2.7. Reverse direction mplsTunnelExtEntry

For Associated bidirectional reverse LSP,
In mplsTunnelExtTable:

```

{
    mplsTunnelExtOppositeDirPtr      = mplsTunnelName.1.1.1.2
    -- Set both the Ingress and Egress LocalId objects to TRUE as
    -- this tunnel entry uses the local identifiers.
    mplsTunnelExtIngressLSRLocalIdValid = true,
    mplsTunnelExtEgressLSRLocalIdValid = true
}

```

9.2.8. Reverse direction mplsInSegmentEntry

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the mplsInSegmentTable and mplsOutSegmentTable.

In mplsInSegmentTable:

```

{
    mplsInSegmentIndex      = 0x00000001
    mplsInSegmentLabel      = 21, -- incoming label
    mplsInSegmentNPop       = 1,
    mplsInSegmentInterface  = 13, -- incoming interface
}

```



```

    -- RowPointer MUST point to the first accessible column.
    mplsInSegmentTrafficParamPtr    = 0.0,
    mplsInSegmentRowStatus          = createAndGo (4)
}

```

Next, two cross-connect entries are created in the mplsXCTable of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.2.9. Reverse direction mplsXCEntry

```

In mplsXCTable:
{
    mplsXCIndex                = 0x01,
    mplsXCInSegmentIndex       = 0x00000001,
    mplsXCOutSegmentIndex      = 0x00000000,
    mplsXCLspId                = 0x0102 -- unique ID
    -- only a single outgoing label
    mplsXCLabelStackIndex      = 0x00,

    mplsXCRowStatus            = createAndGo(4)
}

```

This table entry is extended by entry in the mplsXCExtTable. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the mplsXCExtTable has the same index values as the entry in the mplsXCTable.

9.2.10. Reverse direction mplsXCExtEntry

Next for the reverse direction:

```

In mplsXCExtTable (0x01, 0x00000001, 0x00000000)
{
    -- Back pointer from XC table to Tunnel table
    mplsXCExtTunnelPointer      = mplsTunnelName.2.1.2.1
    mplsXCExtOppositeDirXCPtr   =
                                mplsXCLspId.4.0.0.0.1.1.0.4.0.0.0.1
}

```

9.3. Example of MPLS-TP signaled co-routed bidirectional tunnel setup

The following denotes the co-routed bidirectional tunnel "head" entry and in intermediate and tail-end nodes, the tunnel table and its associated tables are created by the local management subsystem (e.g. agent) when the MPLS TP tunnel is signaled successfully.

Refer [RFC3812] and [RFC4802] for signaled tunnel table configuration examples.

9.3.1. mplstunnelEntry

In mplstunnelTable:

```
{
  mplstunnelIndex          = 1,
  mplstunnelInstance       = 0,
  -- Local map number created in mplstunnelExtNodeConfigTable for
  -- Ingress LSR-Id, for the intermediate and tail-end nodes,
  -- the local management entity is expected to pick a first available
  -- local identifier which is not used in mplstunnelTable.
  mplstunnelIngressLSRId   = 1,

  -- Local map number created in mplstunnelExtNodeConfigTable for
  -- Egress LSR-Id
  mplstunnelEgressLSRId    = 2,
  mplstunnelName           = "TP co-routed bidirectional LSP",
  mplstunnelDescr          = "East to West",
  mplstunnelIsIf           = true (1),

  -- RowPointer MUST point to the first accessible column
  mplstunnelXCPointer      =
    mplstunnelXCLspId.4.0.0.0.1.1.0.4.0.0.0.1,
  mplstunnelSignallingProto = none (1),
  mplstunnelSetupPrio      = 0,
  mplstunnelHoldingPrio    = 0,
  mplstunnelSessionAttributes = 0,
  mplstunnelLocalProtectInUse = false (0),
  -- RowPointer MUST point to the first accessible column
  mplstunnelResourcePointer = mplstunnelResourceMaxRate.5,
  mplstunnelInstancePriority = 1,
  mplstunnelHopTableIndex   = 1,
  mplstunnelIncludeAnyAffinity = 0,
  mplstunnelIncludeAllAffinity = 0,
  mplstunnelExcludeAnyAffinity = 0,
  mplstunnelRole            = head (1),
  -- Mandatory parameters needed to activate the row go here
  mplstunnelRowStatus       = createAndGo (4)
}
```

9.3.2. mplstunnelExtEntry

```
-- An MPLS extension table
In mplstunnelExtTable:
{
```



```
-- This opposite direction tunnel pointer may point to 0.0
-- if co-routed bidirectional tunnel is managed by single tunnel
-- entry
mplsTunnelExtOppositeDirTnlPtr      = 0.0
-- Set both the Ingress and Egress LocalId objects to TRUE as
-- this tunnel entry uses the local identifiers.
mplsTunnelExtIngressLSRLocalIdValid = true,
mplsTunnelExtEgressLSRLocalIdValid = true
}
```

We must next create the appropriate in-segment and out-segment entries. These are done in [RFC3813] using the `mplsInSegmentTable` and `mplsOutSegmentTable`.

9.3.3. Forward direction `mplsOutSegmentEntry`

The forward direction `mplsOutSegmentTable` will be populated automatically based on the information received from the signaling protocol.

9.3.4. Reverse direction `mplsInSegmentEntry`

The reverse direction `mplsOutSegmentTable` will be populated automatically based on the information received from the signaling protocol.

Next, two cross-connect entries are created in the `mplsXCTable` of the MPLS-LSR-STD-MIB [RFC3813], thereby associating the newly created segments together.

9.3.5. Forward direction `mplsXCEntry`

The forward direction `mplsXCEntry` will be populated as soon as the forward path label information is available.

9.3.6. Reverse direction `mplsXCEntry`

The reverse direction `mplsXCEntry` will be populated as soon as the reverse path label information is available.

This table entry is extended by entry in the `mplsXCExtTable`. Note that the nature of the 'extends' relationship is a sparse augmentation so that the entry in the `mplsXCExtTable` has the same index values as the entry in the `mplsXCTable`.

9.3.7. Forward direction `mplsXCExtEntry`

Once the forward path information is negotiated using signaling

protocol, the forward direction mplsXCExtEntry will be created for associating the opposite direction XC entry and tunnel table entry.

9.3.8. Reverse direction mplsXCExtEntry

Once the reverse path information is negotiated using signaling protocol, the reverse direction mplsXCExtEntry will be created for associating the opposite direction XC entry and tunnel table entry.

10. MPLS Textual Convention Extension MIB definitions

```
MPLS-TC-EXT-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, Unsigned32
        FROM SNMPv2-SMI                -- [RFC2578]

    TEXTUAL-CONVENTION
        FROM SNMPv2-TC                -- [RFC2579]

    mplsStdMIB
        FROM MPLS-TC-STD-MIB          -- [RFC3811]

    ;

mplsTcExtStdMIB MODULE-IDENTITY

    LAST-UPDATED
        "201412180000Z" -- December 18, 2014
    ORGANIZATION
        "Multiprotocol Label Switching (MPLS) Working Group"
    CONTACT-INFO
        "
            Venkatesan Mahalingam
            Dell Inc,
            5450 Great America Parkway,
            Santa Clara, CA 95054, USA
            Email: venkat.mahalingams@gmail.com

            Kannan KV Sampath
            Redeem,
            India
            Email: kannankvs@gmail.com

            Sam Aldrin
            Huawei Technologies
            2330 Central Express Way,
            Santa Clara, CA 95051, USA
```


Email: aldrin.ietf@gmail.com

Thomas D. Nadeau

Email: tnadeau@lucidvision.com

"

DESCRIPTION

"Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This MIB module contains Textual Conventions for LSPs of MPLS based transport networks."

-- Revision history.

REVISION

"201412180000Z" -- December 18, 2014

DESCRIPTION

"MPLS Textual Convention Extensions"

::= { mplsStdMIB www } -- www to be replaced with correct value

MplsGlobalId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This object contains the Textual Convention for IP based operator unique identifier (Global_ID), the Global_ID can contain the 2-octet or 4-octet value of the operator's Autonomous System Number (ASN).

When the Global_ID is derived from a 2-octet AS number, the two high-order octets of this 4-octet identifier MUST be set to zero(0x00). Further ASN 0 is reserved. The size of the Global_ID string MUST be zero if the Global_ID is invalid.

Note that a Global_ID of zero is limited to entities contained within a single operator and MUST NOT be used across an Network-to-Network Interface (NNI). A non-zero Global_ID MUST be derived from an ASN owned by the operator."

REFERENCE

"MPLS Transport Profile (MPLS-TP) Identifiers, [RFC6370] Section 3"

SYNTAX OCTET STRING (SIZE (4))

MplsCcId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The CC (Country Code) is a string of two characters, each being an uppercase Basic Latin alphabetic (i.e., A-Z). The characters are encoded using ITU-T Recommendation T.50. The size of the CC string MUST be zero if the CC identifier is invalid."

REFERENCE

"MPLS-TP Identifiers Following ITU-T Conventions, RFC 6923, Section 3. International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information exchange, ITU-T Recommendation T.50, September 1992. "

SYNTAX OCTET STRING (SIZE (0|2))

MplsIccId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The ICC is a string of one to six characters, each an upper case Basic Latin alphabetic (i.e., A-Z) or numeric (i.e., 0-9). The characters are encoded using ITU-T Recommendation T.50. The size of the ICC string MUST be zero if the ICC identifier is invalid."

REFERENCE

"MPLS-TP Identifiers Following ITU-T Conventions, RFC6923, Section 3. International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information exchange, ITU-T Recommendation T.50, September 1992. "

SYNTAX OCTET STRING (SIZE (0|1..6))

MplsNodeId ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"The Node_ID is assigned within the scope of the Global_ID/ICC_Operator_ID.

When IPv4 addresses are in use, the value of this object can be derived from the LSR's IPv4 loop back address. When IPv6 addresses are in use, the value of this object can be a 32-bit value unique within the scope of a Global_ID.

Note that, when IP reachability is not needed, the 32-bit Node_ID is not required to have any association

with the IPv4 address space. The value of 0 indicates the invalid Node identifier."

REFERENCE

"MPLS Transport Profile (MPLS-TP) Identifiers, [RFC6370] Section 4"

SYNTAX Unsigned32 (0|1..4294967295)

-- MPLS-TC-EXT-STD-MIB module ends
END

11. MPLS Identifier MIB definitions

MPLS-ID-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE
FROM SNMPv2-SMI -- [RFC2578]
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF -- [RFC2580]
mplsStdMIB
FROM MPLS-TC-STD-MIB -- [RFC3811]
MplsGlobalId, MplsCcId, MplsIccId, MplsNodeId
FROM MPLS-TC-EXT-STD-MIB
;

mplsIdStdMIB MODULE-IDENTITY

LAST-UPDATED

"201412120000Z" -- December 12, 2014

ORGANIZATION

"Multiprotocol Label Switching (MPLS) Working Group"

CONTACT-INFO

"

Venkatesan Mahalingam
Dell Inc,
5450 Great America Parkway,
Santa Clara, CA 95054, USA
Email: venkat.mahalingams@gmail.com

Kannan KV Sampath

Redeem,
India

Email: kannankvs@gmail.com

Sam Aldrin
Huawei Technologies

2330 Central Express Way,
Santa Clara, CA 95051, USA
Email: aldrin.ietf@gmail.com

Thomas D. Nadeau
Email: tnadeau@lucidvision.com

"

DESCRIPTION

"Copyright (c) 2014 IETF Trust and the persons identified
as the document authors. All rights reserved.

This MIB module contains generic object definitions for
MPLS Traffic Engineering in transport networks."

-- Revision history.

REVISION

"201412120000Z" -- December 12, 2014

DESCRIPTION

"This MIB modules defines the MIB objects for MPLS-TP
identifiers"

::= { mplsStdMIB xxx } -- xxx to be replaced with correct value

-- notifications

mplsIdNotifications OBJECT IDENTIFIER ::= { mplsIdStdMIB 0 }

-- tables, scalars

mplsIdObjects OBJECT IDENTIFIER ::= { mplsIdStdMIB 1 }

-- conformance

mplsIdConformance OBJECT IDENTIFIER ::= { mplsIdStdMIB 2 }

-- MPLS common objects

mplsIdGlobalId OBJECT-TYPE

SYNTAX MplsGlobalId

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object allows the operator or service provider to
assign a unique operator identifier also called MPLS-TP
Global_ID.

If this value is used in mplsTunnelExtNodeConfigGlobalId
for mapping Global_ID::Node_ID with the local identifier
then this object value MUST NOT be changed."

::= { mplsIdObjects 1 }

mplsIdNodeId OBJECT-TYPE

SYNTAX MplsNodeId
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "This object allows the operator or service provider to assign a unique MPLS-TP Node_ID. The Node_ID is assigned within the scope of the Global_ID/ICC_Operator_ID. If this value is used in mplsTunnelExtNodeConfigNodeId for mapping Global_ID::Node_ID with the local identifier then this object value SHOULD NOT be changed. If this value is used in mplsTunnelExtNodeConfigNodeId for mapping ICC_Operator_ID::Node_ID with the local identifier then this object value MUST NOT be changed."
 ::= { mplsIdObjects 2 }

mplsIdCc OBJECT-TYPE

SYNTAX MplsCcId
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "This object allows the operator or service provider to assign a Country Code (CC) to the node. Global uniqueness of ICC is assured by concatenating the ICC with a Country Code (CC). If this value is used in mplsTunnelExtNodeConfigCcId for mapping ICC_Operator_ID::Node_ID with the local identifier then this object value MUST NOT be changed."
REFERENCE
 "MPLS-TP Identifiers Following ITU-T Conventions, [RFC6923] Section 3"
 ::= { mplsIdObjects 3 }

mplsIdIcc OBJECT-TYPE

SYNTAX MplsIccId
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "This object allows the operator or service provider to assign a unique MPLS-TP ITU-T Carrier Code (ICC) to the node. Together, the CC and the ICC form the ICC_Operator_ID as CC::ICC. If this value is used in mplsTunnelExtNodeConfigIccId for mapping ICC_Operator_ID::Node_ID with the local identifier then this object value MUST NOT be changed."
REFERENCE
 "MPLS-TP Identifiers Following ITU-T Conventions, [RFC6923] Section 3"


```
 ::= { mplsIdObjects 4 }

-- Module compliance.

mplsIdCompliances
  OBJECT IDENTIFIER ::= { mplsIdConformance 1 }

mplsIdGroups
  OBJECT IDENTIFIER ::= { mplsIdConformance 2 }

-- Compliance requirement for fully compliant implementations.

mplsIdModuleFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that provide full
    support of the MPLS-ID-STD-MIB module."

  MODULE -- this module

    -- The mandatory group has to be implemented by all LSRs that
    -- originate, terminate, or act as transit for MPLS-TP tunnels.

    GROUP mplsIdIpOperatorGroup
    DESCRIPTION
      "This group is mandatory for devices which support
      IP based identifier configuration."

    GROUP mplsIdIccOperatorGroup
    DESCRIPTION
      "This group is mandatory for devices which support
      ICC based identifier configuration."

    ::= { mplsIdCompliances 1 }

-- Compliance requirement for read-only implementations.

mplsIdModuleReadOnlyCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that only provide
    read-only support for the MPLS-ID-STD-MIB module."

  MODULE -- this module

    GROUP mplsIdIpOperatorGroup
```


DESCRIPTION

"This group is mandatory for devices which support
IP based identifier configuration."

GROUP mplsIdIccOperatorGroup

DESCRIPTION

"This group is mandatory for devices which support
ICC based identifier configuration."

OBJECT mplsIdGlobalId

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsIdNodeId

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsIdCc

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT mplsIdIcc

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

::= { mplsIdCompliances 2 }

-- Units of conformance.

mplsIdIpOperatorGroup OBJECT-GROUP

OBJECTS { mplsIdGlobalId,
mplsIdNodeId
}

STATUS current

DESCRIPTION

"The objects in this group are optional for ICC based
node."

::= { mplsIdGroups 1 }

mplsIdIccOperatorGroup OBJECT-GROUP

OBJECTS { mplsIdNodeId,
mplsIdCc,
mplsIdIcc
}


```

    }
    STATUS current
    DESCRIPTION
        "The objects in this group are optional for IP based
        node."
    ::= { mplsIdGroups 2 }

```

```

-- MPLS-ID-STD-MIB module ends
END

```

12. MPLS LSR Extension MIB definitions

```
MPLS-LSR-EXT-STD-MIB DEFINITIONS ::= BEGIN
```

IMPORTS

```

    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI -- [RFC2578]
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF -- [RFC2580]
    mplsStdMIB
        FROM MPLS-TC-STD-MIB -- [RFC3811]
    RowPointer
        FROM SNMPv2-TC -- [RFC2579]
    mplsXCIndex, mplsXCInSegmentIndex, mplsXCOutSegmentIndex,
    mplsInterfaceGroup, mplsInSegmentGroup, mplsOutSegmentGroup,
    mplsXCGroup, mplsLsrNotificationGroup
        FROM MPLS-LSR-STD-MIB; -- [RFC3813]

```

```
mplsLsrExtStdMIB MODULE-IDENTITY
```

```

    LAST-UPDATED
        "201412120000Z" -- December 12, 2014
    ORGANIZATION
        "Multiprotocol Label Switching (MPLS) Working Group"
    CONTACT-INFO
        "

```

```

            Venkatesan Mahalingam
            Dell Inc,
            5450 Great America Parkway,
            Santa Clara, CA 95054, USA
            Email: venkat.mahalingams@gmail.com

```

```

            Kannan KV Sampath
            Redeem,
            India
            Email: kannankvs@gmail.com

```

```

            Sam Aldrin
            Huawei Technologies

```


2330 Central Express Way,
Santa Clara, CA 95051, USA

Email: aldrin.ietf@gmail.com

Thomas D. Nadeau

Email: tnadeau@lucidvision.com

"

DESCRIPTION

"Copyright (c) 2014 IETF Trust and the persons identified
as the document authors. All rights reserved.

This MIB module contains generic object definitions for
MPLS LSR in transport networks."

-- Revision history.

REVISION

"201412120000Z" -- December 12, 2014

DESCRIPTION

"MPLS LSR specific MIB objects extension"

::= { mplsStdMIB yyy } -- yyy to be replaced with correct value

-- notifications

mplsLsrExtNotifications OBJECT IDENTIFIER ::= { mplsLsrExtStdMIB 0 }

-- tables, scalars

mplsLsrExtObjects OBJECT IDENTIFIER
::= { mplsLsrExtStdMIB 1 }

-- conformance

mplsLsrExtConformance OBJECT IDENTIFIER
::= { mplsLsrExtStdMIB 2 }

-- MPLS LSR common objects

mplsXCExtTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsXCExtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table sparse augments the mplsXCTable of
MPLS-LSR-STD-MIB [RFC3813] to provide MPLS-TP specific
information about associated tunnel information"

REFERENCE

"1. Multiprotocol Label Switching (MPLS) Label Switching
Router (LSR) Management Information Base (MIB), RFC 3813."

::= { mplsLsrExtObjects 1 }


```

mplsXCExtEntry OBJECT-TYPE
    SYNTAX      MplsXCExtEntry
    MAX-ACCESS   not-accessible

    STATUS      current
    DESCRIPTION
        "An entry in this table sparsely extends the cross connect
        information represented by an entry in
        the mplsXCTable in MPLS-LSR-STD-MIB [RFC3813] through
        a sparse augmentation. An entry can be created by
        a network operator via SNMP SET commands, or in
        response to signaling protocol events."
    REFERENCE
        "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."

    INDEX { mplsXCIndex, mplsXCInSegmentIndex,
            mplsXCOutSegmentIndex }
    ::= { mplsXCExtTable 1 }

MplsXCExtEntry ::= SEQUENCE {
    mplsXCExtTunnelPointer      RowPointer,
    mplsXCExtOppositeDirXCPtr   RowPointer
}

mplsXCExtTunnelPointer OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This read-only object indicates the back pointer to
        the tunnel entry segment.
        The only valid value for Tunnel Pointer is
        mplsTunnelTable entry."
    REFERENCE
        "1. Multiprotocol Label Switching (MPLS) Label Switching
        Router (LSR) Management Information Base (MIB), RFC 3813."
    ::= { mplsXCExtEntry 1 }

mplsXCExtOppositeDirXCPtr OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the pointer to the opposite
        direction XC entry. This object cannot be modified if

```



```
mplsXCRowStatus for the corresponding entry in the
mplsXCTable is active(1). If this pointer is not set or
removed, mplsXCOperStatus should be set to down(2)."
```

REFERENCE

```
"1. Multiprotocol Label Switching (MPLS) Label Switching
Router (LSR) Management Information Base (MIB), RFC 3813."
::= { mplsXCExtEntry 2 }
```

```
mplsLsrExtCompliances
  OBJECT IDENTIFIER ::= { mplsLsrExtConformance 1 }
```

```
mplsLsrExtGroups
  OBJECT IDENTIFIER ::= { mplsLsrExtConformance 2 }
```

```
-- Compliance requirement for fully compliant implementations.
```

```
mplsLsrExtModuleFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that provide full support
    for MPLS-LSR-EXT-STD-MIB.
    The mandatory group has to be implemented by all LSRs
    that originate, terminate, or act as transit for
    TE-LSPs/tunnels.
    In addition, depending on the type of tunnels supported,
    other groups become mandatory as explained below."
```

```
MODULE MPLS-LSR-STD-MIB -- The MPLS-LSR-STD-MIB, RFC3813
```

```
MANDATORY-GROUPS {
  mplsInSegmentGroup,
  mplsOutSegmentGroup,
  mplsXCGroup,
  mplsLsrNotificationGroup
}
```

```
MODULE -- this module
```

```
MANDATORY-GROUPS {
  mplsXCExtGroup
}
```

```
::= { mplsLsrExtCompliances 1 }
```

```
-- Compliance requirement for implementations that provide
```



```
-- read-only access.

mplsLsrExtModuleReadOnlyCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Compliance requirement for implementations that only
        provide read-only support for MPLS-LSR-EXT-STD-MIB.
        Such devices can then be monitored but cannot be
        configured using this MIB module."

MODULE MPLS-LSR-STD-MIB

MANDATORY-GROUPS {
    mplsInterfaceGroup,
    mplsInSegmentGroup,
    mplsOutSegmentGroup
}

MODULE -- this module

GROUP mplsXCExtReadOnlyObjectsGroup
DESCRIPTION
    "This group is mandatory for devices which support
    opposite direction XC configuration of tunnels."

-- mplsXCExtTable
OBJECT mplsXCExtOppositeDirXCPtr
MIN-ACCESS read-only
DESCRIPTION
    "Write access is not required.
    This object indicates the pointer to the opposite
    direction XC entry. The only valid value for XC
    Pointer is mplsXCTable entry."
::= { mplsLsrExtCompliances 2 }

-- Units of conformance.

mplsXCExtGroup OBJECT-GROUP
OBJECTS {
    mplsXCExtTunnelPointer,
    mplsXCExtOppositeDirXCPtr
}
STATUS current
DESCRIPTION
    "This object should be supported in order to access
    the tunnel entry from XC entry."
::= { mplsLsrExtGroups 1 }
```



```

mplsXCExtReadOnlyObjectsGroup OBJECT-GROUP
OBJECTS {
    mplsXCExtTunnelPointer,
    mplsXCExtOppositeDirXCPtr
}
STATUS current
DESCRIPTION
    "This Object is needed to associate the opposite direction
    (forward/reverse) XC entry."
 ::= { mplsLsrExtGroups 2 }

-- MPLS-LSR-EXT-STD-MIB module ends
END

```

13. MPLS Tunnel Extension MIB definitions

```

MPLS-TE-EXT-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI -- [RFC2578]
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF -- [RFC2580]
    TruthValue, RowStatus, RowPointer, StorageType
        FROM SNMPv2-TC -- [RFC2579]
    IndexIntegerNextFree
        FROM DIFFSERV-MIB -- [RFC3289]
    MplsGlobalId, MplsNodeId, MplsCcId, MplsIccId
        FROM MPLS-TC-EXT-STD-MIB
    mplsStdMIB, MplsTunnelIndex, MplsTunnelInstanceIndex,
    MplsExtendedTunnelId
        FROM MPLS-TC-STD-MIB -- [RFC3811]
    mplsTunnelIndex, mplsTunnelInstance, mplsTunnelIngressLSRId,
    mplsTunnelEgressLSRId
        FROM MPLS-TE-STD-MIB -- [RFC3812]
;

mplsTeExtStdMIB MODULE-IDENTITY
    LAST-UPDATED
        "201412120000Z" -- December 12, 2014
    ORGANIZATION
        "Multiprotocol Label Switching (MPLS) Working Group"
    CONTACT-INFO
        "
            Venkatesan Mahalingam
            Dell Inc,

```


5450 Great America Parkway,
Santa Clara, CA 95054, USA
Email: venkat.mahalingams@gmail.com

Kannan KV Sampath
Redeem,
India
Email: kannankvs@gmail.com

Sam Aldrin
Huawei Technologies
2330 Central Express Way,
Santa Clara, CA 95051, USA
Email: aldrin.ietf@gmail.com

Thomas D. Nadeau
Email: tnadeau@lucidvision.com
"

DESCRIPTION

"Copyright (c) 2014 IETF Trust and the persons identified
as the document authors. All rights reserved.

This MIB module contains generic object definitions for
MPLS Traffic Engineering in transport networks."

-- Revision history.

REVISION

"201412120000Z" -- December 12, 2014

DESCRIPTION

"MPLS TE MIB objects extension"

::= { mplsStdMIB zzz } -- zzz to be replaced
-- with correct value

-- Top level components of this MIB module.

-- tables, scalars

mplsTeExtObjects OBJECT IDENTIFIER
::= { mplsTeExtStdMIB 0 }

-- conformance

mplsTeExtConformance OBJECT IDENTIFIER
::= { mplsTeExtStdMIB 1 }

-- Start of MPLS Transport Profile Node configuration table

mplsTunnelExtNodeConfigLocalIdNext OBJECT-TYPE

SYNTAX IndexIntegerNextFree (0..16777215)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an unused value for
mplsTunnelExtNodeConfigLocalId, or a zero to indicate
that none exist. Negative values are not allowed,
as they do not correspond to valid values of
mplsTunnelExtNodeConfigLocalId."

::= { mplsTeExtObjects 1 }

mplsTunnelExtNodeConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsTunnelExtNodeConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table allows the operator to map a node or
LSR Identifier (IP compatible [Global_ID::Node_ID] or
ICC based [ICC_Operator_ID::Node_ID]) with a local
identifier.

This table is created to reuse the existing
mplsTunnelTable for MPLS based transport network
tunnels also.

Since the MPLS tunnel's Ingress/Egress LSR identifiers'
size (Unsigned32) value is not compatible for
MPLS-TP tunnel i.e. Global_ID::Node_ID of size 8 bytes and
ICC_Operator_ID::Node_ID of size 12 bytes, there exists a
need to map the Global_ID::Node_ID or ICC_Operator_ID::Node_ID
with the local identifier of size 4 bytes (Unsigned32) value
in order to index (Ingress/Egress LSR identifier)
the existing mplsTunnelTable."

::= { mplsTeExtObjects 2 }

mplsTunnelExtNodeConfigEntry OBJECT-TYPE

SYNTAX MplsTunnelExtNodeConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents a mapping
identification for the operator or service provider
with node or LSR.

As per [RFC6370], IP compatible mapping is represented
as Global_ID::Node_ID.

As per [RFC6923], the CC and the ICC form the ICC_Operator_ID as CC::ICC and ICC compatible mapping is represented as ICC_Operator_ID::Node_ID.

Note: Each entry in this table should have a unique [Global_ID and Node_ID] or [CC::ICC and Node_ID] combination."
 INDEX { mplsTunnelExtNodeConfigLocalId }
 ::= { mplsTunnelExtNodeConfigTable 1 }

```
MplsTunnelExtNodeConfigEntry ::= SEQUENCE {
    mplsTunnelExtNodeConfigLocalId      MplsExtendedTunnelId,
    mplsTunnelExtNodeConfigGlobalId     MplsGlobalId,
    mplsTunnelExtNodeConfigCcId         MplsCcId,
    mplsTunnelExtNodeConfigIccId        MplsIccId,
    mplsTunnelExtNodeConfigNodeId       MplsNodeId,
    mplsTunnelExtNodeConfigIccValid     TruthValue,
    mplsTunnelExtNodeConfigStorageType  StorageType,
    mplsTunnelExtNodeConfigRowStatus    RowStatus
}
```

mplsTunnelExtNodeConfigLocalId OBJECT-TYPE

SYNTAX MplsExtendedTunnelId

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object is used in accommodating the bigger size Global_ID::Node_ID and/or the ICC_Operator_ID::Node_ID with lower size LSR identifier in order to index the mplsTunnelTable.

The Local Identifier is configured between 0 and 16777215, as valid IP address range starts from 16777216(01.00.00.00). This range is chosen to determine whether the mplsTunnelTable's Ingress/Egress LSR-id is an IP address or Local identifier. If the configured range is not an IP address, the operator is expected to retrieve the complete information (Global_ID::Node_ID or ICC_Operator_ID::Node_ID) from mplsTunnelExtNodeConfigTable. This way, existing mplsTunnelTable is reused for bidirectional tunnel extensions for MPLS based transport networks.

The Local Identifier allows the operator to assign a unique identifier to map Global_ID::Node_ID and/or ICC_Operator_ID::Node_ID. As this Local Identifier is unique within the node and the same syntax of this object can be

used for MPLS-TE tunnel also, it is up to the operator/local management entity to choose non-conflicting value for indexing the MPLS and MPLS-TP tunnel entries."

::= { mplsTunnelExtNodeConfigEntry 1 }

mplsTunnelExtNodeConfigGlobalId OBJECT-TYPE

SYNTAX MplsGlobalId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicates the Global Operator Identifier.

This object has no meaning when

mplsTunnelExtNodeConfigIccValid is set true."

REFERENCE

"MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]
Section 3."

::= { mplsTunnelExtNodeConfigEntry 2 }

mplsTunnelExtNodeConfigCcId OBJECT-TYPE

SYNTAX MplsCcId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object allows the operator or service provider to
configure a unique MPLS-TP ITU-T Country Code (CC)
either for Ingress ID or Egress ID.

This object has no meaning when

mplsTunnelExtNodeConfigIccValid is set false."

REFERENCE

"MPLS-TP Identifiers Following ITU-T Conventions,
[RFC6923] Section 3"

::= { mplsTunnelExtNodeConfigEntry 3 }

mplsTunnelExtNodeConfigIccId OBJECT-TYPE

SYNTAX MplsIccId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object allows the operator or service provider to
configure a unique MPLS-TP ITU-T Carrier Code (ICC)
either for Ingress ID or Egress ID.

This object has no meaning when

mplsTunnelExtNodeConfigIccValid is set false."

REFERENCE

"MPLS-TP Identifiers Following ITU-T Conventions,


```

[RFC6923] Section 3"
 ::= { mplsTunnelExtNodeConfigEntry 4 }

mplsTunnelExtNodeConfigNodeId OBJECT-TYPE
    SYNTAX      MplsNodeId
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object indicates the Node_ID within the scope
         of a Global_ID or ICC_Operator_ID."
    REFERENCE
        "MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]
         Section 4."
    ::= { mplsTunnelExtNodeConfigEntry 5 }

mplsTunnelExtNodeConfigIccValid OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "Denotes whether or not this entry uses
         mplsTunnelExtNodeConfigCcId,
         mplsTunnelExtNodeConfigIccId and
         mplsTunnelExtNodeConfigNodeId for mapping
         the ICC based identifiers with the local identifier.
         Note that if this variable is set to false then the
         mplsTunnelExtNodeConfigGlobalId and
         mplsTunnelExtNodeConfigNodeId objects should have
         the valid information."
    DEFVAL { false }
    ::= { mplsTunnelExtNodeConfigEntry 6 }

mplsTunnelExtNodeConfigStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This variable indicates the storage type for this
         object.
         Conceptual rows having the value 'permanent'
         need not allow write-access to any columnar
         objects in the row."
    DEFVAL { volatile }
    ::= { mplsTunnelExtNodeConfigEntry 7 }

mplsTunnelExtNodeConfigRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create

```



```

        STATUS          current
        DESCRIPTION
            "This object allows the operator to create, modify,
            and/or delete a row in this table."
        ::= { mplstunnelExtNodeConfigEntry 8 }

-- End of MPLS Transport Profile Node configuration table

-- Start of MPLS Transport Profile Node IP compatible
-- mapping table

mplstunnelExtNodeIpMapTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MplstunnelExtNodeIpMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This read-only table allows the operator to retrieve
        the local identifier for a given Global_ID::Node_ID in an IP
        compatible operator environment.

        This table MAY be used in on-demand and/or proactive

        OAM operations to get the Ingress/Egress LSR identifier
        (Local Identifier) from Src-Global_Node_ID
        or Dst-Global_Node_ID. The Ingress and Egress LSR
        identifiers are used to retrieve the tunnel entry.

        This table returns nothing when the associated entry
        is not defined in mplstunnelExtNodeConfigTable."
    ::= { mplstunnelExtNodeConfigTable 3 }

mplstunnelExtNodeIpMapEntry OBJECT-TYPE
    SYNTAX          MplstunnelExtNodeIpMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in this table represents a mapping of
        Global_ID::Node_ID with the local identifier.

        An entry in this table is created automatically when
        the Local identifier is associated with Global_ID and
        Node_Id in the mplstunnelExtNodeConfigTable.

        Note: Each entry in this table should have a unique
        Global_ID and Node_ID combination."
    INDEX { mplstunnelExtNodeIpMapGlobalId,
```



```

        mplstunnelExtNodeIpMapNodeId
    }
 ::= { mplstunnelExtNodeIpMapTable 1 }

MplstunnelExtNodeIpMapEntry ::= SEQUENCE {
    mplstunnelExtNodeIpMapGlobalId      MplsGlobalId,
    mplstunnelExtNodeIpMapNodeId        MplsNodeId,
    mplstunnelExtNodeIpMapLocalId       MplsExtendedTunnelId
}

mplstunnelExtNodeIpMapGlobalId  OBJECT-TYPE
    SYNTAX          MplsGlobalId
    MAX-ACCESS      not-accessible
    STATUS           current
    DESCRIPTION
        "This object indicates the Global_ID."
    ::= { mplstunnelExtNodeIpMapEntry 1 }

mplstunnelExtNodeIpMapNodeId  OBJECT-TYPE
    SYNTAX          MplsNodeId
    MAX-ACCESS      not-accessible
    STATUS           current
    DESCRIPTION
        "This object indicates the Node_ID within the

        operator."
    ::= { mplstunnelExtNodeIpMapEntry 2 }

mplstunnelExtNodeIpMapLocalId  OBJECT-TYPE
    SYNTAX          MplsExtendedTunnelId
    MAX-ACCESS      read-only
    STATUS           current
    DESCRIPTION
        "This object contains an IP compatible local identifier
        which is defined in mplstunnelExtNodeConfigTable."
    ::= { mplstunnelExtNodeIpMapEntry 3 }

-- End MPLS Transport Profile Node IP compatible table

-- Start of MPLS Transport Profile Node ICC based table

mplstunnelExtNodeIccMapTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MplstunnelExtNodeIccMapEntry
    MAX-ACCESS      not-accessible
    STATUS           current
    DESCRIPTION
        "This read-only table allows the operator to retrieve
        the local identifier for a given ICC_Operator_ID::Node_ID"

```


in an ICC operator environment.

This table MAY be used in on-demand and/or proactive OAM operations to get the Ingress/Egress LSR identifier (Local Identifier) from Src-ICC or Dst-ICC. The Ingress and Egress LSR identifiers are used to retrieve the tunnel entry. This table returns nothing when the associated entry is not defined in `mplsTunnelExtNodeConfigTable`.
`::= { mplsTeExtObjects 4 }`

`mplsTunnelExtNodeIccMapEntry` OBJECT-TYPE

SYNTAX `MplsTunnelExtNodeIccMapEntry`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents a mapping of `ICC_Operator_ID::Node_ID` with the local identifier.

An entry in this table is created automatically when the Local identifier is associated with `ICC_Operator_ID::Node_ID` in the `mplsTunnelExtNodeConfigTable`."

INDEX { `mplsTunnelExtNodeIccMapCcId`,
`mplsTunnelExtNodeIccMapIccId`,
`mplsTunnelExtNodeIccMapNodeId` }

`::= { mplsTunnelExtNodeIccMapTable 1 }`

`MplsTunnelExtNodeIccMapEntry` ::= SEQUENCE {

`mplsTunnelExtNodeIccMapCcId` `MplsCcId`,

`mplsTunnelExtNodeIccMapIccId` `MplsIccId`,

`mplsTunnelExtNodeIccMapNodeId` `MplsNodeId`,

`mplsTunnelExtNodeIccMapLocalId` `MplsExtendedTunnelId`

}

`mplsTunnelExtNodeIccMapCcId` OBJECT-TYPE

SYNTAX `MplsCcId`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object allows the operator or service provider to configure a unique MPLS-TP ITU-T Country Code (CC) either for Ingress or Egress LSR ID.

The CC is a string of two alphabetic characters represented with upper case letters (i.e., A-Z)."

`::= { mplsTunnelExtNodeIccMapEntry 1 }`

mplsTunnelExtNodeIccMapIccId OBJECT-TYPE

SYNTAX MplsIccId
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"This object allows the operator or service provider to configure a unique MPLS-TP ITU-T Carrier Code (ICC) either for Ingress or Egress LSR ID.

The ICC is a string of one to six characters, each character being either alphabetic (i.e. A-Z) or numeric (i.e. 0-9) characters. Alphabetic characters in the ICC should be represented with upper case letters."

::= { mplsTunnelExtNodeIccMapEntry 2 }

mplsTunnelExtNodeIccMapNodeId OBJECT-TYPE

SYNTAX MplsNodeId
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"This object indicates the Node_ID within the ICC based operator."

::= { mplsTunnelExtNodeIccMapEntry 3 }

mplsTunnelExtNodeIccMapLocalId OBJECT-TYPE

SYNTAX MplsExtendedTunnelId
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"This object contains an ICC based local identifier which is defined in mplsTunnelExtNodeConfigTable."

::= { mplsTunnelExtNodeIccMapEntry 4 }

-- End MPLS Transport Profile Node ICC based table

-- Start of MPLS Tunnel table extension

mplsTunnelExtTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsTunnelExtEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"This table represents extensions to mplsTunnelTable in order to support MPLS-TP tunnels.

As per MPLS-TP Identifiers [RFC6370], LSP_ID for IP based co-routed bidirectional tunnel,

A1-{{Global_ID::Node_ID::Tunnel_Num}}::Z9-{{Global_ID::Node_ID::Tunnel_Num}}::LSP_Num

LSP_ID for IP based associated bidirectional tunnel,
 A1-{{Global_ID::Node_ID::Tunnel_Num::LSP_Num}}::
 Z9-{{Global_ID::Node_ID::Tunnel_Num::LSP_Num}}

mplsTunnelTable is reused for forming the LSP_ID
 as follows,

Source Tunnel_Num is mapped with mplsTunnelIndex,
 Source Node_ID is mapped with
 mplsTunnelIngressLSRId, Destination Node_ID is
 mapped with mplsTunnelEgressLSRId LSP_Num is mapped with
 mplsTunnelInstance.

Source Global_ID::Node_ID and/or ICC_Operator_ID::Node_ID and
 Destination Global_ID::Node_ID and/or ICC_Operator_ID::Node-ID
 are maintained in the mplsTunnelExtNodeConfigTable and
 mplsTunnelExtNodeConfigLocalId is used to create an entry
 in mplsTunnelTable."

REFERENCE

"MPLS Transport Profile (MPLS-TP) Identifiers [RFC6370]."
 ::= { mplsTeExtObjects 5 }

mplsTunnelExtEntry OBJECT-TYPE

SYNTAX MplsTunnelExtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table represents MPLS-TP
 specific additional tunnel configurations."

INDEX {

mplsTunnelIndex,
 mplsTunnelInstance,
 mplsTunnelIngressLSRId,
 mplsTunnelEgressLSRId

}

::= { mplsTunnelExtTable 1 }

MplsTunnelExtEntry ::= SEQUENCE {

mplsTunnelExtOppositeDirPtr	RowPointer,
mplsTunnelExtOppositeDirTnlValid	TruthValue,
mplsTunnelExtDestTnlIndex	MplsTunnelIndex,
mplsTunnelExtDestTnlLspIndex	MplsTunnelInstanceIndex,
mplsTunnelExtDestTnlValid	TruthValue,
mplsTunnelExtIngressLSRLocalIdValid	TruthValue,
mplsTunnelExtEgressLSRLocalIdValid	TruthValue


```
}

mplsTunnelExtOppositeDirPtr OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object points to the opposite direction tunnel entry."
    ::= { mplsTunnelExtEntry 1 }

mplsTunnelExtOppositeDirTnlValid OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "Denotes whether or not this tunnel uses
         mplsTunnelExtOppositeDirPtr for identifying the opposite
         direction tunnel information. Note that if this variable
         is set to true then the mplsTunnelExtOppositeDirPtr should
         point to the first accessible row of the valid opposite
         direction tunnel."
    DEFVAL { false }
    ::= { mplsTunnelExtEntry 2 }

mplsTunnelExtDestTnlIndex OBJECT-TYPE
    SYNTAX      MplsTunnelIndex
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object is applicable only for the bidirectional
         tunnel that has the forward and reverse LSPs in the
         different tunnel entries.

         The values of this object and the
         mplsTunnelExtDestTnlLspIndex object together can be used
         to identify an opposite direction LSP i.e. if the
         mplsTunnelIndex and mplsTunnelInstance hold the value
         for forward LSP, this object and
         mplsTunnelExtDestTnlLspIndex can be used to retrieve
         the reverse direction LSP and vice versa.

         This object and mplsTunnelExtDestTnlLspIndex values
         provide the first two indices of tunnel entry and
         the remaining indices can be derived as follows,
         the Ingress and Egress Identifiers should be
         swapped in order to index the other direction tunnel."
    ::= { mplsTunnelExtEntry 3 }
```



```

mplsTunnelExtDestTnlLspIndex OBJECT-TYPE
    SYNTAX      MplsTunnelInstanceIndex
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object is applicable only for the bidirectional
        tunnel that has the forward and reverse LSPs in the
        different tunnel entries. This object holds
        the instance index of the opposite direction tunnel."
        ::= { mplsTunnelExtEntry 4 }

mplsTunnelExtDestTnlValid OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "Denotes whether or not this tunnel uses
        mplsTunnelExtDestTnlIndex and
        mplsTunnelExtDestTnlLspIndex for identifying
        the opposite direction tunnel information. Note that if
        this variable is set to true then the
        mplsTunnelExtDestTnlIndex and
        mplsTunnelExtDestTnlLspIndex objects should have
        the valid opposite direction tunnel indices."
    DEFVAL { false }
    ::= { mplsTunnelExtEntry 5 }

mplsTunnelExtIngressLSRLocalIdValid OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "This object denotes whether the mplsTunnelIngressLSRId
        contains the local value, which is used to reference
        the complete Ingress Global_ID::Node_ID or ICC_Operator_ID
        from the mplsTunnelExtNodeConfigTable.

        If this object is set to FALSE, mplsTunnelExtNodeConfigTable
        will not contain an entry to reference local identifier with
        Global_ID::Node_ID or ICC_Operator_ID::Node_ID value.

        This object is set to FALSE for legacy implementations like
        MPLS TE tunnels where mplsTunnelIngressId itself provides
        complete Ingress LSRId."
    REFERENCE
        "MPLS-TE-STD-MIB [RFC3812], Section 11.
        mplsTunnelIngressLSRId object in mplsTunnelTable."
    DEFVAL { false }

```



```
 ::= { mplsTunnelExtEntry 6 }

mplsTunnelExtEgressLSRLocalIdValid OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object denotes whether the mplsTunnelEgressLSRId
        contains the local value, which is used to reference
        the complete Egress Global_ID::Node_ID or
        ICC_Operator_ID::Node_ID from
        the mplsTunnelExtNodeConfigTable.

        If this object is set to FALSE, mplsTunnelExtNodeConfigTable
        will not contain an entry to reference local identifier with
        Global_ID::Node_ID or ICC_Operator_ID::Node_ID value.

        This object is set to FALSE for legacy implementations like
        MPLS TE tunnels where mplsTunnelEgressId itself provides
        complete Egress LSRId."
    REFERENCE
        "MPLS-TE-STD-MIB [RFC3812], Section 11.
        mplsTunnelEgressLSRId object in mplsTunnelTable."
    DEFVAL { false }
    ::= { mplsTunnelExtEntry 7 }

-- End of MPLS Tunnel table extension

-- Module compliance.

mplsTeExtCompliances
    OBJECT IDENTIFIER ::= { mplsTeExtConformance 1 }

mplsTeExtGroups
    OBJECT IDENTIFIER ::= { mplsTeExtConformance 2 }

-- Compliance requirement for fully compliant implementations.

mplsTeExtModuleFullCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Compliance statement for agents that provide full
        support the MPLS-TE-EXT-STD-MIB module."

    MODULE -- this module

        -- The mandatory group has to be implemented by all
        -- LSRs that originate/terminate MPLS-TP tunnels.
```



```
-- In addition, depending on the type of tunnels
-- supported, other groups become mandatory as
-- explained below.

MANDATORY-GROUPS {
    mplSTunnelExtGroup
}

GROUP mplSTunnelExtIpOperatorGroup
DESCRIPTION
    "This group is mandatory for devices which support
    configuration of IP based identifier tunnels."

GROUP mplSTunnelExtIccOperatorGroup
DESCRIPTION
    "This group is mandatory for devices which support
    configuration of ICC based tunnels."

 ::= { mplSTeExtCompliances 1 }

-- Compliance requirement for read-only implementations.

mplSTeExtModuleReadOnlyCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Compliance statement for agents that only provide
        read-only support for MPLS-TE-EXT-STD-MIB module."

    MODULE -- this module

MANDATORY-GROUPS {
    mplSTunnelExtGroup
}

GROUP mplSTunnelExtIpOperatorGroup
DESCRIPTION
    "This group is mandatory for devices which support
    configuration of IP based identifier tunnels."

GROUP mplSTunnelExtIccOperatorGroup
DESCRIPTION
    "This group is mandatory for devices which support
    configuration of ICC based tunnels."

-- mplSTunnelExtTable

OBJECT      mplSTunnelExtOppositeDirPtr
```


MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtOppositeDirTnlValid
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtDestTnlIndex
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtDestTnlLspIndex
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtDestTnlValid
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtIngressLSRLocalIdValid
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtEgressLSRLocalIdValid
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtNodeConfigGlobalId
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtNodeConfigNodeId
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtNodeConfigStorageType
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT mplsTunnelExtNodeConfigRowStatus
 SYNTAX RowStatus { active(1) }
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required."

OBJECT mplsTunnelExtNodeConfigCcId
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required."

OBJECT mplsTunnelExtNodeConfigIccId
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required."

OBJECT mplsTunnelExtNodeConfigIccValid
 MIN-ACCESS read-only
 DESCRIPTION
 "Write access is not required."

::= { mplsTeExtCompliances 2 }

-- Units of conformance.

mplsTunnelExtGroup OBJECT-GROUP
 OBJECTS {
 mplsTunnelExtOppositeDirPtr,
 mplsTunnelExtOppositeDirTnlValid,
 mplsTunnelExtDestTnlIndex,
 mplsTunnelExtDestTnlLspIndex,
 mplsTunnelExtDestTnlValid,
 mplsTunnelExtIngressLSRLocalIdValid,
 mplsTunnelExtEgressLSRLocalIdValid
 }

STATUS current
 DESCRIPTION
 "Necessary, but not sufficient, set of objects to
 implement tunnels. In addition, depending on the
 operating environment, the following groups are
 mandatory."
 ::= { mplsTeExtGroups 1 }

mplsTunnelExtIpOperatorGroup OBJECT-GROUP
 OBJECTS { mplsTunnelExtNodeConfigLocalIdNext,


```

        mplsTunnelExtNodeConfigGlobalId,
        mplsTunnelExtNodeConfigNodeId,
        mplsTunnelExtNodeIpMapLocalId,
        mplsTunnelExtNodeConfigStorageType,
        mplsTunnelExtNodeConfigRowStatus
    }
    STATUS current
    DESCRIPTION
        "Object(s) needed to implement IP compatible tunnels."
    ::= { mplsTeExtGroups 2 }

mplsTunnelExtIccOperatorGroup OBJECT-GROUP
    OBJECTS { mplsTunnelExtNodeConfigLocalIdNext,
        mplsTunnelExtNodeConfigCcId,
        mplsTunnelExtNodeConfigIccId,
        mplsTunnelExtNodeConfigNodeId,
        mplsTunnelExtNodeConfigIccValid,
        mplsTunnelExtNodeIccMapLocalId,
        mplsTunnelExtNodeConfigStorageType,
        mplsTunnelExtNodeConfigRowStatus
    }
    STATUS current
    DESCRIPTION
        "Object(s) needed to implement ICC based tunnels."
    ::= { mplsTeExtGroups 3 }

-- MPLS-TE-EXT-STD-MIB module ends
END

```

14. Security Consideration

This document follows the security consideration mentioned in the section 12 of [RFC3812]. These security considerations are also applicable to the MIB objects and tables defined in this draft, which are identified as below.

- The common objects mplsIdGlobalId, mplsIdNodeId, mplsIdCc, and mplsIdIcc are used to define the identity of an MPLS-TP node for OAM purposes. If write-access is allowed to these objects it offers the possibility for incorrect values to be entered that will confuse the information returned by OAM functions and possibly prevent OAM from operating correctly. Furthermore, there is the possibility of inducing one node to impersonate another with confusing results.
- mplsTunnelExtNodeConfigTable, mplsTunnelExtTable and mplsXCExtTable collectively contain objects to provision MPLS-TP

tunnels, tunnel hops, and tunnel resources.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- mplsTunnelExtNodeConfigTable, mplsTunnelExtTable, and mplsXCExtTable collectively show the MPLS-TP tunnel network topology characteristics. If an Administrator does not want to reveal this information, then these tables should be considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

15. IANA Considerations

As described in [RFC4221], [RFC6639] and as requested in the MPLS-TC-STD-MIB [RFC3811], MPLS related standards track MIB modules should be rooted under the mplsStdMIB subtree. There are 4 MPLS MIB Modules contained in this document, each of the following "IANA Considerations" subsections requests IANA for a new assignment under the mplsStdMIB subtree. New assignments can only be made via a

Standards Action as specified in [RFC5226].

15.1. IANA Considerations for MPLS-TC-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-TC-EXT-STD-MIB module specified in this document.

15.2. IANA Considerations for MPLS-ID-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-ID-STD-MIB module specified in this document.

15.3. IANA Considerations for MPLS-LSR-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-LSR-EXT-STD-MIB module specified in this document.

15.4. IANA Considerations for MPLS-TE-EXT-STD-MIB

IANA is requested to assign an OID { mplsStdMIB OID } to the MPLS-TE-EXT-STD-MIB module specified in this document.

16. References

16.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.
- [RFC3811] Nadeau, T., Ed., and J. Cucchiara, Ed., "Definitions of

Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management", RFC 3811, June 2004.

- [RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.
- [RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching (LSR) Router Management Information Base (MIB)", RFC 3813, June 2004.
- [RFC4802] Nadeau, T., Ed., and A. Farrel, Ed., "Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base", RFC 4802, February 2007.
- [RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, September 2011.
- [RFC6923] Winter, R., Gray, E., Helvoort, H., and M. Betts, "MPLS-TP Identifiers Following ITU-T Conventions", RFC 6923, May 2013
- [T.50] "International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information exchange", ITU-T Recommendation T.50, September 1992.

16.2. Informative References

- [RFC3410] J. Case, R. Mundy, D. pertain, B.Stewart, "Introduction and Applicability Statement for Internet Standard Management Framework", RFC 3410, December 2002.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3826] Blumenthal, U., F. Maino and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.
- [RFC4221] Nadeau, T., Srinivasan, C., and A. Farrel, "Multiprotocol Label Switching (MPLS) Management Overview", RFC 4221, November 2005.

- [RFC5226] Narten, T. and H. Alvestrand., "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.
- [RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, September 2009.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, July 2011.
- [RFC6639] Venkatesan, M., King, D., "Multiprotocol Label Switching Transport Profile (MPLS-TP) MIB-Based Management Overview", RFC 6639, June 2012

17. Acknowledgments

The authors would like to thank Francesco Fondelli, Josh Littlefield, Agrahara Kiran Koushik, Metrri Jain, Muly Ilan, Randy Presuhn, Elwyn Davies, Tom Taylor and Pete Resnick for their valuable review and comments. A special thanks to Joan Cucchiara and Adrian Farrel for really getting the MIB modules into shape.

18. Authors' Addresses

Venkatesan Mahalingam
Dell Inc.
5450 Great America Parkway,
Santa Clara, CA 95054, USA
Email: venkat.mahalingams@gmail.com

Sam Aldrin
Huawei Technologies
2330 Central Express Way,
Santa Clara, CA 95051, USA
Email: aldrin.ietf@gmail.com

Thomas D. Nadeau

Brocade
Email: tnadeau@lucidvision.com

Kannan KV Sampath
Redeem
India
Email: kannankvs@gmail.com

MPLS WG
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

K. Kompella
R. Balaji
Juniper Networks, Inc.
February 14, 2014

Label Distribution Using ARP
draft-kompella-larp-00

Abstract

This document describes extensions to the Address Resolution Protocol to distribute MPLS labels for IP host addresses. Distribution of labels via ARP enables simple plug-and-play operation of MPLS, which is among the key goals of "MPLS Fabric" architecture.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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	2
1.1. Approach	2
2. Overview of Ethernet ARP	3
3. L-ARP Protocol Operation	4
3.1. Basic Operation	4
3.2. Asynchronous operation	5
3.3. Applicability	5
4. For Future Study	5
5. L-ARP Message Format	6
6. Security Considerations	8
7. IANA Considerations	8
8. Acknowledgments	8
9. Normative References	8
Authors' Addresses	8

1. Introduction

This document describes extensions to the Address Resolution Protocol (ARP) [RFC0826] to advertise label bindings for IP host addresses. While there are well-established protocols, such as LDP, RSVP and BGP, that provide robust mechanisms for label distribution, these protocols tend to be relatively complex, and often require detailed configuration for proper operation. There are situations where a simpler protocol may be more suitable from an operational standpoint. An example is where an MPLS Fabric is the underlay technology in a Data Centre; here, MPLS tunnels originate from host machines. The host thus needs a mechanism to acquire label bindings to participate in the MPLS Fabric, but in a simple, plug-and-play manner. Existing signaling/routing protocols do not always meet this need. Labeled ARP (L-ARP) is a proposal to fill that gap.

[TODO-MPLS-FABRIC] describes the motivation for using MPLS as the fabric technology.

1.1. Approach

ARP is a nearly ubiquitous protocol; every device with an Ethernet interface, from hand-helds to servers, have an implementation of ARP. ARP is plug-and-play; ARP clients do not need configuration to use ARP. That suggests that ARP may be a good fit for devices that want

to source and sink MPLS tunnels, but do so in a zero-config, plug-and-play manner, with minimal impact to their code.

The approach taken here is to create a minor variant of the ARP protocol, labeled ARP (L-ARP), which is distinguished by a new hardware type, MPLS-over-Ethernet. Regular (Ethernet) ARP (E-ARP) and L-ARP can coexist; a device, as an ARP client, can choose to send out an E-ARP or an L-ARP request, depending on whether it needs Ethernet or MPLS connectivity. Another device may choose to function as an E-ARP server and/or an L-ARP server, depending on its ability to provide an IP-to-Ethernet and/or IP-to-MPLS mapping.

2. Overview of Ethernet ARP

In the most straightforward mode of operation [RFC0826], ARP queries are sent to resolve "directly connected" IP addresses. The ARP query is broadcast, with the target-protocol-address field carrying the IP address of another node in the same subnet. All the nodes in the LAN receive this ARP query. All the nodes, except the node that owns the IP address, ignore the ARP query. The IP address owner learns the MAC address of the sender from the source-hardware-address field in the ARP request, and unicasts an ARP reply to the sender. The ARP reply carries the replying node's MAC address in the source-hardware-address field, thus enabling two-way communication between the two nodes.

A variation of this scheme, known as "proxy ARP" [RFC2002], allows a node to respond to an ARP request with its own MAC address, even when the responding node does not own the requested IP address. Generally, the proxy ARP response is generated by routers to attract traffic for prefixes they can forward packets to. This scheme requires the host to send ARP queries for the IP address the host is trying to reach, rather than the IP address of the router. When there is more than one router connected to a network, proxy ARP enables a host to automatically select an exit router without running any routing protocol to determine IP reachability. Unlike regular ARP, a proxy ARP request can elicit multiple responses, e.g., when more than one router has connectivity to the address being resolved. The sender must be prepared to select one of the responding routers.

Yet another variation of the ARP protocol, called 'Gratuitous ARP' [RFC2002], allows a node to update the ARP cache of other nodes in an unsolicited fashion. Gratuitous ARP is sent as either an ARP request or an ARP reply. In either case, the Source Protocol Address and Target Protocol Address contain the sender's address, and the Source Hardware Address is set to the sender's hardware address. In case of a gratuitous ARP reply, the Target Hardware Address is also set to the sender's address.

3. L-ARP Protocol Operation

The L-ARP protocol builds on the proxy ARP model, and also leverages gratuitous ARP model for asynchronous updates.

In this memo, we will refer to L-ARP clients (that make L-ARP requests) and L-ARP servers (that send L-ARP responses). In Figure 1, C1, C2 and C3 are L-ARP clients, and S1, S2 and S3 are L-ARP servers. T is a member of the MPLS Fabric that may not be an L-ARP server. Within the MPLS Fabric, the usual MPLS protocols (IGP, LDP, RSVP-TE) are run. Say C1, C2 and C3 want to establish MPLS tunnels to each other (for example, they are using BGP MPLS VPNs as the overlay virtual network technology). C1 might also want to talk to a member of the MPLS Fabric, say T.

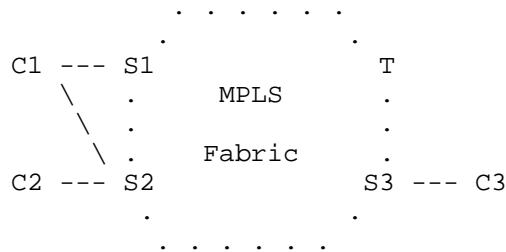


Figure 1

3.1. Basic Operation

A node (say C1) that needs an MPLS tunnel to a destination (say C3) broadcasts an L-ARP query with the Target Protocol Address set to C3. A node that has reachability to C3 (such as S1 or S2) sends an L-ARP reply with the Source Hardware Address set to a locally-allocated MPLS label plus its Ethernet MAC address. After receiving one or more L-ARP replies, C1 can select either S1 or S2 to send MPLS packets that are destined to C3. As described later, the L-ARP response may contain certain parameters that enable the client to make an informed choice of the routers.

As with standard ARP, the validity of the MPLS label obtained using L-ARP is time-bound. The client should periodically resend its L-ARP requests to obtain the latest information, and time out entries in its ARP cache if such an update is not forthcoming. Once an L-ARP server has advertised a label binding, it MUST NOT change the binding until expiry of the binding's validity time.

The mechanism defined here is simplistic; see Section 4.

3.2. Asynchronous operation

The preceding sections described a request-response based model. In some cases, the L-ARP server may want to asynchronously update its clients. L-ARP uses the gratuitous ARP model [RFC2002] to "push" such changes.

In a pure "push" model, a device may send out updates for all prefixes it knows about. This naive approach will not scale well. This memo specifies a mode of operation that is somewhere between "push" and "pull" model. An L-ARP server does not advertise any binding for a prefix until at least one L-ARP client expresses interest in that prefix (by initiating an L-ARP query). As long as the server has at least one interested client for a prefix, the server sends unsolicited (aka gratuitous, though the term is less appropriate in this context) L-ARP replies when a prefix's reachability changes. The server will deem the client's interest in a prefix to have ceased when it does not hear any L-ARP queries for some configured timeout period.

3.3. Applicability

L-ARP can be used between a server and its Top-of-Rack switch in a Data Center. L-ARP can also be used between a DSLAM and its aggregation switch going to the B-RAS. More generally, L-ARP can be used between an "access node" and its first hop MPLS-enabled device in the context of Seamless MPLS [reference]. In all these cases, L-ARP can handle the presence of multiple connections between the access device and its first hop devices.

ARP is not a routing protocol. The use of L-ARP should be limited to cases where the L-ARP client has a small number of one-hop connections to L-ARP servers. The presence of a complex topology between the L-ARP client and server suggests to use a different protocol.

4. For Future Study

The L-ARP specification is quite simple, and the goal is to keep it that way. However, inevitably, there will be questions and features that will be requested. Some of these are:

1. Keeping L-ARP clients and servers in sync. In particular, dealing with:
 - A. client and/or server restart
 - B. lost packets

C. timeouts

2. Withdrawing a response.
3. Dealing with scale.
4. If there are many servers, which one to pick?
5. How can a client make best use of underlying ECMP paths?
6. and probably many more.

In all of these, it is important to realize that, whenever possible, a solution that places most of the burden on the server rather than on the client is preferable.

5. L-ARP Message Format

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ar$hrd                               | ar$pro                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ar$hln                               | ar$pln                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
//                                     ar$sha (variable...)                                     //
+-----+-----+-----+-----+-----+-----+-----+-----+
//                                     ar$spa (variable...)                                     //
+-----+-----+-----+-----+-----+-----+-----+-----+
//                                     ar$tha (variable...)                                     //
+-----+-----+-----+-----+-----+-----+-----+-----+
//                                     ar$tpa (variable...)                                     //
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 2: L-ARP Packet Format

ar\$hrd Hardware Type: MPLS-over-Ethernet. The value of the field used here is [HTYPE-MPLS-TBD]. To start with, we will use the experimental value HW_EXP2 (256)

ar\$pro Protocol Type: IP. The value of the field used here is 0x0800.

ar\$hln Hardware Length: the value of the field used here is 12.

ar\$pln Protocol Address Length: the value is 4.

ar\$op Operation Code: set to 1 for request, and 2 for reply.

ar\$sha Source Hardware Address: In an L-ARP query message, Source Hardware Address is irrelevant, and set to all-zeroes. In an L-ARP reply message, the address follows the 'hardware address' format specified below.

ar\$spa Source Protocol Address: In an L-ARP query message, this field carries the sender's IP address. In an L-ARP reply message, this field carries the target protocol address received in the corresponding query message.

ar\$tha Target Hardware Address: This field is invalid in both request and reply messages.

ar\$tpa Target Protocol Address: In an L-ARP query message, this field carries the IP address for which the client is seeking an MPLS label. In an L-ARP reply message, this field carries the Source Protocol Address received in the corresponding L-ARP query.

The following diagram describes the format of 'Hardware Address' carried in L-ARP.

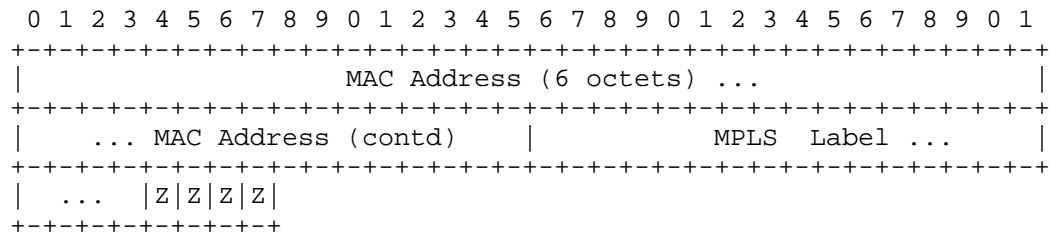


Figure 3: MPLS Hardware Address Format

MAC Address This field contains the Ethernet hardware address that data packets should be directed to.

MPLS Label This field contains the MPLS label allocated by the server. This field is valid only in an L-ARP request message. This field is 20 bits wide, left-justified.

Z These bits are not used, and SHOULD be set to zero on sending and ignored on receipt.

If other parameters are deemed useful in the L-ARP reply, they will be added as needed.

6. Security Considerations

TODO

7. IANA Considerations

TODO

8. Acknowledgments

Many thanks to Shane Amante for his detailed comments and suggestions. Many thanks to the team in Juniper prototyping this work for their suggestions on making this variant workable in the context of existing ARP implementations.

9. Normative References

- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [RFC2002] Perkins, C., "IP Mobility Support", RFC 2002, October 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

Authors' Addresses

Kireeti Kompella
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: kireeti@juniper.net

Balaji Rajagopalan
Juniper Networks, Inc.
Prestige Electra, Exora Business Park
Marathahalli - Sarjapur Outer Ring Road
Bangalore 560103
India

Email: balajir@juniper.net

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

Z. Li
Q. Zhao
Huawei Technologies
T. Yang
China Mobile
February 14, 2014

A Framework of MPLS Global Label
draft-li-mpls-global-label-framework-01

Abstract

The document defines the framework of MPLS global label including: representation of MPLS global label, process of control plane for MPLS global label, and process of data plane for MPLS global label.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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	2
2. Terminology	3
3. Representation of MPLS Global Label	3
3.1. Option A -- Traditional MPLS Label	3
3.2. Option B -- Expansions of MPLS Label	3
4. Control Plane	4
4.1. Overview	4
4.1.1. Shared MPLS Global Label Range Calculation	4
4.1.2. MPLS Global Label Allocation	4
4.1.3. MPLS Global Label Withdraw	5
4.1.4. Error Process	5
4.1.5. Redundancy	5
4.2. BGP-Based Control Plane	6
4.3. IGP-Based Control Plane	6
4.4. PCE-Based Control Plane	8
5. IANA Considerations	9
6. Security Considerations	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Authors' Addresses	10

1. Introduction

[I-D.li-mpls-global-label-usecases] proposes possible usecases of MPLS global label. MPLS global label can be used for identification of the location, the service and the network in different application scenarios. The new solutions based on MPLS global label can gain advantage over the existing solutions to facilitate service provisions.

This document defines the framework for MPLS global label. The framework includes the representation of MPLS global label, the process of control plane and data plane for MPLS global label.

2. Terminology

BDR: Backup Designated Router

DR: Designated Router

FEC: Forward Equivalence Class

MVPN: Multicast VPN

NBMA: Non-broadcast multi-access

PCE: Path Computation Element

PCC: Path Computation Client

RR: Route Reflector

3. Representation of MPLS Global Label

3.1. Option A -- Traditional MPLS Label

Current MPLS label uses 20 bits to represent the label range from 0 to $2^{20} - 1$. Since the existing MPLS label is always allocated locally, in order to guarantee a specific label is allocated globally, the available label values of the network nodes should be reported to a central control point. The central control point can calculate the COMMON label space which is available for all network nodes. Then the network nodes must reserve the common label space for the global label. When the global label is requested to allocate for specific service, the central control point can allocate the label from the common label space.

3.2. Option B -- Expansions of MPLS Label

[I-D.mpls-big-label-ucase-req] proposes the usecases and requirements for MPLS big label. It could also be a reasonable way to define a new label range or segment for MPLS global label which is independent from the existing MPLS label range. The label stack mechanism can be introduced to expand the MPLS label range. For example, the MPLS global label can be represented as the following figure. The global label value is achieved by adding the actual base label value indicated by the base label and the remainder label value.

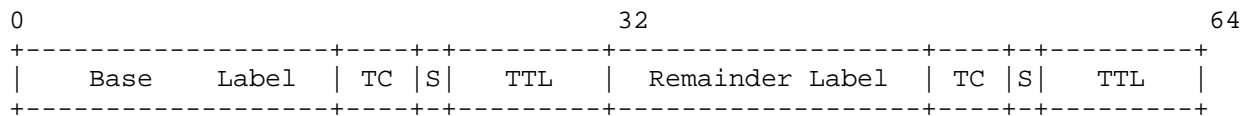


Figure 1 Representation of MPLS Global Label

If the new label range is used for the global label, it can reduce the possible confliction with the existing label range.

4. Control Plane

4.1. Overview

MPLS global label should be allocated concentratedly to guarantee all nodes can understand the same meaning for a specific global label. It should adopt a server/client model in the control plane for MPLS global label allocation. The procedures for the global label are described as follows.

4.1.1. Shared MPLS Global Label Range Calculation

1. Clients nodes should report MPLS global label capability to the central controller.
2. The central controller collects MPLS global label capability and MPLS global label range of all nodes. Then it can calculate the shared MPLS global label range for all nodes.
3. The centralized controller should notify the shared global label range to all client nodes. Client nodes reserve the shared global label range.

4.1.2. MPLS Global Label Allocation

1. The client node should send the global label request for specific usage to the central controller. FEC(Forward Equivalence Class) should be incorporated in the MPLS global label request message.
2. When the central controllers receives the MPLS global label request, it should allocate the label from the shared MPLS global label segment of all nodes.
3. The central controller sends the MPLS global label mapping message to all client nodes. Thus the MPLS global label for specific usage can be understood by all client nodes.

4. The client node receives the MPLS global label mapping message and installs the corresponding MPLS forwarding entry for the global label.

4.1.2.1. Process of Duplicate MPLS Global Label Request

Since MPLS global label is used for specific usage globally, it is possible that multiple MPLS global label requests for the same usage are sent by different client nodes to the central controller. The controller needs to count the MPLS global label requests for the same usage. It can send multiple global label mapping messages to respond these requests. It can also send only one copy of the global label mapping message to all nodes in order to reduce the unnecessary protocol operation. If the controller sends multiple copies of the global label mapping message to respond each label request, client nodes need to ignore the subsequent messages.

4.1.3. MPLS Global Label Withdraw

TBD.

4.1.4. Error Process

TBD.

4.1.5. Redundancy

Since MPLS global label is allocated concentratedly, it is important to guarantee the high availability of the central controller. Redundant backup needs to be introduced for the high availability. The backup central controller needs to learn global label requests sent by client nodes and corresponding label mapping sent by the primary central controller. The backup central controller will not send any global label mapping to client nodes until failure happens for the primary central controller.

The primary role and the backup role of the central controllers can be specified administratively. They can also be elected dynamically based on auto-discovery of these controllers.

The failure detection mechanism also needs to be introduced between the primary controller and the backup controller. It can be the keepalive-like mechanism, the fast detection mechanism like BFD, or mixing use of both mechanisms.

4.2. BGP-Based Control Plane

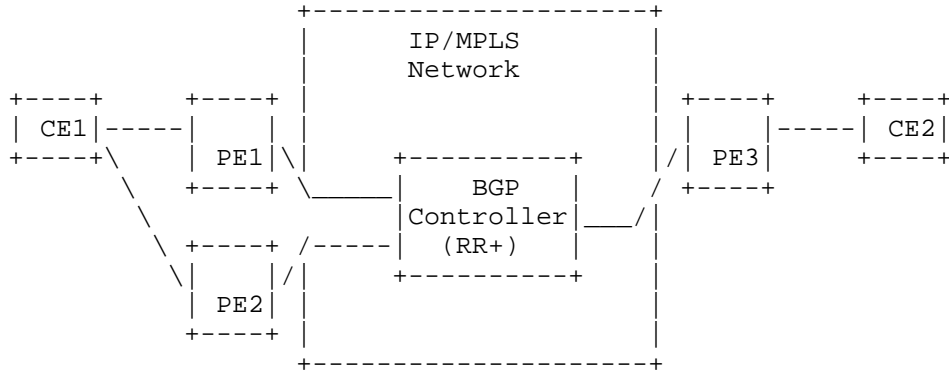


Figure 2: BGP-based Control Plane

Many types of services such as VPLS[RFC4761], Multicast VPN[RFC6514] and E-VPN[I-D.ietf-l2vpn-evpn] are based on MP-BGP. If new solutions based on MPLS global label are introduced for such services, the architecture shown in the figure 2 can be applied.

In the BGP-based control plane, Route Reflector (RR) of BGP [RFC4456] can act as the role of the central controller. We call this type of RR as RR+. For VPLS, Multicast VPN and E-VPN services, auto-discovery mechanisms based on MP-BGP are introduced. So the RR+ can learn the necessary membership information through these A-D routes. RR+ can also learn the MPLS label capability information through necessary MP-BGP extensions. When MPLS global label is necessary, the BGP client on the PE node can send label request to RR+ and the label mapping for the allocated MPLS global label will be advertised to all PEs. Thus all PEs can learn the binding between the service and the MPLS global and the forwarding entry for the MPLS global label can be installed accordingly.

4.3. IGP-Based Control Plane

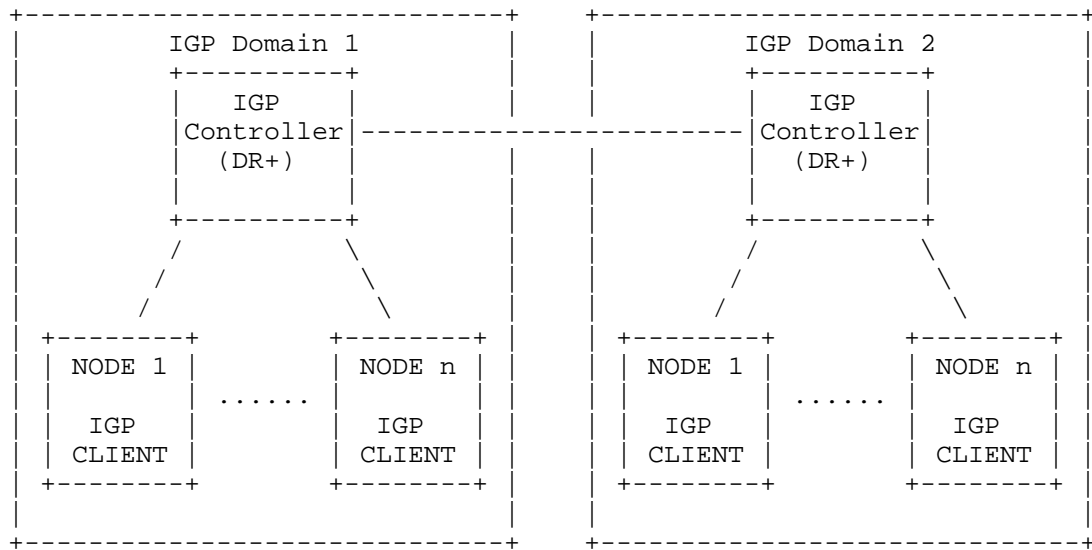


Figure 3: IGP-based Control Plane

If the internal nodes of the network support MPLS global label, IGP-based control plane can be introduced. IGP has ever introduced the DR(Designated Router) and BDR(Backup Designated Router) concepts for broadcast and NBMA network([RFC2328]). The Designated Router is elected in the broadcast or NBMA network to act as a centralized control point to advertise adjacencies among DR and DR others. In the IGP-base control plane for MPLS global label, we can adopt the DR concept which can act as the central controller for the MPLS global label. We called this type of DR ad DR+. The DR+ can collect the MPLS global label capability of all client nodes. If MPLS global label is necessary for specific usage, the MPLS global label will be allocated by the DR+ and the corresponding label mapping can be advertised to all network nodes through IGP extensions. Thus all network nodes in the IGP area can learn the label binding between the specific usage and the MPLS global label and the forwarding entry for the MPLS global label can be installed accordingly.

MPLS global label binding information should be always advertised in a specific IGP domains. There may be multiple IGP domains and nodes in other IGP domains may be necessary to learn the MPLS global label information. There are two possible solutions:

1. The global label information can be advertised by IGP to span multiple domains. It is like leaking the information from the native area to other areas.

2. There can exist direct connections between IGP DR+. The MPLS global label information can be advertised from the native IGP DR+ to the other IGP DR+ using possible protocol extensions other than IGP(e.g. PCEP extensions or BGP extensions). The other IGP DR+ can learn the MPLS global label information and advertise it in its own area through IGP extensions.

4.4. PCE-Based Control Plane

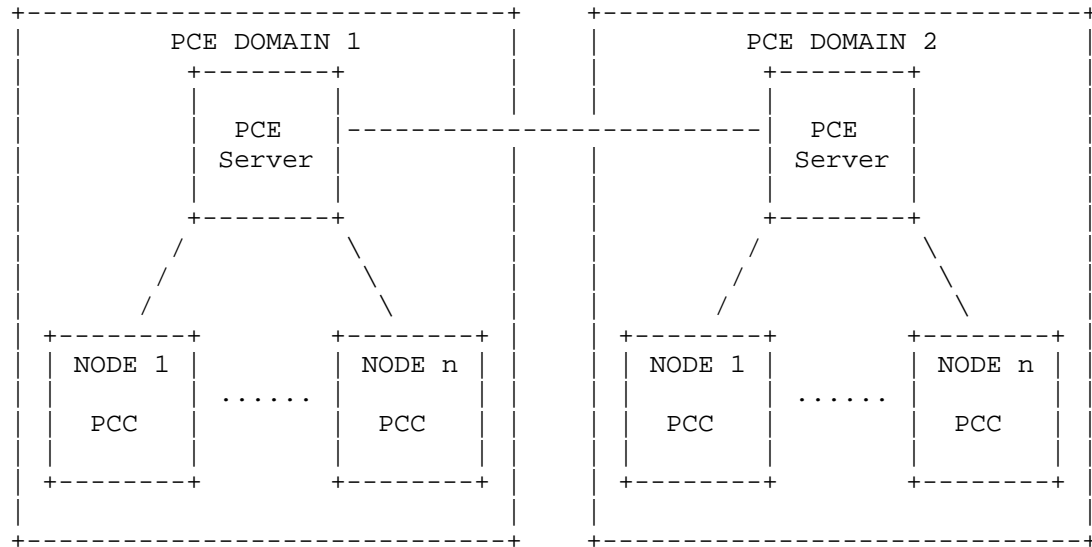


Figure 4: PCE-based Control Plane

PCE[RFC4655] is another choice to implement the control plane for MPLS global label. The PCE servers can act as the role of the centralized controller and the PCC can act the role of the client for process of MPLS global label. PCE servers can collect the MPLS global label capability of all nodes through PCEP extensions. If MPLS global label is necessary for specific usage, the label request can be sent from PCC to PCE server. MPLS global label will be allocated by the PCE server and the corresponding label mapping will be advertised to all network nodes through PCEP extensions. Thus all network nodes in the domain can learn the label binding between the specific usage and the MPLS global label and the forwarding entry for the MPLS global label can be installed accordingly.

If MPLS global label information needs to be advertised in different domain, it can be advertised from the native PCE server to other PCE servers through PCEP extensions. Then other PCE servers can

advertise the MPLS global information to PCC through PCEP in its own domain.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

TBD.

7. References

7.1. Normative References

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

7.2. Informative References

[I-D.ietf-l2vpn-evpn]
Sajassi, A., Aggarwal, R., Henderickx, W., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-05 (work in progress), February 2014.

[I-D.li-mpls-global-label-usecases]
Li, Z., Zhao, Q., and T. Yang, "Usecases of MPLS Global Label", draft-li-mpls-global-label-usecases-01 (work in progress), February 2014.

[I-D.mpls-big-label-ucase-req]
Li, R. and K. Zhao, "MPLS Big Label Usecases and Requirements", draft-mpls-big-label-ucase-req-00 (work in progress), October 2013.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

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

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
US

Email: quintin.zhao@huawei.com

Tianle Yang
China Mobile
32, Xuanwumenxi Ave.
Beijing 01719
China

Email: yangtianle@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2014

Z. Li
T. Huang
Huawei Technologies
L. Chen
Ericsson
October 14, 2013

Alternative Constraints for Point-to-Multipoint Traffic-Engineered MPLS
Label Switched Path(LSP)
draft-li-mpls-p2mp-te-alt-path-01

Abstract

The document proposes a solution to be able to set up the alternative path for specific leaf nodes of a P2MP TE LSP. Corresponding RSVP-TE protocol extension is also defined. The solution is used to cope with the issue that in some scenarios traffic loss happens even if there exists possible path for the leaf nodes.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 April 17, 2014.

Copyright Notice

Copyright (c) 2013 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	2
2. Terminology	3
3. Problem Statement	3
4. Mechanisms	4
4.1. Path Computation in Root Node	4
4.2. Alternative Constraints Propagation	5
4.3. Resource and Label	5
5. Method of Separate Messages	5
6. Method of Single message	6
6.1. Path Message Format	6
6.2. Path Message Processing	7
6.3. Other Messages	7
7. IANA Considerations	8
8. Security Considerations	8
9. Normative References	8
Authors' Addresses	9

1. Introduction

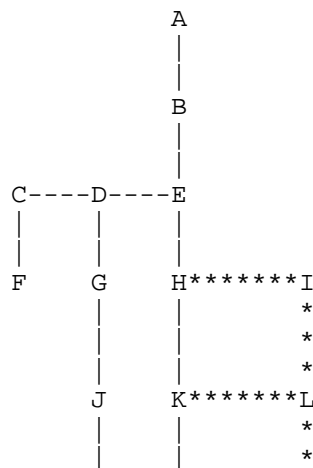
[RFC4461] presents a set of requirements for the establishment and maintenance of Point-to-Multipoint (P2MP) Traffic-Engineered (TE) Multi-protocol Label Switching (MPLS) Label Switched Paths (LSPs). [RFC4875] defines extensions to the RSVP-TE protocol for setup of P2MP TE LSPs. P2MP TE LSPs are set up with a series of traffic engineering constraints. These constraints are applied to all S2L sub-LSPs. This may cause the issue that some S2L sub-LSPs can be set up while others cannot set up according to the constraints. There may be worse case that some S2L sub-LSPs cannot be restored after link failure according to the constraints. When P2MP TE LSPs are used for specific applications, it will cause continuous traffic loss. This document identifies the applicability issue and proposes the solution and corresponding protocol extension.

2. Terminology

This document uses terminologies defined in [RFC2205], [RFC3031], [RFC3209], [RFC3473], [RFC4090], [RFC4461] and [RFC4875].

3. Problem Statement

The P2MP TE LSP is set up with a series of traffic engineering constraints such as bandwidth, explicit path, affinity property(color), etc. These traffic engineering constraints are applied to path computation for all S2L sub-LSPs. Owing to the network provision some leaves of the P2MP LSP are not reachable according to the required constraints (it will be called primary constraints in the following text). There may be the worse case that all leaves are reachable at the beginning and they are not reachable when failure happens. In fact in the scenario these leaves can be reachable if ignore some or all of the primary constraints .



N M*****O

Figure 1. Constraints for P2MP TE LSP

An example for P2MP TE LSP setup is shown in the figure 1. A is the root node and F, N and M are leaf nodes. The link with '|' means the link with red color and the link with '*' means the link with green color. The constraint is that the link with red color should be chosen for the path. For the leaf node M, the path is A->B->E->H->K-M. When link between H and K fails, there is no path with red color can be found from A to M. This will cause the initial available traffic break until the link between H and K restores. The continuous traffic loss can cause bad user experience if the P2MP TE LSP is used for IPTV or other applications. In fact, during the course of failure, there is an alternative path from A to M (A->B->E->H->I->L->K->M) if the link with green color can be chosen.

4. Mechanisms

In order to solve the above applicability issue for P2MP TE LSP, alternative constraints can be specified for the P2MP TE LSP to calculate paths to specific leaf nodes if the path with the primary constraints is not available. The P2MP TE LSP is set up with some S2L sub-LSPs using the primary constraints while the other S2L sub-LSPs using the alternative constraints. The constraints may be used in the downstream nodes, such as ASBR node, and the alternative constraints MUST be propagated to keep the consistence through RSVP-TE protocol extensions.

4.1. Path Computation in Root Node

When alternative constraints is allowed for a specific P2MP TE LSP in the root node, the node MUST try to compute paths for all leaf nodes using the primary constraints. If paths with the primary constraints are available for all leaf nodes, the alternative constraints MUST NOT be used.

When paths with the primary constraints are not available for specific leaf nodes, the alternative constraints SHOULD be used to calculate paths for these leaf nodes. In order to get available paths, the alternative constraints should be looser than the primary constraints. The alternative constraints can be set as zero to simplify the process and the best-effort path as routing is calculated.

When calculate paths with the alternative constraints, the constraints MUST be applied to the whole S2L sub-LSP. That is, it is

prohibited that some parts of the S2L sub-LSP satisfies the primary constraints while other parts satisfies the alternative constraints. If the root node can not calculate the whole S2L sub-LSP (abstract node exists in the calculated path), the alternative constraints MUST be used in the downstream nodes path calculation.

The root node will keep trying to re-optimize to a better path to meet the primary constraints, and it is outside the scope of this document.

4.2. Alternative Constraints Propagation

When setup P2MP LSP, the primary constraint is carried according to the RSVP-TE protocol extension which is defined in [RFC4875]. If the paths to specific leaf nodes are computed using alternative constraints, the alternative constraints MUST be carried corresponding to the S2L sub-LSPs to these leaf nodes in the Path message. These alternative constraints corresponding to S2L sub-LSPs are propagated along the paths from the root node to the leaf nodes.

There are two methods for RSVP-TE protocol to propagate the alternative constraints. One is to propagate alternative constraints in separate message from primary constraints. This method can reuse current P2MP RSVP-TE Message, and does not introduce any extension. The other method is to propagate primary and alternative constraints in single RSVP Message, and need some extension on the Path Message.

When alternative constraints are received for one or more S2L sub-LSPs, they MUST be used when calculating for those S2L sub-LSPs, while the primary constraints MUST be used for other S2L sub-LSPs without alternative constraints. This will be described in detail in the section 5 and 6.

4.3. Resource and Label

When the Resv message is propagated from the leaf nodes to the root node, the transit node MUST reserve resource according to the traffic parameters specified by the required constraints. However, the common upstream node, such as A, B node in figure 1, may have different traffic parameters required if both the primary and alternative constraints exist. But no matter the parameters are same or different, all sub-LSPs in one P2MP LSP MUST share the resource and use same incoming Label on the common nodes.

5. Method of Separate Messages

Propagating alternative constraints through separate messages does not need to introduce any extension on RSVP messages based

on[RFC4875]. However, it needs to change on Path and Resv Message processing. According to [RFC4875], the constraints for all sub-LSPs that belongs to one P2MP LSP should be the same. This document introduces that sub-LSPs can have different constraints in the same P2MP LSP. In this case, a node supporting alternative sub-LSPs MUST accept such different constraints for local processing and continue to propagate them to downstream nodes. The resource reservation and Label processing are as described in Section 4.3.

Exception for the LSP attributes defined by alternative constraints, the S2L sub-LSP descriptors and Sub-Group identifier, the separate Path Message has the same objects with other Path messages for same P2MP LSP.

If a node cannot support alternative sub-LSPs, it MUST send PathErr Message back to Ingress and stop the establishment for such sub-LSPs. But other sub-LSPs with primary constraints SHOULD not be impacted.

6. Method of Single message

This method needs to extend Path Message based on [RFC4875] to carry both primary and alternative constraints in single message.

6.1. Path Message Format

```
<Path Message> ::=
    <Common Header> [ <INTEGRITY> ]
    [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
    [ <MESSAGE_ID> ]
    <SESSION> <RSVP_HOP>
    <TIME_VALUES>
    [ <EXPLICIT_ROUTE> ]
    <LABEL_REQUEST>
    [ <PROTECTION> ]
    [ <LABEL_SET> ... ]
    [ <SESSION_ATTRIBUTE> ]
    [ <NOTIFY_REQUEST> ]
    [ <ADMIN_STATUS> ]
    [ <POLICY_DATA> ... ]
    <sender descriptor>
    [ <S2L sub-LSP descriptor list> ]
```

The following is the format of the S2L sub-LSP descriptor list.

```
<S2L sub-LSP descriptor list> ::= <S2L sub-LSP descriptor>
    [ <S2L sub-LSP descriptor list> ]

<S2L sub-LSP descriptor> ::= <S2L_SUB_LSP>
    [ <P2MP_SECONDARY_EXPLICIT_ROUTE> ]
```



```
[ <P2MP SECONDARY_SESSION_ATTRIBUTE> ]  
[ <P2MP SECONDARY_SENDER_TSPEC> ]
```

In the Path message, S2L_SUB_LSP for specific leaf nodes can carry the alternative constraints besides the explicit route. <P2MP SECONDARY_SESSION_ATTRIBUTE> and <P2MP SECONDARY_SENDER_TSPEC> are added to specify the alternative constraints such as resource affinity, setup and holding priority and traffic parameters. The format, Class Num and C-Type of <P2MP SECONDARY_SESSION_ATTRIBUTE> and <P2MP SECONDARY_SENDER_TSPEC> are all the same as <SESSION_ATTRIBUTE> defined by [RFC3209] and <SENDER_TSPEC> defined by [RFC2210]. The downstream node can judge that the SESSION_ATTRIBUTE and SENDER_TSPEC objects are for alternative constraints of specific S2L sub-LSP when they are placed following corresponding S2L_SUB_LSP object. For convenience, we still use the names, P2MP SECONDARY_SESSION_ATTRIBUTE and P2MP SECONDARY_SENDER_TSPEC, to represent these two objects for specific sub-LSPs.

6.2. Path Message Processing

When a node receives a Path Message with P2MP SECONDARY_SESSION_ATTRIBUTE and P2MP SECONDARY_SENDER_TSPEC objects following one or more S2L_SUB_LSP objects, it can judge that such sub-LSPs are alternative sub-LSPs which have attributes identified by these two objects.

If after a branch node, the alternative sub-LSP will become alone, then the branch node will signal a new Path Message for that alternative sub-LSP in the normal way. This means, for this new path message, the content of P2MP SECONDARY_SESSION_ATTRIBUTE and P2MP SECONDARY_SENDER_TSPEC objects will be carried by the primary SESSION_ATTRIBUTE and SENDER_TSPEC like a normal P2MP Path Message, and these two new objects will not be carried any more to downstream. The SUB-Group ID for that path message will also be a new value different from the original Primary sub-LSP for the same egress.

If a transit node cannot support alternative sub-LSPs, it MUST send a PathErr Message back to ingress.

6.3. Other Messages

The format of Resv Message based on [RFC4875] does not need to be modified. But a new case for Resv Message processing is introduced that, a branch node may receive different traffic parameters in FLOWSPEC of the same P2MP LSP from different downstream nodes. It

MUST calculate the shared resource for resource reservation and carry the result as FLOWSPEC to upstream.

For other RSVP Messages based on [RFC4875], the message format and processing have no change.

7. IANA Considerations

TBD.

8. Security Considerations

This document does not introduce any security issues above those identified in[RFC4875].

9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 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.
- [RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC4461] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.
- [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.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Tieying Huang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: huangtieying@huawei.com

Lei Chen
Ericsson
CDK building, No.1 Wangjing North Rd.
Beijing 100102
China

Email: charles.c.chen@ericsson.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

Kishore Tiruveedhula, Ed.
Juniper Networks
Uwe Joerde
Deutsche Telekom
Arvind Venkateswaran
Cisco Systems
February 14, 2014

Definitions of Managed Objects for the LDP Point-to-Multipoint and
Multipoint-to-Multipoint Label Switched Paths
draft-tiruveedhula-mpls-mldp-mib-02

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular it defines objects for managing multicast LDP point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) Label Switched Paths. The MIB module defined in this document is extension of LDP MIB defined in RFC3815 which supports only for LDP point-to-point LSPs.

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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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	4
2. The Internet-Standard Management Framework	4
3. Conventions	4
4. Overview	5
5. Future Considerations	5
6. Structure of the MIB Module	5
6.1. Summary of mLDP Scalar Objects	6
6.2. Summary of mLDP Table Objects	6
7. mLDP Scalar Objects	6
7.1. mplsMldpP2mpCapable	6
7.2. mplsMldpMp2mpCapable	7
7.3. mplsMldpMbbCapable	7
7.4. mplsMldpMbbTime	7
7.5. mplsMldpNumFecs	7
7.6. mplsMldpNumFecsActive	7
7.7. mplsMldpPlrCapable	7
7.8. mplsMldpMptCapable	7
7.9. mplsMldpProtLsrCapable	7
7.10. mplsMldpNodeProtCapable	8
8. mLDP Table Objects	8
8.1. LDP Peer Capability Table mplsLdpPeerCapabilityTable	8
8.2. mLDP Session Stats Table: mplsMldpSessionStatsTable	8
8.3. mLDP Fec Table: mplsMldpFecTable	8
8.4. mLDP Fec Branch Traffic statistics Table: mplsMldpFecBranchStatsTable	8
8.5. mLDP Fec Upstream Session Table: mplsMldpFecUpstreamSessTable	8
8.6. mLDP Interface Traffic statistics Table: mplsMldpInterfaceStatsTable	8
9. The mLDP Notifications	9
10. Relationship to Other MIB Modules	9
10.1. Diagrammatic Representation	10
10.2. Relationship to the LSR MIB	10
10.3. Relationship to the LDP MIB	11
11. Multicast MPLS Label Distribution Protocol MIB Definitions	11
12. Security Considerations	32
13. IANA Considerations	34
14. Acknowledgments	34
15. References	34
15.1. Normative References	34

15.2. Informative References	35
Appendix A. Change Log	36
Appendix B. Open Issues	36

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular it defines objects for managing multicast LDP point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) Label Switched Paths. The MIB module defined in this document is extension of LDP MIB defined in RFC3815 which supports only for LDP point-to-point LSPs.

The RFC3815 describes only unicast Managed objects for the Label distribution protocol. The RFC6388 describes LDP protocol extensions for the point to multipoint and multipoint to multipoint LSPs. The RFC 6826 describes multicast LDP inband signalling for P2MP and MP2MP LSPs.

This document defines a MIB module for managing and controlling mLDP P2MP and MP2MP LSPs. It builds on the objects and tables defined in [RFC3815] for mLDP MIB.

2. The Internet-Standard Management Framework

[[anchor3: The title and text for this section has been copied from the official boilerplate, and should not be modified unless the official boilerplate text from the OPS Area web site has changed. See RFC4818 section 3.1 for a discussion of the boilerplate section.]]

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Conventions

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 BCP 14, RFC 2119 [RFC2119].

4. Overview

This document focusses on the management of following multicast LDP (mLDP) features, which were defined after unicast LDP [RFC5036].

RFC6388: Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths.

RFC6826: Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths.

RFC7060: Using LDP Multipoint Extensions on Targeted LDP Sessions.

[MoFRR] Multicast only Fast Re-Route draft-ietf-rtgwg-mofrr-03 .

[MLDP_NODE_PROT] mLDP Node Protection.

For all the above features, the mLDP MIB needs to include the following information:

- Session Capability (P2MP, MP2MP) information: configured capability, negotiated capability.
- mLDP FECs: include opaque information (Generic LSP Identifier, source and group address) and MoFRR enable.
- Primary and backup upstream session when mLDP MoFRR enabled.
- Active and inactive upstream session for make before break.
- mLDP Traffic stats per mLDP Fec: The traffic stats for mLDP fec.
- mLDP Traffic stats per per Interface: The mLDP traffic stats per Interface.
- Traps when mLDP Fec LSP up, down.

5. Future Considerations

Any new opaque TLVs added for any other mLDP fetures, the opaque value object in the mplsMldpFecTable need to be enhanced accordingly.

6. Structure of the MIB Module

This section describes the structure of the mLDP MIB. In this MIB MPLS-MLDP-STD-MIB, scalar objects, table objects and notifications are defined. Following section describes in details about each object.

6.1. Summary of mLDP Scalar Objects

New scalar objects `mplsMldpP2mpCapable` and `mplsMldpMp2mpCapable` are defined to provide the mLDP capabilities of P2MP, MP2MP support.

New scalar objects `mplsMldpMbbCapable` and `mplsMldpMbbTime` are defined to provide MBB capability information.

New scalar object `mplsMldpNumFecs` which will give the total number of mLDP FECs setup on the LSR.

Another New scalar object `mplsMldpNumFecsActive`, which will give the total number of active mLDP FECs.

New scalar objects `mplsMldpPlrCapable`, `mplsMldpMptCapable`, `mplsMldpProtLsrCapable` and `mplsMldpNodeProtCapable` are defined to provide mLDP node protection capabilities.

6.2. Summary of mLDP Table Objects

`mplsLdpPeerCapabilityTable` to include peer capability information.

`mplsMldpSessionStatsTable` : This table contains the number of mLDP FECs received and advertised to particular LDP session.

`mplsMldpFecTable`: This table is similar to point to point `mplsLdpFecTable` and will have mLDP specific Fec information.

`mplsMldpFecBranchStatsTable` : This table contains the traffic statistics for the given mLDP FECs on particular interface.

`mplsMldpFecUpstreamSessTable` : Includes the upstream session info for the particular mLDP Fec and also includes the primary or backup upstream session, that may be used for mLDP MoFRR.

`mplsMldpInterfaceStatsTable` : This table contains the traffic statistics for all mLDP related FECs.

7. mLDP Scalar Objects

There are ten scalars, listed below are defined for this MIB module.

7.1. `mplsMldpP2mpCapable`

The `mplsMldpP2mpCapable` scalar object denotes whether the LSR is capable of supporting multicast LDP with Point-to-Multipoint capability.

7.2. mplsMldpMp2mpCapable

The mplsMldpMp2mpCapable scalar object denotes whether the LSR is capable of supporting multicast LDP with Multipoint-to-Multipoint LSPs.

7.3. mplsMldpMbbCapable

The mplsMldpMbbCapable scalar object denotes whether the LSR is capable of supporting multicast LDP with MBB (make before break) feature mentioned in the section 8 of RFC 6388 .

7.4. mplsMldpMbbTime

The mplsMldpMbbTime scalar object denotes MBB time for which LSR is waiting for MBB Ack from upstream node. This timer helps LSR to prevent waiting indefinitely for the MBB Notification from upstream node.

7.5. mplsMldpNumFecs

The mplsMldpNumFecs provides a read-only counter of the number of mLDP FECs setup on this LSR.

7.6. mplsMldpNumFecsActive

The mplsMldpNumFecsActive provides a read-only counter of the number of mLDP FECs Active on this LSR.

7.7. mplsMldpPlrCapable

The mplsMldpPlrCapable scalar object denotes whether the LSR is capable of supporting PLR capability as specified in the section 5.1 of [MLDP_NODE_PROT]

7.8. mplsMldpMptCapable

The mplsMldpMptCapable scalar object denotes whether the LSR is capable of supporting MPT capability as specified in the section 5.2 of [MLDP_NODE_PROT]

7.9. mplsMldpProtLsrCapable

The mplsMldpProtLsrCapable scalar object denotes whether the LSR is capable of supporting the "Protected LSR" capability as specified in the section 5.3 of [MLDP_NODE_PROT]

7.10. mplsMldpNodeProtCapable

The mplsMldpNodeProtCapable scalar object denotes whether the LSR is capable of supporting the "Node Protection" capability as specified in the section 5.4 of [MLDP_NODE_PROT]

8. mLDP Table Objects

8.1. LDP Peer Capability Table mplsLdpPeerCapabilityTable

The new table mplsLdpPeerCapabilityTable is read-only table, which contains learned capability information from LDP peer. This table augments the mplsLdpPeerTable, which is defined in RFC 3815.

8.2. mLDP Session Stats Table: mplsMldpSessionStatsTable

The mplsMldpSessionStatsTable is a read-only table which contains mLDP statistical information on sessions. This table augments the mplsLdpSessionStatsTable, which is defined in the RFC 3815.

8.3. mLDP Fec Table: mplsMldpFecTable

The mplsMldpFecTable is a table which contains FEC (Forwarding Equivalence Class) information relating to point to multi-point and multipoint to multipoint LDP LSP. Each entry/row represents a single FEC Element. This table is similar LDP LSP FEC Table, mplsLdpLspFecTable, which is defined in the RFC 3815, which associates FECs with the LSPs.

8.4. mLDP Fec Branch Traffic statistics Table: mplsMldpFecBranchStatsTable

This table mplsMldpFecBranchStatsTable gives the information about number of packets and number of bytes sent out on particular downstream session or on outgoing interface.

8.5. mLDP Fec Upstream Session Table: mplsMldpFecUpstreamSessTable

The mplsMldpFecUpstreamSessTable is a read-only table which contains mLDP upstream session information for mLDP Fec. This table is similar to mplsInSegmentLdpLspTable. This table will also have information about primary, backup upstream session, and also indicates whether the label is in MBB request or MBB Ack received state.

8.6. mLDP Interface Traffic statistics Table: mplsMldpInterfaceStatsTable

This table mplsMldpInterfaceStatsTable gives the information about

number of mLDP packets and number of mLDP bytes sent and received on particular interface for all mLDP FECs.

9. The mLDP Notifications

The RFC 3815 defined some of the notifications related to session and P2P Fec. In this MIB, the following notification added to support mLDP features.

The `mplsMldpFecUp` and `mplsMldpFecDown` notifications are generated when mLDP FEC changes the state to UP and Down.

10. Relationship to Other MIB Modules

This section describes relationships between MIB tables defined in this document as part of MPLS-MLDP-STD-MIB, and the tables defined in MPLS-LDP-STD-MIB [RFC3815] and MPLS-LSR-STD-MIB [RFC3813].

The Figure 1 shows the diagrammatic representation of the relationship between MPLS-MLDP-STD-MIB, MPLS-LDP-STD-MIB and MPLS-LSR-STD-MIB. An arrow in the Figure shows that the MIB table pointed from contains a reference to the MIB table pointed to.

10.1. Diagrammatic Representation

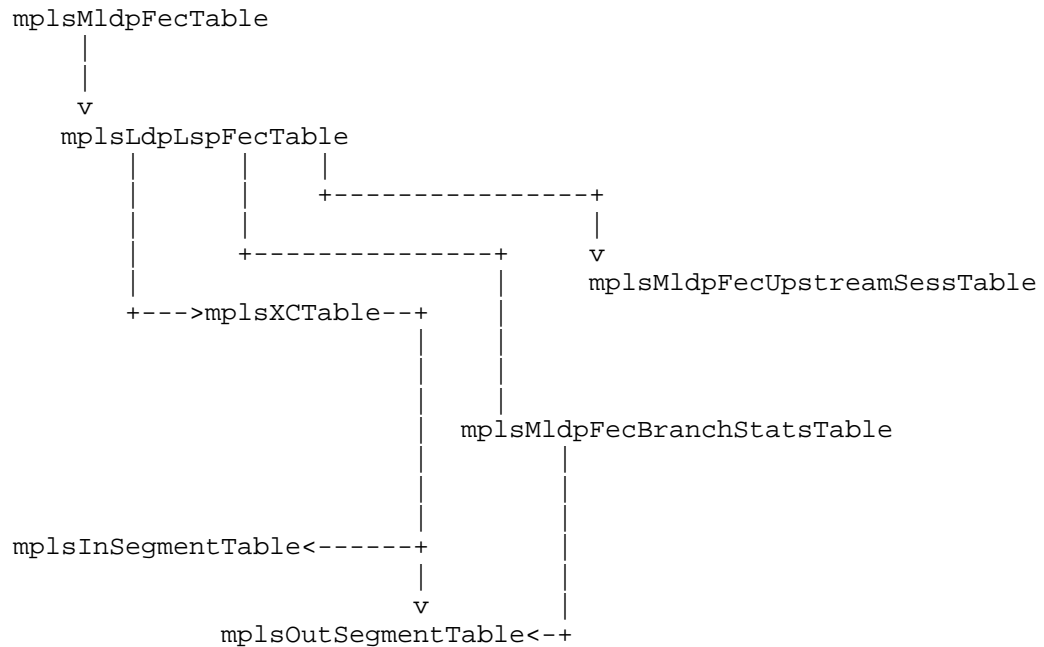


Figure 1 : Dependencies Between MIB Tables

Figure 1

10.2. Relationship to the LSR MIB

The LSR MIB [RFC3813] have below tables, which cross connects the incoming label to outgoing label. Below Tables will be used for mLDP also in the similar way as in the point to point LDP LSPs.

```

mplsXCTable

mplsInSegmentTable

mplsOutSegmentTable

```


10.3. Relationship to the LDP MIB

The MIB module defined in this document is extension of MPLS-LDP-STD-MIB to support multicast LDP features.

Below optional tables in MPLS-LDP-STD-MIB, will also be used in mLDP for associating the mLDP LSPs to LSR-MIB tables.

mplsLdpLspFecTable

mplsInSegmentLdpLspTable

mplsOutSegmentLdpLspTable

11. Multicast MPLS Label Distribution Protocol MIB Definitions

```

MPLS-MLDP-STD-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Unsigned32, Counter32, Counter64, TimeTicks
        FROM SNMPv2-SMI
        -- RFC 2578
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
        -- RFC 2580
    TruthValue, RowStatus, StorageType, TimeStamp
        FROM SNMPv2-TC
        -- RFC 2579

    InterfaceIndex
        FROM IF-MIB
        -- [RFC2020]

    mplsStdMIB, MplsLdpIdentifier
        FROM MPLS-TC-STD-MIB
        -- RFC 3811

    MplsIndexType
        FROM MPLS-LSR-STD-MIB
        -- RFC 3813

    IndexInteger, IndexIntegerNextFree
        FROM DIFFSERV-MIB
        -- RFC 3289

    InetAddress, InetAddressType
        FROM INET-ADDRESS-MIB
        -- RFC 4001

    mplsLdpStdMIB
        FROM MPLS-LDP-STD-MIB
        -- RFC 3815
    ;

mplsMldpStdMIB MODULE-IDENTITY

```


LAST-UPDATED "201402140000Z" -- Feb 14, 2014
ORGANIZATION "Multiprotocol Label Switching (mpls)
Working Group"

CONTACT-INFO

" Kishore Tiruveedhula
Juniper Networks
Email: kishoret@juniper.net

Uwe Joerde
Deutsche Telekom
Email: Uwe.Joerde@telekom.de

Arvind Venkateswaran
Cisco Systems
EMail: arvvenka@cisco.com

Comments about this document should be emailed
directly to the MPLS working group mailing list at
mpls@lists.ietf.org"

DESCRIPTION

"Copyright (c) 2009 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 in effect on the
date of publication of this document
(<http://trustee.ietf.org/license-info>). Please review these
documents carefully, as they describe your rights and
restrictions with respect to this document.

The initial version of this MIB module was published in
RFC XXXX. For full legal notices see the RFC itself or see:
<http://www.ietf.org/copyrights/ianamib.html>

-- RFC Editor. Please replace XXXX with the RFC number for this
-- document and remove this note.

This MIB module contains managed object definitions for mLDP LSPS
defined in Label Distribution Protocol Extensions Point-to-Multipoin
t and

Multipoint-to-Multipoint Label Switched Paths, RFC 6388, November
2011."

REVISION "201402140000Z" -- Feb 14, 2014
DESCRIPTION

"Initial version issued as part of RFC XXXX."
-- RFC Editor. Please replace XXXX with the RFC number for this
-- document and remove this note.


```
 ::= { mplsStdMIB 99 }
-- RFC Editor. Please replace 99 with the codepoint issued by IANA
-- and remove this note.

-- Top level components of this MIB module.

-- notifications
mplsMldpNotifications OBJECT IDENTIFIER ::= { mplsMldpStdMIB 0 }
-- tables, scalars
mplsMldpScalars          OBJECT IDENTIFIER ::= { mplsMldpStdMIB 1 }
mplsMldpObjects          OBJECT IDENTIFIER ::= { mplsMldpStdMIB 2 }

-- MPLS mLDP LSP scalars.

mplsMldpP2mpCapable OBJECT-TYPE
    SYNTAX          INTEGER {
                        enable(1),
                        disable(2)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object provides the P2MP capability of the LSR."

    REFERENCE
        "Section 2.1 of [RFC6388]."
```

```
 ::= { mplsMldpScalars 1 }
```

```
mplsMldpMp2mpCapable OBJECT-TYPE
    SYNTAX          INTEGER {
                        enable(1),
                        disable(2)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object provides MP2MP capability of the LSR."

    REFERENCE
        "Section 3.1 of [RFC6388]."
```

```
 ::= { mplsMldpScalars 2 }
```

```
mplsMldpMbbCapable OBJECT-TYPE
```



```
SYNTAX      INTEGER {
                enable(1),
                disable(2)
            }
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "This object provides MBB (make before break) capability of the LSR."

REFERENCE
    "Section 8.3 of [RFC6388]."
```

::= { mplsMldpScalars 3 }

mplsMldpMbbTime OBJECT-TYPE

```
SYNTAX      Unsigned32 (1..300)
UNITS       "seconds"
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "The 32-bit unsigned integer value provides the time for waiting MBB
```

Ack

```
    from upstream node."

DEFVAL { 30 }
::= { mplsMldpScalars 4 }
```

mplsMldpNumFecs OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "The number of mLDp FECs setup on this device. "
```

::= { mplsMldpScalars 5 }

mplsMldpNumFecsActive OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
    "The number of mLDp FECs Active on this device. The mLDP FEC is
    considered active if the mplsMldpFecOperStatus is up(1)."
```

::= { mplsMldpScalars 6 }

mplsMldpPlrCapable OBJECT-TYPE

```
SYNTAX      INTEGER {
```



```

        enable(1),
        disable(2)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This object provides Point of Local Repair (PLR)
    capability of the LSR."

REFERENCE
    "Section 5.1 of [MLDP_NODE_PROT]."
```

::= { mplsMldpScalars 7 }

```
mplsMldpMptCapable OBJECT-TYPE
    SYNTAX      INTEGER {
        enable(1),
        disable(2)
    }
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "This object provides Merge Point (MPT) capability of the LSR."

    REFERENCE
        "Section 5.2 of [MLDP_NODE_PROT]."
```

::= { mplsMldpScalars 8 }

```
mplsMldProtLsrCapable OBJECT-TYPE
    SYNTAX      INTEGER {
        enable(1),
        disable(2)
    }
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "This object provides Protected LSR capability."

    REFERENCE
        "Section 5.3 of [MLDP_NODE_PROT]."
```

::= { mplsMldpScalars 9 }

```
mplsMldProtNodeProtCapable OBJECT-TYPE
    SYNTAX      INTEGER {
        enable(1),
        disable(2)
    }
```



```

        }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object provides Node Protection capability of the LSR."

    REFERENCE
        "Section 5.3 of [MLDP_NODE_PROT]."
```

::= { mplsMldpScalars 10 }

-- End of MPLS mLDP scalars.

-- MPLS mLDP tables.

--

-- The MPLS LDP Peer Capability Table

--

mplsLdpPeerCapabilityTable OBJECT-TYPE

```

    SYNTAX          SEQUENCE OF MplsLdpPeerCapabilityEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table will have learned information relating to Mldp."
    ::= { mplsMldpObjects 1 }
```

mplsLdpPeerCapabilityEntry OBJECT-TYPE

```

    SYNTAX          MplsLdpPeerCapabilityEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Information about a single Peer which is related
        to a Session. This table is augmented by
        the mplsLdpSessionTable."
    INDEX
        { mplsLdpEntityLdpId,
          mplsLdpEntityIndex,
          mplsLdpPeerLdpId }

    ::= { mplsLdpPeerCapabilityTable 1 }
```

mplsLdpPeerCapabilityEntry ::= SEQUENCE {

```

    mplsLdpPeerLdpId          MplsLdpIdentifier,
    mplsLdpPeerCapability     Integer32,
}
```

mplsLdpPeerCapability OBJECT-TYPE

```

    SYNTAX          BITS {
```



```

        none (0),
        p2mp (1),
        mp2mp(2),
        mbb (3),
        upstream-label-assignment (4),
        dynamic (5),
        plr (6),
        mpt (7),
        prot-lsr (8),
        node-prot (9)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    " This will indicate the LDP capability information about peer.
    p2mp indicates peer supports P2MP Capability.
    mp2mp indicates peer supports MP2MP Capability.
    mbb indicates peer supports MBB Capability.
    upstream-label-assignment indicates peer supports Upstream label
    assignment Capability.
    dynamic indicates peer supports dynamic Capability.
    "

REFERENCE
    "RFC6388, Section 2.1 for P2MP Capability TLV.
    and the section 3.1 for MP2MP Capability TLV.
    The RFC6388 for MBB Capability TLV.
    RFC5561 Section 9 for Dynamic Capability Announcement TLV.
    RFC6389 Section 3 for Upstream Label Assignment Capability TLV.
    MLDP_NODE_PROT section 5 for PLR capability, MPT capability,
    The Protected LSR and The Node Protection Capability. "

 ::= { mplsLdpPeerCapability 2 }

--
-- The MPLS mLDP Session Statistics Table
--

mplsMldpSessionStatsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF MplsMldpSessionStatsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A table of statistics related to mLDP on Sessions.
        This table AUGMENTS the mplsLdpSessionStatsTable."
    ::= { mplsMldpObjects 2 }

mplsMldpSessionStatsEntry OBJECT-TYPE
```



```

SYNTAX      MplsMldpSessionStatsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in this table represents mLDP statistical
    information on a single session between an LDP
    Entity and LDP Peer."

AUGMENTS    { mplsLdpSessionStatsEntry }
::= { mplsMldpSessionStatsTable 1 }

MplsMldpSessionStatsEntry ::= SEQUENCE {
    mplsMldpSessionStatsNumFecsSent          Counter32,
    mplsMldpSessionStatsNumMbbReqSentState   Counter32,
    mplsMldpSessionStatsNumFecsRcvd         Counter32,
    mplsMldpSessionStatsNumMbbReqRcvdState   Counter32,
    mplsMldpSessionStatsNumMbbResetAckByTimer Counter32
}

mplsMldpSessionStatsNumFecsSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts the number of mLDP FECs sent on this
        session. If the FEC is withdrawn, then this number is
        decremented.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        mplsLdpSessionDiscontinuityTime."

    ::= { mplsMldpSessionStatsEntry 1 }

mplsMldpSessionStatsNumMbbReqSentState OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts the number of mLDP FECs sent on this
        session and waiting for MBB Ack. This counter will get incremented
        when MBB req sent for a label on this session and will get
        decremented when the MBB Ack received.

    ::= { mplsMldpSessionStatsEntry 2 }

```


mplsMldpSessionStatsNumFecsRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object counts the number of mLDP FECs received on this session. If the FEC is withdrawn from the downstream session, then this is decremented.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of mplsLdpSessionDiscontinuityTime."

::= { mplsMldpSessionStatsEntry 3 }

mplsMldpSessionStatsNumMbbReqRcvdState OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object counts the number of mLDP FECs received on this session and waiting for sending MBB Ack. This counter will get incremented when MBB req is received for a label on this session and will get decremented when the MBB Ack sent."

::= { mplsMldpSessionStatsEntry 4 }

mplsMldpSessionStatsNumMbbResetAckByTimer OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object counts the number mLDP FECs for which the MBB Ack is reset by MBB timer, in which the LSR is waiting for MBB ack.

::= { mplsMldpSessionStatsEntry 5 }

--

-- Mpls mLDP FEC Table

--

mplsMldpFecTable OBJECT-TYPE

SYNTAX SEQUENCE OF MplsFecEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table represents the FEC
(Forwarding Equivalence Class)
Information associated with an mLDP LSP."

::= { mplsMldpObjects 3 }

mplsMldpFecEntry OBJECT-TYPE
SYNTAX MplsMldpFecEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Each row represents a single mLDP FEC Element."
INDEX { mplsMldpFecIndex }

::= { mplsMldpFecTable 1 }

MplsMldpFecEntry ::= SEQUENCE {
mplsMldpFecIndex IndexInteger,
mplsMldpFecType INTEGER,
mplsMldpFecRootAddrType InetAddressType,
mplsMldpFecRootAddr InetAddress,
mplsMldpFecOpaqueType INTEGER,
mplsMldpFecOpaqueGenLspId Unsigned32,
mplsMldpFecOpaqueTransitSourceOrBidirAddrType InetAddressType,
mplsMldpFecOpaqueTransitSourceOrBidirAddr InetAddress,
mplsMldpFecOpaqueTransitGroupAddrType InetAddressType,
mplsMldpFecOpaqueTransitGroupAddr InetAddress,
mplsMldpFecAdminStatus INTEGER,
mplsMldpFecOperStatus INTEGER,
mplsMldpFecMoFrr INTEGER,
mplsMldpFecLsrState INTEGER,
mplsMldpFecUpTime TimeStamp
}

mplsMldpFecIndex OBJECT-TYPE
SYNTAX IndexInteger
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The index which uniquely identifies this entry."

::= { mplsMldpFecEntry 1 }

mplsMldpFecType OBJECT-TYPE
SYNTAX INTEGER {
p2mp(6),


```

        mp2mpUpstream(7),
        mp2mpDownstream(8)
    }
MAX-ACCESS read-only
STATUS current
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The type of the FEC. If the value of this object
    is 6, then it is P2MP Fec Type, and 7, 8 are correspond to
    MP2MP upstream and downstream type."

REFERENCE
    "RFC6388, Section 2.2. The P2MP FEC Element and the section 3.3
    for the MP2MP Fec elements."

::= { mplsMldpFecEntry 2 }

mplsMldpFecRootAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is the type of the
        Internet address. The value of this object,
        decides how the value of the mplsMldpFecRootAddr object
        is interpreted."
    REFERENCE
        "RFC6388, Section 2.2. The P2MP FEC Element and the section 3.3
        for the MP2MP Fec elements."

    ::= { mplsMldpFecEntry 3 }

mplsMldpFecRootAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is interpreted based
        on the value of the mplsMldpFecRootAddrType object.
        This is ingress node address for the mLDP LSP."
    REFERENCE
        "RFC6388, Section 2.2. The P2MP FEC Element and the section 3.3
        for the MP2MP Fec elements."

    ::= { mplsMldpFecEntry 4 }
```



```

mplsMldpFecOpaqueType OBJECT-TYPE
    SYNTAX      INTEGER {
                    genericLspId(1),
                    transitIpv4Source(3),
                    transitIpv6Source(4),
                    transitIpv4Bidir(5),
                    transitIpv6Bidir(6)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This is opaque type of the mLDP FEC. The value of this object is
        shown below.

        1 - The Generic LSP Identifier
        3 - Transit IPv4 Source TLV
        4 - Transit IPv6 Source TLV
        5 - Transit IPv4 Bidir TLV
        6 - Transit IPv6 Bidir TLV.
        "
    ::= { mplsMldpFecEntry 5 }

mplsMldpFecOpaqueGenLspId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The 32-bit unsigned integer value which is to represent Generic
        LSP ID. This value is only valid if the mplsMldpFecOpaqueType is
        genericLspId(1), otherwise 0 must be returned."

    REFERENCE
        "RFC6388, Section 2.3.1."

    ::= { mplsMldpFecEntry 6 }

mplsMldpFecOpaqueTransitSourceOrBidirAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of this object is the type of the
        Internet address. The value of this object,
        decides how the value of the mplsMldpFecOpaqueTransitSourceOrBidir
        Addr
        object is interpreted."
    REFERENCE
        "RFC6826, Section 3.1."

```



```
::= { mplsMldpFecEntry 7 }
```

```
mplsMldpFecOpaqueTransitSourceOrBidirAddr OBJECT-TYPE
```

```
SYNTAX      InetAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The value of this object is interpreted based
on the value of the mplsMldpFecOpaqueTransitSourceOrBidirAddrType
object. This is source node address for the mLDP inband LSP."
```

```
REFERENCE
```

```
"RFC6826, Section 3.1."
```

```
::= { mplsMldpFecEntry 8 }
```

```
mplsMldpFecOpaqueTransitGroupAddrType          OBJECT-TYPE
```

```
SYNTAX      InetAddressType
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The value of this object is the type of the
Internet address. The value of this object,
decides how the value of the mplsMldpFecOpaqueTransitGroupAddr
object is interpreted."
```

```
REFERENCE
```

```
"RFC6826, Section 3.2."
```

```
::= { mplsMldpFecEntry 9 }
```

```
mplsMldpFecOpaqueTransitGroupAddr OBJECT-TYPE
```

```
SYNTAX      InetAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The value of this object is interpreted based
on the value of the mplsMldpFecOpaqueTransitGroupAddrType
object. This is group node address for the mLDP inband LSP."
```

```
REFERENCE
```

```
"RFC6826, Section 3.2."
```

```
::= { mplsMldpFecEntry 10 }
```

```
mplsMldpFecAdminStatus OBJECT-TYPE
```

```
SYNTAX      INTEGER {
```



```

        up(1),          -- ready to pass data
        down(2)         -- out of service
    }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Indicates the admin status of this mLDP FEC."

DEFVAL { up }

 ::= { mplsMldpFecEntry 11 }

mplsMldpFecOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        up(1),          -- ready to pass data
        down(2)         -- out of service
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the actual operational status of this mLDP Fec."

 ::= { mplsMldpFecEntry 12 }

mplsMldpFecMoFrr OBJECT-TYPE
    SYNTAX      INTEGER {
        enable(1),
        disable(2)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object provides whether MoFRR enabled for this mLDP FEC.
        on this mLDP FEC. As mentioned in the section 3.2 of [MoFRR],
        When this is enabled, then mLDP may select two upstream sessions,
        one is priamry and other one is backup. The backup traffic is
        discarded when the primary upstream session is UP. When the
        primary upstream session goes down, the traffic from the backup
        upstream session will be forwarded to downsteam.
        "

 ::= { mplsMldpFecEntry 13 }

mplsMldpFecLsrState OBJECT-TYPE
    SYNTAX      INTEGER {
        egress(1),
        bud(2),
        transit(3),

```



```

        ingress(4)
    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates the role of FEC either egress, bud, transit or ingress"

    ::= { mplsMldpFecEntry 14 }

mplsMldpFecUpTime OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This values shows Fec UP time. This is time since mplsMldpFecOperStat
us is UP."

    ::= { mplsMldpFecEntry 15 }

-- MPLS mLDP LSP Branch Traffic Stats Table.

mplsMldpFecBranchStatsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MplsMldpFecBranchStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table provides mLDP Fec branch MPLS Traffic Stats
information."

    ::= { mplsMldpObjects 4 }

mplsMldpFecBranchStatsEntry OBJECT-TYPE
    SYNTAX          MplsMldpFecBranchStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in this table is created by the LSR for each
        downstream branch (out-segment) from this LSR for this mLDP
        LSP. Each downstream session may represent a single out-segment.

        Each entry in the table is indexed by the four identifiers
        of the mLDP LSP, and the out-segment that identifies the
        outgoing branch."

    INDEX          { mplsLdpEntityLdpId,
                    mplsLdpEntityIndex,
                    mplsLdpPeerLdpId,
                    mplsMldpFecBranchFecIndex,
                    mplsMldpFecBranchOutSegIndex

```



```

    }

    ::= { mplsMldpFecBranchStatsTable 1 }

MplsMldpFecBranchStatsEntry ::= SEQUENCE {
    mplsMldpFecBranchFecIndex          MplsIndexType,
    mplsMldpFecBranchOutSegIndex       MplsIndexType,
    mplsMldpFecBranchStatsPackets      Counter64,
    mplsMldpFecBranchStatsBytes        Counter64,
    mplsMldpFecBranchStatsDiscontinuityTime TimeStamp
}

mplsMldpFecBranchFecIndex          OBJECT-TYPE
    SYNTAX          MplsIndexType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This index identifies the mLDP FEC entry in the
        mplsMldpFecTable.  This is same as mplsMldpFecIndex."

    ::= { mplsMldpFecBranchStatsEntry 1 }

mplsMldpFecBranchOutSegIndex       OBJECT-TYPE
    SYNTAX          MplsIndexType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object identifies an outgoing branch from this mLDP LSP
        Its value is unique within the context of the mLDP LSP.

        This contains the same value as the mplsOutSegmentIndex in the
        MPLS-LSR-STD-MIBs mplsOutSegmentTable."

    ::= { mplsMldpFecBranchStatsEntry 2 }

mplsMldpFecBranchStatsPackets      OBJECT-TYPE
    SYNTAX          Counter64
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object represent the 64-bit value, which gives the number
        of packets forwarded by the mLDP LSP onto this branch.
        This object should be read in conjunction with
        mplsMldpFecBranchStatsDiscontinuityTime."

    ::= { mplsMldpFecBranchStatsEntry 3 }

```



```
mplsMldpFecBranchStatsBytes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This object represent the 64-bit value, which gives the number
        of bytes forwarded by the mLDP LSP onto this branch.
        This object should be read in conjunction with
        mplsMldpFecBranchStatsDiscontinuityTime."

    ::= { mplsMldpFecBranchStatsEntry 4 }

mplsMldpFecBranchStatsDiscontinuityTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion at which
        any one or more of this rows Counter32 or Counter64 objects
        experienced a discontinuity. If no such discontinuity has
        occurred since the last re-initialization of the local
        management subsystem, then this object contains a zero
        value."

    ::= { mplsMldpFecBranchStatsEntry 5 }

-- End of mplsMldpFecBranchStatsTable

-- MPLS mLDP LSP Upstream Session Table.

mplsMldpFecUpstreamSessTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MplsMldpFecUpstreamSessEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides mLDP Fec upstream Session information."

    ::= { mplsMldpObjects 5 }

mplsMldpFecUpstreamSessEntry OBJECT-TYPE
    SYNTAX      MplsMldpFecUpstreamSessEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in this table is created by the LSR for each
        upstream session (in-segment) from this LSR for this mLDP
        LSP. Each upstream session may represent a single in-segment."
```


Each entry in the table is indexed by the four identifiers of the mLDP LSP, and the in-segment that identifies the incoming traffic."

```

INDEX      { mplsLdpEntityLdpId,
              mplsLdpEntityIndex,
              mplsLdpPeerLdpId,
              mplsMldpFecUpstreamSessFecIndex,
              mplsMldpFecUpstreamSessInSegIndex
            }

```

```
 ::= { mplsMldpFecUpstreamSessTable 1 }
```

```

MplsmLdpFecUpstreamSessEntry ::= SEQUENCE {
    mplsMldpFecUpstreamSessFecIndex      MplsIndexType,
    mplsMldpFecUpstreamSessInSegIndex    MplsIndexType,
    mplsMldpFecUpstreamSessPrimary       INTEGER,
    mplsMldpFecUpstreamSessActive        INTEGER,
    mplsMldpFecUpstreamSessPackets       Counter64,
    mplsMldpFecUpstreamSessBytes         Counter64,
    mplsMldpFecUpstreamSessDiscontinuityTime TimeStamp
}

```

```

mplsMldpFecUpstreamSessFecIndex          OBJECT-TYPE
SYNTAX      MplsIndexType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This index identifies the mLDP FEC entry in the
     mplsMldpFecTable."

```

```
 ::= { mplsMldpFecUpstreamSessEntry 1 }
```

```

mplsMldpFecUpstreamSessInSegIndex        OBJECT-TYPE
SYNTAX      MplsIndexType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This object identifies an upstream session from this mLDP LSP
     Its value is unique within the context of the mLDP LSP.

     This contains the same value as the mplsInSegmentIndex in the
     MPLS-LSR-STD-MIBs mplsInSegmentTable."

```

```
 ::= { mplsMldpFecUpstreamSessEntry 2 }
```

```

mplsMldpFecUpstreamSessPrimary          OBJECT-TYPE

```



```

SYNTAX          INTEGER {
                    primary(1),
                    backup(2)
                  }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This indicated wether the received traffic from upstream is
    primary or backup. This is valid only if the MoFRR
    (mplsMldpFecMoFrr) is enabled on this FEC."

 ::= { mplsMldpFecUpstreamSessEntry 3 }

mplsMldpFecUpstreamSessActive OBJECT-TYPE
SYNTAX          INTEGER {
                    active(1),
                    inactive(2)
                  }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This indicates whether the upstream session is active, means the
    LSR programmed the forwarding engine to receive the traffic from
    this upstream session. This will be Inactive if the LSR is wating
    for MBB Ack."

 ::= { mplsMldpFecUpstreamSessEntry 4 }

mplsMldpFecUpstreamSessPackets OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object represent the 64-bit value, which gives the number
    of packets received by the mLDP LSP from this upstream
    session. This object should be read in conjunction with
    mplsMldpFecUpstreamSessDiscontinuityTime."

 ::= { mplsMldpFecUpstreamSessEntry 5 }

mplsMldpFecUpstreamSessBytes OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object represent the 64-bit value, which gives the number
    of bytes received by the mLDP LSP from this upstream
    session. This object should be read in conjunction with

```



```

        mplsMldpFecUpstreamSessDiscontinuityTime."

 ::= { mplsMldpFecUpstreamSessEntry 6 }

mplsMldpFecUpstreamSessDiscontinuityTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion at which
        any one or more of this rows Counter32 or Counter64 objects
        experienced a discontinuity. If no such discontinuity has
        occurred since the last re-initialization of the local
        management subsystem, then this object contains a zero
        value."
 ::= { mplsMldpFecUpstreamSessEntry 7 }

-- End of mplsMldpFecBranchStatsTable

-- MPLS mLDP Interface Traffic Stats Table.

mplsMldpInterfaceStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MplsMldpInterfaceStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides mLDP Traffic Stats on specified interface."

 ::= { mplsMldpObjects 6 }

mplsMldpInterfaceStatsEntry OBJECT-TYPE
    SYNTAX      MplsMldpInterfaceStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry in this table is created by the LSR for each
        downstream branch (out-segment) from this LSR for this mLDP
        LSP. Each downstream session may represent a single out-segment.

        Each entry in the table is indexed by the four identifiers
        of the mLDP LSP, and the out-segment that identifies the
        outgoing branch."

    INDEX      { mplsMldpInterfaceIndex
                }

 ::= { mplsMldpInterfaceStatsTable 1 }

```



```

MplsMldpInterfaceStatsEntry ::= SEQUENCE {
    mplsMldpInterfaceIndex          InterfaceIndex,
    mplsMldpInterfaceStatsSentPackets Counter64,
    mplsMldpInterfaceStatsSentBytes Counter64,
    mplsMldpInterfaceStatsRecvPackets Counter64,
    mplsMldpInterfaceStatsRecvBytes Counter64
}

mplsMldpInterfaceIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This index identifies the specific interface. "

    ::= { mplsMldpInterfaceStatsEntry 1 }

mplsMldpInterfaceStatsSentPackets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is 64 bit value, which gives the number of packets
        forwarded by all mLDP LSPs onto this interface."

    ::= { mplsMldpInterfaceStatsEntry 2 }

mplsMldpInterfaceStatsSentBytes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is 64 bit value, which gives the number of bytes
        forwarded by all mLDP LSPs onto this interface."

    ::= { mplsMldpInterfaceStatsEntry 3 }

mplsMldpInterfaceStatsRecvPackets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is 64 bit value, which gives the number of packets
        received by all mLDP LSPs from this interface."

    ::= { mplsMldpInterfaceStatsEntry 4 }

```



```
mplsMldpInterfaceStatsRecvBytes OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This is 64 bit value, which gives the number of bytes
        received by all mLDP LSPs from this interface."

    ::= { mplsMldpInterfaceStatsEntry 5 }

-- End of mplsMldpInterfaceStatsTable

-- Notifications.

mplsMldpFecUp NOTIFICATION-TYPE
    OBJECTS      {
        mplsMldpFecAdminStatus,
        mplsMldpFecOperStatus
    }
    STATUS       current
    DESCRIPTION
        "This notification is generated when a mplsMldpFecOperStatus
        object changes from down to up."

    ::= { mplsMldpNotifications 1 }

mplsMldpFecDown NOTIFICATION-TYPE
    OBJECTS      {
        mplsMldpFecAdminStatus,
        mplsMldpFecOperStatus
    }
    STATUS       current
    DESCRIPTION
        "This notification is generated when a mplsMldpFecOperStatus
        object changes from up to down."

    ::= { mplsMldpNotifications 2 }

-- End of notifications.
```

12. Security Considerations

This MIB module is useful for the configuration of certain objects and monitoring of mLDP LSPs.

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this

MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o mplsMldpFecTable
- o mplsLdpPeerCapabilityTable
- o mplsMldpSessionStatsTable
- o mplsMldpFecBranchStatsTable
- o mplsMldpFecUpstreamSessTable
- o mplsMldpInterfaceStatsTable
- o mplsMldpNumFecsConfigured
- o mplsMldpNumFecsActive
- o mplsMldpMbbTime

Above listed tables and objects show information about the mLDP LSPs, its route through the network, and its traffic statistics. Knowledge of this information could be used to compromise the network, or simply to breach confidentiality. If an Administrator does not want to reveal this information, these tables and objects should be considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations

MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

13. IANA Considerations

This is new MPLS MIB module, contained in this document and IANA is requested to assign an oid under the mplsStdMIB subtree to the MPLS-MDLP-STD-MIB module specified in this document.

14. Acknowledgments

The authors wish to thank Santosh Esale, Alia Atlas and Martin Ehlers for doing the detailed review. Thanks to Adrian Farrel and Raveendra Torvi for their input to this work and for many helpful suggestions.

15. References

15.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

- [RFC3811] Nadeau, T. and J. Cucchiara, "Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management", RFC 3811, June 2004.
- [RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)", RFC 3813, June 2004.
- [RFC3815] Cucchiara, J., Sjostrand, H., and J. Luciani, "Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)", RFC 3815, June 2004.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009.
- [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.
- [RFC6826] Wijnands, IJ., Eckert, T., Leymann, N., and M. Napierala, "Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6826, January 2013.
- [RFC7060] Napierala, M., Rosen, E., and IJ. Wijnands, "Using LDP Multipoint Extensions on Targeted LDP Sessions", RFC 7060, November 2013.
- [RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, November 2011.

15.2. Informative References

- [RFC2223] Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC 2223, October 1997.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC4181] Heard, C., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, September 2005.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [MoFRR] Karan, Filsfils, Farinacci, Leymann, Joerde, and Henderickx, "Multicast only Fast Re-Route", draft-ietf-rtgwg-mofrr-03.txt (work in progress), 2012.
- [MLDP_NODE_PROT] Wijnands, Rosen, Raza, Tantsura, Leymann, and Zhao, "mLDP Node Protection", draft-ietf-mpls-mldp-node-protection-00.txt (work in progress), 2013.

Appendix A. Change Log

Appendix B. Open Issues

Authors' Addresses

Kishore Tiruveedhula (editor)
Juniper Networks
10 Technology Park Drive
Westford MA 01886
USA

Phone: +1 9785898861
EMail: kishoret@juniper.net

Uwe Joerde
Deutsche Telekom
Dahlweg 100
Munster 48153
Germany

EMail: Uwe.Joerde@telekom.de

Arvind Venkateswaran
Cisco Systems
510 McCarthy Blvd
Milpitas CA 95035
USA

EMail: arvvenka@cisco.com

MPLS Working Group
Internet-Draft
Intended status: Informational
Expires: July 28, 2014

Tarek Saad
Rakesh Gandhi
Zafar Ali
Cisco Systems, Inc.
Robert H. Venator
Defense Information Systems Agency
Yuji Kamite
NTT Communications Corporation
January 24, 2014

Reoptimization of Point-to-Multipoint Traffic Engineering
Loosely Routed LSPs

draft-tsaad-mpls-p2mp-loose-path-reopt-00.txt

Abstract

This document defines signaling extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for reoptimizing loosely routed point-to-multipoint (P2MP) Traffic Engineered (TE) Label Switched Path (LSP) in an Multi-Protocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks.

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."

Copyright Notice

Copyright (c) 2014 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
2. Terminology	4
2.1 Conventions used in this document	4
3. Procedure for Reoptimization of a Loosely Routed P2MP-TE LSP	4
4. RSVP Signaling Extensions	5
4.1. P2MP-TE Tree Re-evaluation Request	5
4.2. Preferable P2MP-TE Tree Exists Error sub-code	5
5. Compatibility	6
6. Security Considerations	6
7. IANA Considerations	6
7.1 P2MP-TE Tree Re-evaluation Request Flag	6
7.2 Preferable P2MP-TE Tree Exists sub-code	7
8. Acknowledgments	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Author's Addresses	8

1. Introduction

This document defines RSVP signaling extensions for the reoptimization of loosely routed point-to-multipoint (P2MP) MPLS and GMPLS (Generalized Multiprotocol Label Switching) Traffic Engineered (TE) Label Switched Path (LSP).

A P2MP-TE LSP is comprised of one or more source-to-leaf (S2L) sub-LSPs. A loosely routed P2MP-TE S2L sub-LSP is defined as one whose path does not contain the full explicit route identifying each node along the path to the egress at the time of its signaling by the ingress node. Such an S2L sub-LSP is signaled with no Explicit Route Object (ERO), or with an ERO that contains at least one loose hop, or with an ERO that contains an abstract node that is not a simple abstract node (that is, an abstract node that identifies more than one node).

[RFC4736] defines RSVP signaling extensions for reoptimizing loosely routed P2P TE LSP(s). Specifically, an ingress node sends a "path re-evaluation request" to a border node by setting a flag (0x20) in SESSION_ATTRIBUTES object in the Path message. A border node sends a PathErr code 25 (notify error defined in [RFC3209]) with sub-code 6 to indicate "preferable path exists" to the ingress node. The ingress node upon receiving this PathErr initiates reoptimization of the LSP.

As per [RFC4875], an ingress node may reoptimize the entire P2MP-TE LSP by resignaling all its S2L sub-LSP(s) or may reoptimize individual S2L sub-LSP(s) i.e. individual destination(s).

[RFC4736] does not define signaling extensions specific for reoptimizing entire P2MP-TE LSP tree. Mechanisms defined in [RFC4736] can be used for signaling the reoptimization of individual S2L sub-LSP(s). However, to use [RFC4736] mechanisms for reoptimizing an entire P2MP-TE LSP tree, an ingress node needs to send the query on all (typically 100s of) S2L sub-LSPs and a border node needs to notify PathErrs for all S2L sub-LSPs. In addition, a border node has to accumulate the received queries on all S2L sub-LSPs (using a wait timer) and interpret them as a reoptimization request for the P2MP-TE LSP tree. Furthermore, when the ingress node gradually receives unsolicited PathErr(s) notifications for individual S2L sub-LSP(s), it may prematurely start reoptimizing these sub-set of sub-LSPs. However, as mentioned in [RFC4875] Section 14.2, such reoptimization procedure may result in data duplication that can be avoided if the entire P2MP-TE LSP tree is reoptimized, especially if the ingress node eventually receives PathErr(s) notifications for all S2L sub-LSP(s) of the P2MP-TE LSP tree. In such cases, the ingress node may have to heuristically determine when to perform P2MP-TE LSP tree reoptimization or per S2L sub-LSP reoptimization, for example, to

wait long enough time to accumulate all PathErr(s) to be received. Such methods may produce undesired results that can be avoided by the proposed RSVP signaling extensions in this draft.

This document defines required RSVP signaling extensions to query and notify for reoptimizing loosely routed P2MP-TE LSP tree.

2. Terminology

ABR: Area Border Router.

ERO: Explicit Route Object.

TE LSP: Traffic Engineering Label Switched Path.

TE LSP ingress: head/source of the TE LSP.

TE LSP egress: tail/destination of the TE LSP.

S2L: Source-to-leaf.

Interior Gateway Protocol Area (IGP Area): OSPF Area or IS-IS level.

Inter-area TE LSP: A TE LSP whose path transits across at least two different IGP areas.

Inter-AS MPLS TE LSP: A TE LSP whose path transits across at least two different Autonomous Systems (ASes) or sub-ASes (BGP confederations).

2.1 Conventions used in this document

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 reader is assumed to be familiar with the terminology in [RFC4875] and [RFC4736].

3. Procedure for Reoptimization of a Loosely Routed P2MP-TE LSP

As per [RFC4875], an ingress node may prefer to reoptimize the entire P2MP-TE LSP by resignaling all its S2L sub-LSP(s) (Section 14.1, "Make-before-Break") or reoptimize individual S2L sub-LSP(s) i.e. individual destination(s) (Section 14.2 "Sub-Group-Based Re-Optimization").

Procedures defined in [RFC4736] are used by an ingress node to reoptimize the S2L sub-LSP individually.

To reoptimize entire P2MP-TE LSP tree, in order to query border nodes to check if a preferable P2MP-TE LSP tree exists, an ingress node sends a Path message with "P2MP-TE Tree Re-evaluation Request" defined in this document.

A border node receiving the "P2MP-TE Tree Re-evaluation Request" checks for a preferable P2MP-TE LSP tree by re-evaluating loosely expanded paths for all S2L sub-LSP(s) of the P2MP-TE LSP. If a preferable P2MP-TE LSP tree is found, the border node immediately sends an RSVP PathErr to the ingress node with Error code 25 (Notify defined in [RFC3209] and Error sub-code defined in this document "Preferable P2MP-TE Tree Exists". At this point, the border node does not propagate this bit in subsequent RSVP Path messages sent downstream for the re-evaluated TE LSP. The sending of an RSVP PathErr Notify message "Preferable P2MP-TE Tree Exists" to the ingress node will notify the ingress node of the existence of a preferable P2MP-TE LSP tree. If no preferable path can be found, the recommended mode is for the border node to relay the request (by setting the "P2MP-TE Tree Re-evaluation Request" bit in the LSP_ATTRIBUTES TLV of RSVP path message sent downstream).

A border node MAY also send "Preferable P2MP-TE Tree Exists" with PathErr code 25 to the ingress node to reoptimize the entire P2MP-TE LSP tree with an unsolicited PathErr message.

4. RSVP Signaling Extensions

4.1. P2MP-TE Tree Re-evaluation Request

In order to query border nodes to check if a preferable P2MP-TE LSP tree exists, a new flag is defined in Attributes Flags TLV of the LSP_ATTRIBUTES object [RFC5420] as follows:

Bit Number (to be assigned by IANA): P2MP-TE Tree Re-evaluation Request flag

The "P2MP-TE Tree Re-evaluation Request" flag is meaningful in a Path message of an S2L sub-LSP and is inserted by the ingress node.

4.2. Preferable P2MP-TE Tree Exists Error sub-code

In order to indicate to an ingress node that a preferable P2MP-TE LSP tree is available, following new sub-code for PathErr code 25 (Notify Error) [RFC3209] is defined:

Sub-code (to be assigned by IANA): Preferable P2MP-TE Tree Exists sub-code

When a preferable P2MP-TE LSP tree is found, the border node MUST send "Preferable P2MP-TE Tree Exists" PathErr to the ingress node in order to reoptimize the entire P2MP-TE LSP.

5. Compatibility

The LSP_ATTRIBUTES TLV has been defined in [RFC5420] with class numbers in the form 11bbbbbb, which ensures compatibility with non-supporting nodes. Per [RFC2205], nodes not supporting this extension will ignore the new flag defined in this document but forward it, unexamined and unmodified, in all messages resulting from this message.

6. Security Considerations

This document does not introduce any additional security issues above those identified in [RFC3209] and [RFC4875].

7. IANA Considerations

IANA maintains a name space for RSVP-TE TE parameters "Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters". From the registries in this name space "Attribute Flags" allocation of new flag is requested (sections 4.1).

IANA also maintains a name space for RSVP protocol parameters "Resource Reservation Protocol (RSVP) Parameters". From the sub-registry "Sub-Codes - 25 Notify Error" in registry "Error Codes and Globally-Defined Error Value Sub-Codes" allocation of a new error code is requested (section 4.2).

7.1 P2MP-TE Tree Re-evaluation Request Flag

The following new flag is defined for the Attributes Flags TLV in the LSP_ATTRIBUTES object [RFC5420]. The numeric values are to be assigned by IANA.

- o P2MP-TE Tree Re-evaluation Request Flag:
 - Bit Number: To be assigned by IANA.
 - Attribute flag carried in Path message: Yes
 - Attribute flag carried in Resv message: No

7.2 Preferable P2MP-TE Tree Exists sub-code

As defined in [RFC3209], the Error Code 25 in the ERROR SPEC object corresponds to a Notify Error PathErr. This document adds a new sub-code as follows for this PathErr:

- o Preferable P2MP-TE Tree Exists sub-code:
 - Sub-code for Notify PathErr code 25. To be assigned by IANA.

8. Acknowledgments

TBA.

8. References

8.1. Normative References

- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [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.
- [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.
- [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.

8.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4736] Vasseur, JP., Ikejiri, Y. and Zhang, R, "Reoptimization of Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Loosely Routed Label Switched Path (LSP)", RFC 4736, November 2006.

Author's Addresses

Tarek Saad
Cisco Systems

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems

Email: rgandhi@cisco.com

Zafar Ali
Cisco Systems

Email: zali@cisco.com

Robert H. Venator
Defense Information Systems Agency

Email: robert.h.venator.civ@mail.mil

Yuji Kamite
NTT Communications Corporation

Email: y.kamite@ntt.com