

Network
Internet-Draft
Updates: 5036 (if approved)
Intended status: Standards Track
Expires: March 6, 2020

M. Anush
T. Anupkumar
Ericsson
September 3, 2019

LDP behaviour on link-shut scenarios
draft-aa-mpls-ldp-link-shut-00

Abstract

This document is intended as clarification of LDP behaviour in link-down scenarios. Base LDP RFC5036 lacks sufficient clarity on what an LDP enabled node should be doing when a link down event is received, and the only LDP adjacency for an LDP peer is over this link. Different vendors have handled this scenario differently, with some immediately resetting tcp session with neighbor and some waiting for igp reconvergence instead of reacting directly to link events. With this document we intend to clarify the expected behaviour explicitly so that any interop issues can be avoided.

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 <https://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 6, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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	2
3. Problem Description	2
4. Solutions	3
5. Security Considerations	4
6. IANA Considerationss	4
7. Acknowledgments	4
8. Normative References	4
Authors' Addresses	4

1. Introduction

[RFC5036] details LDP specification and procedures to be followed by LDP implementations. However, for some scenarios like link down, the rfc isn't particularly clear as to what an implementation is supposed to do. This could lead to interop issues when routers from different vendors are part of the network. More details are given in the problem description section below. A possible solution is also suggested in the subsequent sections.

2. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

- o LDP: Label distribution protocol.
- o GR: Graceful-restart.

3. Problem Description

Consider the following topology:

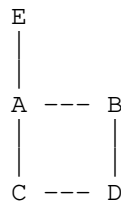


Figure 1: Example topology for LDP

All the nodes here are LDP enabled and also support GR. We have an lsp from C-E via C-A-E path (this is igp bestpath). IGP has also computed LFA backup for this primary-lsp via C-D-B-A-E path and we have LDP lfa backup as well (taking this path).

Now, let's bring down A-C link. Node A has detected link-shut event and since this link is the only adjacency to LDP-neighbor C, it resets the LDP session and sends shutdown to neighbour C.

At C, the link-down event is detected bit late and subsequently the IGP update is also delayed. Meanwhile, C has received shutdown from peer A, and it results in C flushing all labels received from A. Since the primary-label for C-E lsp is no longer available (from A), the lsp itself is deleted by LDP, as LDP can't be congruent with IGP. This LDP-lsp flap can in turn impact l3vpn/l2vpn traffic which are dependent on this LSP.

We can definitely reduce traffic-loss by running BFD and switching traffic to lfa backup in forwarding, but the intention above is to highlight that IGP updates and subsequently LDP updates would be asynchronous at nodes A and C, which may be more prominent if there are routers with different capabilities (and maybe from different vendors) in the network. So even if traffic has moved to lfa-backup lsp in forwarding, the primary-lsp itself could be deleted by the shutdown message (which is a fatal error).

4. Solutions

When a node has LDP adjacency to its neighbor (With GR [RFC3478] enabled on both the node and its neighbor) over a 'single' directly connected link and that link goes down, the node MAY reset the tcp session with neighbor. However, it MUST NOT send shutdown message, which flushes advertised labels at neighbor immediately.

The neighbor itself could have different backup mechanisms (ldp-lfa, rsvp-bypass etc) to ensure minimal traffic loss in forwarding for lsps having this node as active(primary)-path. Transmitting shutdown

message immediately could result in neighbor prematurely deleting LSPs instead of letting IGP reconverge.

Another approach could be to avoid reacting immediately to link down events. Instead, let hello timeout bringdown the session and update LSP-paths as soon as IGP reconverges.

Both approaches can help to avoid traffic loss by accounting for asynchronous ordering of events in LDP-peering routers.

5. Security Considerations

The security considerations described in RFC5036 apply to this document.

6. IANA Considerationss

7. Acknowledgments

.

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, DOI 10.17487/RFC3478, February 2003, <<https://www.rfc-editor.org/info/rfc3478>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Anush Mohan
Ericsson
Bangalore
India

Email: anush.mohan@ericsson.com

Anupkumar T
Ericsson
Bangalore
India

Email: anupkumar.t@ericsson.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 5, 2020

W. Cheng
China Mobile
X. Min
ZTE
T. Zhou
Huawei
X. Dong
FiberHome
Y. Peleg
Broadcom
November 2, 2019

Encapsulation For MPLS Performance Measurement with Alternate Marking
Method
draft-cheng-mpls-inband-pm-encapsulation-02

Abstract

This document defines the encapsulation for MPLS performance measurement with alternate marking method, which performs flow-based packet loss, delay, and jitter measurements on live traffic.

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 <https://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 May 5, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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	3
1.1.1. Terminology	3
1.1.2. Requirements Language	3
2. Flow-based PM Encapsulation in MPLS	3
2.1. Examples for Applying Flow-ID in a label stack	4
3. Procedures of Encapsulation, Look-up and Decapsulation	7
4. Procedures of Flow-ID allocation	8
5. Security Considerations	9
6. IANA Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Authors' Addresses	10

1. Introduction

[I-D.fioccola-spring-flow-label-alt-mark] describes how the alternate marking method can be used as the passive performance measurement method in an IPv6 domain, actually the alternate marking method can also be applied to an MPLS domain, and what's missed is the encapsulation for MPLS performance measurement with alternate marking method.

[RFC8372] discusses the desired capabilities for MPLS flow identification, in order to perform a better in-band performance monitoring of user data packets. Synonymous Flow Label (SFL), which is introduced in [I-D.ietf-mpls-sfl-framework], is identified as a method of accomplishing MPLS flow identification. This document employs a method, other than SFL, to accomplish MPLS flow identification. The method described in this document is simple and flexible, furthermore, it complies with the current MPLS forwarding paradigm.

The method described in this document is complementary to the SFL method, the former targets at hop-by-hop performance measurement, and the latter targets at end-to-end performance measurement, furthermore, the former supports the application scenario where Flow-

ID is applied to MPLS LSP and MPLS VPN synchronously, and the latter doesn't support this kind of application scenario.

This document defines the encapsulation for MPLS performance measurement with alternate marking method, which performs flow-based packet loss, delay, and jitter measurements on live traffic.

1.1. Conventions Used in This Document

1.1.1. Terminology

LSP: Label Switched Path

MPLS: Multi-Protocol Label Switching

NMS: Network Management System

PM: Performance Measurement

PW: PseudoWire

SFL: Synonymous Flow Label

TC: Traffic Class

TTL: Time to Live

VC: Virtual Channel

VPN: Virtual Private Network

1.1.2. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Flow-based PM Encapsulation in MPLS

Flow-based MPLS performance measurement encapsulation with alternate marking method has the following format:

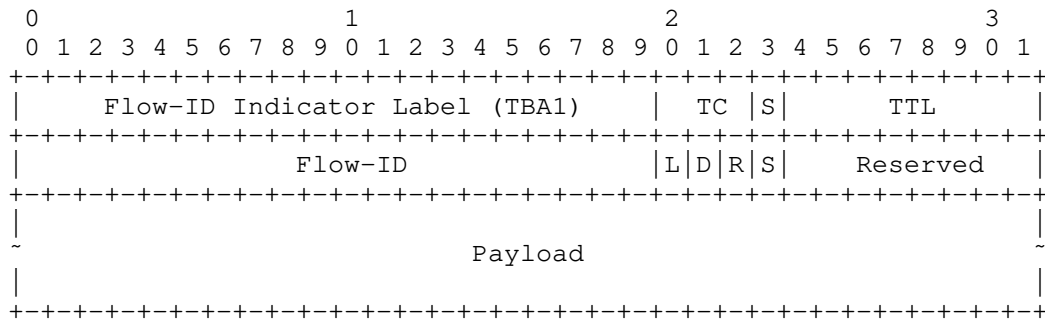


Figure 1: Flow-based PM Encapsulation in MPLS

Where Flow-ID Indicator Label is defined in this document as value TBA1, and the other fields related to the Flow-based PM encapsulation are defined as follows:

- o Flow-ID - an MPLS label value used as MPLS flow identification [RFC8372], it should be unique within the administrative domain. Flow-ID values can be allocated by an external NMS or a controller, based on measurement object instance such as LSP and PW. There is a one-to-one mapping between Flow-ID and flow. The specific method on how to allocate the Flow-ID values is described in Section 4. Note that the Flow-ID Label can be placed either at the bottom of the MPLS label stack or not, and the Flow-ID Indicator Label MAY appear multiple times in a label stack, which means more than one Flow-ID can be present within an MPLS label stack. Section 2.1 of this document provides several examples to illustrate how to apply Flow-ID in a label stack.
- o L and D - L(oss) bit and D(elay) bit are used for coloring the packets (called double-marking methodology), which is required by the alternate marking method.
- o R - R bit is reserved for future use and MUST be set to zero.
- o Reserved - one octet long field reserved for future use and MUST be set to zero.

2.1. Examples for Applying Flow-ID in a label stack

Three examples on different layout of Flow-ID label (4 octets) are illustrated as follows:

- (1) Layout of Flow-ID label when applied to MPLS LSP.

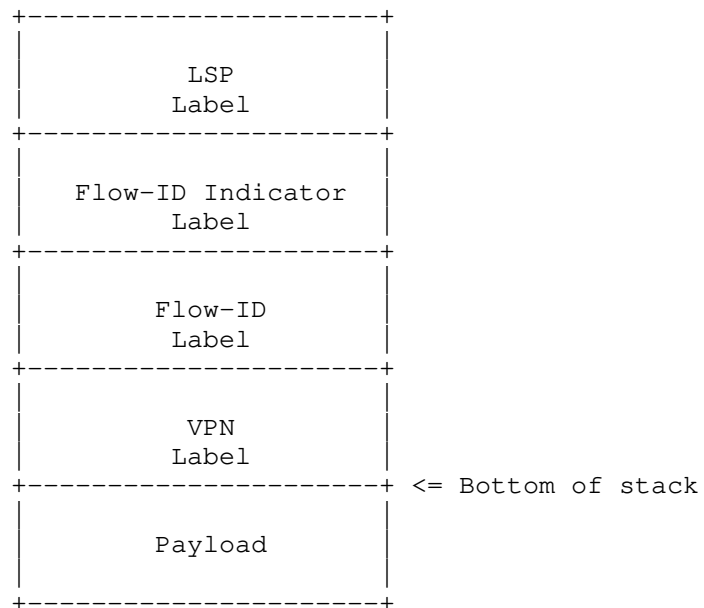


Figure 2: Applying Flow-ID to MPLS LSP

(2) Layout of Flow-ID label when applied to MPLS VPN traffic.

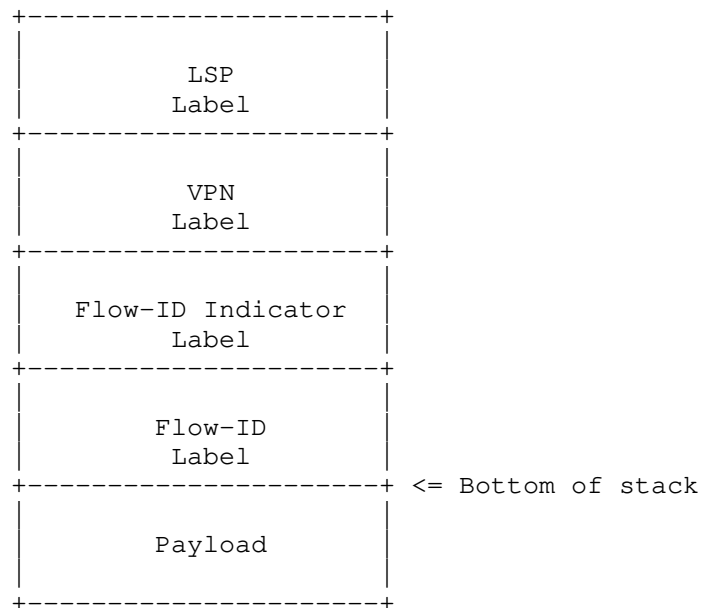


Figure 3: Applying Flow-ID to MPLS VPN

(3) Layout of Flow-ID label when applied to both MPLS LSP and MPLS VPN traffic.

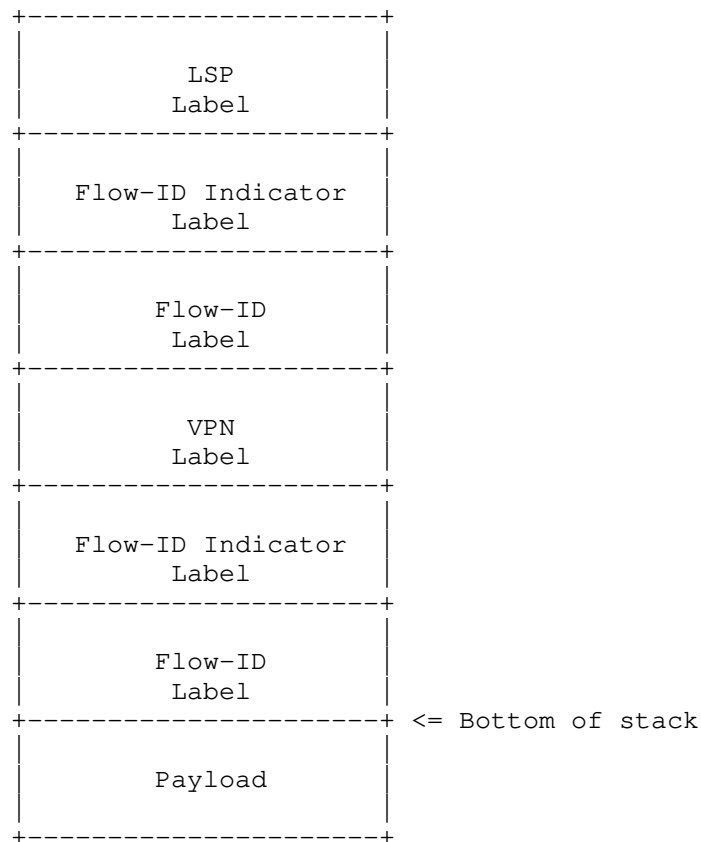


Figure 4: Applying Flow-ID to both MPLS LSP and MPLS VPN

Note that here VPN label can be MPLS PW label, MPLS Ethernet VPN label or MPLS IP VPN label, and it's also called VC label as defined in [RFC4026].

Also note that for this example the two Flow-ID values appearing in a label stack MUST be different, that is to say, Flow-ID applied to MPLS LSP and Flow-ID applied to MPLS VPN share the same value space.

3. Procedures of Encapsulation, Look-up and Decapsulation

The procedures for Flow-ID label encapsulation, look-up and decapsulation are summarized as follows:

- o The ingress node inserts the Flow-ID Indicator Label, alongside with the Flow-ID label, into the MPLS label stack. At the same

time, the ingress node sets the L bit and D bit, as needed by alternate-marking technique, and sets the Flow-ID value, as defined in this document.

- o The transit nodes look up the Flow-ID label with the help of the Flow-ID Indicator Label, and transmit the collected information to an external NMS or a controller, which includes the values of the block counters and the timestamps of the marked packets, along with the value of the Flow-ID, referring to the procedures of alternate marking method.
- o The egress node pops the Flow-ID Indicator Label, alongside with the Flow-ID label, from the MPLS label stack. This document doesn't introduce any new procedure regarding to the process of the decapsulated packet.

4. Procedures of Flow-ID allocation

There are two ways of allocating Flow-ID, one way is to allocate Flow-ID by manual trigger from the network operator, and the other way is to allocate Flow-ID by automatic trigger from the ingress node, details are as follows:

- o In the case of manual trigger, the network operator would manually input the characteristics (e.g. IP five tuples and IP DSCP) of the measured IP traffic flow, then the NMS or the controller would generate one or two Flow-IDs based on the input from the network operator, and provision the ingress node with the characteristics of the measured IP traffic flow and the corresponding allocated Flow-ID(s).
- o In the case of automatic trigger, the ingress node would identify the IP traffic flow entering the measured path, export the characteristics of the identified IP traffic flow to the NMS or the controller by IPFIX [RFC7011], then the NMS or the controller would generate one or two Flow-IDs based on the export from the ingress node, and provision the ingress node with the characteristics of the identified IP traffic flow and the corresponding allocated Flow-ID(s).

The policy pre-configured at the NMS or the controller decides whether one Flow-ID or two Flow-IDs would be generated. If the performance measurement on VPN traffic is enabled, then one Flow-ID applied to MPLS VPN would be generated; if the performance measurement on LSP tunnel is enabled, then one Flow-ID applied to MPLS LSP would be generated; if both of them are enabled, then two Flow-IDs respectively applied to MPLS VPN and MPLS LSP would be generated.

Whether using manual trigger or using automatic trigger, the NMS or the controller MUST guarantee every generated Flow-ID is unique within the administrative domain.

5. Security Considerations

This document does not introduce additional security requirements and mechanisms.

6. IANA Considerations

In the Special-Purpose MPLS Label Values registry defined in [SP-MPLS-Label], a new Special-Purpose MPLS Label Value for Flow-ID Indicator is requested from IANA as follows:

Special-Purpose MPLS Label Value	Description	Semantics Definition	Reference
TBA1	Flow-ID Indicator Label	Section 2	This Document

Table 1: New Special-Purpose MPLS Label Value for Flow-ID Indicator

7. Acknowledgements

The authors would like to acknowledge Greg Mirsky, Aihua Liu, Shuangping Zhan and Ming Ke for their careful review and very helpful comments.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [SP-MPLS-Label] "Special-Purpose MPLS Label Values", 2014, <<https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>>.

8.2. Informative References

- [I-D.fioccola-spring-flow-label-alt-mark]
Fioccola, G., Velde, G., Cociglio, M., and P. Muley,
"Using the IPv6 Flow Label for Performance Measurement
with Alternate Marking Method in Segment Routing", draft-
fioccola-spring-flow-label-alt-mark-01 (work in progress),
October 2017.
- [I-D.ietf-mpls-sfl-framework]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,
and G. Mirsky, "Synonymous Flow Label Framework", draft-
ietf-mpls-sfl-framework-06 (work in progress), October
2019.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual
Private Network (VPN) Terminology", RFC 4026,
DOI 10.17487/RFC4026, March 2005,
<<https://www.rfc-editor.org/info/rfc4026>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
"Specification of the IP Flow Information Export (IPFIX)
Protocol for the Exchange of Flow Information", STD 77,
RFC 7011, DOI 10.17487/RFC7011, September 2013,
<<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC8372] Bryant, S., Pignataro, C., Chen, M., Li, Z., and G.
Mirsky, "MPLS Flow Identification Considerations",
RFC 8372, DOI 10.17487/RFC8372, May 2018,
<<https://www.rfc-editor.org/info/rfc8372>>.

Authors' Addresses

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Xiao Min
ZTE
Nanjing
China

Email: xiao.min2@zte.com.cn

Tianran Zhou
Huawei
Beijing
China

Email: zhoutianran@huawei.com

Ximing Dong
FiberHome
Wuhan
China

Email: dxm@fiberhome.com

Yoav Peleg
Broadcom
USA

Email: yoav.peleg@broadcom.com

MPLS Working Group
Internet-Draft
Intended Status: Standards Track
Expires: April 20, 2020

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
M. Chen
Huawei
October 18, 2019

Performance Measurement for
Segment Routing Networks with MPLS Data Plane
draft-gandhi-mpls-rfc6374-sr-00

Abstract

Segment Routing (SR) leverages the source routing paradigm. RFC 6374 specifies protocol mechanisms to enable the efficient and accurate measurement of packet loss, one-way and two-way delay, as well as related metrics such as delay variation in MPLS networks using probe messages. This document utilizes these mechanisms for Performance Delay and Loss Measurements in Segment Routing (SR) networks with MPLS data plane (SR-MPLS), for both SR links and end-to-end SR Policies. In addition, this document defines Return Path TLV for two-way performance measurement and Block Number TLV for loss measurement.

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) 2019 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. Conventions Used in This Document	4
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
3. Overview	5
4. Probe Query and Response Packets	6
4.1. Probe Packet Header for SR-MPLS Policies	6
4.2. Probe Packet Header for SR-MPLS Links	6
4.3. Probe Response Message for SR-MPLS Links and Policies	7
4.3.1. One-way Measurement Mode	7
4.3.2. Two-way Measurement Mode	7
4.3.2.1. Return Path TLV	7
4.3.3. Loopback Measurement Mode	9
5. Performance Delay Measurement	9
5.1. Delay Measurement Message Format	9
5.2. Timestamps	9
6. Performance Loss Measurement	10
6.1. Loss Measurement Message Format	10
6.1.1. Block Number TLV	11
7. Performance Measurement for P2MP SR Policies	11
8. ECMP for SR-MPLS Policies	12
9. SR Link Extended TE Metrics Advertisements	12
10. Security Considerations	13
11. IANA Considerations	13
12. References	14
12.1. Normative References	14
12.2. Informative References	14
Acknowledgments	17
Contributors	17
Authors' Addresses	17

1. Introduction

Service provider's ability to satisfy Service Level Agreements (SLAs) depend on the ability to measure and monitor performance metrics for packet loss and one-way and two-way delay, as well as related metrics such as delay variation. The ability to monitor these performance metrics also provides operators with greater visibility into the performance characteristics of their networks, thereby facilitating planning, troubleshooting, and network performance evaluation.

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics in MPLS networks using probe messages. The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. However, mechanisms defined in [RFC6374] are more suitable for Segment Routing (SR) when using MPLS data plane (SR-MPLS). [RFC6374] also supports IEEE 1588 timestamps [IEEE1588] and "direct mode" Loss Measurement (LM), which are required in SR networks.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe replies over an UDP return path when receiving RFC 6374 based probe queries. These procedures can be used to send out-of-band PM replies for both SR-MPLS links and Policies [I-D.spring-segment-routing-policy] for one-way measurement.

This document utilizes the probe-based mechanisms defined in [RFC6374] for Performance Delay and Loss Measurements in SR networks with MPLS data plane, for both SR links and end-to-end SR Policies. In addition, this document defines Return Path TLV for two-way performance measurement and Block Number TLV for loss measurement. The Performance Measurements (PM) for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and can be advertised in the network using the routing protocol extensions.

2. Conventions Used in This Document

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

ACH: Associated Channel Header.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

G-ACh: Generic Associated Channel (G-ACh).

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

TC: Traffic Class.

TE: Traffic Engineering.

URO: UDP Return Object.

2.3. Reference Topology

In the reference topology shown in Figure 1, the sender node R1 initiates a performance measurement probe query and the responder node R5 sends a probe response for the query message received. The probe response is typically sent back to the sender node R1. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].

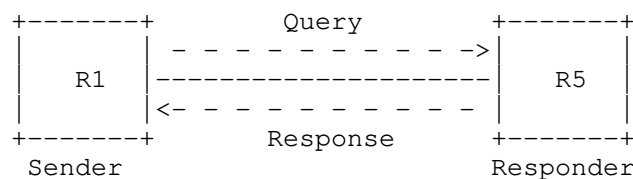


Figure 1: Reference Topology

3. Overview

One-way delay and two-way delay measurement procedure defined in Section 2.4 of [RFC6374] are used. Transmit and Receive packet loss measurement procedures defined in Section 2.2 and Section 2.6 of [RFC6374] are used. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss. For both links and end-to-end SR Policies, no PM session for delay or loss measurement is created on the responder node R5 [RFC6374].

For Performance Measurement, probe query and response messages are sent as following:

- o For Delay Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the links and SR Policies.
- o For Loss Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the responder node (incoming link or incoming SID needed since the

responder node does not have PM session state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.mpls-ioam-sr] are used to carry PM information in-band as part of the data traffic, and are outside the scope of this document.

4. Probe Query and Response Packets

4.1. Probe Packet Header for SR-MPLS Policies

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for an end-to-end measurement for SR Policy contains SR-MPLS label stack [I-D.spring-segment-routing-policy], with the G-ACh Label (GAL) at the bottom of the stack (with S=1). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 2.

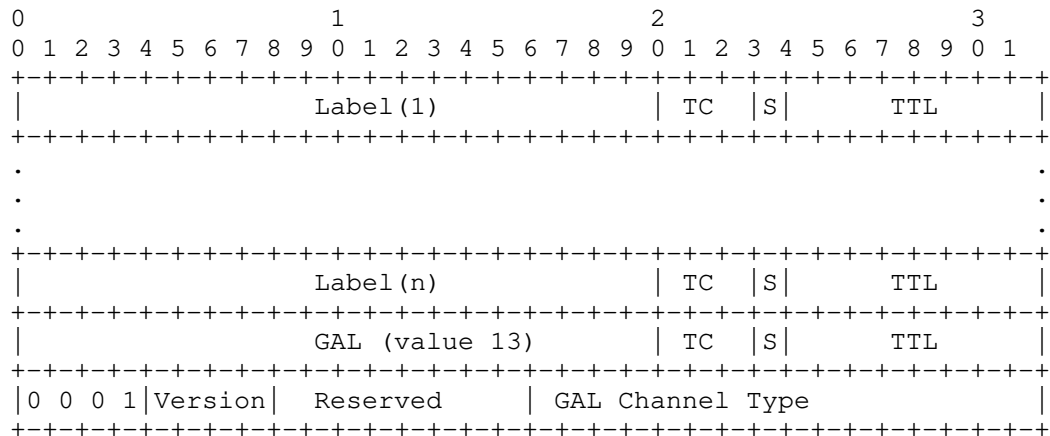


Figure 2: Probe Packet Header for an End-to-end SR-MPLS Policy

The SR-MPLS label stack can be empty (as shown in Figure 3) to indicate Implicit NULL label case.

4.2. Probe Packet Header for SR-MPLS Links

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for SR-MPLS links contains G-ACh Label (GAL)

(with S=1). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 3.

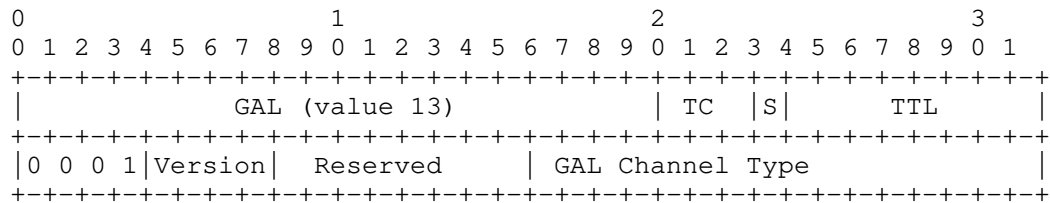


Figure 3: Probe Packet Header for an SR-MPLS Link

4.3. Probe Response Message for SR-MPLS Links and Policies

4.3.1. One-way Measurement Mode

In one-way performance measurement mode [RFC7679], the PM sender node can receive "out-of-band" probe replies by properly setting the UDP Return Object (URO) TLV in the probe query message. The URO TLV (Type=131) is defined in [RFC7876] and includes the UDP-Destination-Port and IP Address. In particular, if the sender sets its own IP address in the URO TLV, the probe response is sent back by the responder node to the sender node. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

4.3.2. Two-way Measurement Mode

In two-way performance measurement mode [RFC6374], when using a bidirectional path, the probe response message is sent back to the sender node on the congruent path of the data traffic on the reverse direction SR Link or SR Policy using a message with format similar to their probe query message. In this case, the "control code" in the probe query message is set to "in-band response requested".

A Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] of the forward SR-MPLS Policy can be used to find the reverse SR-MPLS Policy and to send back the probe response message for two-way measurement.

4.3.2.1. Return Path TLV

For two-way performance measurement, the responder node needs to send

the probe response message on a specific reverse path. This way the destination node does not require any additional SR Policy state. The sender node can request the responder node to send a response message back on a given reverse path (e.g. co-routed path for two-way measurement).

[RFC6374] defines DM and LM probe query messages that can include one or more optional TLVs. New TLV Type (TBA1) is defined in this document for Return Path to carry reverse path for probe response messages (in the payload of the message). The format of the Return Path TLV is shown in Figure 7A and 7B:

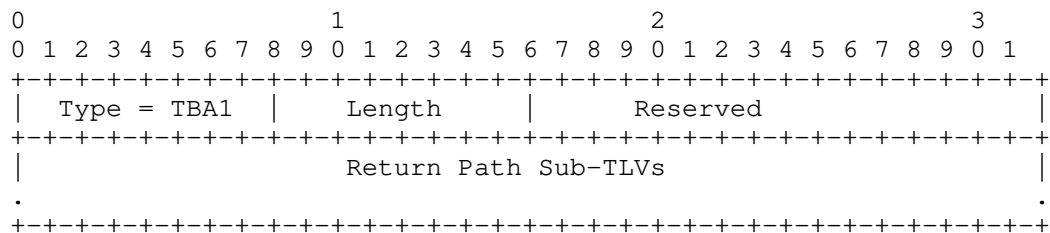


Figure 7A: Return Path TLV

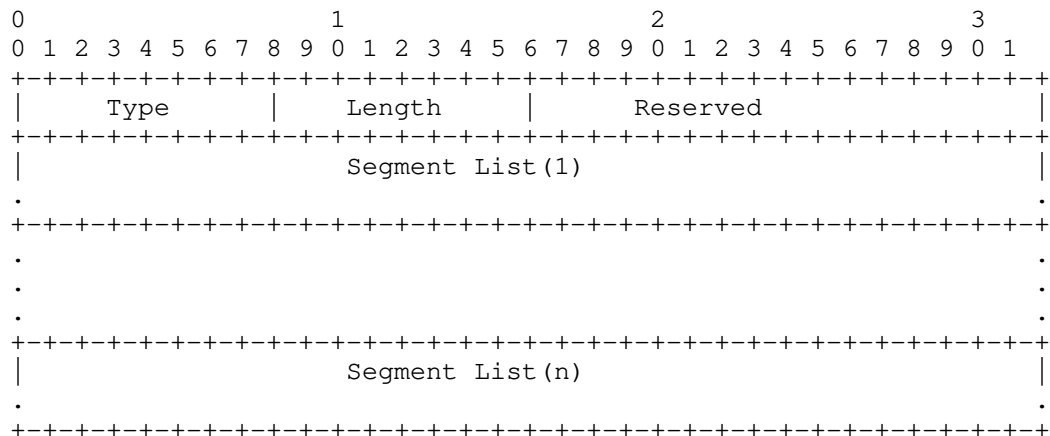


Figure 7B: Segment List Sub-TLV in Return Path TLV

The Segment List Sub-TLV in the Return Path TLV can be one of the following Types:

- o Type (value 1): Respond back on Incoming Interface (Layer-3 and

Layer-2) (Segment List is Empty)

- o Type (value 2): SR-MPLS Segment List (Label Stack) of the Reverse SR Path
- o Type (value 3): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

The Return Path TLV is optional. The PM sender node MUST only insert one Return Path TLV in the probe query message and the responder node MUST only process the first Return Path TLV in the probe query message and ignore other Return Path TLVs if present. The responder node MUST send probe response message back on the reverse path specified in the Return Path TLV and MUST NOT add Return Path TLV in the probe response message.

4.3.3. Loopback Measurement Mode

The Loopback measurement mode defined in Section 2.8 of [RFC6374] can be used to measure round-trip delay for a bidirectional SR Path. The probe query messages in this case carries the reverse SR Path label stack as part of the MPLS header. The GAL is still carried at the bottom of the label stack (with S=1). The responder node does not process the PM probe messages and generate response messages.

5. Performance Delay Measurement

5.1. Delay Measurement Message Format

As defined in [RFC6374], MPLS DM probe query and response messages use Associated Channel Header (ACH) (value 0x000C for delay measurement) [RFC6374], which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR-MPLS Policies, the same MPLS DM ACH value is used.

The DM message payload as defined in Section 3.2 of [RFC6374] is used for SR-MPLS delay measurement, for both SR links and end-to-end SR Policies.

5.2. Timestamps

The Section 3.4 of [RFC6374] defines timestamp format that can be used for delay measurement. The IEEE 1588 Precision Time Protocol (PTP) timestamp format [IEEE1588] is used by default as described in Appendix A of [RFC6374], preferred with hardware support. As an alternative, Network Time Protocol (NTP) timestamp format can also be

used [RFC6374].

Note that for one-way delay measurement mode, clock synchronization between the sender and responder nodes using the methods detailed in [RFC6374] is required. The two-way delay measurement mode and loopback measurement mode do not require clock synchronization between the sender and responder nodes.

6. Performance Loss Measurement

The LM protocol can perform two distinct kinds of loss measurement as described in Section 2.9.8 of [RFC6374].

- o In inferred mode, LM will measure the loss of specially generated test messages in order to infer the approximate data plane loss level. Inferred mode LM provides only approximate loss accounting.
- o In direct mode, LM will directly measure data plane packet loss. Direct mode LM provides perfect loss accounting, but may require hardware support.

For both of these modes of LM, Path Segment Identifier (PSID) [I-D.spring-mpls-path-segment] is used for accounting received traffic on the egress node of the SR-MPLS Policy as shown in Figure 4. Different values of PSID can be used to measure packet loss per SR-MPLS Policy, per Candidate Path or per Segment List of the SR Policy.

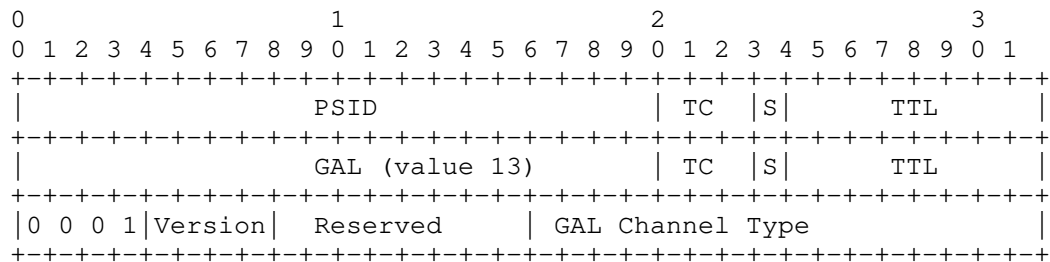


Figure 4: With Path Segment Identifier for SR-MPLS Policy

6.1. Loss Measurement Message Format

As defined in [RFC6374], MPLS LM probe query and response messages use Associated Channel Header (ACH) (value 0x000A for direct loss

measurement or value 0x000B for inferred loss measurement), which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR-MPLS Policies, the same MPLS LM ACH value is used.

The LM message payload as defined in Section 3.1 of [RFC6374] is used for SR-MPLS loss measurement, for both SR links and end-to-end SR Policies.

6.1.1. Block Number TLV

The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (or color) of the traffic counters carried by the probe query and response messages. Probe query and response messages specified in [RFC6374] for Loss Measurement do not define any means to carry the Block Number.

[RFC6374] defines probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA2) is defined in this document to carry Block Number (16-bit) for the traffic counters in the probe query and response messages for loss measurement. The format of the Block Number TLV is shown in Figure 5:

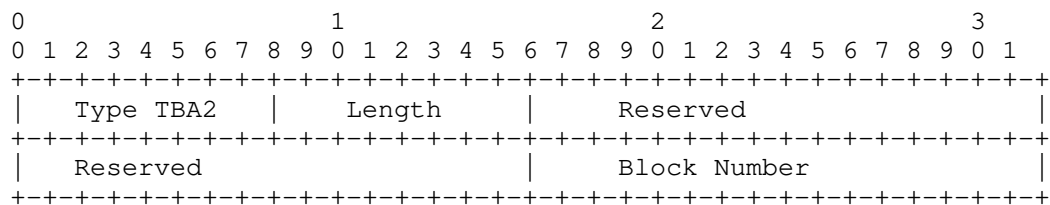


Figure 5: Block Number TLV

The Block Number TLV is optional. The PM sender node SHOULD only insert one Block Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Block Number TLV from the probe query messages and ignore other Block Number TLVs if present. In both probe query and response messages, the counters MUST belong to the same Block Number.

7. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement reviewed in this document for Point-to-Point (P2P) SR-MPLS Policies [I-D.spring-segment-routing-policy] are also equally applicable to the Point-to-Multipoint (P2MP) SR-MPLS Policies [I-D.spring-sr-p2mp-policy] as following:

- o The sender root node sends probe query messages using the either Spray P2MP segment or TreeSID P2MP segment defined in [I-D.spring-sr-p2mp-policy] over the P2MP SR Policy as shown in Figure 6.
- o Each responder leaf node adds the "Source Address" TLV (Type 130) [RFC6374] with its IP address in the probe response messages. This TLV allows the sender root node to identify the responder leaf nodes of the P2MP SR Policy.
- o The P2MP root node measures the end-to-end delay and loss performance for each P2MP leaf node.

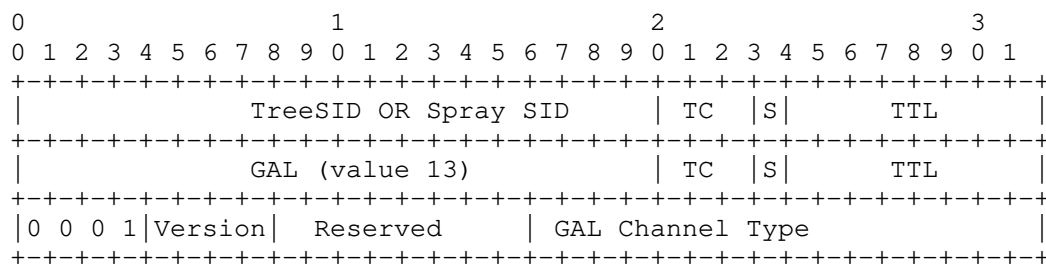


Figure 6: With P2MP Segment Identifier for SR-MPLS Policy

8. ECMP for SR-MPLS Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. For SR-MPLS Policy, entropy label [RFC6790] can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the ECMP path taken by them.

9. SR Link Extended TE Metrics Advertisements

The extended TE metrics for SR link delay and loss computed using the performance measurement procedures reviewed in this document can be advertised in the routing domain as follows:

- o For OSPF, ISIS, and BGP-LS, protocol extensions defined in [RFC7471], [RFC8570], and [RFC8571] are used, respectively for advertising the extended TE link metrics in the network.
- o The extended TE link delay metrics advertised are minimum-delay, maximum-delay, average-delay, and delay-variance for one-way.
- o The delay-variance metric is computed as specified in Section 4.2 of [RFC5481].
- o The one-way delay metrics can be computed using two-way delay measurement or round-trip delay measurement from loopback mode by dividing the measured delay values by 2.
- o The extended TE link loss metric advertised is one-way percentage packet loss.
- o Similarly, the extended TE link delay and loss metrics are advertised for Layer 2 bundle members in ISIS [I-D.lsr-ospf-l2bundles] and OSPF [I-D.isis-l2bundles] using the same mechanisms defined in [RFC8570] and [RFC7471], respectively.

10. Security Considerations

This document reviews the procedures for performance delay and loss measurement for SR-MPLS networks, for both links and end-to-end SR Policies using the mechanisms defined in [RFC6374] and [RFC7876]. This document does not introduce any additional security considerations other than those covered in [RFC6374], [RFC7471], [RFC8570], [RFC8571], and [RFC7876].

11. IANA Considerations

IANA is requested to allocate a value for the following Return Path TLV Type for RFC 6374 to be carried in PM probe query messages:

- o Type TBA1: Return Path TLV

IANA is requested to allocate the values for the following Sub-TLV Types for the Return Path TLV for RFC 6374.

- o Type (value 1): Respond back on Incoming Interface (Layer-3 and Layer-2) (Segment List is Empty)
- o Type (value 2): SR-MPLS Segment List (Label Stack) of the Reverse

SR Path

- o Type (value 3): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

IANA is also requested to allocate a value for the following Block Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for loss measurement:

- o Type TBA2: Block Number TLV

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks", RFC 6374, September 2011.
- [RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

12.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarez, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and

- L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [RFC7679] Almes, G., et al., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", RFC 7679, January 2016.
- [RFC7471] Giacalone, S., et al., "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.
- [RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.
- [RFC8402] Filts, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, July 2018.
- [RFC8570] Ginsberg, L. Ed., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, March 2019.
- [RFC8571] Ginsberg, L. Ed., et al., "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, March 2019.
- [I-D.spring-segment-routing-policy] Filts, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.pce-binding-label-sid] Filts, C., et al., "Carrying Binding Label/Segment-ID in PCE-based Networks", draft-ietf-pce-binding-label-sid, work in progress.
- [I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment, work in progress.
- [I-D.mpls-ioam-sr] Gandhi, R. Ed., et al., "Segment Routing with MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr, work in progress.
- [I-D.lsr-ospf-l2bundles] Talaulikar, K., et al., "Advertising L2 Bundle Member Link Attributes in OSPF", draft-ketant-lsr-ospf-l2bundles, work in progress.

[I-D.isis-l2bundles] Ginsberg, L., et al., "Advertising L2 Bundle
Member Link Attributes in IS-IS",
draft-ietf-isis-l2bundles, work in progress.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for the performance measurement in segment routing networks. The authors would like to thank Greg Mirsky for providing many useful comments and suggestions. The authors would also like to thank Stewart Bryant, Sam Aldrin, and Rajiv Asati for their review comments.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 23, 2020

R. Gandhi, Ed.
Z. Ali
C. Filsfils
F. Brockners
Cisco Systems, Inc.
B. Wen
V. Kozak
Comcast
August 22, 2019

Segment Routing with MPLS Data Plane Encapsulation
for In-situ OAM Data
draft-gandhi-spring-ioam-sr-mpls-02

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the data packet while the packet traverses a path between two nodes in the network. Segment Routing (SR) technology leverages the source routing paradigm. This document defines how IOAM data fields are transported with the Segment Routing with MPLS data plane (SR-MPLS) encapsulation. The procedures defined are also equally applicable to all other MPLS data plane encapsulations.

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) 2019 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. Conventions	3
2.1. Requirement Language	3
2.2. Abbreviations	3
3. IOAM Data Field Encapsulation in SR-MPLS Header	3
4. Procedure for Edge-to-Edge IOAM	5
4.1. Edge-to-Edge IOAM Indicator Labels	6
5. Procedure for Hop-by-Hop IOAM	7
6. Considerations for ECMP	7
7. Node Capability	7
8. IANA Considerations	7
9. Security Considerations	8
10. Acknowledgements	8
11. References	8
11.1. Normative References	8
11.2. Informative References	9
Contributors	9
Authors' Addresses	9

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than being sent within the probe packets specifically dedicated to OAM or Performance Measurement (PM). The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data], and can be used for various use-cases for OAM and PM.

Segment Routing (SR) technology leverages the source routing paradigm [I-D.ietf-spring-segment-routing-mpls]. A node steers a packet through a controlled set of instructions, called segments, by pre-pending the packet with an SR header. In the MPLS data plane,

the SR header is instantiated through a label stack.

This document defines how IOAM data fields are transported with the SR with MPLS data plane (SR-MPLS) encapsulation. The procedures defined are also equally applicable to all other MPLS data plane encapsulations.

2. Conventions

2.1. Requirement 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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

Abbreviations used in this document:

ECMP	Equal Cost Multi-Path
IOAM	In-situ Operations, Administration, and Maintenance
MPLS	Multiprotocol Label Switching
OAM	Operations, Administration, and Maintenance
PBT	Postcard Based Telemetry
PM	Performance Measurement
PoT	Proof-of-Transit
SR	Segment Routing
SR-MPLS	Segment Routing with MPLS Data plane

3. IOAM Data Field Encapsulation in SR-MPLS Header

SR-MPLS encapsulation is defined in [I-D.ietf-spring-segment-routing-mpls]. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. IOAM data fields are carried in the SR-MPLS header as shown in Figure 1 and Figure 2. More than one trace options can be present in the IOAM data fields. The Indicator Label is added at the bottom of the MPLS label stack (S

flag set to 1) to indicate the presence of the IOAM data field(s) in the MPLS header.

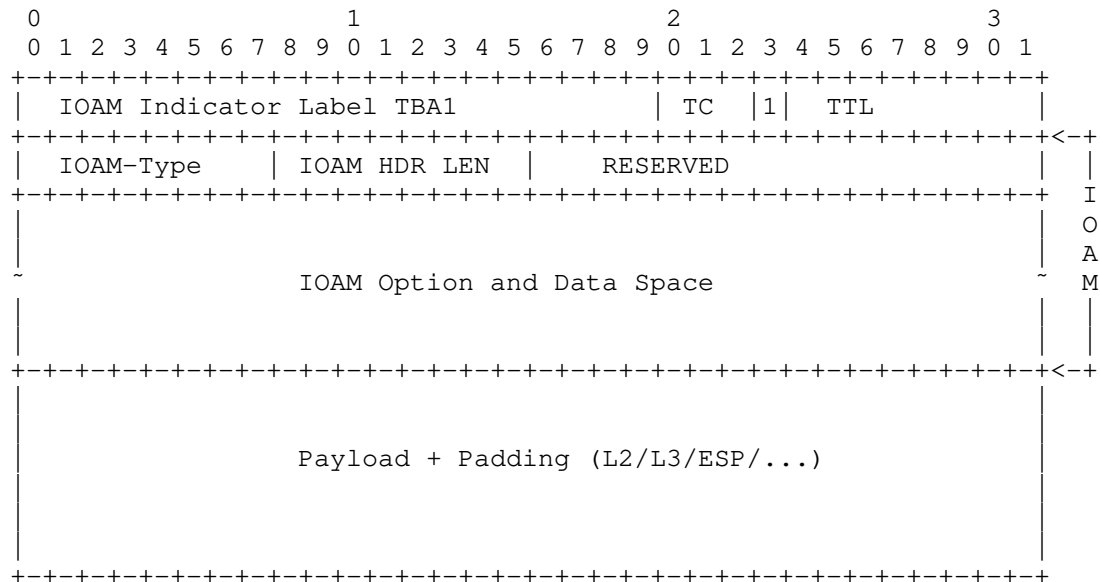
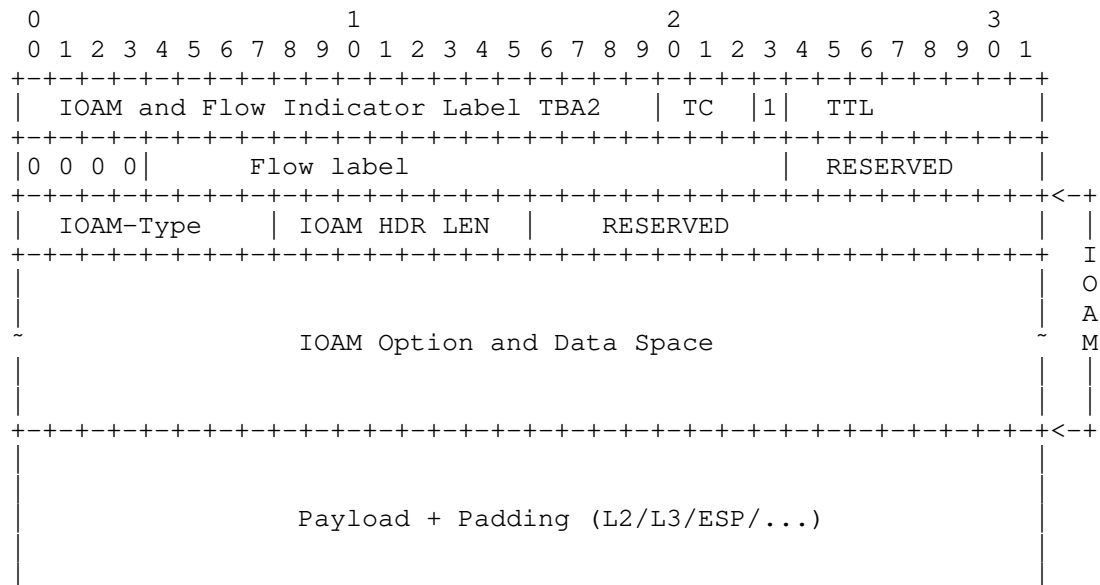


Figure 1: IOAM data encapsulation in SR-MPLS Header



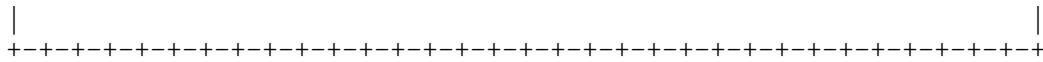


Figure 2: IOAM data encapsulation with Flow Label in SR-MPLS Header

Indicator Label and Flow Label as defined in this document.

The fields related to the encapsulation of IOAM data fields in the SR-MPLS header are defined as follows:

IOAM-Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

IOAM HDR LEN: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

RESERVED: 8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.

IOAM Option and Data Space: IOAM option header and data is present as defined by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

4. Procedure for Edge-to-Edge IOAM

This section summarizes the procedure for data encapsulation and decapsulation for IOAM Edge-to-Edge Option Type [I-D.ietf-ippm-ioam-data] in SR-MPLS header.

- o The encapsulating node inserts the IOAM Indicator Label or IOAM Flow Indicator Label with Flow Label and one or more IOAM data field(s) in the MPLS header. The procedure to generate the Flow Label is outside the scope of this document.
- o The decapsulating node "forwards and punts the timestamped copy" of the data packet including IOAM data fields when the node recognizes the IOAM Indicator Label and IOAM Flow Indicator Label. The copy of the data packet is punted to the slow path for OAM processing and is not necessarily punted to the control-plane. The receive timestamp is required by various OAM use-cases.
- o The decapsulating node processes the IOAM data field(s) using the procedures defined in [I-D.ietf-ippm-ioam-data]. An example of IOAM processing may be to export the data fields, send data fields via Telemetry, etc.

- o The decapsulating node also pops the Indicator Label and the IOAM data fields from the MPLS header.

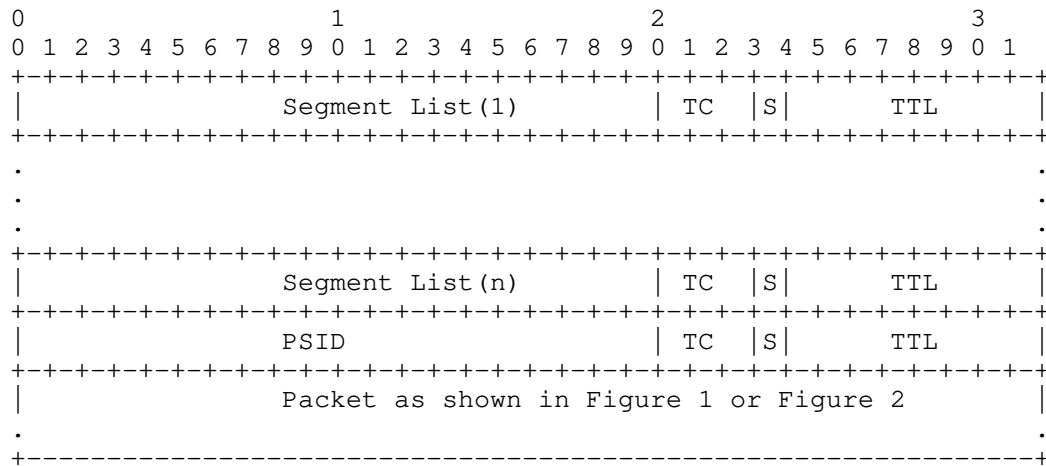


Figure 3: Data Packet over SR-MPLS Policy

4.1. Edge-to-Edge IOAM Indicator Labels

IOAM Indicator Label (value TBA1) and IOAM and Flow Indicator Label (value TBA2) are used to indicate the presence of the IOAM data field in the MPLS header.

The Indicator Label with value TBA2 is used to carry a second label underneath with protocol value 0000b and 20-bit Flow Label. The protocol value 0000b allows to avoid incorrect IP header based hashing over ECMP paths that uses the value 0x4 (for IPv4) and value 0x6 (for IPv6) [RFC4928]. The Flow Label identifies the traffic flow that can be used for IOAM purpose as well as for hashing over ECMP paths.

The IOAM Indicator Label and IOAM and Flow Indicator Label can be allocated using one of the following methods:

- o Labels assigned by IANA with value TBA1 and TBA2 from the Extended Special-Purpose MPLS Values [mpls-spl-terminology].
- o Labels allocated by a controller from the global table of the decapsulating node. The controller provisions the label on both encapsulating and decapsulating nodes.
- o Labels allocated by the decapsulating node. The signaling

extension for this is outside the scope of this document.

5. Procedure for Hop-by-Hop IOAM

The hop-by-hop IOAM includes IOAM-Types IOAM Pre-allocated Trace Option Type, IOAM Incremental Trace Option Type and IOAM POT Option Type.

Different Indicator Labels (TBA3 and TBA4) are used for hop-by-hop IOAM.

The details for hop-by-hop IOAM will be added in a future version of the document.

6. Considerations for ECMP

The encapsulating node needs to make sure the IOAM data field does not start with a well known IP protocol value (e.g. 0x4 for IPv4 and 0x6 for IPv6) as it can alter the hashing function for ECMP that uses the IP header. This can be achieved by using the IOAM and Flow Indicator Label (value TBA2 and TBA4) that follows by protocol value 0000b. This approach is consistent with the use of utilizing 0000b as the first nibble after the MPLS label stack, as described in [RFC4928] [RFC4385].

Note that the hashing function for ECMP that uses the labels from the MPLS header may also now include the Indicator Label.

The entropy label can be used for hashing function for ECMP as defined in [RFC6790].

7. Node Capability

The decapsulating node that has to pop the Indicator Label, data fields, and perform the IOAM function may not be capable of supporting it. The encapsulating node needs to know if the decapsulating node can support the IOAM function. The signaling extension for this capability exchange is outside the scope of this document.

8. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see

<<https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>>). IANA is requested to allocate IOAM Indicator Label value and IOAM and Flow Indicator value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	E2E IOAM Indicator Label	This document
TBA2	E2E IOAM and Flow Indicator Label	This document
TBA3	HbH IOAM Indicator Label	This document
TBA4	HbH IOAM and Flow Indicator Label	This document

9. Security Considerations

The security considerations of SR-MPLS are discussed in [I-D.ietf-spring-segment-routing-mpls], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

10. Acknowledgements

The authors would like to thank Shwetha Bhandari and Vengada Prasad Govindan for the discussions on IOAM. The authors would also like to thank Tarek Saad, Loa Andersson and Cheng Li for providing many useful comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.
- [I-D.ietf-spring-segment-routing-mpls] Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls, work in progress.
- [I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and Bernier, D., "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data, work in progress.

11.2. Informative References

- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.
- [RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, June 2007.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [mpls-spl-terminology] L. Andersson, et al. "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology, work in progress.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Voitek Kozak
Comcast

Email: Voitek_Kozak@comcast.com

Routing area
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2020

S. Hegde
K. Arora
M. Srivastava
S. Ninan
Juniper Networks Inc.
October 31, 2019

Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR)
Egress Peer engineering Segment Identifiers (SIDs) with MPLS Data Planes
draft-hegde-mpls-spring-epe-oam-03

Abstract

Egress Peer Engineering is an application of Segment Routing to solve the problem of egress peer selection. The SR-based BGP-EPE solution allows a centralized (Software Defined Network, SDN) controller to program any egress peer. The EPE solution requires a node to program PeerNodeSID, PeerAdjSID, PeerSetSID as described in [I-D.ietf-spring-segment-routing-central-epe]. This document provides new sub-TLVs for EPE SIDs that would be used in Target stack TLV (Type 1) as defined in [RFC8029] for the EPE SIDs.

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 <https://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 May 3, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. FEC Definitions	3
2.1. PeerAdjSID Sub-TLV	3
2.2. PeerNodeSID Sub-TLV	4
2.3. PeerSetSID Sub-TLV	6
3. Security Considerations	9
4. IANA Considerations	9
5. Acknowledgments	9
6. References	9
6.1. Normative References	9
6.2. Informative References	10
Authors' Addresses	10

1. Introduction

Egress Peer Engineering (EPE) as defined in [I-D.ietf-spring-segment-routing-central-epe] is an effective mechanism to select the egress peer link based on different criteria. The EPE SIDs provide means to represent egress peer links. Many network deployments have built their networks consisting of multiple Autonomous Systems either for ease of operations or as a result of network mergers and acquisitions. The inter-AS links connecting the two Autonomous Systems could be traffic engineered using EPE-SIDs in this case as well. It is important to be able to validate the control plane to forwarding plane synchronization for these SIDs so that any anomaly can be detected easily by the operator.

This document provides Target FEC stack TLV definitions for EPE SIDs. Other procedures for mpls ping and traceroute as defined in [RFC8287] are applicable for EPE-SIDs as well.

2. FEC Definitions

As described in [RFC8287] sec 5, 3 new type of sub-TLVs for the Target FEC Stack TLV are defined for the Target FEC stack TLV corresponding to each label in the label stack. If a malformed FEC sub-TLV is received, then a return code of 1, "Malformed echo request received" as defined in [RFC8029] SHOULD be sent.

2.1. PeerAdjSID Sub-TLV

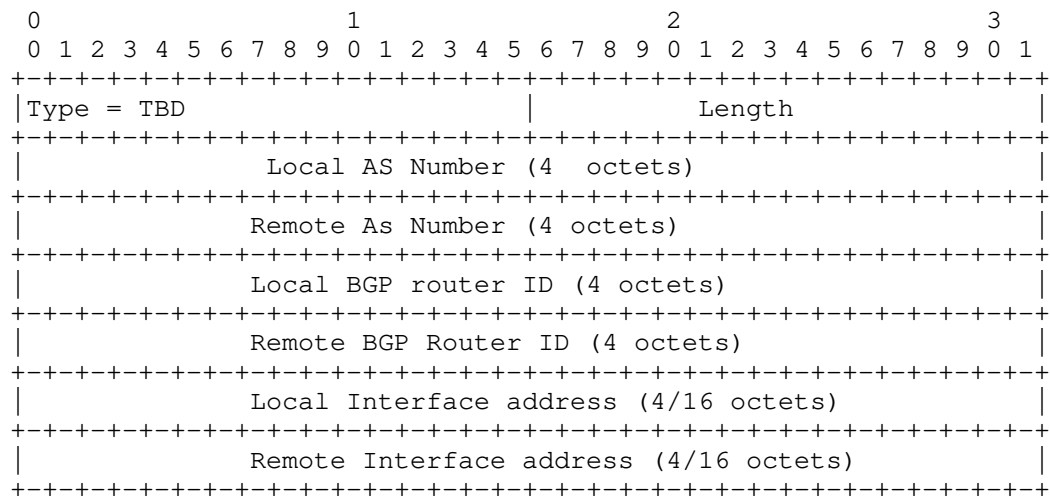


Figure 1: PeerAdjSID Sub-TLV

Type : TBD

Length : variable based on ipv4/ipv6 interface address

Local AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS to which PeerAdjSID advertising node belongs to.

Remote AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS of the remote node for which the PeerAdjSID is advertised.

Local BGP Router ID :

4 octet unsigned integer of the advertising node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

Remote BGP Router ID :

4 octet unsigned integer of the receiving node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

Local Interface Address :

In case of PeerAdjSID Local interface address corresponding to the PeerAdjSID should be specified in this field. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

Remote Interface Address :

In case of PeerAdjSID Remote interface address corresponding to the PeerAdjSID should be specified in this field. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

2.2. PeerNodeSID Sub-TLV

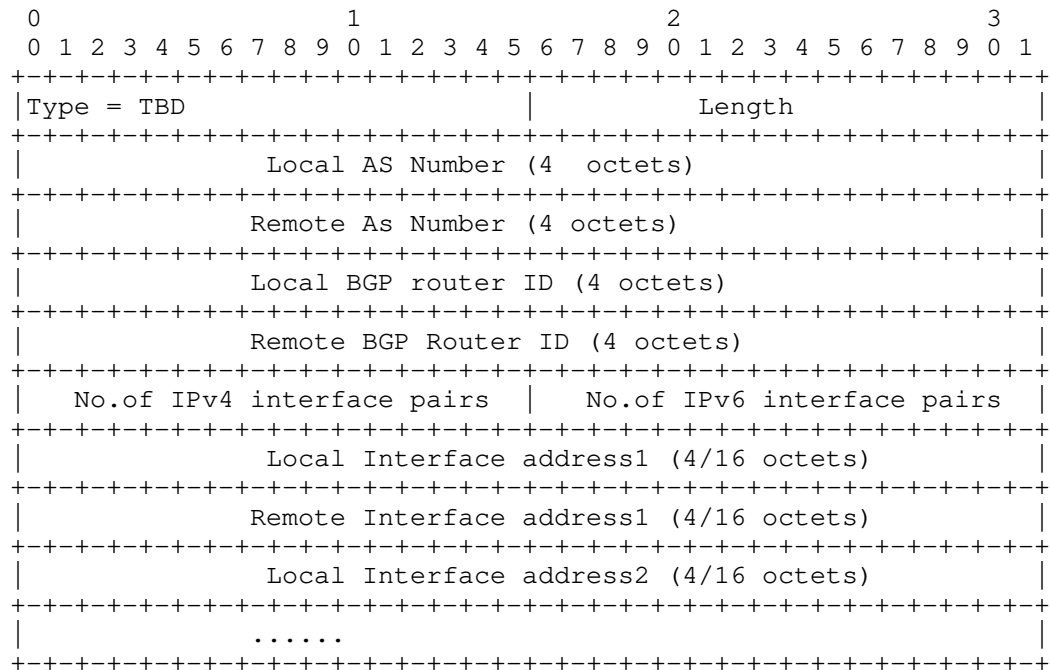


Figure 2: PeerNodeSID Sub-TLV

Type : TBD

Length : variable based on ipv4/ipv6 interface address

Local AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS to which PeerNodeSID advertising node belongs to.

Remote AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS of the remote node for which the PeerNodeSID is advertised.

Local BGP Router ID :

4 octet unsigned integer of the advertising node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

Remote BGP Router ID :

4 octet unsigned integer of the receiving node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

Number of IPv4 interface pairs:

Total number of IPV4 local and remote interface address pairs.

Number of IPv6 interface pairs:

Total number of IPV6 local and remote interface address pairs.

There can be multiple Layer 3 interfaces on which a peerNodeSID loadbalances the traffic. All such interfaces local/remote address MUST be included in the FEC.

When a PeerNodeSID load-balances over few interfaces with IPv4 only address and few interfaces with IPv6 address then the FEC definition should list all IPv4 address pairs together followed by IPv6 address pairs.

Local Interface Address :

In case of PeerNodeSID, the interface local address ipv4/ipv6 which corresponds to the PeerNodeSID MUST be specified. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

Remote Interface Address :

In case of PeerNodeSID, the interface remote address ipv4/ipv6 which corresponds to the PeerNodeSID MUST be specified. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

2.3. PeerSetSID Sub-TLV

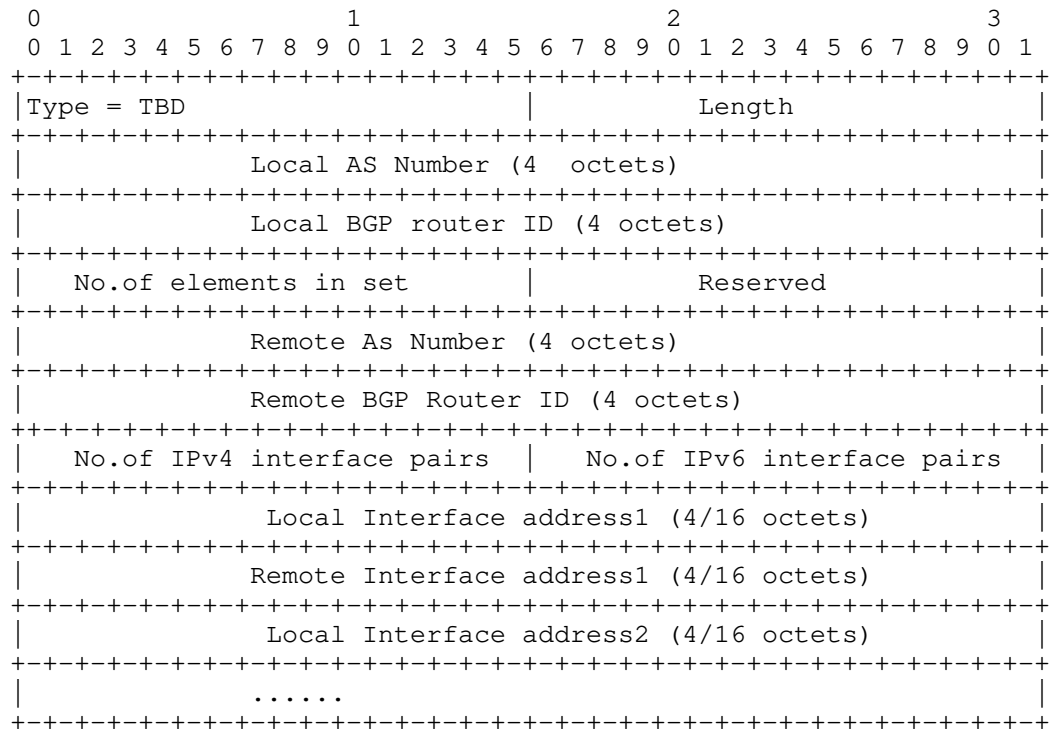


Figure 3: PeerSetSID Sub-TLV

Type : TBD

Length : variable based on ipv4/ipv6 interface address and number of elements in the set

Local AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS to which PeerSetSID advertising node belongs to.

Remote AS Number :

4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]. The AS number corresponds to the AS of the remote node for which the PeerSetSID is advertised.

Advertising BGP Router ID :

4 octet unsigned integer of the advertising node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

Receiving BGP Router ID :

4 octet unsigned integer of the receiving node representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

No.of elements in set:

Number of remote ASes, the set SID load-balances on.

PeerSetSID may be associated with a number of PeerNodeSIDs and PeerAdjSIDs. Link address details of all these SIDs should be included in the peerSetSID FEC so that the data-plane can be correctly verified on the remote node.

Number of IPv4 interface pairs:

Total number of IPV4 local and remote interface address pairs.

Number of IPv6 interface pairs:

Total number of IPV6 local and remote interface address pairs.

There can be multiple Layer 3 interfaces on which a peerNodeSID loadbalances the traffic. All such interfaces local/remote address MUST be included in the FEC.

When a PeerSetSID load-balances over few interfaces with IPv4 only address and few interfaces with IPv6 address then the Link address TLV should list all IPv4 address pairs together followed by IPv6 address pairs.

Local Interface Address :

In case of PeerNodeSID/PeerAdjSID, the interface local address ipv4/ipv6 which corresponds to the PeerNodeSID/PeerAdjSID MUST be specified. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

Remote Interface Address :

In case of PeerNodeSID/PeerAdjSID, the interface remote address ipv4/ipv6 which corresponds to the PeerNodeSID/PeerAdjSID MUST be

specified. For IPv4, this field is 4 octets; for IPv6, this field is 16 octets.

3. Security Considerations

The EPE SIDs are advertised for egress links for Egress Peer Engineering purposes or for inter-As links between co-operating ASes. When co-operating domains are involved, they can allow the packets arriving on trusted interfaces to reach the control plane and get processed. When EPE SIDs which are created for egress TE links where the neighbor AS is an independent entity, it may not allow packets arriving from external world to reach the control plane. In such deployments mpls OAM packets will be dropped by the neighboring AS.

4. IANA Considerations

New Target FEC stack sub-TLV from the "sub-TLVs for TLV types 1,16 and 21" subregistry of the "Multi-Protocol Label switching (MPLS) Label Switched Paths (LSPs) Ping parameters" registry

PeerAdjSID segment ID Sub-TLV : TBD

PeerNode segment ID Sub-TLV : TBD

PeerSetSID segment ID Sub-TLV : TBD

5. Acknowledgments

6. References

6.1. Normative References

- [I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", draft-ietf-spring-segment-routing-central-epe-10 (work in progress), December 2017.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.

6.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.

Authors' Addresses

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Kapil Arora
Juniper Networks Inc.

Email: kapilaro@juniper.net

Mukul Srivastava
Juniper Networks Inc.

Email: msri@juniper.net

Samson Ninan
Juniper Networks Inc.

Email: samsonn@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 15, 2020

T. Saad
Juniper Networks
K. Raza
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
September 12, 2019

A YANG Data Model for MPLS Base
draft-ietf-mpls-base-yang-11

Abstract

This document contains a specification of the MPLS base YANG model. The MPLS base YANG model serves as a base framework for configuring and managing an MPLS switching subsystem on an MPLS-enabled router. It is expected that other MPLS YANG models (e.g. MPLS LSP Static, LDP or RSVP-TE YANG models) will augment the MPLS base YANG model.

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 <https://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 15, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Terminology	3
1.2. Acronyms and Abbreviations	3
2. MPLS Base Model	3
2.1. Model Overview	3
2.2. Model Organization	4
2.3. Model Tree Diagram	5
2.4. Model YANG Module	7
3. IANA Considerations	15
4. Security Considerations	16
5. Acknowledgement	16
6. Contributors	16
7. References	17
7.1. Normative References	17
7.2. Informative References	18
Authors' Addresses	18

1. Introduction

A core routing data model is defined in [RFC8349], and it provides a basis for the development of data models for routing protocols. The MPLS base model augments core routing data model with additional data specific to MPLS technology as described in the MPLS architecture document [RFC3031].

The MPLS base model serves as a basis for future development of MPLS data models covering more-sophisticated MPLS feature(s) and sub-system(s). The main purpose is to provide essential building blocks for the more-complicated data models involving different control-plane protocols, and advanced MPLS functions.

To this end, it is expected that the MPLS base data model will be augmented by a number of other modules developed at IETF (e.g. by TEAS and MPLS working groups).

The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

1.1. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology for describing YANG data models is found in [RFC7950].

1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

RIB: Routing Information Base

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

ILM: Incoming Label Map

2. MPLS Base Model

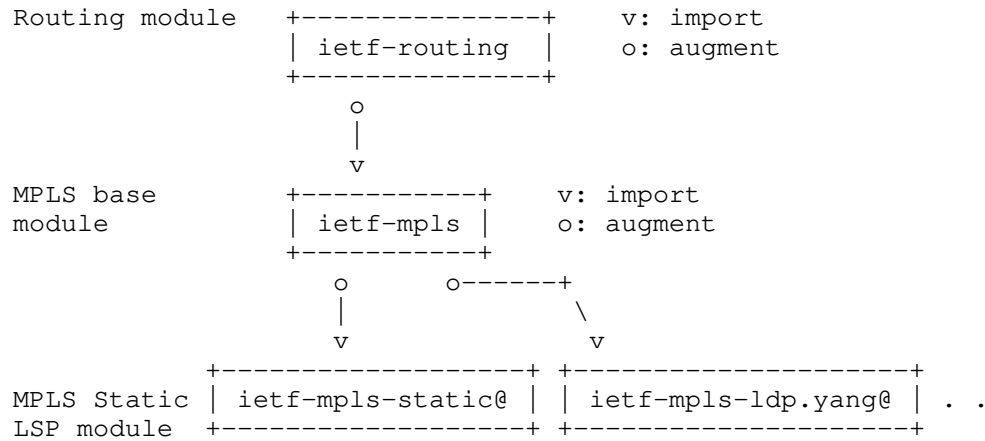
This document describes the ietf-mpls YANG module that provides base components of the MPLS data model. It is expected that other MPLS YANG modules will augment the ietf-mpls base module for other MPLS extension to provision LSP(s) (e.g. MPLS Static, MPLS LDP or MPLS RSVP-TE LSP(s)).

2.1. Model Overview

This document defines a mechanism to model MPLS labeled routes as an augmentation of the routing RIB data model defined in [RFC8349] for IP prefix routes that are MPLS labeled.

The other MPLS route(s) that are non-IP prefix routes are modelled by introducing a new "mpls" address-family RIB as per recommendation .

2.2. Model Organization



@: not in this document, shown for illustration only

Figure 1: Relationship between MPLS modules

ietf-mpls module contains the following high-level types and groupings:

label-block-alloc-mode:

A base YANG identity for supported label block allocation mode(s).

mpls-operations-type:

An enumeration type that represents support possible MPLS operation types (impose-and-forward, pop-and-forward, pop-impose-and-forward, and pop-and-lookup)

nhlfe-role:

An enumeration type that represents the role of the NHLFE entry.

nhlfe-single-contents:

A YANG grouping that describes single NHLFE and its associated parameters as described in the MPLS architecture document [RFC3031].

nhlfe-multiple-contents:

A YANG grouping that describes a set of NHLFE(s) and their associated parameters as described in the MPLS architecture document [RFC3031].

interface-mpls-properties:

A YANG grouping that describes the properties of an MPLS interface on a device.

interfaces-mpls:

A YANG grouping that describes the list of MPLS enabled interfaces on a device.

label-block-properties:

A YANG grouping that describes the properties of an MPLS label block.

label-blocks:

A YANG grouping that describes the list of MPLS enabled interfaces on a device.

2.3. Model Tree Diagram

The MPLS base tree diagram that follows the notation defined in [RFC8340] is shown in Figure 2.

```

module: ietf-mpls
  augment /rt:routing:
    +--rw mpls
      +--rw ttl-propagate?  boolean
      +--rw label-blocks
        +--rw label-block* [index]
          +--rw index                string
          +--rw start-label?         rt-types:mpls-label
          +--rw end-label?           rt-types:mpls-label
          +--rw block-allocation-mode? identityref
          +--ro free-labels-count?   yang:counter32
          +--ro inuse-labels-count?  yang:counter32
      +--rw interface* [name]
        +--rw name                  if:interface-ref
        +--rw enabled?              boolean
        +--rw mtu?                  uint32
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
    +--ro mpls-enabled?            boolean
    +--ro local-label?             rt-types:mpls-label

```

```

augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
  /rt:next-hop-options/rt:simple-next-hop:
  +--ro mpls-label-stack
    +--ro entry* [id]
      +--ro id          uint8
      +--ro label?      rt-types:mpls-label
      +--ro ttl?        uint8
      +--ro traffic-class? uint8
augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
  /rt:next-hop-options/rt:next-hop-list/rt:next-hop-list
  /rt:next-hop:
  +--ro index?          string
  +--ro backup-index?   string
  +--ro loadshare?      uint16
  +--ro role?           nhlfe-role
  +--ro mpls-label-stack
    +--ro entry* [id]
      +--ro id          uint8
      +--ro label?      rt-types:mpls-label
      +--ro ttl?        uint8
      +--ro traffic-class? uint8
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input:
  +---w local-label?    rt-types:mpls-label
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route
  /rt:next-hop/rt:next-hop-options/rt:simple-next-hop:
  +-- mpls-label-stack
    +-- entry* [id]
      +-- id          uint8
      +-- label?      rt-types:mpls-label
      +-- ttl?        uint8
      +-- traffic-class? uint8
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route
  /rt:next-hop/rt:next-hop-options/rt:next-hop-list
  /rt:next-hop-list/rt:next-hop:
  +-- index?          string
  +-- backup-index?   string
  +-- loadshare?      uint16
  +-- role?           nhlfe-role
  +-- mpls-label-stack
    +-- entry* [id]
      +-- id          uint8
      +-- label?      rt-types:mpls-label
      +-- ttl?        uint8
      +-- traffic-class? uint8

```

Figure 2: MPLS Base tree diagram

2.4. Model YANG Module

This section describes the "ietf-mpls" YANG module that provides base components of the MPLS data model. Other YANG module(s) may import and augment the base MPLS module to add feature specific data.

The ietf-mpls module imports the following modules:

- o ietf-routing defined in [RFC8349]
- o ietf-routing-types defined in [RFC8294]
- o ietf-interfaces defined in [RFC8343]

```
<CODE BEGINS> file "ietf-mpls@2019-09-11.yang"
module ietf-mpls {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls";

  /* Replace with IANA when assigned */
  prefix "mpls";

  import ietf-routing {
    prefix "rt";
    reference "RFC8349: A YANG Data Model for Routing Management";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference "RFC8294: Common YANG Data Types for the Routing Area";
  }

  import ietf-yang-types {
    prefix "yang";
    reference "RFC6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference "RFC8343: A YANG Data Model for Interface Management";
  }

  organization "IETF MPLS Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/mpls/>"

```

WG List: <mailto:mpls@ietf.org>

Editor: Tarek Saad
<mailto:tsaad@cisco.com>

Editor: Kamran Raza
<mailto:skraza@cisco.com>

Editor: Rakesh Gandhi
<mailto:rgandhi@cisco.com>

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Vishnu Pavan Beeram
<mailto:vbeeram@juniper.net>";

description

"This YANG module defines the essential components for the management of the MPLS subsystem. The model fully conforms to the Network Management Datastore Architecture (NMDA).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>). This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.

// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

```
revision "2019-09-11" {  
  description  
    "Latest revision:  
    - Addressed review comments";  
  reference "RFC XXXX: A YANG Data Model for base MPLS";  
}
```

/* Identities */

```
identity mpls {
  base rt:address-family;
  description
    "This identity represents the MPLS address family.";
}

identity label-block-alloc-mode {
  description
    "Base identity label-block allocation mode";
}

identity label-block-alloc-mode-manager {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by label manager";
}

identity label-block-alloc-mode-application {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by application";
}

/**
 * Typedefs
 */
typedef mpls-operations-type {
  type enumeration {
    enum impose-and-forward {
      description
        "Operation impose outgoing label(s) and forward to
        next-hop";
    }
    enum pop-and-forward {
      description
        "Operation pop incoming label and forward to next-hop";
    }
    enum pop-impose-and-forward {
      description
        "Operation pop incoming label, impose one or more
        outgoing label(s) and forward to next-hop";
    }
    enum swap-and-forward {
      description
        "Operation swap incoming label, with outgoing label and
        forward to next-hop";
    }
  }
}
```

```
    }
    enum pop-and-lookup {
        description
            "Operation pop incoming label and perform a lookup";
    }
}
description "MPLS operations types";
}

typedef nhlfe-role {
    type enumeration {
        enum PRIMARY {
            description
                "Next-hop acts as primary traffic carrying";
        }
        enum BACKUP {
            description
                "Next-hop acts as backup";
        }
        enum PRIMARY_AND_BACKUP {
            description
                "Next-hop acts as primary and backup simultaneously";
        }
    }
    description "The next-hop role";
}

grouping nhlfe-single-contents {
    description
        "MPLS simple NHLFE contents";
    uses rt-types:mpls-label-stack;
}

grouping nhlfe-multiple-contents {
    description
        "MPLS NHLFE contents";
    leaf index {
        type string;
        description
            "A user-specified identifier utilised to uniquely
            reference the next-hop entry in the next-hop list.
            The value of this index has no semantic meaning
            other than for referencing the entry.";
    }

    leaf backup-index {
        type string;
        description

```



```
    "A user-specified identifier utilised to uniquely
    reference the backup next-hop entry in the NHLFE list.
    The value of this index has no semantic meaning
    other than for referencing the entry.";
  }

  leaf loadshare {
    type uint16;
    description
      "This value is used to compute a loadshare to perform un-equal
      load balancing when multiple outgoing next-hop(s) are
      specified. A share is computed as a ratio of this number to the
      total under all next-hops(s).";
  }

  leaf role {
    type nhlfe-role;
    description "NHLFE role";
  }

  uses nhlfe-single-contents;
}

grouping interface-mpls-properties {
  description "MPLS interface contents grouping";
  leaf enabled {
    type boolean;
    description
      "'true' if mpls encapsulation is enabled on the
      interface. 'false' if mpls encapsulation is enabled
      on the interface.";
  }
  leaf mtu {
    type uint32;
    description
      "MPLS Maximum Transmission Unit (MTU) in bytes";
  }
}

grouping interfaces-mpls {
  description "List of MPLS interfaces";
  list interface {
    key "name";
    description "List of MPLS interfaces";
    leaf name {
      type if:interface-ref;
      description
        "The name of a configured MPLS interface";
    }
  }
}
```

```
    }
    uses interface-mpls-properties;
  }
}

grouping label-block-properties {
  description "Label-block configuration items";
  leaf index {
    type string;
    description
      "A user-specified identifier utilised to uniquely
      reference an MPLS label block";
  }
  leaf start-label {
    type rt-types:mpls-label;
    must '.. >= ../end-label' {
      error-message
        "The start-label must be less than or equal " +
        "to end-label";
    }
    description "Label-block start";
  }
  leaf end-label {
    type rt-types:mpls-label;
    must '.. <= ../start-label' {
      error-message
        "The end-label must be greater than or equal " +
        "to start-label";
    }
    description "Label-block end";
  }
  leaf block-allocation-mode {
    type identityref {
      base label-block-alloc-mode;
    }
    description "Label-block allocation mode";
  }
}

grouping label-block_state {
  description "Label-block state items";
  leaf free-labels-count {
    when "../block-allocation-mode = " +
      "'label-block-alloc-mode-manager'";
    type yang:counter32;
    config false;
    description "Label-block free labels count";
  }
}
```

```
    leaf inuse-labels-count {
        when "../block-allocation-mode = " +
            "'label-block-alloc-mode-manager'";
        type yang:counter32;
        config false;
        description "Label-block inuse labels count";
    }
}

grouping globals {
    description "MPLS global configuration grouping";
    leaf ttl-propagate {
        type boolean;
        default 'true';
        description "Propagate TTL between IP and MPLS";
    }
}

grouping label-blocks {
    description "Label-block allocation grouping";
    container label-blocks {
        description "Label-block allocation container";
        list label-block {
            key index;
            unique "start-label end-label";
            description "List of MPLS label-blocks";
            uses label-block-properties;
            uses label-block_state;
        }
    }
}

augment "/rt:routing" {
    description "MPLS augmentation.";
    container mpls {
        description
            "MPLS container, to be used as an augmentation target node
            other MPLS sub-features config, e.g. MPLS static LSP, MPLS
            LDP LSPs, and Traffic Engineering MPLS LSP Tunnels, etc.";
        uses globals;
        uses label-blocks;
        uses interfaces-mpls;
    }
}

/* MPLS routes augmentation */
augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
    description
        "This is augmentation for all MPLS routes.";
}
```

```
leaf mpls-enabled {
    type boolean;
    default 'false';
    description
        "Indicates whether MPLS is enabled for this route";
}
leaf local-label {
    when "../mpls-enabled = 'true'";
    type rt-types:mpls-label;
    description "MPLS local label associated with the route.";
}
}

/* MPLS simple-next-hop augmentation */
augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
    description
        "Augment 'simple-next-hop' case in IP unicast routes.";
    uses nhlfe-single-contents {
        when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" +
            "/mpls:mpls-enabled = 'true'";
    }
}

/* MPLS next-hop-list augmentation */
augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
    + "rt:next-hop-list/rt:next-hop" {
    description
        "This leaf augments the 'next-hop-list' case of IP unicast
        routes.";
    uses nhlfe-multiple-contents {
        when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" +
            "/mpls:mpls-enabled = 'true'";
    }
}

/* MPLS RPC input augmentation */
augment
    "/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" {
    description
        "Input MPLS augmentation for the 'active-route' action
        statement.";
    leaf local-label {
        type rt-types:mpls-label;
        description
            "MPLS local label.";
    }
}
```

```
}

/* MPLS RPC output augmentation */
augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  description
    "Output MPLS augmentation for the 'active-route' action
    statement.";
  uses nhlfe-single-contents;
}
augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
  + "rt:output/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
  + "rt:next-hop-list/rt:next-hop" {
  description
    "Output MPLS augmentation for the 'active-route' action
    statement.";
  uses nhlfe-multiple-contents;
}
}
<CODE ENDS>
```

Figure 3: MPLS base YANG module

3. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls
Registrant Contact: The MPLS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-mpls
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls
prefix: ietf-mpls
// RFC Ed.: replace XXXX with RFC number and remove this note
reference: RFCXXXX

4. Security Considerations

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

Some of the readable data nodes in these YANG modules may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route: this path is augmented by additional MPLS leaf(s) defined in this model. Access to this information may disclose the per prefix and/or other information.

/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route/rt:next-hop/rt:next-hop-options/rt:simple-next-hop: this path is augmented by additional MPLS leaf(s) defined in this model. Access to this information may disclose the next-hop or path per prefix and/or other information.

5. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of this model.

6. Contributors

Igor Bryskin
Huawei Technologies
email: Igor.Bryskin@huawei.com

Himanshu Shah
Ciena
email: hshah@ciena.com

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

7.2. Informative References

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Kamran Raza
Cisco Systems Inc

Email: skraza@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

K. Raza
R. Asati
Cisco Systems

X. Liu
Volta Networks

S. Esale
Juniper Networks

X. Chen
Huawei Technologies

H. Shah
Ciena Corporation

November 4, 2019

YANG Data Model for MPLS LDP
draft-ietf-mpls-ldp-yang-07

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP). The model also serves as the base model to define Multipoint LDP (mLDP) model.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

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 <https://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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Base and Extended	3
2. Specification of Requirements	4
3. Overview	4
4. Consolidated Tree	7
5. Configuration	16
5.1. Configuration Tree	19
5.1.1. Base	19
5.1.2. Extended	20
5.2. Configuration Hierarchy	23
5.2.1. Per-VRF parameters	24
6. Operational State	25
6.1. Operational Tree	25
6.1.1. Base	26
6.1.2. Extended	29
6.2. States	31
6.2.1. Adjacency state	31
6.2.2. Peer state	32
6.2.3. Bindings state	33
6.2.4. Capabilities state	35
7. Notifications	35
8. Actions	36
9. YANG Specification	37
9.1. Base	37
9.2. Extended	65
10. Security Considerations	86
11. IANA Considerations	87
12. Acknowledgments	87
13. References	88

13.1. Normative References	88
13.2. Informative References	90
Appendix A. Data Tree Example	91
Appendix B. Additional Contributors	96
Authors' Addresses	96

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is one of the network management protocols that defines mechanisms to manage network devices. YANG [RFC6020] [RFC7950] is a modular language that represents data structures in an XML tree format, and is used as a data modelling language for the NETCONF.

This document introduces a YANG data model for MPLS Label Distribution Protocol (LDP) [RFC5036]. This model also covers LDP IPv6 [RFC7552] and LDP capabilities [RFC5561] specifications.

The data model is defined for following constructs that are used for managing the protocol:

- o Configuration
- o Operational State
- o Executables (Actions)
- o Notifications

This document is organized to define the data model for each of the above constructs in the sequence as listed above.

1.1. Base and Extended

The configuration and state items are divided into following two broad categories:

- o Base
- o Extended

The "base" category contains the basic and fundamental features that are covered in LDP base specification [RFC5036] and constitute the minimum requirements for a typical base LDP deployment. Whereas, the "extended" category contains all other non-base features. All the items in a base category are mandatory and hence no "if-feature" is allowed under the "base" category model. The base and extended categories are defined in their own modules as described later.

The example of base feature includes the configuration of LDP lsr-id, enabling LDP interfaces, setting password for LDP session etc., whereas the examples of extended feature include inbound/outbound label policies, igp sync, downstream-on-demand etc. This is worth highlighting that LDP IPv6 [RFC7552] is also categorized as an extended feature.

While "base" model support will suffice for small deployments, it is expected that large deployments will require not only the "base" module support from the vendors but also the support for "extended" model for some extended feature(s) of interest.

2. Specification of Requirements

In this document, the word "IP" is used to refer to both IPv4 and IPv6, unless otherwise explicitly stated. For example, "IP address family" means and be read as "IPv4 and/or IPv6 address family"

3. Overview

This document defines two new modules for LDP YANG support:

- o "ietf-mpls-ldp" module that models the base LDP features and augments /rt:routing/rt:control-plane-protocols defined in [RFC8349]
- o extended "ietf-mpls-ldp-extended" module that models the extended LDP features and augments the base LDP

It is to be noted that mLDP data model [I-D.ietf-mpls-mldp-yang] augments LDP base and extended models to model the base and extended mLDP features respectively.

There are four main containers in our module(s):

- o Read-Write parameters for configuration (Discussed in Section 5)
- o Read-only parameters for operational state (Discussed in Section 6)
- o Notifications for events (Discussed in Section 7)
- o RPCs for executing commands to perform some action (Discussed in Section 8)

The modeling in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342]. The operational state data is combined with the associated configuration data in the

same hierarchy [RFC8407]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

Following diagram depicts high level LDP yang tree organization and hierarchy:

```

+-- rw routing
  +-- rw control-plane-protocols
    +-- rw mpls-ldp
      +-- rw ...
        +-- rw ... // base
        |   +-- rw ...
        |   +-- ro ...
        |   +--
        +-- ro ...
        |   +-- ro ...
        |   +-- ro ...
        |   +--
        +-- rw ldp-ext: .... // extended
        |   +-- rw ...
        |   +-- ro ...
        |   +--
        +-- ro ...
            +-- ro ...
            +-- ro ...

```

```

rpcs:
  +-- x mpls-ldp-some_action
  +-- x . . . . .

notifications:
  +--- n mpls-ldp-some_event
  +--- n ...

```

Figure 1

Before going into data model details, it is important to take note of the following points:

- o This module aims to address only the core LDP parameters as per RFC specification, as well as some widely deployed non-RFC features (such as label policies, session authentication etc).

Any vendor specific feature should be defined in a vendor-specific augmentation of this model.

- o Multi-topology LDP [RFC7307] is beyond the scope of this document.
- o This module does not cover any applications running on top of LDP, nor does it cover any OAM procedures for LDP.
- o This model is a VPN Forwarding and Routing (VRF)-centric model. It is important to note that [RFC4364] defines VRF tables and default forwarding tables as different, however from a yang modelling perspective this introduces unnecessary complications, hence we are treating the default forwarding table as just another VRF.
- o A "network-instance", as defined in [RFC8529], refers to a VRF instance (both default and non-default) within the scope of this model.
- o This model supports two address-families, namely "ipv4" and "ipv6".
- o This model assumes platform-wide label space (i.e. label space Id of zero). However, when Upstream Label assignment [RFC6389] is in use, an upstream assigned label is looked up in a Context-Specific label space as defined in [RFC5331].
- o The label and peer policies (including filters) are defined using a prefix-list. When used for a peer policy, the prefix refers to the LSR Id of the peer. The prefix-list is referenced from routing-policy model as defined in [I-D.ietf-rtgwg-policy-model].
- o This model uses the terms LDP "neighbor"/"adjacency", "session", and "peer" with the following semantics:
 - * Neighbor/Adjacency: An LDP enabled LSR that is discovered through LDP discovery mechanisms.
 - * Session: An LDP neighbor with whom a TCP connection has been established.
 - * Peer: An LDP session which has successfully progressed beyond its initialization phase and is either already exchanging the bindings or is ready to do so.

It is to be noted that LDP Graceful Restart mechanisms defined in [RFC3478] allow keeping the exchanged bindings for some time after a session goes down with a peer. We call such a state belonging

to a "stale" peer -- i.e. keeping peer bindings from a peer with whom currently there is either no connection established or connection is established but GR session is in recovery state. When used in this document, the above terms will refer strictly to the semantics and definitions defined for them.

A simplified graphical tree representation of full LDP YANG data model is presented in Figure 2, whereas LDP configuration (base and extended), state (base and extended), notification, and rpc are graphically represented in Figure 5, Figure 6, Figure 8, Figure 9, Figure 15, and Figure 16 respectively. The meaning of the symbols in these tree diagrams is defined in [RFC8340].

The actual base and extended model definition in YANG is captured in Section 9.

While presenting the YANG tree view and actual .yang specification, this document assumes readers' familiarity with the concepts of YANG modeling, its presentation and its compilation.

4. Consolidated Tree

Following is a consolidated tree representation of configuration, state, notification, and rpc items under LDP base and extended.

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--rw capability
        +--rw ldp-ext:end-of-lib {capability-end-of-lib}?
          | +--rw ldp-ext:enable?    boolean
        +--rw ldp-ext:typed-wildcard-fec
          | {capability-typed-wildcard-fec}?
          | +--rw ldp-ext:enable?    boolean
        +--rw ldp-ext:upstream-label-assignment
          | {capability-upstream-label-assignment}?
          | +--rw ldp-ext:enable?    boolean
      +--rw graceful-restart
        +--rw enable?                boolean
        +--rw reconnect-time?        uint16
        +--rw recovery-time?         uint16
        +--rw forwarding-holdtime?   uint16
        +--rw ldp-ext:helper-enable? boolean
          | {graceful-restart-helper-mode}?
      +--rw lsr-id?
        | rt-types:router-id

```



```

+--rw address-families
|   +--rw ipv4!
|   |   +--rw enable?                               boolean
|   |   +--ro label-distribution-controlmode?        enumeration
|   |   +--ro bindings
|   |   |   +--ro address* [address]
|   |   |   |   +--ro address                        inet:ipv4-address
|   |   |   |   +--ro advertisement-type?            advertised-received
|   |   |   |   +--ro peer
|   |   |   |   |   +--ro lsr-id?                    leafref
|   |   |   |   |   +--ro label-space-id?            leafref
|   |   |   +--ro fec-label* [fec]
|   |   |   |   +--ro fec                            inet:ipv4-prefix
|   |   |   |   +--ro peer*
|   |   |   |   |   [lsr-id label-space-id advertisement-type]
|   |   |   |   |   +--ro lsr-id                    leafref
|   |   |   |   |   +--ro label-space-id            leafref
|   |   |   |   |   +--ro advertisement-type
|   |   |   |   |   |   advertised-received
|   |   |   |   |   +--ro label?
|   |   |   |   |   |   rt-types:mpls-label
|   |   |   |   |   +--ro used-in-forwarding?        boolean
|   |   +--rw ldp-ext:label-policy
|   |   |   +--rw ldp-ext:advertise
|   |   |   |   +--rw ldp-ext:egress-explicit-null
|   |   |   |   |   +--rw ldp-ext:enable?            boolean
|   |   |   |   +--rw ldp-ext:prefix-list?
|   |   |   |   |   prefix-list-ref
|   |   |   +--rw ldp-ext:accept
|   |   |   |   +--rw ldp-ext:prefix-list?            prefix-list-ref
|   |   |   +--rw ldp-ext:assign
|   |   |   |   {policy-label-assignment-config}?
|   |   |   +--rw ldp-ext:independent-mode
|   |   |   |   +--rw ldp-ext:prefix-list?            prefix-list-ref
|   |   |   +--rw ldp-ext:ordered-mode
|   |   |   |   {policy-ordered-label-config}?
|   |   |   |   +--rw ldp-ext:egress-prefix-list?
|   |   |   |   |   prefix-list-ref
|   |   +--rw ldp-ext:transport-address?
|   |   |   inet:ipv4-address
|   +--rw ldp-ext:ipv6!
|   |   +--rw ldp-ext:enable?
|   |   |   boolean
|   |   +--rw ldp-ext:label-policy
|   |   |   +--rw ldp-ext:advertise
|   |   |   |   +--rw ldp-ext:egress-explicit-null
|   |   |   |   |   +--rw ldp-ext:enable?            boolean
|   |   |   |   +--rw ldp-ext:prefix-list?

```

```

|         prefix-list-ref
+--rw ldp-ext:accept
|   +--rw ldp-ext:prefix-list?   prefix-list-ref
+--rw ldp-ext:assign
|   {policy-label-assignment-config}?
+--rw ldp-ext:independent-mode
|   +--rw ldp-ext:prefix-list?   prefix-list-ref
+--rw ldp-ext:ordered-mode
|   {policy-ordered-label-config}?
+--rw ldp-ext:egress-prefix-list?
|   prefix-list-ref
+--rw ldp-ext:transport-address
|   inet:ipv6-address
+--ro ldp-ext:label-distribution-controlmode?
|   enumeration
+--ro ldp-ext:bindings
+--ro ldp-ext:address* [address]
|   +--ro ldp-ext:address
|   |   inet:ipv6-address
+--ro ldp-ext:advertisement-type?
|   |   advertised-received
+--ro ldp-ext:peer
|   +--ro ldp-ext:lsr-id?           leafref
|   +--ro ldp-ext:label-space-id?  leafref
+--ro ldp-ext:fec-label* [fec]
+--ro ldp-ext:fec      inet:ipv6-prefix
+--ro ldp-ext:peer*
|   [lsr-id label-space-id advertisement-type]
+--ro ldp-ext:lsr-id           leafref
+--ro ldp-ext:label-space-id   leafref
+--ro ldp-ext:advertisement-type
|   |   advertised-received
+--ro ldp-ext:label?
|   |   rt-types:mpls-label
+--ro ldp-ext:used-in-forwarding?  boolean
+--rw ldp-ext:forwarding-nexthop
|   {forwarding-nexthop-config}?
+--rw ldp-ext:interfaces
+--rw ldp-ext:interface* [name]
+--rw ldp-ext:name           if:interface-ref
+--rw ldp-ext:address-family* [afi]
+--rw ldp-ext:afi
|   |   ldp:ldp-address-family
+--rw ldp-ext:ldp-disable?  boolean
+--rw ldp-ext:igp-synchronization-delay?  uint16
+--rw discovery
+--rw interfaces
|   +--rw hello-holdtime?  uint16

```

```

+--rw hello-interval?   uint16
+--rw interface* [name]
  +--rw name
    |   if:interface-ref
  +--ro next-hello?      uint16
+--rw address-families
  +--rw ipv4!
    +--rw enable?        boolean
    +--ro hello-adjacencies
      +--ro hello-adjacency* [adjacent-address]
        +--ro adjacent-address
          |   inet:ipv4-address
        +--ro flag*       identityref
        +--ro hello-holdtime
          +--ro adjacent?   uint16
          +--ro negotiated? uint16
          +--ro remaining?  uint16
        +--ro next-hello?   uint16
        +--ro statistics
          +--ro discontinuity-time
            |   yang:date-and-time
          +--ro hello-received?
            |   yang:counter64
          +--ro hello-dropped?
            |   yang:counter64
        +--ro peer
          +--ro lsr-id?      leafref
          +--ro label-space-id? leafref
      +--rw ldp-ext:transport-address? union
+--rw ldp-ext:ipv6!
  +--rw ldp-ext:enable?      boolean
  +--ro ldp-ext:hello-adjacencies
    +--ro ldp-ext:hello-adjacency*
      [adjacent-address]
    +--ro ldp-ext:adjacent-address
      |   inet:ipv6-address
    +--ro ldp-ext:flag*
      |   identityref
    +--ro ldp-ext:hello-holdtime
      +--ro ldp-ext:adjacent?   uint16
      +--ro ldp-ext:negotiated? uint16
      +--ro ldp-ext:remaining?  uint16
    +--ro ldp-ext:next-hello?   uint16
    +--ro ldp-ext:statistics
      +--ro ldp-ext:discontinuity-time
        |   yang:date-and-time
      +--ro ldp-ext:hello-received?
        |   yang:counter64

```

```

    +--ro ldp-ext:hello-dropped?
        yang:counter64
    +--ro ldp-ext:peer
        +--ro ldp-ext:lsr-id?          leafref
        +--ro ldp-ext:label-space-id?  leafref
    +--rw ldp-ext:transport-address?  union
+--rw ldp-ext:hello-holdtime?          uint16
+--rw ldp-ext:hello-interval?          uint16
+--rw ldp-ext:igp-synchronization-delay? uint16
    {per-interface-timer-config}?
+--rw targeted
    +--rw hello-holdtime?          uint16
    +--rw hello-interval?          uint16
    +--rw hello-accept
        +--rw enable?              boolean
        +--rw ldp-ext:neighbor-list? neighbor-list-ref
            {policy-targeted-discovery-config}?
+--rw address-families
    +--rw ipv4!
        +--ro hello-adjacencies
            +--ro hello-adjacency*
                [local-address adjacent-address]
                +--ro local-address          inet:ipv4-address
                +--ro adjacent-address        inet:ipv4-address
                +--ro flag*                  identityref
                +--ro hello-holdtime
                    +--ro adjacent?          uint16
                    +--ro negotiated?        uint16
                    +--ro remaining?         uint16
                +--ro next-hello?            uint16
                +--ro statistics
                    +--ro discontinuity-time
                        yang:date-and-time
                    +--ro hello-received?
                        yang:counter64
                    +--ro hello-dropped?
                        yang:counter64
                +--ro peer
                    +--ro lsr-id?            leafref
                    +--ro label-space-id?    leafref
        +--rw target* [adjacent-address]
            +--rw adjacent-address          inet:ipv4-address
            +--rw enable?                   boolean
            +--rw local-address?            inet:ipv4-address
+--rw ldp-ext:ipv6!
    +--ro ldp-ext:hello-adjacencies
        +--ro ldp-ext:hello-adjacency*
            [local-address adjacent-address]

```

```

    +---ro ldp-ext:local-address
    |   inet:ipv6-address
    +---ro ldp-ext:adjacent-address
    |   inet:ipv6-address
    +---ro ldp-ext:flag*
    |   identityref
    +---ro ldp-ext:hello-holdtime
    |   +---ro ldp-ext:adjacent?      uint16
    |   +---ro ldp-ext:negotiated?    uint16
    |   +---ro ldp-ext:remaining?     uint16
    +---ro ldp-ext:next-hello?        uint16
    +---ro ldp-ext:statistics
    |   +---ro ldp-ext:discontinuity-time
    |   |   yang:date-and-time
    |   +---ro ldp-ext:hello-received?
    |   |   yang:counter64
    |   +---ro ldp-ext:hello-dropped?
    |   |   yang:counter64
    +---ro ldp-ext:peer
    |   +---ro ldp-ext:lsr-id?         leafref
    |   +---ro ldp-ext:label-space-id? leafref
    +---rw ldp-ext:target* [adjacent-address]
    |   +---rw ldp-ext:adjacent-address
    |   |   inet:ipv6-address
    |   +---rw ldp-ext:enable?         boolean
    |   +---rw ldp-ext:local-address?  inet:ipv6-address
+---rw peers
+---rw authentication
|   +---rw (auth-type-selection)?
|   |   +---: (auth-key)
|   |   |   +---rw md5-key?          string
|   |   +---: (ldp-ext:auth-key-chain) {key-chain}?
|   |   |   +---rw ldp-ext:key-chain? key-chain:key-chain-ref
+---rw session-ka-holdtime?          uint16
+---rw session-ka-interval?          uint16
+---rw peer* [lsr-id label-space-id]
|   +---rw lsr-id                    rt-types:router-id
|   +---rw label-space-id            uint16
|   +---rw authentication
|   |   +---rw (auth-type-selection)?
|   |   |   +---: (auth-key)
|   |   |   |   +---rw md5-key?          string
|   |   |   +---: (ldp-ext:auth-key-chain) {key-chain}?
|   |   |   |   +---rw ldp-ext:key-chain? key-chain:key-chain-ref
+---rw capability
+---rw address-families

```

```

+--rw ipv4!
|   +--ro hello-adjacencies
|   |   +--ro hello-adjacency*
|   |   |   [local-address adjacent-address]
|   |   |   +--ro local-address      inet:ipv4-address
|   |   |   +--ro adjacent-address   inet:ipv4-address
|   |   |   +--ro flag*              identityref
|   |   |   +--ro hello-holdtime
|   |   |   |   +--ro adjacent?      uint16
|   |   |   |   +--ro negotiated?    uint16
|   |   |   |   +--ro remaining?     uint16
|   |   |   +--ro next-hello?        uint16
|   |   +--ro statistics
|   |   |   +--ro discontinuity-time
|   |   |   |   yang:date-and-time
|   |   |   +--ro hello-received?
|   |   |   |   yang:counter64
|   |   |   +--ro hello-dropped?
|   |   |   |   yang:counter64
|   |   +--ro interface?             if:interface-ref
|   +--rw ldp-ext:label-policy
|   |   +--rw ldp-ext:advertise
|   |   |   +--rw ldp-ext:prefix-list?  prefix-list-ref
|   |   +--rw ldp-ext:accept
|   |   |   +--rw ldp-ext:prefix-list?  prefix-list-ref
+--rw ldp-ext:ipv6!
|   +--ro ldp-ext:hello-adjacencies
|   |   +--ro ldp-ext:hello-adjacency*
|   |   |   [local-address adjacent-address]
|   |   |   +--ro ldp-ext:local-address
|   |   |   |   inet:ipv6-address
|   |   |   +--ro ldp-ext:adjacent-address
|   |   |   |   inet:ipv6-address
|   |   |   +--ro ldp-ext:flag*
|   |   |   |   identityref
|   |   |   +--ro ldp-ext:hello-holdtime
|   |   |   |   +--ro ldp-ext:adjacent?    uint16
|   |   |   |   +--ro ldp-ext:negotiated?  uint16
|   |   |   |   +--ro ldp-ext:remaining?    uint16
|   |   |   +--ro ldp-ext:next-hello?      uint16
|   |   +--ro ldp-ext:statistics
|   |   |   +--ro ldp-ext:discontinuity-time
|   |   |   |   yang:date-and-time
|   |   |   +--ro ldp-ext:hello-received?
|   |   |   |   yang:counter64
|   |   |   +--ro ldp-ext:hello-dropped?
|   |   |   |   yang:counter64
|   |   +--ro ldp-ext:interface?

```

```

|         |         if:interface-ref
|         +---rw ldp-ext:label-policy
|         |         +---rw ldp-ext:advertise
|         |         |         +---rw ldp-ext:prefix-list?    prefix-list-ref
|         |         +---rw ldp-ext:accept
|         |         |         +---rw ldp-ext:prefix-list?    prefix-list-ref
+---ro label-advertisement-mode
|   +---ro local?          label-adv-mode
|   +---ro peer?           label-adv-mode
|   +---ro negotiated?     label-adv-mode
+---ro next-keep-alive?    uint16
+---ro received-peer-state
|   +---ro graceful-restart
|   |   +---ro enable?      boolean
|   |   +---ro reconnect-time?  uint16
|   |   +---ro recovery-time?  uint16
|   +---ro capability
|   |   +---ro end-of-lib
|   |   |   +---ro enable?    boolean
|   |   +---ro typed-wildcard-fec
|   |   |   +---ro enable?    boolean
|   |   +---ro upstream-label-assignment
|   |   |   +---ro enable?    boolean
+---ro session-holdtime
|   +---ro peer?           uint16
|   +---ro negotiated?     uint16
|   +---ro remaining?      uint16
+---ro session-state?      enumeration
+---ro tcp-connection
|   +---ro local-address?   inet:ip-address
|   +---ro local-port?      inet:port-number
|   +---ro remote-address?  inet:ip-address
|   +---ro remote-port?     inet:port-number
+---ro up-time?            string
+---ro statistics
|   +---ro discontinuity-time    yang:date-and-time
|   +---ro received
|   |   +---ro total-octets?      yang:counter64
|   |   +---ro total-messages?   yang:counter64
|   |   +---ro address?           yang:counter64
|   |   +---ro address-withdraw?  yang:counter64
|   |   +---ro initialization?    yang:counter64
|   |   +---ro keepalive?         yang:counter64
|   |   +---ro label-abort-request? yang:counter64
|   |   +---ro label-mapping?     yang:counter64
|   |   +---ro label-release?     yang:counter64
|   |   +---ro label-request?     yang:counter64
|   |   +---ro label-withdraw?    yang:counter64

```

```

| | | +--ro notification?          yang:counter64
| | | +--ro sent
| | | | +--ro total-octets?        yang:counter64
| | | | +--ro total-messages?     yang:counter64
| | | | +--ro address?            yang:counter64
| | | | +--ro address-withdraw?   yang:counter64
| | | | +--ro initialization?     yang:counter64
| | | | +--ro keepalive?          yang:counter64
| | | | +--ro label-abort-request? yang:counter64
| | | | +--ro label-mapping?      yang:counter64
| | | | +--ro label-release?      yang:counter64
| | | | +--ro label-request?      yang:counter64
| | | | +--ro label-withdraw?     yang:counter64
| | | | +--ro notification?       yang:counter64
| | | +--ro total-addresses?      uint32
| | | +--ro total-labels?         uint32
| | | +--ro total-fec-label-bindings? uint32
| | +--rw ldp-ext:admin-down?     boolean
| | | {per-peer-admin-down}?
| | +--rw ldp-ext:graceful-restart
| | | +--rw ldp-ext:enable?        boolean
| | | +--rw ldp-ext:reconnect-time? uint16
| | | +--rw ldp-ext:recovery-time? uint16
| | +--rw ldp-ext:session-ka-holdtime? uint16
| | +--rw ldp-ext:session-ka-interval? uint16
+--rw ldp-ext:session-downstream-on-demand
| | {session-downstream-on-demand-config}?
| | +--rw ldp-ext:enable?          boolean
| | +--rw ldp-ext:peer-list?       peer-list-ref
+--rw ldp-ext:dual-stack-transport-pereference
| | {dual-stack-transport-pereference}?
| | +--rw ldp-ext:max-wait?        uint16
| | +--rw ldp-ext:prefer-ipv4!
| | | +--rw ldp-ext:peer-list?     peer-list-ref

rpcs:
+---x mpls-ldp-clear-peer
| +---w input
| | +---w lsr-id?                  leafref
| | +---w label-space-id?         leafref
+---x mpls-ldp-clear-hello-adjacency
| +---w input
| | +---w hello-adjacency
| | | +---w (hello-adjacency-type)?
| | | | +--:(targeted)
| | | | | +---w targeted!
| | | | | | +---w target-address?   inet:ip-address
| | | | +--:(link)

```



```

|               +---w link!
|               +---w next-hop-interface?  leafref
|               +---w next-hop-address?    inet:ip-address
+---x mpls-ldp-clear-peer-statistics
|   +---w input
|       +---w lsr-id?          leafref
|       +---w label-space-id?  leafref
notifications:
+---n mpls-ldp-peer-event
|   +---ro event-type?  oper-status-event-type
|   +---ro peer
|       +---ro lsr-id?          leafref
|       +---ro label-space-id?  leafref
+---n mpls-ldp-hello-adjacency-event
|   +---ro event-type?          oper-status-event-type
|   +---ro (hello-adjacency-type)?
|       +---:(targeted)
|           +---ro targeted
|               +---ro target-address?  inet:ip-address
|       +---:(link)
|           +---ro link
|               +---ro next-hop-interface?  if:interface-ref
|               +---ro next-hop-address?    inet:ip-address
+---n mpls-ldp-fec-event
|   +---ro event-type?  oper-status-event-type
|   +---ro prefix?      inet:ip-prefix

```

Figure 2

5. Configuration

This specification defines the configuration parameters for base LDP as specified in [RFC5036] and LDP IPv6 [RFC7552]. Moreover, it incorporates provisions to enable LDP Capabilities [RFC5561], and defines some of the most significant and commonly used capabilities such as Typed Wildcard FEC [RFC5918], End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

This model augments /rt:routing/rt:control-plane-protocols that is defined in [RFC8349] and follows NMDA as mentioned earlier.

Following is the high-level configuration organization for base LDP:

```

augment /rt:routing/rt:control-plane-protocols:
  +-- mpls-ldp
    +-- global
      +-- ...
      +-- ...
      +-- address-families
        +-- ipv4
          +-- . . .
          +-- . . .
      +-- capability
        +-- ...
        +-- ...
    +-- discovery
      +-- interfaces
        +-- ...
        +-- ...
        +-- interface* [interface]
          +-- ...
          +-- address-families
            +-- ipv4
              +-- ...
              +-- ...
      +-- targeted
        +-- ...
        +-- address-families
          +-- ipv4
            +-- target* [adjacent-address]
              +- ...
              +- ...
    +-- peers
      +-- ...
      +-- ...
      +-- peer* [lsr-id label-space-id]
        +-- ...
        +-- ...

```

Figure 3

Following is the high-level configuration organization for extended LDP:

```

augment /rt:routing/rt:control-plane-protocols:
  +-- mpls-ldp
    +-- global
      +-- ...
      +-- ...

```

```

+-- address-families
|   +-- ipv4
|   |   +-- . . .
|   |   +-- . . .
|   |   +-- label-policy
|   |   |   +-- ...
|   |   |   +-- ...
|   |   +-- ipv6
|   |   |   +-- . . .
|   |   |   +-- . . .
|   |   |   +-- label-policy
|   |   |   |   +-- ...
|   |   |   |   +-- ...
|   |   +-- capability
|   |   |   +-- ...
|   |   |   +-- ...
|   +-- discovery
|   |   +-- interfaces
|   |   |   +-- ...
|   |   |   +-- ...
|   |   |   +-- interface* [interface]
|   |   |   |   +-- ...
|   |   |   |   +-- address-families
|   |   |   |   |   +-- ipv4
|   |   |   |   |   |   +-- ...
|   |   |   |   |   |   +-- ...
|   |   |   |   |   +-- ipv6
|   |   |   |   |   |   +-- ...
|   |   |   |   |   |   +-- ...
|   |   |   +-- targetted
|   |   |   |   +-- ...
|   |   |   |   +-- address-families
|   |   |   |   |   +-- ipv6
|   |   |   |   |   |   +-- target* [adjacent-address]
|   |   |   |   |   |   |   +-- ...
|   |   |   |   |   |   |   +-- ...
+-- forwarding-nextthop
|   +-- ...
|   +-- ...
+-- peers
|   +-- ...
|   +-- ...
|   +-- peer*
|   |   +-- ...
|   |   +-- ...
|   |   +-- label-policy
|   |   |   +-- ..
|   +-- address-families

```

```

+-- ipv4
|   +-- ...
+-- ipv6
    +-- ...

```

Figure 4

Given the configuration hierarchy, the model allows inheritance such that an item in a child tree is able to derive value from a similar or related item in one of the parent. For instance, hello holdtime can be configured per-VRF or per-VRF-interface, thus allowing inheritance as well flexibility to override with a different value at any child level.

5.1. Configuration Tree

5.1.1. Base

Following is a simplified graphical representation of the data model for LDP base configuration

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--rw graceful-restart
        +--rw enable?                boolean
        +--rw reconnect-time?       uint16
        +--rw recovery-time?        uint16
        +--rw forwarding-holdtime?   uint16
      +--rw lsr-id?                  rt-types:router-id
      +--rw address-families
        +--rw ipv4!
          +--rw enable?              boolean
      +--rw discovery
        +--rw interfaces
          +--rw hello-holdtime?      uint16
          +--rw hello-interval?      uint16
          +--rw interface* [name]
            +--rw name                if:interface-ref
            +--rw address-families
              +--rw ipv4!
                +--rw enable?        boolean
          +--rw targeted
            +--rw hello-holdtime?    uint16

```

```

    |
    |   +--rw hello-interval?      uint16
    |   +--rw hello-accept
    |   |   +--rw enable?      boolean
    |   +--rw address-families
    |   |   +--rw ipv4!
    |   |   |   +--rw target* [adjacent-address]
    |   |   |   |   +--rw adjacent-address      inet:ipv4-address
    |   |   |   |   +--rw enable?              boolean
    |   |   |   |   +--rw local-address?       inet:ipv4-address
    |   +--rw peers
    |   |   +--rw authentication
    |   |   |   +--rw (auth-type-selection)?
    |   |   |   |   +--:(auth-key)
    |   |   |   |   |   +--rw md5-key?      string
    |   |   +--rw session-ka-holdtime?  uint16
    |   |   +--rw session-ka-interval?  uint16
    |   |   +--rw peer* [lsr-id label-space-id]
    |   |   |   +--rw lsr-id                      rt-types:router-id
    |   |   |   +--rw label-space-id              uint16
    |   |   |   +--rw authentication
    |   |   |   |   +--rw (auth-type-selection)?
    |   |   |   |   |   +--:(auth-key)
    |   |   |   |   |   |   +--rw md5-key?      string

```

Figure 5

5.1.2. Extended

Following is a simplified graphical representation of the data model for LDP extended configuration

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
  +--rw global
  |   +--rw capability
  |   |   +--rw ldp-ext:end-of-lib {capability-end-of-lib}?
  |   |   |   +--rw ldp-ext:enable?      boolean
  |   |   +--rw ldp-ext:typed-wildcard-fec
  |   |   |   {capability-typed-wildcard-fec}?
  |   |   |   +--rw ldp-ext:enable?      boolean
  |   |   +--rw ldp-ext:upstream-label-assignment
  |   |   |   {capability-upstream-label-assignment}?
  |   |   |   +--rw ldp-ext:enable?      boolean
  |   +--rw graceful-restart

```

```

|   +---rw ldp-ext:helper-enable?   boolean
|                                   {graceful-restart-helper-mode}?
+---rw address-families
|   +---rw ipv4!
|       +---rw ldp-ext:label-policy
|           +---rw ldp-ext:advertise
|               +---rw ldp-ext:egress-explicit-null
|                   |   +---rw ldp-ext:enable?   boolean
|                   |   +---rw ldp-ext:prefix-list? prefix-list-ref
|       +---rw ldp-ext:accept
|           |   +---rw ldp-ext:prefix-list?   prefix-list-ref
|       +---rw ldp-ext:assign
|                   {policy-label-assignment-config}?
|       +---rw ldp-ext:independent-mode
|           |   +---rw ldp-ext:prefix-list?   prefix-list-ref
|       +---rw ldp-ext:ordered-mode {
|                               policy-ordered-label-config}?
|                               +---rw ldp-ext:egress-prefix-list?
|                                       prefix-list-ref
|       +---rw ldp-ext:transport-address? inet:ipv4-address
+---rw ldp-ext:ipv6!
|   +---rw ldp-ext:enable?   boolean
|   +---rw ldp-ext:label-policy
|       +---rw ldp-ext:advertise
|           +---rw ldp-ext:egress-explicit-null
|               |   +---rw ldp-ext:enable?   boolean
|               |   +---rw ldp-ext:prefix-list? prefix-list-ref
|       +---rw ldp-ext:accept
|           |   +---rw ldp-ext:prefix-list?   prefix-list-ref
|       +---rw ldp-ext:assign
|                   {policy-label-assignment-config}?
|       +---rw ldp-ext:independent-mode
|           |   +---rw ldp-ext:prefix-list?   prefix-list-ref
|       +---rw ldp-ext:ordered-mode
|                   {policy-ordered-label-config}?
|                   +---rw ldp-ext:egress-prefix-list?
|                           prefix-list-ref
|       +---rw ldp-ext:transport-address   inet:ipv6-address
+---rw ldp-ext:forwarding-nexthop {forwarding-nexthop-config}?
|   +---rw ldp-ext:interfaces
|       +---rw ldp-ext:interface* [name]
|           +---rw ldp-ext:name           if:interface-ref
|           +---rw ldp-ext:address-family* [afi]
|               +---rw ldp-ext:afi       ldp:ldp-address-family
|               +---rw ldp-ext:ldp-disable?   boolean
+---rw ldp-ext:igp-synchronization-delay?   uint16
+---rw discovery
|   +---rw interfaces

```

```

+--rw interface* [name]
  +--rw name if:interface-ref
  +--rw address-families
    +--rw ipv4!
      | +--rw ldp-ext:transport-address?  union
    +--rw ldp-ext:ipv6!
      | +--rw ldp-ext:enable?              boolean
      | +--rw ldp-ext:transport-address?  union
    +--rw ldp-ext:hello-holdtime?          uint16
    +--rw ldp-ext:hello-interval?          uint16
    +--rw ldp-ext:igp-synchronization-delay? uint16
                                          {per-interface-timer-config}?
+--rw targeted
  +--rw hello-accept
    | +--rw ldp-ext:neighbor-list?  neighbor-list-ref
    |                               {policy-targeted-discovery-config}?
  +--rw address-families
    +--rw ldp-ext:ipv6!
      +--rw ldp-ext:target* [adjacent-address]
        +--rw ldp-ext:adjacent-address  inet:ipv6-address
        +--rw ldp-ext:enable?            boolean
        +--rw ldp-ext:local-address?     inet:ipv6-address
+--rw peers
  +--rw authentication
    | +--rw (auth-type-selection)?
    |   +--:(ldp-ext:auth-key-chain) {key-chain}?
    |   +--rw ldp-ext:key-chain?  key-chain:key-chain-ref
  +--rw peer* [lsr-id label-space-id]
    +--rw lsr-id                      rt-types:router-id
    +--rw label-space-id              uint16
    +--rw authentication
      | +--rw (auth-type-selection)?
      |   +--:(ldp-ext:auth-key-chain) {key-chain}?
      |   +--rw ldp-ext:key-chain?  key-chain:key-chain-ref
    +--rw address-families
      +--rw ipv4!
        | +--rw ldp-ext:label-policy
        |   +--rw ldp-ext:advertise
        |     | +--rw ldp-ext:prefix-list?  prefix-list-ref
        |     +--rw ldp-ext:accept
        |       | +--rw ldp-ext:prefix-list?  prefix-list-ref
      +--rw ldp-ext:ipv6!
        | +--rw ldp-ext:label-policy
        |   +--rw ldp-ext:advertise
        |     | +--rw ldp-ext:prefix-list?  prefix-list-ref
        |     +--rw ldp-ext:accept
        |       | +--rw ldp-ext:prefix-list?  prefix-list-ref
      +--rw ldp-ext:admin-down?  boolean {per-peer-admin-down}?

```

```

    |--rw ldp-ext:graceful-restart
    |   |--rw ldp-ext:enable?          boolean
    |   |--rw ldp-ext:reconnect-time?  uint16
    |   |--rw ldp-ext:recovery-time?   uint16
    |--rw ldp-ext:session-ka-holdtime? uint16
    |--rw ldp-ext:session-ka-interval? uint16
    |--rw ldp-ext:session-downstream-on-demand
    |   {session-downstream-on-demand-config}?
    |   |--rw ldp-ext:enable?          boolean
    |   |--rw ldp-ext:peer-list?       peer-list-ref
    |--rw ldp-ext:dual-stack-transport-pereference
    |   {dual-stack-transport-pereference}?
    |--rw ldp-ext:max-wait?            uint16
    |--rw ldp-ext:prefer-ipv4!
    |   |--rw ldp-ext:peer-list?       peer-list-ref

```

Figure 6

5.2. Configuration Hierarchy

The LDP configuration container is logically divided into following high-level config areas:

- Per-VRF parameters
 - o Global parameters
 - o Per-address-family parameters
 - o LDP Capabilities parameters
 - o Hello Discovery parameters
 - interfaces
 - Per-interface:
 - Global
 - Per-address-family
 - targeted
 - Per-target
 - o Peer parameters
 - Global
 - Per-peer
 - Per-address-family
 - o Forwarding parameters

Figure 7

Following subsections briefly explain these configuration areas.

5.2.1. Per-VRF parameters

LDP module resides under an network-instance and the scope of any LDP configuration defined under this tree is per network-instance (per-VRF). This configuration is further divided into sub categories as follows.

5.2.1.1. Per-VRF global parameters

There are configuration items that are available directly under a VRF instance and do not fall under any other sub tree. Example of such a parameter is LDP LSR id that is typically configured per VRF. To keep legacy LDP features and applications working in an LDP IPv4 networks with this model, this document recommends an operator to pick a routable IPv4 unicast address as an LSR Id.

5.2.1.2. Per-VRF Capabilities parameters

This container falls under global tree and holds the LDP capabilities that are to be enabled for certain features. By default, an LDP capability is disabled unless explicitly enabled. These capabilities are typically used to negotiate with LDP peer(s) the support/non-support related to a feature and its parameters. The scope of a capability enabled under this container applies to all LDP peers in the given VRF instance. There is also a peer level capability container that is provided to override a capability that is enabled/specified at VRF level.

5.2.1.3. Per-VRF Per-Address-Family parameters

Any LDP configuration parameter related to IP address family (AF) whose scope is VRF wide is configured under this tree. The examples of per-AF parameters include enabling LDP for an address family, prefix-list based label policies, and LDP transport address.

5.2.1.4. Per-VRF Hello Discovery parameters

This container is used to hold LDP configuration related to Hello and discovery process for both basic (link) and extended (targeted) discovery.

The "interfaces" is a container to configure parameters related to VRF interfaces. There are parameters that apply to all interfaces (such as hello timers), as well as parameters that can be configured per-interface. Hence, an interface list is defined under "interfaces" container. The model defines parameters to configure per-interface non AF related items, as well as per-interface per-AF

items. The example of former is interface hello timers, and example of latter is enabling hellos for a given AF under an interface.

The "targeted" container under a VRF instance allows to configure LDP targeted discovery related parameters. Within this container, the "target" list provides a mean to configure multiple target addresses to perform extended discovery to a specific destination target, as well as to fine-tune the per-target parameters.

5.2.1.5. Per-VRF Peer parameters

This container is used to hold LDP configuration related to LDP sessions and peers under a VRF instance. This container allows to configure parameters that either apply on VRF's all peers or a subset (peer-list) of VRF peers. The example of such parameters include authentication password, session KA timers etc. Moreover, the model also allows per-peer parameter tuning by specifying a "peer" list under the "peers" container. A peer is uniquely identified using its LSR Id and hence LSR Id is the key for peer list

Like per-interface parameters, some per-peer parameters are AF-agnostic (i.e. either non AF related or apply to both IP address families), and some that belong to an AF. The example of former is per-peer session password configuration, whereas the example of latter is prefix-list based label policies (inbound and outbound) that apply to a given peer.

5.2.1.6. Per-VRF Forwarding parameters

This container is used to hold configuration used to control LDP forwarding behavior under a VRF instance. One example of a configuration under this container is when a user wishes to enable neighbor discovery on an interface but wishes to disable use of the same interface as forwarding nexthop. This example configuration makes sense only when there are more than one LDP enabled interfaces towards the neighbor.

6. Operational State

Operational state of LDP can be queried and obtained from read-only state containers that fall under the same tree (/rt:routing/rt:control-plane-protocols/) as the configuration.

6.1. Operational Tree

6.1.1.1. Base

Following is a simplified graphical representation of the base data model for LDP operational state.

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--rw address-families
        +--rw ipv4!
          +--ro label-distribution-controlmode? enumeration
          +--ro bindings
            +--ro address* [address]
              +--ro address inet:ipv4-address
              +--ro advertisement-type? advertised-received
              +--ro peer
                +--ro lsr-id? leafref
                +--ro label-space-id? leafref
            +--ro fec-label* [fec]
              +--ro fec inet:ipv4-prefix
              +--ro peer*
                [lsr-id label-space-id advertisement-type]
                +--ro lsr-id leafref
                +--ro label-space-id leafref
                +--ro advertisement-type advertised-received
                +--ro label? rt-types:mpls-label
                +--ro used-in-forwarding? boolean
          +--rw discovery
            +--rw interfaces
              +--rw interface* [name]
                +--rw name if:interface-ref
                +--ro next-hello? uint16
                +--rw address-families
                  +--rw ipv4!
                    +--ro hello-adjacencies
                      +--ro hello-adjacency* [adjacent-address]
                        +--ro adjacent-address inet:ipv4-address
                        +--ro flag* identityref
                        +--ro hello-holdtime
                          +--ro adjacent? uint16
                          +--ro negotiated? uint16
                          +--ro remaining? uint16
                        +--ro next-hello? uint16
                      +--ro statistics
                        +--ro discontinuity-time
                          yang:date-and-time

```

```

+--ro hello-received?          yang:counter64
+--ro hello-dropped?          yang:counter64
+--ro peer
+--ro lsr-id?                  leafref
+--ro label-space-id?         leafref
+--rw targeted
+--rw address-families
+--rw ipv4!
+--ro hello-adjacencies
+--ro hello-adjacency*
+--ro [local-address adjacent-address]
+--ro local-address            inet:ipv4-address
+--ro adjacent-address         inet:ipv4-address
+--ro flag*                    identityref
+--ro hello-holdtime
+--ro adjacent?                uint16
+--ro negotiated?              uint16
+--ro remaining?               uint16
+--ro next-hello?              uint16
+--ro statistics
+--ro discontinuity-time
+--ro yang:date-and-time
+--ro hello-received?          yang:counter64
+--ro hello-dropped?           yang:counter64
+--ro peer
+--ro lsr-id?                  leafref
+--ro label-space-id?         leafref
+--rw peers
+--rw peer* [lsr-id label-space-id]
+--rw lsr-id                    rt-types:router-id
+--rw label-space-id            uint16
+--rw address-families
+--rw ipv4!
+--ro hello-adjacencies
+--ro hello-adjacency*
+--ro [local-address adjacent-address]
+--ro local-address            inet:ipv4-address
+--ro adjacent-address         inet:ipv4-address
+--ro flag*                    identityref
+--ro hello-holdtime
+--ro adjacent?                uint16
+--ro negotiated?              uint16
+--ro remaining?               uint16
+--ro next-hello?              uint16
+--ro statistics
+--ro discontinuity-time
+--ro yang:date-and-time
+--ro hello-received?          yang:counter64

```

```

|           | +--ro hello-dropped?          yang:counter64
|           +--ro interface?                if:interface-ref
+--ro label-advertisement-mode
|   +--ro local?          label-adv-mode
|   +--ro peer?           label-adv-mode
|   +--ro negotiated?     label-adv-mode
+--ro next-keep-alive?    uint16
+--ro received-peer-state
|   +--ro graceful-restart
|   |   +--ro enable?      boolean
|   |   +--ro reconnect-time?  uint16
|   |   +--ro recovery-time?  uint16
|   +--ro capability
|   |   +--ro end-of-lib
|   |   |   +--ro enable?  boolean
|   |   +--ro typed-wildcard-fec
|   |   |   +--ro enable?  boolean
|   |   +--ro upstream-label-assignment
|   |   |   +--ro enable?  boolean
+--ro session-holdtime
|   +--ro peer?          uint16
|   +--ro negotiated?    uint16
|   +--ro remaining?     uint16
+--ro session-state?      enumeration
+--ro tcp-connection
|   +--ro local-address?  inet:ip-address
|   +--ro local-port?     inet:port-number
|   +--ro remote-address? inet:ip-address
|   +--ro remote-port?    inet:port-number
+--ro up-time?            string
+--ro statistics
|   +--ro discontinuity-time      yang:date-and-time
|   +--ro received
|   |   +--ro total-octets?      yang:counter64
|   |   +--ro total-messages?    yang:counter64
|   |   +--ro address?           yang:counter64
|   |   +--ro address-withdraw?  yang:counter64
|   |   +--ro initialization?     yang:counter64
|   |   +--ro keepalive?         yang:counter64
|   |   +--ro label-abort-request? yang:counter64
|   |   +--ro label-mapping?     yang:counter64
|   |   +--ro label-release?     yang:counter64
|   |   +--ro label-request?     yang:counter64
|   |   +--ro label-withdraw?    yang:counter64
|   |   +--ro notification?      yang:counter64
|   +--ro sent
|   |   +--ro total-octets?      yang:counter64
|   |   +--ro total-messages?    yang:counter64

```

	+++ro address?	yang:counter64
	+++ro address-withdraw?	yang:counter64
	+++ro initialization?	yang:counter64
	+++ro keepalive?	yang:counter64
	+++ro label-abort-request?	yang:counter64
	+++ro label-mapping?	yang:counter64
	+++ro label-release?	yang:counter64
	+++ro label-request?	yang:counter64
	+++ro label-withdraw?	yang:counter64
	+++ro notification?	yang:counter64
	+++ro total-addresses?	uint32
	+++ro total-labels?	uint32
	+++ro total-fec-label-bindings?	uint32

Figure 8

6.1.2. Extended

Following is a simplified graphical representation of the extended data model for LDP operational state.

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--rw address-families
        +--rw ldp-ext:ipv6!
          +--ro ldp-ext:label-distribution-controlmode?
            |
            enumeration
          +--ro ldp-ext:bindings
            +--ro ldp-ext:address* [address]
              +--ro ldp-ext:address
                |
                inet:ipv6-address
              +--ro ldp-ext:advertisement-type?
                |
                advertised-received
            +--ro ldp-ext:peer
              +--ro ldp-ext:lsr-id? leafref
              +--ro ldp-ext:label-space-id? leafref
            +--ro ldp-ext:fec-label* [fec]
              +--ro ldp-ext:fec inet:ipv6-prefix
            +--ro ldp-ext:peer*
              [lsr-id label-space-id advertisement-type]
              +--ro ldp-ext:lsr-id leafref
              +--ro ldp-ext:label-space-id leafref
              +--ro ldp-ext:advertisement-type
                |
                advertised-received

```

```

        +---ro ldp-ext:label?
        |   rt-types:mpls-label
        +---ro ldp-ext:used-in-forwarding?   boolean
+--rw discovery
+--rw interfaces
+--rw interface* [name]
+--rw name if:interface-ref
+--rw address-families
+--rw ldp-ext:ipv6!
+--ro ldp-ext:hello-adjacencies
+--ro ldp-ext:hello-adjacency*
    [adjacent-address]
+--ro ldp-ext:adjacent-address
    inet:ipv6-address
+--ro ldp-ext:flag*
    identityref
+--ro ldp-ext:hello-holdtime
    +---ro ldp-ext:adjacent?      uint16
    +---ro ldp-ext:negotiated?    uint16
    +---ro ldp-ext:remaining?     uint16
+--ro ldp-ext:next-hello?        uint16
+--ro ldp-ext:statistics
    +---ro ldp-ext:discontinuity-time
        yang:date-and-time
    +---ro ldp-ext:hello-received?
        yang:counter64
    +---ro ldp-ext:hello-dropped?
        yang:counter64
+--ro ldp-ext:peer
    +---ro ldp-ext:lsr-id?        leafref
    +---ro ldp-ext:label-space-id? leafref
+--rw targeted
+--rw address-families
+--rw ldp-ext:ipv6!
+--ro ldp-ext:hello-adjacencies
+--ro ldp-ext:hello-adjacency*
    [local-address adjacent-address]
+--ro ldp-ext:local-address
    inet:ipv6-address
+--ro ldp-ext:adjacent-address
    inet:ipv6-address
+--ro ldp-ext:flag*
    identityref
+--ro ldp-ext:hello-holdtime
    +---ro ldp-ext:adjacent?      uint16
    +---ro ldp-ext:negotiated?    uint16
    +---ro ldp-ext:remaining?     uint16
+--ro ldp-ext:next-hello?        uint16
+--ro ldp-ext:statistics
    +---ro ldp-ext:discontinuity-time
        yang:date-and-time

```

```

|
|      +--ro ldp-ext:hello-received? yang:counter64
|      +--ro ldp-ext:hello-dropped? yang:counter64
|      +--ro ldp-ext:peer
|          +--ro ldp-ext:lsr-id?          leafref
|          +--ro ldp-ext:label-space-id?  leafref
+--rw peers
    +--rw address-families
        +--rw ldp-ext:ipv6!
            +--ro ldp-ext:hello-adjacencies
            +--ro ldp-ext:hello-adjacency*
                | [local-address adjacent-address]
                +--ro ldp-ext:local-address  inet:ipv6-address
                +--ro ldp-ext:adjacent-address
                |                               inet:ipv6-address
                +--ro ldp-ext:flag*            identityref
            +--ro ldp-ext:hello-holdtime
                |
                |      +--ro ldp-ext:adjacent?      uint16
                |      +--ro ldp-ext:negotiated?    uint16
                |      +--ro ldp-ext:remaining?     uint16
                +--ro ldp-ext:next-hello?          uint16
            +--ro ldp-ext:statistics
                |
                |      +--ro ldp-ext:discontinuity-time
                |          |                               yang:date-and-time
                |      +--ro ldp-ext:hello-received? yang:counter64
                |      +--ro ldp-ext:hello-dropped? yang:counter64
            +--ro ldp-ext:interface? if:interface-ref

```

Figure 9

6.2. States

Following are main areas for which LDP operational state is defined:

Neighbor Adjacencies

Peer

Bindings (FEC-label and address)

Capabilities

6.2.1. Adjacency state

Neighbor adjacencies are per address-family hello adjacencies that are formed with neighbors as result of LDP basic or extended discovery. In terms of organization, there is a source of discovery (e.g. interface or target address) along with its associated

parameters and one or more discovered neighbors along with neighbor discovery related parameters. For the basic discovery, there could be more than one discovered neighbor for a given source (interface), whereas there is at most one discovered neighbor for an extended discovery source (local-address and target-address). This is also to be noted that the reason for a targeted neighbor adjacency could be either an active source (locally configured targeted) or passive source (to allow any incoming extended/targeted hellos). A neighbor/adjacency record also contains session-state that helps highlight whether a given adjacency has progressed to subsequent session level or to eventual peer level.

Following captures high level tree hierarchy for neighbor adjacency state.

```

+--rw mpls-ldp!
  +--rw discovery
    +--rw interfaces
      |
      |   +--rw interface* [interface]
      |   |
      |   |   +--rw address-families
      |   |   |
      |   |   |   +--rw ipv4 (or ipv6)
      |   |   |   |
      |   |   |   |   +--ro hello-adjacencies
      |   |   |   |   |
      |   |   |   |   |   +--ro hello-adjacencies* [adjacent-address]
      |   |   |   |   |   |
      |   |   |   |   |   |   +--ro adjacent-address
      |   |   |   |   |   |   |
      |   |   |   |   |   |   |   . . . .
      |   |   |   |   |   |   |   . . . .
      |   |
      |   |   +--rw targeted
      |   |   |
      |   |   |   +--rw address-families
      |   |   |   |
      |   |   |   |   +--rw ipv4 (or ipv6)
      |   |   |   |   |
      |   |   |   |   |   +--ro hello-adjacencies
      |   |   |   |   |   |
      |   |   |   |   |   |   +--ro hello-adjacencies*
      |   |   |   |   |   |   |
      |   |   |   |   |   |   |   [local-address adjacent-address]
      |   |   |   |   |   |   |   |
      |   |   |   |   |   |   |   |   +--ro local-address
      |   |   |   |   |   |   |   |   |
      |   |   |   |   |   |   |   |   |   +--ro adjacent-address
      |   |   |   |   |   |   |   |   |   |
      |   |   |   |   |   |   |   |   |   |   . . . .
      |   |   |   |   |   |   |   |   |   |   . . . .

```

Figure 10

6.2.2. Peer state

Peer related state is presented under peers tree. This is one of the core state that provides info on the session related parameters (mode, authentication, KA timeout etc.), TCP connection info, hello adjacencies for the peer, statistics related to messages and bindings, and capabilities exchange info.

Following captures high level tree hierarchy for peer state.

```

+--rw mpls-ldp!
  +--rw peers
    +--rw peer* [lsr-id label-space-id]
      +--rw lsr-id
      +--rw label-space-id
      +--ro label-advertisement-mode
      +--ro session-state
      +--ro tcp-connection
      +--ro session-holdtime?
      +--ro up-time
      +-- . . . .
      +--ro address-families
        +--ro ipv4 (or ipv6)
          +--ro hello-adjacencies
            +--ro hello-adjacencies*
              [local-address adjacent-address]
          . . . .
      +--ro received-peer-state
        +--ro . . . .
        +--ro capability
          +--ro . . . .
      +--ro statistics
        +-- . . . .
        +-- received
          +-- . . .
        +-- sent
          +-- . . .

```

Figure 11

6.2.3. Bindings state

Binding state provides information on LDP FEC-label bindings as well as address binding for both inbound (received) as well as outbound (advertised) direction. FEC-label bindings are presented as a FEC-centric view, and address bindings are presented as an address-centric view:

```
FEC-Label bindings:
  FEC 203.0.113.1/32:
    advertised: local-label 16000
      peer 192.0.2.1:0
      peer 192.0.2.2:0
      peer 192.0.2.3:0
    received:
      peer 192.0.2.1:0, label 16002, used-in-forwarding=Yes
      peer 192.0.2.2:0, label 17002, used-in-forwarding=No
  FEC 203.0.113.2/32:
    . . . .
  FEC 198.51.100.0/24:
    . . . .

Address bindings:
  Addr 192.0.2.10:
    advertised
  Addr 192.0.2.1:
    received, peer 192.0.2.1:0
  Addr 192.0.2.2:
    received, peer 192.0.2.2:0
  Addr 192.0.2.3:
    received, peer 192.0.2.3:0
```

Figure 12

Note that all local addresses are advertised to all peers and hence no need to provide per-peer information for local address advertisement. Furthermore, note that it is easy to derive a peer-centric view for the bindings from the information already provided in this model.

Following captures high level tree hierarchy for bindings state.

```

+--rw mpls-ldp!
  +--rw global
    +--rw address-families
      +--rw ipv4 (or ipv6)
        +--ro bindings
          +--ro address* [address]
            |   +--ro address (ipv4-address or ipv6-address)
            |   +--ro advertisement-type?   advertised-received
            |   +--ro peer?                 leafref
          +--ro fec-label* [fec]
            +--ro fec (ipv4-prefix or ipv6-prefix)
            +--ro peer* [peer advertisement-type]
              +--ro peer                 leafref
              +--ro advertisement-type? advertised-received
              +--ro label?                mpls:mpls-label
              +--ro used-in-forwarding?   boolean

```

Figure 13

6.2.4. Capabilities state

LDP capabilities state comprise two types of information - global information (such as timer etc.), and per-peer information.

Following captures high level tree hierarchy for LDP capabilities state.

```

+--rw mpls-ldp!
  +--rw peers
    +--rw peer* [lsr-id label-space-id]
      +--rw lsr-id yang:dotted-quad
      +--rw label-space-id
      +--ro received-peer-state
        +--ro capability
          +--ro . . . .
          +--ro . . . .

```

Figure 14

7. Notifications

This model defines a list of notifications to inform client of important events detected during the protocol operation. These events include events related to changes in the operational state of an LDP peer, hello adjacency, and FEC etc. It is to be noted that an

LDP FEC is treated as operational (up) as long as it has at least 1 NHLFE with outgoing label.

Following is a simplified graphical representation of the data model for LDP notifications.

```

module: ietf-mpls-ldp
  notifications:
    +---n mpls-ldp-peer-event
    |   +--ro event-type?   oper-status-event-type
    |   +--ro peer
    |   |   +---ro lsr-id?           leafref
    |   |   +---ro label-space-id?  leafref
    +---n mpls-ldp-hello-adjacency-event
    |   +--ro event-type?   oper-status-event-type
    |   +--ro (hello-adjacency-type)?
    |   |   +---:(targeted)
    |   |   |   +--ro targeted
    |   |   |   |   +--ro target-address?  inet:ip-address
    |   |   +---:(link)
    |   |   |   +--ro link
    |   |   |   |   +--ro next-hop-interface?  if:interface-ref
    |   |   |   |   +--ro next-hop-address?   inet:ip-address
    +---n mpls-ldp-fec-event
    |   +--ro event-type?   oper-status-event-type
    |   +--ro prefix?      inet:ip-prefix

```

Figure 15

8. Actions

This model defines a list of rpcs that allow performing an action or executing a command on the protocol. For example, it allows to clear (reset) LDP peers, hello-adjacencies, and statistics. The model makes an effort to provide different level of control so that a user is able to either clear all, or clear all for a given type, or clear a specific entity.

Following is a simplified graphical representation of the data model for LDP actions.

```

module: ietf-mpls-ldp
  rpcs:
    +---x mpls-ldp-clear-peer
    |   +---w input
    |   |   +---w lsr-id?          leafref
    |   |   +---w label-space-id? leafref
    +---x mpls-ldp-clear-hello-adjacency
    |   +---w input
    |   |   +---w hello-adjacency
    |   |   |   +---w (hello-adjacency-type)?
    |   |   |   |   +--:(targeted)
    |   |   |   |   |   +---w targeted!
    |   |   |   |   |   |   +---w target-address?  inet:ip-address
    |   |   |   |   +--:(link)
    |   |   |   |   |   +---w link!
    |   |   |   |   |   |   +---w next-hop-interface? leafref
    |   |   |   |   |   |   +---w next-hop-address?  inet:ip-address
    +---x mpls-ldp-clear-peer-statistics
    |   +---w input
    |   |   +---w lsr-id?          leafref
    |   |   +---w label-space-id? leafref

```

Figure 16

9. YANG Specification

Following sections specify the actual YANG (module) specification for LDP constructs defined earlier in the document.

9.1. Base

This YANG module imports types defined in [RFC6991], [RFC8349], [RFC8294], [RFC8343], and [RFC8344].

```
<CODE BEGINS> file "ietf-mpls-ldp@2018-10-22.yang"
```

```
// RFC Editor: replace the above date 2018-10-22 with the date of
// publication and remove this note.
```

```

module ietf-mpls-ldp {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp";
  prefix "ldp";

  import ietf-inet-types {
    prefix "inet";

```

```
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-routing {
    prefix "rt";
    reference
        "RFC 8349: A YANG Data Model for Routing Management (NMDA
        version)";
}

import ietf-routing-types {
    prefix "rt-types";
    reference
        "RFC 8294: Common YANG Data Types for the Routing Area";
}

import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface Management";
}

import ietf-ip {
    prefix "ip";
    reference "RFC 8344: A YANG Data Model for IP Management";
}

organization
    "IETF MPLS Working Group";
contact
    "WG Web:    <http://tools.ietf.org/wg/mpls/>
    WG List:    <mailto:mpls@ietf.org>

    Editor:     Kamran Raza
                <mailto:skraza@cisco.com>

    Editor:     Rajiv Asati
                <mailto:rajiva@cisco.com>

    Editor:     Xufeng Liu
                <mailto:xufeng.liu.ietf@gmail.com>

    Editor:     Santosh Esale
                <mailto:sesale@juniper.net>
```

Editor: Xia Chen
<mailto:jescia.chenxia@huawei.com>

Editor: Himanshu Shah
<mailto:hshah@ciena.com>;

description

"This YANG module defines the essential components for the management of Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP). It is also the base model to be augmented for Multipoint LDP (mLDP).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

revision 2018-10-22 {

// RFC Editor: replace the above date 2018-10-22 with the date of
// publication and remove this note.

description

"Initial revision.";

reference

"RFC XXXX: YANG Data Model for MPLS LDP.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

}

/*

* Typedefs

*/

typedef ldp-address-family {

type identityref {

base rt:address-family;

}

description


```
    "LDP address family type.";
}

typedef duration32-inf {
    type union {
        type uint32;
        type enumeration {
            enum "infinite" {
                description "The duration is infinite.";
            }
        }
    }
    units seconds;
    description
        "Duration represented as 32 bit seconds with infinite.";
}

typedef advertised-received {
    type enumeration {
        enum advertised {
            description "Advertised information.";
        }
        enum received {
            description "Received information.";
        }
    }
    description
        "Received or advertised.";
}

typedef downstream-upstream {
    type enumeration {
        enum downstream {
            description "Downstream information.";
        }
        enum upstream {
            description "Upstream information.";
        }
    }
    description
        "Received or advertised.";
}

typedef label-adv-mode {
    type enumeration {
        enum downstream-unsolicited {
            description "Downstream Unsolicited.";
        }
    }
}
```

```
        enum downstream-on-demand {
            description "Downstream on Demand.";
        }
    }
    description
        "Label Advertisement Mode.";
}

typedef oper-status-event-type {
    type enumeration {
        enum up {
            value 1;
            description
                "Operational status changed to up.";
        }
        enum down {
            value 2;
            description
                "Operational status changed to down.";
        }
    }
    description "Operational status event type for notifications.";
}

/*
 * Identities
 */
identity adjacency-flag-base {
    description "Base type for adjacency flags.";
}

identity adjacency-flag-active {
    base adjacency-flag-base;
    description
        "This adjacency is configured and actively created.";
}

identity adjacency-flag-passive {
    base adjacency-flag-base;
    description
        "This adjacency is not configured and passively accepted.";
}

/*
 * Groupings
 */

grouping adjacency-state-attributes {
```

```
description
  "The operational state attributes of an LDP hello adjacency,
  which can used for basic and extended discoveris, in IPv4 and
  IPv6 address families.";

leaf-list flag {
  type identityref {
    base adjacency-flag-base;
  }
  description
    "On or more flags to indicate whether the adjacency is
    actively created, passively accepted, or both.";
}
container hello-holdtime {
  description
    "Containing hello holdtime state information.";
  leaf adjacent {
    type uint16;
    units seconds;
    description
      "The holdtime value learned from the adjacent LSR.";
  }
  leaf negotiated {
    type uint16;
    units seconds;
    description
      "The holdtime negotiated between this LSR and the adjacent
      LSR.";
  }
  leaf remaining {
    type uint16;
    units seconds;
    description
      "The time remaining until the holdtime timer expires.";
  }
}

leaf next-hello {
  type uint16;
  units seconds;
  description
    "The time when the next Hello message will be sent.";
}

container statistics {
  description
    "Statistics objects.";
```

```
leaf discontinuity-time {
    type yang:date-and-time;
    mandatory true;
    description
        "The time on the most recent occasion at which any one or
        more of this interface's counters suffered a
        discontinuity. If no such discontinuities have occurred
        since the last re-initialization of the local management
        subsystem, then this node contains the time the local
        management subsystem re-initialized itself.";
}

leaf hello-received {
    type yang:counter64;
    description
        "The number of Hello messages received.";
}
leaf hello-dropped {
    type yang:counter64;
    description
        "The number of Hello messages dropped.";
}
} // statistics
} // adjacency-state-attributes

grouping basic-discovery-timers {
    description
        "The timer attributes for basic discovery, used in the
        per-interface setting and in the all-interface setting.";

    leaf hello-holdtime {
        type uint16 {
            range 15..3600;
        }
        units seconds;
        default 15;
        description
            "The time interval for which a LDP link Hello adjacency
            is maintained in the absence of link Hello messages from
            the LDP neighbor";
    }
    leaf hello-interval {
        type uint16 {
            range 5..1200;
        }
        units seconds;
        default 5;
        description
```

```
        "The interval between consecutive LDP link Hello messages
        used in basic LDP discovery";
    }
} // basic-discovery-timers

grouping binding-address-state-attributes {
    description
        "Operational state attributes of an address binding, used in
        IPv4 and IPv6 address families.";

    leaf advertisement-type {
        type advertised-received;
        description
            "Received or advertised.";
    }
    container peer {
        when "../advertisement-type = 'received'" {
            description
                "Applicable for received address.";
        }
        description
            "LDP peer from which this address is received.";
        uses ldp-peer-ref;
    }
} // binding-address-state-attributes

grouping binding-label-state-attributes {
    description
        "Operational state attributes for a FEC-label binding, used in
        IPv4 and IPv6 address families.";

    list peer {
        key "lsr-id label-space-id advertisement-type";
        description
            "List of advertised and received peers.";
        uses ldp-peer-ref {
            description
                "The LDP peer from which this binding is received, or to
                which this binding is advertised.
                The peer is identified by its LDP ID, which consists of
                the LSR ID and the Label Space ID.";
        }
        leaf advertisement-type {
            type advertised-received;
            description
                "Received or advertised.";
        }
        leaf label {
```

```
    type rt-types:mpls-label;
    description
      "Advertised (outbound) or received (inbound)
       label.";
  }
  leaf used-in-forwarding {
    type boolean;
    description
      "'true' if the lable is used in forwarding.";
  }
} // peer
} // binding-label-state-attributes

grouping graceful-restart-attributes-per-peer {
  description
    "Per peer graceful restart attributes.
     On the local side, these attributes are configuration and
     operational state data. One the peer side, these attributes
     are operational state data reveiced from the peer.";

  container graceful-restart {
    description
      "Attributes for graceful restart.";
    leaf enable {
      type boolean;
      default false;
      description
        "Enable or disable graceful restart.";
    }
    leaf reconnect-time {
      type uint16 {
        range 10..1800;
      }
      units seconds;
      default 120;
      description
        "Specifies the time interval that the remote LDP peer
         must wait for the local LDP peer to reconnect after the
         remote peer detects the LDP communication failure.";
    }
    leaf recovery-time {
      type uint16 {
        range 30..3600;
      }
      units seconds;
      default 120;
      description
        "Specifies the time interval, in seconds, that the remote
```

```
        LDP peer preserves its MPLS forwarding state after
        receiving the Initialization message from the restarted
        local LDP peer.";
    }
} // graceful-restart
} // graceful-restart-attributes-per-peer

grouping ldp-interface-ref {
  description
    "Defining a reference to LDP interface.";

  leaf name {
    type if:interface-ref;
    must "(/if:interfaces/if:interface[if:name=current()]/ip:ipv4) "
      + " or "
      + "(/if:interfaces/if:interface[if:name=current()]/ip:ipv6) "
    {
      description "Interface is IPv4 or IPv6.";
    }
    description
      "The name of an LDP interface.";
  }
}

grouping ldp-peer-ref {
  description
    "An absolute reference to an LDP peer, by the LDP ID, which
    consists of the LSR ID and the Label Space ID.";

  leaf lsr-id {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
        + "ldp:peers/ldp:peer/ldp:lsr-id";
    }
    description
      "The LSR ID of the peer, as a portion of the peer LDP ID.";
  }
  leaf label-space-id {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
        + "ldp:peers/ldp:peer/ldp:label-space-id";
    }
    description
      "The Label Space ID of the peer, as a portion of the peer
      LDP ID.";
  }
} // ldp-peer-ref
```

```
grouping ldp-peer-ref-container {
  description
    "A container containing an absolute reference to an LDP peer.";

  container peer {
    description
      "Reference to an LDP peer, by the LDP ID, which consists of
       the LSR ID and the Label Space ID.";
    uses ldp-peer-ref;
  } // peer
} // ldp-peer-ref

grouping peer-attributes {
  description
    "Peer configuration attributes, used in the per-peer setting
     can in the all-peer setting.";

  leaf session-ka-holdtime {
    type uint16 {
      range 45..3600;
    }
    units seconds;
    default 180;
    description
      "The time interval after which an inactive LDP session
       terminates and the corresponding TCP session closes.
       Inactivity is defined as not receiving LDP packets from the
       peer.";
  }
  leaf session-ka-interval {
    type uint16 {
      range 15..1200;
    }
    units seconds;
    default 60;
    description
      "The interval between successive transmissions of keepalive
       packets. Keepalive packets are only sent in the absence of
       other LDP packets transmitted over the LDP session.";
  }
} // peer-attributes

grouping peer-authentication {
  description
    "Peer authentication container, used in the per-peer setting
     can in the all-peer setting.";

  container authentication {
```



```
    description
      "Containing authentication information.";
    choice auth-type-selection {
      description
        "Options for expressing authentication setting.";
      case auth-key {
        leaf md5-key {
          type string;
          description
            "MD5 Key string.";
        }
      }
    }
  } // authentication
} // peer-authentication

grouping peer-state-derived {
  description
    "The peer state information derived from the LDP protocol
    operatoins.";

  container label-advertisement-mode {
    config false;
    description "Label advertisement mode state.";
    leaf local {
      type label-adv-mode;
      description
        "Local Label Advertisement Mode.";
    }
    leaf peer {
      type label-adv-mode;
      description
        "Peer Label Advertisement Mode.";
    }
    leaf negotiated {
      type label-adv-mode;
      description
        "Negotiated Label Advertisement Mode.";
    }
  }
}
leaf next-keep-alive {
  type uint16;
  units seconds;
  config false;
  description "Time to send the next KeepAlive message.";
}

container received-peer-state {
```

```
    config false;
    description
        "Operational state information learned from the peer.";

    uses graceful-restart-attributes-per-peer;

    container capability {
        description "Configure capability.";
        container end-of-lib {
            description
                "Configure end-of-lib capability.";
            leaf enable {
                type boolean;
                description
                    "Enable end-of-lib capability.";
            }
        }
        container typed-wildcard-fec {
            description
                "Configure typed-wildcard-fec capability.";
            leaf enable {
                type boolean;
                description
                    "Enable typed-wildcard-fec capability.";
            }
        }
        container upstream-label-assignment {
            description
                "Configure upstream label assignment capability.";
            leaf enable {
                type boolean;
                description
                    "Enable upstream label assignment.";
            }
        }
    } // capability
} // received-peer-state

container session-holdtime {
    config false;
    description "Session holdtime state.";
    leaf peer {
        type uint16;
        units seconds;
        description "Peer holdtime.";
    }
    leaf negotiated {
        type uint16;
    }
}
```

```
        units seconds;
        description "Negotiated holdtime.";
    }
    leaf remaining {
        type uint16;
        units seconds;
        description "Remaining holdtime.";
    }
} // session-holdtime

leaf session-state {
    type enumeration {
        enum non-existent {
            description "NON EXISTENT state. Transport disconnected.";
        }
        enum initialized {
            description "INITIALIZED state.";
        }
        enum openrec {
            description "OPENREC state.";
        }
        enum opensent {
            description "OPENSENT state.";
        }
        enum operational {
            description "OPERATIONAL state.";
        }
    }
}
config false;
description
    "Representing the operational status of the LDP session.";
reference
    "RFC5036, Sec. 2.5.4.";
}

container tcp-connection {
    config false;
    description "TCP connection state.";
    leaf local-address {
        type inet:ip-address;
        description "Local address.";
    }
    leaf local-port {
        type inet:port-number;
        description "Local port number.";
    }
    leaf remote-address {
        type inet:ip-address;
    }
}
```

```
        description "Remote address.";
    }
    leaf remote-port {
        type inet:port-number;
        description "Remote port number.";
    }
} // tcp-connection

leaf up-time {
    type string;
    config false;
    description "Up time. The interval format in ISO 8601.";
}

container statistics {
    config false;
    description
        "Statistics objects.";

    leaf discontinuity-time {
        type yang:date-and-time;
        mandatory true;
        description
            "The time on the most recent occasion at which any one or
            more of this interface's counters suffered a
            discontinuity.  If no such discontinuities have occurred
            since the last re-initialization of the local management
            subsystem, then this node contains the time the local
            management subsystem re-initialized itself.";
    }

    container received {
        description "Inbound statistics.";
        uses statistics-peer-received-sent;
    }
    container sent {
        description "Outbound statistics.";
        uses statistics-peer-received-sent;
    }

    leaf total-addresses {
        type uint32;
        description
            "The number of learned addresses.";
    }
    leaf total-labels {
        type uint32;
        description
```

```
        "The number of learned labels.";
    }
    leaf total-fec-label-bindings {
        type uint32;
        description
            "The number of learned label-address bindings.";
    }
} // statistics
} // peer-state-derived

grouping statistics-peer-received-sent {
    description
        "Inbound and outbound statistic counters.";
    leaf total-octets {
        type yang:counter64;
        description
            "The total number of octets sent or received.";
    }
    leaf total-messages {
        type yang:counter64;
        description
            "The number of messages sent or received.";
    }
    leaf address {
        type yang:counter64;
        description
            "The number of address messages sent or received.";
    }
    leaf address-withdraw {
        type yang:counter64;
        description
            "The number of address-withdraw messages sent or received.";
    }
    leaf initialization {
        type yang:counter64;
        description
            "The number of initialization messages sent or received.";
    }
    leaf keepalive {
        type yang:counter64;
        description
            "The number of keepalive messages sent or received.";
    }
    leaf label-abort-request {
        type yang:counter64;
        description
            "The number of label-abort-request messages sent or
            received.";
```

```
    }
    leaf label-mapping {
        type yang:counter64;
        description
            "The number of label-mapping messages sent or received.";
    }
    leaf label-release {
        type yang:counter64;
        description
            "The number of label-release messages sent or received.";
    }
    leaf label-request {
        type yang:counter64;
        description
            "The number of label-request messages sent or received.";
    }
    leaf label-withdraw {
        type yang:counter64;
        description
            "The number of label-withdraw messages sent or received.";
    }
    leaf notification {
        type yang:counter64;
        description
            "The number of messages sent or received.";
    }
} // statistics-peer-received-sent

/*
 * Configuration data and operational state data nodes
 */

augment "/rt:routing/rt:control-plane-protocols" {
    description "LDP augmentation.";

    container mpls-ldp {
        presence
            "Enables the LDP protocol.";
        description
            "Containing configuration and operational data for the LDP
            protocol.";

        container global {
            description
                "Global attributes for LDP.";

            container capability {
                description
```

```
        "Containing the LDP capability data. The container is
        used for augmentations.";
    reference
        "RFC5036: Sec. 1.5.";
} // capability

container graceful-restart {
    description
        "Attributes for graceful restart.";
    leaf enable {
        type boolean;
        default false;
        description
            "Enable or disable graceful restart.";
    }
    leaf reconnect-time {
        type uint16 {
            range 10..1800;
        }
        units seconds;
        default 120;
        description
            "Specifies the time interval that the remote LDP peer
            must wait for the local LDP peer to reconnect after
            the remote peer detects the LDP communication
            failure.";
    }
    leaf recovery-time {
        type uint16 {
            range 30..3600;
        }
        units seconds;
        default 120;
        description
            "Specifies the time interval, in seconds, that the
            remote LDP peer preserves its MPLS forwarding state
            after receiving the Initialization message from the
            restarted local LDP peer.";
    }
    leaf forwarding-holdtime {
        type uint16 {
            range 30..3600;
        }
        units seconds;
        default 180;
        description
            "Specifies the time interval, in seconds, before the
            termination of the recovery phase.";
    }
}
```

```
    }  
  } // graceful-restart  
  
  leaf lsr-id {  
    type rt-types:router-id;  
    description  
      "Specify the value to act as the LDP LSR ID.  
      If this attribute is not specified, LDP uses the router  
      ID as determined by the system."  
  }  
  
  container address-families {  
    description  
      "Per address family configuration and operational state.  
      The address family can be either IPv4 or IPv6."  
    container ipv4 {  
      presence  
        "Present if IPv4 is enabled, unless the 'enable'  
        leaf is set to 'false'";  
      description  
        "Containing data related to the IPv4 address family."  
  
      leaf enable {  
        type boolean;  
        default true;  
        description  
          "'true' to enable the address family."  
      }  
  
      leaf label-distribution-controlmode {  
        type enumeration {  
          enum independent {  
            description  
              "Independent label distribution control."  
          }  
          enum ordered {  
            description  
              "Ordered label distribution control."  
          }  
        }  
        config false;  
        description  
          "Label distribution control mode."  
        reference  
          "RFC5036: LDP Specification. Sec 2.6."  
      }  
    }  
  
    // ipv4 bindings
```



```
    container bindings {
      config false;
      description
        "LDP address and label binding information.";
      list address {
        key "address";
        description
          "List of address bindings learned by LDP.";
        leaf address {
          type inet:ipv4-address;
          description
            "The IPv4 address learned from an Address
              message received from or advertised to a peer.";
        }
        uses binding-address-state-attributes;
      } // binding-address

      list fec-label {
        key "fec";
        description
          "List of FEC-label bindings learned by LDP.";
        leaf fec {
          type inet:ipv4-prefix;
          description
            "The prefix FEC value in the FEC-label binding,
              learned in a Label Mapping message received from
              or advertised to a peer.";
        }
        uses binding-label-state-attributes;
      } // fec-label
    } // bindings
  } // ipv4
} // address-families
} // global

container discovery {
  description
    "Neighbor discovery configuration and operational state.";

  container interfaces {
    description
      "A list of interfaces for LDP Basic Discovery.";
    reference
      "RFC5036: LDP Specification. Sec 2.4.1.";

    uses basic-discovery-timers;

    list interface {
```

```
key "name";
description
  "List of LDP interfaces used for LDP Basic Discovery.";
uses ldp-interface-ref;
leaf next-hello {
  type uint16;
  units seconds;
  config false;
  description "Time to send the next hello message.";
}

container address-families {
  description
    "Container for address families.";
  container ipv4 {
    presence
      "Present if IPv4 is enabled, unless the 'enable'
       leaf is set to 'false'";
    description
      "IPv4 address family.";

    leaf enable {
      type boolean;
      default true;
      description
        "Enable the address family on the interface.";
    }

    // ipv4
    container hello-adjacencies {
      config false;
      description
        "Containing a list of hello adjacencies.";

      list hello-adjacency {
        key "adjacent-address";
        config false;
        description "List of hello adjacencies.";

        leaf adjacent-address {
          type inet:ipv4-address;
          description
            "Neighbor address of the hello adjacency.";
        }

        uses adjacency-state-attributes;
        uses ldp-peer-ref-container;
      } // hello-adjacency
    }
  }
}
```

```
        } // hello-adjacencies
    } // ipv4
    } // address-families
} // list interface
} // interfaces

container targeted
{
    description
        "A list of targeted neighbors for extended discovery.";

    leaf hello-holdtime {
        type uint16 {
            range 15..3600;
        }
        units seconds;
        default 45;
        description
            "The time interval for which LDP targeted Hello
            adjacency is maintained in the absence of targeted
            Hello messages from an LDP neighbor.";
    }
    leaf hello-interval {
        type uint16 {
            range 5..3600;
        }
        units seconds;
        default 15;
        description
            "The interval between consecutive LDP targeted Hello
            messages used in extended LDP discovery.";
    }
}

container hello-accept {
    description
        "LDP policy to control the acceptance of extended
        neighbor discovery Hello messages.";

    leaf enable {
        type boolean;
        default false;
        description
            "'true' to accept; 'false' to deny.";
    }
} // hello-accept

container address-families {
    description
```

```
    "Container for address families.";
container ipv4 {
  presence
    "Present if IPv4 is enabled.";
  description
    "IPv4 address family.";

  container hello-adjacencies {
    config false;
    description
      "Containing a list of hello adjacencies.";

    list hello-adjacency {
      key "local-address adjacent-address";
      description "List of hello adjacencies.";

      leaf local-address {
        type inet:ipv4-address;
        description
          "Local address of the hello adjacency.";
      }
      leaf adjacent-address {
        type inet:ipv4-address;
        description
          "Neighbor address of the hello adjacency.";
      }

      uses adjacency-state-attributes;
      uses ldp-peer-ref-container;
    } // hello-adjacency
  } // hello-adjacencies

  list target {
    key "adjacent-address";
    description
      "Targeted discovery params.";

    leaf adjacent-address {
      type inet:ipv4-address;
      description
        "Configures a remote LDP neighbor and enables
        extended LDP discovery of the specified
        neighbor.";
    }

    leaf enable {
      type boolean;
      default true;
    }
  }
}
```

```
        description
            "Enable the target.";
    }
    leaf local-address {
        type inet:ipv4-address;
        description
            "The local address used as the source address to
            send targeted Hello messages.
            If the value is not specified, the
            transport-address is used as the source
            address.";
    }
    } // target
    } // ipv4
    } // address-families
    } // targeted
} // discovery

container peers {
    description
        "Peers configuration attributes.";

    uses peer-authentication;
    uses peer-attributes;

    list peer {
        key "lsr-id label-space-id";
        description
            "List of peers.";

        leaf lsr-id {
            type rt-types:router-id;
            description
                "The LSR ID of the peer, to identify the globally
                unique LSR. This is the first four octets of the LDP
                ID. This leaf is used together with the leaf
                'label-space-id' to form the LDP ID.";
            reference
                "RFC5036. Sec 2.2.2.";
        }
        leaf label-space-id {
            type uint16;
            description
                "The Label Space ID of the peer, to identify a specific
                label space within the LSR. This is the last two
                octets of the LDP ID. This leaf is used together with
                the leaf 'lsr-id' to form the LDP ID.";
            reference
```

```
        "RFC5036. Sec 2.2.2.";
    }

    uses peer-authentication;
    container capability {
        description
            "Per peer capability";
    }

    container address-families {
        description
            "Per-vrf per-af params.";
        container ipv4 {
            presence
                "Present if IPv4 is enabled.";
            description
                "IPv4 address family.";

            container hello-adjacencies {
                config false;
                description
                    "Containing a list of hello adjacencies.";

                list hello-adjacency {
                    key "local-address adjacent-address";
                    description "List of hello adjacencies.";

                    leaf local-address {
                        type inet:ipv4-address;
                        description
                            "Local address of the hello adjacency.";
                    }
                    leaf adjacent-address {
                        type inet:ipv4-address;
                        description
                            "Neighbor address of the hello adjacency.";
                    }
                }

                uses adjacency-state-attributes;

                leaf interface {
                    type if:interface-ref;
                    description "Interface for this adjacency.";
                }
            } // hello-adjacency
        } // hello-adjacencies
    } // ipv4
} // address-families
```

```
        uses peer-state-derived;
    } // list peer
  } // peers
} // container mpls-ldp
}

/*
 * RPCs
 */
rpc mpls-ldp-clear-peer {
  description
    "Clears the session to the peer.";
  input {
    uses ldp-peer-ref {
      description
        "The LDP peer to be cleared. If this is not provided
        then all peers are cleared.
        The peer is identified by its LDP ID, which consists of
        the LSR ID and the Label Space ID.";
    }
  }
}

rpc mpls-ldp-clear-hello-adjacency {
  description
    "Clears the hello adjacency";
  input {
    container hello-adjacency {
      description
        "Link adjacency or targettted adjacency. If this is not
        provided then all hello adjacencies are cleared";
      choice hello-adjacency-type {
        description "Adjacency type.";
        case targeted {
          container targeted {
            presence "Present to clear targeted adjacencies.";
            description
              "Clear targeted adjacencies.";
            leaf target-address {
              type inet:ip-address;
              description
                "The target address. If this is not provided then
                all targeted adjacencies are cleared";
            }
          } // targeted
        }
        case link {
          container link {

```

```

        presence "Present to clear link adjacencies.";
        description
            "Clear link adjacencies.";
        leaf next-hop-interface {
            type leafref {
                path "/rt:routing/rt:control-plane-protocols/"
                    + "mpls-ldp/discovery/interfaces/interface/name";
            }
            description
                "Interface connecting to next-hop. If this is not
                provided then all link adjacencies are cleared.";
        }
        leaf next-hop-address {
            type inet:ip-address;
            must "../next-hop-interface" {
                description
                    "Applicable when interface is specified.";
            }
            description
                "IP address of next-hop. If this is not provided
                then adjacencies to all next-hops on the given
                interface are cleared.";
        } // next-hop-address
    } // link
}

}
}
}

rpc mpls-ldp-clear-peer-statistics {
    description
        "Clears protocol statistics (e.g. sent and received
        counters).";
    input {
        uses ldp-peer-ref {
            description
                "The LDP peer whose statistics are to be cleared.
                If this is not provided then all peers' statistics are
                cleared.
                The peer is identified by its LDP ID, which consists of
                the LSR ID and the Label Space ID.";
        }
    }
}

/*
 * Notifications

```



```
*/
notification mpls-ldp-peer-event {
    description
        "Notification event for a change of LDP peer operational
        status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    uses ldp-peer-ref-container;
}

notification mpls-ldp-hello-adjacency-event {
    description
        "Notification event for a change of LDP adjacency operational
        status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    choice hello-adjacency-type {
        description
            "Interface or targeted adjacency.";
        case targeted {
            container targeted {
                description
                    "Targeted adjacency through LDP extended discovery.";
                leaf target-address {
                    type inet:ip-address;
                    description
                        "The target adjacent address learned.";
                }
            } // targeted
        }
        case link {
            container link {
                description
                    "Link adjacency through LDP basic discovery.";
                leaf next-hop-interface {
                    type if:interface-ref;
                    description
                        "The interface connecting to the adjacent next hop.";
                }
                leaf next-hop-address {
                    type inet:ip-address;
                    must "../next-hop-interface" {
                        description

```

```

        "Applicable when interface is specified.";
    }
    description
        "IP address of the next hop. This can be IPv4 or IPv6
        address.";
    }
    } // link
}
}
}

notification mpls-ldp-fec-event {
    description
        "Notification event for a change of FEC status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    leaf prefix {
        type inet:ip-prefix;
        description
            "The address prefix element of the FEC whose status
            has changed.";
    }
}
}
}

<CODE ENDS>

```

Figure 17

9.2. Extended

This YANG module imports types defined in [RFC6991], [RFC8349], [RFC8177], and [RFC8343].

```

<CODE BEGINS> file "ietf-mpls-ldp-extended@2018-10-22.yang"

// RFC Editor: replace the above date 2018-02-28 with the date of
// publication and remove this note.

module ietf-mpls-ldp-extended {
    yang-version 1.1;

    namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp-extended";

```

```
prefix "ldp-ext";

import ietf-inet-types {
  prefix "inet";
  reference "RFC 6991: Common YANG Data Types";
}

import ietf-routing {
  prefix "rt";
  reference
    "RFC 8349: A YANG Data Model for Routing Management (NMDA
    version)";
}

import ietf-key-chain {
  prefix "key-chain";
  reference "RFC 8177: YANG Data Model for Key Chains";
}

import ietf-mpls-ldp {
  prefix "ldp";
  reference "RFC XXXX: YANG Data Model for MPLS LDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

import ietf-interfaces {
  prefix "if";
  reference "RFC 8343: A YANG Data Model for Interface Management";
}

organization
  "IETF MPLS Working Group";
contact
  "WG Web:    <http://tools.ietf.org/wg/mpls/>
  WG List:    <mailto:mpls@ietf.org>

  Editor:     Kamran Raza
               <mailto:skraza@cisco.com>

  Editor:     Rajiv Asati
               <mailto:rajiva@cisco.com>

  Editor:     Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

  Editor:     Santosh Esale
               <mailto:sesale@juniper.net>
```

Editor: Xia Chen
<mailto:jescia.chenxia@huawei.com>

Editor: Himanshu Shah
<mailto:hshah@ciena.com>;

description

"This YANG module defines the extended components for the management of Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP). It is also the model to be augmented for extended Multipoint LDP (mLDP).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

revision 2018-10-22 {

// RFC Editor: replace the above date 2018-10-22 with the date of
// publication and remove this note.

description

"Initial revision.";

reference

"RFC XXXX: YANG Data Model for MPLS LDP.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

}

/*

* Features

*/

feature dual-stack-transport-pereference {

description

"This feature indicates that the system allows to configure the transport connection pereference in a dual-stack setup.";

```
}

feature capability-end-of-lib {
  description
    "This feature indicates that the system allows to configure
    LDP end-of-lib capability.";
}

feature capability-typed-wildcard-fec {
  description
    "This feature indicates that the system allows to configure
    LDP typed-wildcard-fec capability.";
}

feature capability-upstream-label-assignment {
  description
    "This feature indicates that the system allows to configure
    LDP upstream label assignment capability.";
}

feature forwarding-nexthop-config {
  description
    "This feature indicates that the system allows to configure
    forwarding nexthop on interfaces.";
}

feature graceful-restart-helper-mode {
  description
    "This feature indicates that the system supports graceful
    restart helper mode.";
}

feature key-chain {
  description
    "This feature indicates that the system supports keychain for
    authentication.";
}

feature per-interface-timer-config {
  description
    "This feature indicates that the system allows to configure
    interface hello timers at the per-interface level.";
}

feature per-peer-admin-down {
  description
    "This feature indicates that the system allows to
    administratively disable a peer.";
```

```
}

feature per-peer-graceful-restart-config {
  description
    "This feature indicates that the system allows to configure
    graceful restart at the per-peer level.";
}

feature per-peer-session-attributes-config {
  description
    "This feature indicates that the system allows to configure
    session attributes at the per-peer level.";
}

feature policy-label-assignment-config {
  description
    "This feature indicates that the system allows to configure
    policies to assign labels according to certain prefixes.";
}

feature policy-ordered-label-config {
  description
    "This feature indicates that the system allows to configure
    ordered label policies.";
}

feature policy-targeted-discovery-config {
  description
    "This feature indicates that the system allows to configure
    policies to control the acceptance of targeted neighbor
    discovery hello messages.";
}

feature session-downstream-on-demand-config {
  description
    "This feature indicates that the system allows to configure
    session downstream-on-demand";
}

/*
 * Typedefs
 */
typedef neighbor-list-ref {
  type string;
  description
    "A type for a reference to a neighbor address list.
    The string value is the name identifier for uniquely
    identifying the referenced address list, which contains a list
```

```
        of addresses that a routing policy can applied. The definition
        of such an address list is outside the scope of this
        document.";
    }

    typedef prefix-list-ref {
        type string;
        description
            "A type for a reference to a prefix list.
            The string value is the name identifier for uniquely
            identifying the referenced prefix set, which contains a list
            of prefixes that a routing policy can applied. The definition
            of such a prefix set is outside the scope of this document.";
    }

    typedef peer-list-ref {
        type string;
        description
            "A type for a reference to a peer address list.
            The string value is the name identifier for uniquely
            identifying the referenced address list, which contains a list
            of addresses that a routing policy can applied. The definition
            of such an address list is outside the scope of this
            document.";
    }

    /*
     * Identities
     */

    /*
     * Groupings
     */
    grouping address-family-ipv4-augment {
        description "Augmentation to address family IPv4.";

        uses policy-container;

        leaf transport-address {
            type inet:ipv4-address;
            description
                "The transport address advertised in LDP Hello messages.
                If this value is not specified, the LDP LSR ID is used as
                the transport address.";
            reference
                "RFC5036. Sec. 3.5.2.";
        }
    }
```

```
    } // address-family-ipv4-augment

    grouping authentication-keychain-augment {
        description "Augmentation to authentication to add keychain.";

        leaf key-chain {
            type key-chain:key-chain-ref;
            description
                "key-chain name.
                If not specified, no key chain is used.";
        }
    } // authentication-keychain-augment

    grouping capability-augment {
        description "Augmentation to capability.";

        container end-of-lib {
            if-feature capability-end-of-lib;
            description
                "Configure end-of-lib capability.";
            leaf enable {
                type boolean;
                default false;
                description
                    "Enable end-of-lib capability.";
            }
        }
        container typed-wildcard-fec {
            if-feature capability-typed-wildcard-fec;
            description
                "Configure typed-wildcard-fec capability.";
            leaf enable {
                type boolean;
                default false;
                description
                    "Enable typed-wildcard-fec capability.";
            }
        }
        container upstream-label-assignment {
            if-feature capability-upstream-label-assignment;
            description
                "Configure upstream label assignment capability.";
            leaf enable {
                type boolean;
                default false;
                description
                    "Enable upstream label assignment.";
            }
        }
    }
```



```
    }
  } // capability-augment

  grouping global-augment {
    description "Augmentation to global attributes.";

    leaf igp-synchronization-delay {
      type uint16 {
        range "0 | 3..300";
      }
      units seconds;
      default 0;
      description
        "Sets the interval that the LDP waits before notifying the
        Interior Gateway Protocol (IGP) that label exchange is
        completed so that IGP can start advertising the normal
        metric for the link.
        If the value is not specified, there is no delay.";
    }
  } // global-augment

  grouping global-forwarding-nexthop-augment {
    description
      "Augmentation to global forwarding nexthop interfaces.";

    container forwarding-nexthop {
      if-feature forwarding-nexthop-config;
      description
        "Configuration for forwarding nexthop.";

      container interfaces {
        description
          "A list of interfaces on which forwarding is disabled.";

        list interface {
          key "name";
          description
            "List of LDP interfaces used for LDP Basic Discovery.";
          uses ldp:ldp-interface-ref;
          list address-family {
            key "afi";
            description
              "Per-vrf per-af params.";
            leaf afi {
              type ldp:ldp-address-family;
              description
                "Address family type value.";
            }
          }
        }
      }
    }
  }
}
```

```
        leaf ldp-disable {
            type boolean;
            default false;
            description
                "'true' to disable LDP forwarding on the interface.";
        }
    } // address-family
} // list interface
} // interfaces
} // forwarding-nexthop
} // global-forwarding-nexthop-augment

grouping graceful-restart-augment {
    description "Augmentation to graceful restart.";

    leaf helper-enable {
        if-feature graceful-restart-helper-mode;
        type boolean;
        default false;
        description
            "Enable or disable graceful restart helper mode.";
    }
} // graceful-restart-augment

grouping interface-address-family-ipv4-augment {
    description "Augmentation to interface address family IPv4.";

    leaf transport-address {
        type union {
            type enumeration {
                enum "use-global-transport-address" {
                    description
                        "Use the transport address set at the global level
                         common for all interfaces for this address family.";
                }
                enum "use-interface-address" {
                    description
                        "Use interface address as the transport address.";
                }
            }
            type inet:ipv4-address;
        }
        default "use-global-transport-address";
        description
            "IP address to be advertised as the LDP transport address.";
    }
} // interface-address-family-ipv4-augment
```

```
grouping interface-address-family-ipv6-augment {
  description "Augmentation to interface address family IPv6.";

  leaf transport-address {
    type union {
      type enumeration {
        enum "use-global-transport-address" {
          description
            "Use the transport address set at the global level
             common for all interfaces for this address family.";
        }
        enum "use-interface-address" {
          description
            "Use interface address as the transport address.";
        }
      }
      type inet:ipv6-address;
    }
    default "use-global-transport-address";
    description
      "IP address to be advertised as the LDP transport address.";
  }
} // interface-address-family-ipv6-augment

grouping interface-augment {
  description "Augmentation to interface.";

  uses ldp:basic-discovery-timers {
    if-feature per-interface-timer-config;
  }
  leaf igp-synchronization-delay {
    if-feature per-interface-timer-config;
    type uint16 {
      range "0 | 3..300";
    }
    units seconds;
    default 0;
    description
      "Sets the interval that the LDP waits before notifying the
       Interior Gateway Protocol (IGP) that label exchange is
       completed so that IGP can start advertising the normal
       metric for the link.
       If the value is not specified, there is no delay.";
  }
} // interface-augment

grouping peer-af-policy-container {
  description
```

```
    "LDP policy attribute container under peer address-family.";
  container label-policy {
    description
      "Label policy attributes.";
    container advertise {
      description
        "Label advertising policies.";
      leaf prefix-list {
        type prefix-list-ref;
        description
          "Applies the prefix list to filter outgoing label
           advertisements.
           If the value is not specified, no prefix filter
           is applied.";
      }
    }
    container accept {
      description
        "Label advertisement acceptance policies.";
      leaf prefix-list {
        type prefix-list-ref;
        description
          "Applies the prefix list to filter incoming label
           advertisements.
           If the value is not specified, no prefix filter
           is applied.";
      }
    } // accept
  } // label-policy
} // peer-af-policy-container

grouping peer-augment {
  description "Augmentation to each peer list entry.";

  leaf admin-down {
    if-feature per-peer-admin-down;
    type boolean;
    default false;
    description
      "'true' to disable the peer.";
  }

  uses ldp:graceful-restart-attributes-per-peer {
    if-feature per-peer-graceful-restart-config;
  }

  uses ldp:peer-attributes {
    if-feature per-peer-session-attributes-config;
  }
}
```

```
    }  
  } // peer-augment  
  
  grouping peers-augment {  
    description "Augmentation to peers container.";  
  
    container session-downstream-on-demand {  
      if-feature session-downstream-on-demand-config;  
      description  
        "Session downstream-on-demand attributes.";  
      leaf enable {  
        type boolean;  
        default false;  
        description  
          "'true' if session downstream-on-demand is enabled.";  
      }  
      leaf peer-list {  
        type peer-list-ref;  
        description  
          "The name of a peer ACL, to be applied to the  
          downstream-on-demand sessions.  
          If this value is not specified, no filter is applied to  
          any downstream-on-demand sessions.";  
      }  
    }  
  }  
  
  container dual-stack-transport-pereference {  
    if-feature dual-stack-transport-pereference;  
    description  
      "The settings of peers to establish TCP connection in a  
      dual-stack setup.";  
    leaf max-wait {  
      type uint16 {  
        range "0..60";  
      }  
      default 30;  
      description  
        "The maximum wait time in seconds for preferred transport  
        connection establishment. 0 indicates no preference.";  
    }  
  }  
  
  container prefer-ipv4 {  
    presence  
      "Present if IPv4 is preferred for transport connection  
      establishment, subject to the 'peer-list' in this  
      container.";  
    description  
      "Uses IPv4 as the preferred address family for transport  
      connection establishment, subject to the 'peer-list' in  
      this container."
```

```
        If this container is not present, as a default, IPv6 is
        the preferred address family for transport connection
        establishment.";
    leaf peer-list {
        type peer-list-ref;
        description
            "The name of a peer ACL, to be applied to the IPv4
            transport connections.
            If this value is not specified, no filter is applied,
            and the IPv4 is preferred for all peers.";
    }
}
} // peers-augment

grouping policy-container {
    description
        "LDP policy attributes.";
    container label-policy {
        description
            "Label policy attributes.";
        container advertise {
            description
                "Label advertising policies.";
            container egress-explicit-null {
                description
                    "Enables an egress router to advertise an
                    explicit null label (value 0) in place of an
                    implicit null label (value 3) to the
                    penultimate hop router.";
                leaf enable {
                    type boolean;
                    default false;
                    description
                        "'true' to enable explicit null.";
                }
            }
        }
        leaf prefix-list {
            type prefix-list-ref;
            description
                "Applies the prefix list to filter outgoing label
                advertisements.
                If the value is not specified, no prefix filter
                is applied.";
        }
    } // advertise
    container accept {
        description
```

```
        "Label advertisement acceptance policies.";
    leaf prefix-list {
        type prefix-list-ref;
        description
            "Applies the prefix list to filter incoming label
             advertisements.
             If the value is not specified, no prefix filter
             is applied.";
    }
} // accept
container assign {
    if-feature policy-label-assignment-config;
    description
        "Label assignment policies";
    container independent-mode {
        description
            "Independent label policy attributes.";
        leaf prefix-list {
            type prefix-list-ref;
            description
                "Assign labels according to certain prefixes.
                 If the value is not specified, no prefix filter
                 is applied (labels are assigned to all learned
                 routes).";
        }
    } // independent-mode
    container ordered-mode {
        if-feature policy-ordered-label-config;
        description
            "Ordered label policy attributes.";
        leaf egress-prefix-list {
            type prefix-list-ref;
            description
                "Assign labels according to certain prefixes for
                 egress LSR.";
        }
    } // ordered-mode
} // assign
} // label-policy
} // policy-container

/*
 * Configuration and state data nodes
 */
// Forwarding nexthop augmentation to the global tree
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
    + "ldp:global" {
```

```
    description "Graceful forwarding nexthop augmentation.";
    uses global-forwarding-nexthop-augment;
}

// global/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:global/ldp:address-families" {
    description "Global IPv6 augmentation.";

    container ipv6 {
        presence
            "Present if IPv6 is enabled, unless the 'enable'
            leaf is set to 'false'";
        description
            "Containing data related to the IPv6 address family.";

        leaf enable {
            type boolean;
            default true;
            description
                "'true' to enable the address family.";
        }

        uses policy-container;

        leaf transport-address {
            type inet:ipv6-address;
            mandatory true;
            description
                "The transport address advertised in LDP Hello messages.";
        }

        leaf label-distribution-controlmode {
            type enumeration {
                enum independent {
                    description
                        "Independent label distribution control.";
                }
                enum ordered {
                    description
                        "Ordered label distribution control.";
                }
            }
            config false;
            description
                "Label distribution control mode.";
            reference
                "RFC5036: LDP Specification. Sec 2.6.";
        }
    }
}
```



```
    }

    // ipv6 bindings
    container bindings {
        config false;
        description
            "LDP address and label binding information.";
        list address {
            key "address";
            description
                "List of address bindings learned by LDP.";
            leaf address {
                type inet:ipv6-address;
                description
                    "The IPv6 address learned from an Address
                     message received from or advertised to a peer.";
            }
            uses ldp:binding-address-state-attributes;
        } // binding-address

        list fec-label {
            key "fec";
            description
                "List of FEC-label bindings learned by LDP.";
            leaf fec {
                type inet:ipv6-prefix;
                description
                    "The prefix FEC value in the FEC-label binding,
                     learned in a Label Mapping message received from
                     or advertised to a peer.";
            }
            uses ldp:binding-label-state-attributes;
        } // fec-label
    } // bindings
} // ipv6
}

// discovery/interfaces/interface/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:interfaces/ldp:interface/"
+ "ldp:address-families" {
    description "Interface IPv6 augmentation.";

    container ipv6 {
        presence
            "Present if IPv6 is enabled, unless the 'enable'
             leaf is set to 'false'";
        description
    }
}
```

```
        "IPv6 address family.";

    leaf enable {
        type boolean;
        default true;
        description
            "Enable the address family on the interface.";
    }

    // ipv6
    container hello-adjacencies {
        config false;
        description
            "Containing a list of hello adjacencies.";

        list hello-adjacency {
            key "adjacent-address";
            config false;
            description "List of hello adjacencies.";

            leaf adjacent-address {
                type inet:ipv6-address;
                description
                    "Neighbor address of the hello adjacency.";
            }

            uses ldp:adjacency-state-attributes;
            uses ldp:ldp-peer-ref-container;
        } // hello-adjacency
    } // hello-adjacencies
} // ipv6

// discovery/targeted/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:targeted/ldp:address-families" {
    description "Targeted discovery IPv6 augmentation.";

    container ipv6 {
        presence
            "Present if IPv6 is enabled.";
        description
            "IPv6 address family.";

        container hello-adjacencies {
            config false;
            description
                "Containing a list of hello adjacencies.";
```

```
list hello-adjacency {
  key "local-address adjacent-address";
  config false;
  description "List of hello adjacencies.";

  leaf local-address {
    type inet:ipv6-address;
    description
      "Local address of the hello adjacency.";
  }
  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Neighbor address of the hello adjacency.";
  }

  uses ldp:adjacency-state-attributes;
  uses ldp:ldp-peer-ref-container;
} // hello-adjacency
} // hello-adjacencies

list target {
  key "adjacent-address";
  description
    "Targeted discovery params.";

  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Configures a remote LDP neighbor and enables
       extended LDP discovery of the specified
       neighbor.";
  }
  leaf enable {
    type boolean;
    default true;
    description
      "Enable the target.";
  }
  leaf local-address {
    type inet:ipv6-address;
    description
      "The local address used as the source address to send
       targeted Hello messages.
       If the value is not specified, the transport-address
       is used as the source address.";
  }
} // target
```

```
    } // ipv6
}

// /peers/peer/state/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:peer/ldp:address-families" {
  description "Peer state IPv6 augmentation.";

  container ipv6 {
    presence
      "Present if IPv6 is enabled.";
    description
      "IPv6 address family.";

    container hello-adjacencies {
      config false;
      description
        "Containing a list of hello adjacencies.";

      list hello-adjacency {
        key "local-address adjacent-address";
        description "List of hello adjacencies.";

        leaf local-address {
          type inet:ipv6-address;
          description
            "Local address of the hello adjacency.";
        }
        leaf adjacent-address {
          type inet:ipv6-address;
          description
            "Neighbor address of the hello adjacency.";
        }
      }

      uses ldp:adjacency-state-attributes;

      leaf interface {
        type if:interface-ref;
        description "Interface for this adjacency.";
      }
    } // hello-adjacency
  } // hello-adjacencies
} // ipv6
}

/*
 * Configuration data and operational state data nodes
 */
```

```
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:global" {
  description "Graceful restart augmentation.";
  uses global-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:global/ldp:capability" {
  description "Capability augmentation.";
  uses capability-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:global/ldp:graceful-restart" {
  description "Graceful restart augmentation.";
  uses graceful-restart-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:global/ldp:address-families/ldp:ipv4" {
  description "Address family IPv4 augmentation.";
  uses address-family-ipv4-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:interfaces/ldp:interface" {
  description "Interface augmentation.";
  uses interface-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:interfaces/ldp:interface/"
+ "ldp:address-families/ldp:ipv4" {
  description "Interface address family IPv4 augmentation.";
  uses interface-address-family-ipv4-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:interfaces/ldp:interface/"
+ "ldp:address-families/ldp-ext:ipv6" {
  description "Interface address family IPv6 augmentation.";
  uses interface-address-family-ipv6-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:discovery/ldp:targeted/ldp:hello-accept" {
  description "Targeted discovery augmentation.";
  leaf neighbor-list {
```

```
    if-feature policy-targeted-discovery-config;
    type neighbor-list-ref;
    description
      "The name of a neighbor ACL, to accept Hello messages from
      LDP peers as permitted by the neighbor-list policy.
      If this value is not specified, targeted Hello messages from
      any source are accepted.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers" {
  description "Peers augmentation.";
  uses peers-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:authentication/ldp:auth-type-selection" {
  if-feature key-chain;
  description "Peers authentication augmentation.";
  case auth-key-chain {
    uses authentication-keychain-augment;
  }
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:peer" {
  description "Peer list entry augmentation.";
  uses peer-augment;
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:peer/ldp:authentication/"
+ "ldp:auth-type-selection" {
  if-feature key-chain;
  description "Peer list entry authentication augmentation.";
  case auth-key-chain {
    uses authentication-keychain-augment;
  }
}

augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:peer/ldp:address-families/ldp:ipv4" {
  description
    "Peer list entry IPv4 augmentation.";
  uses peer-af-policy-container;
}
```

```
augment "/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/"
+ "ldp:peers/ldp:peer/ldp:address-families/ldp-ext:ipv6" {
  description
    "Peer list entry IPv6 augmentation.";
  uses peer-af-policy-container;
}
}
```

<CODE ENDS>

Figure 18

10. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations.

It goes without saying that this specification also inherits the security considerations captured in the actual protocol specification documents, namely base LDP [RFC5036], LDP IPv6 [RFC7552], LDP

Capabilities [RFC5561], Typed Wildcard FEC [RFC5918], LDP End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

11. IANA Considerations

This document requests the registration of the following URIs in the IETF "XML registry" [RFC3688]:

URI	Registrant	XML
urn:ietf:params:xml:ns:yang:ietf-mpls-ldp	The IESG	N/A
urn:ietf:params:xml:ns:yang:ietf-mpls-ldp-extended	The IESG	N/A

This document requests the registration of the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name	Namespace	Prefix	Reference
ietf-mpls-ldp	urn:ietf:params:xml:ns:yang:ietf-mpls-ldp	ldp	This document
ietf-mpls-ldp-extended	urn:ietf:params:xml:ns:yang:ietf-mpls-ldp-extended	ldp-ext	This document

-- RFC Editor: Replace "This document" with the document RFC number at time of publication, and remove this note.

12. Acknowledgments

The authors would like to acknowledge Eddie Chami, Nagendra Kumar, Mannan Venkatesan, and Pavan Beeram for their contribution to this document. We also acknowledge Ladislav Lhotka for his useful comments as the YANG Doctor.

The review comments from Tom Petch, as part of WGLC of this document, were very useful. Some of those comments were also applied to some other YANG modules in the Routing area.

13. References

13.1. Normative References

- [RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, DOI 10.17487/RFC3478, February 2003, <<https://www.rfc-editor.org/info/rfc3478>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<https://www.rfc-editor.org/info/rfc5331>>.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, DOI 10.17487/RFC5561, July 2009, <<https://www.rfc-editor.org/info/rfc5561>>.
- [RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, DOI 10.17487/RFC5918, August 2010, <<https://www.rfc-editor.org/info/rfc5918>>.
- [RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas, "Signaling LDP Label Advertisement Completion", RFC 5919, DOI 10.17487/RFC5919, August 2010, <<https://www.rfc-editor.org/info/rfc5919>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, DOI 10.17487/RFC6389, November 2011, <<https://www.rfc-editor.org/info/rfc6389>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <<https://www.rfc-editor.org/info/rfc7552>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.

- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.

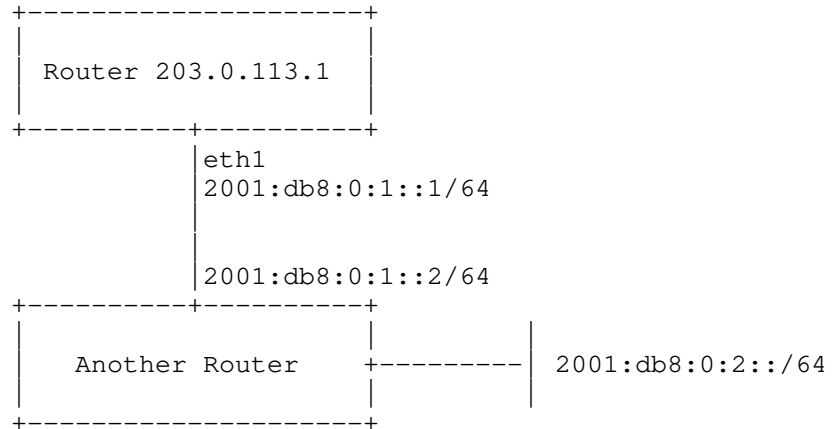
13.2. Informative References

- [I-D.ietf-mpls-mlldp-yang]
Raza, K., Liu, X., Esale, S., Andersson, L., Tantsura, J., and S. Krishnaswamy, "YANG Data Model for MPLS mLDP", draft-ietf-mpls-mlldp-yang-06 (work in progress), May 2019.
- [I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy Management", draft-ietf-rtgwg-policy-model-07 (work in progress), September 2019.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC7307] Zhao, Q., Raza, K., Zhou, C., Fang, L., Li, L., and D. King, "LDP Extensions for Multi-Topology", RFC 7307, DOI 10.17487/RFC7307, July 2014, <<https://www.rfc-editor.org/info/rfc7307>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951], containing both configuration and state data.



The configuration instance data tree for Router 203.0.113.1 in the above figure could be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "description": "An interface with LDP enabled.",
        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:1::1",
              "prefix-length": 64
            }
          ],
          "forwarding": true
        }
      }
    ]
  },
  "ietf-routing:routing": {
    "router-id": "203.0.113.1",
    "control-plane-protocols": {
      "ietf-mpls-ldp:mpls-ldp": {
        "global": {
          "address-families": {
            "ietf-mpls-ldp-extended:ipv6": {
              "enable": true
            }
          }
        },
        "discovery": {
          "interfaces": {
            "interface": [
              {
                "name": "eth1",
                "address-families": {
                  "ietf-mpls-ldp-extended:ipv6": {
                    "enable": true
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

The cooresponding operational state data for Router 203.0.113.1 could be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "description": "An interface with RIPng enabled.",
        "type": "iana-if-type:ethernetCsmacd",
        "phys-address": "00:00:5e:00:53:01",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2018-09-10T15:16:27-05:00"
        },
        "ietf-ip:ipv6": {
          "forwarding": true,
          "mtu": 1500,
          "address": [
            {
              "ip": "2001:db8:0:1::1",
              "prefix-length": 64,
              "origin": "static",
              "status": "preferred"
            },
            {
              "ip": "fe80::200:5eff:fe00:5301",
              "prefix-length": 64,
              "origin": "link-layer",
              "status": "preferred"
            }
          ],
          "neighbor": [
            {
              "ip": "2001:db8:0:1::2",
              "link-layer-address": "00:00:5e:00:53:02",
              "origin": "dynamic",
              "is-router": [null],
              "state": "reachable"
            },
            {
              "ip": "fe80::200:5eff:fe00:5302",
              "link-layer-address": "00:00:5e:00:53:02",
              "origin": "dynamic",
              "is-router": [null],
              "state": "reachable"
            }
          ]
        }
      ]
    }
  }
}
```

```

    }
  }
]
},
"ietf-routing:routing": {
  "router-id": "203.0.113.1",
  "interfaces": {
    "interface": [
      "eth1"
    ]
  },
  "control-plane-protocols": {
    "ietf-mpls-ldp:mpls-ldp": {
      "global": {
        "address-families": {
          "ietf-mpls-ldp-extended:ipv6": {
            "enable": true
          }
        }
      },
      "discovery": {
        "interfaces": {
          "interface": [
            {
              "name": "eth1",
              "address-families": {
                "ietf-mpls-ldp-extended:ipv6": {
                  "enable": true,
                  "hello-adjacencies": {
                    "hello-adjacency": [
                      {
                        "adjacent-address":
                          "fe80::200:5eff:fe00:5302",
                        "flag": ["adjacency-flag-active"],
                        "hello-holdtime": {
                          "adjacent": 15,
                          "negotiated": 15,
                          "remaining": 9
                        },
                        "next-hello": 3,
                        "statistics": {
                          "discontinuity-time":
                            "2018-09-10T15:16:27-05:00"
                        },
                        "peer": {
                          "lsr-id": "203.0.113.2",
                          "label-space-id": 0
                        }
                      }
                    ]
                  }
                }
              }
            }
          ]
        }
      }
    }
  }
}

```

```

    }
  ]
}
},
"peers": {
  "peer": [
    {
      "lsr-id": "203.0.113.2",
      "label-space-id": 0,
      "label-advertisement-mode": {
        "local": "downstream-unsolicited",
        "peer": "downstream-unsolicited",
        "negotiated": "downstream-unsolicited"
      },
      "next-keep-alive": 5,
      "session-holdtime": {
        "peer": 180,
        "negotiated": 180,
        "remaining": 78
      },
      "session-state": "operational",
      "tcp-connection": {
        "local-address": "fe80::200:5eff:fe00:5301",
        "local-port": 646,
        "remote-address": "fe80::200:5eff:fe00:5302",
        "remote-port": 646
      },
      "up-time": "P2H33M5S",
      "statistics": {
        "discontinuity-time": "2018-09-10T15:16:27-05:00"
      }
    }
  ]
}
}
}
```


Appendix B. Additional Contributors

Reshad Rahman
Cisco Systems
Email: rrahman@cisco.com

Stephane Litkowski
Cisco Systems
Email: slitkows@cisco.com

Authors' Addresses

Kamran Raza
Cisco Systems
Email: skraza@cisco.com

Rajiv Asati
Cisco Systems
Email: rajiva@cisco.com

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Xia Chen
Huawei Technologies
Email: jescia.chenxia@huawei.com

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

Danial Johari
Cisco Systems
Email: dajohari@cisco.com

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Jeff Tantsura
Apstra
Email: jefftant.ietf@gmail.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

MPLS Working Group
Internet-Draft
Updates: 8029, 8611 (if approved)
Intended status: Standards Track
Expires: April 19, 2020

L. Andersson
Bronze Dragon Consulting
T. Saad
Juniper Networks
M. Chen
Huawei Technologies
C. Pignataro
Cisco Systems
October 17, 2019

Updating the IANA MPLS LSP Ping Parameters
draft-ietf-mpls-lsp-ping-registries-update-00

Abstract

This document updates RFC 8029 and RFC 8611 that define IANA registries for MPLS LSP Ping. The updates are mostly for clarification and to align this registry with recent developments..

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 <https://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 19, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Requirement Language	3
2. Updating the Message Types, Reply Mode and Return Codes Registries	3
3. Updating the TLV and sub-TLV registries	4
3.1. General principles the LSP Ping TLV and sub-TLV registries	5
3.1.1. Unrecognized Experimental and Private TLVs and sub-TLVs	5
3.2. Changes to the LSP Ping registries	6
3.2.1. Common changes to the TLV and sub-TLV registries	6
4. Security Considerations	7
5. IANA Considerations	7
5.1. New Message Type, Reply Mode and Return Codes registries	7
5.2. Common Registration Procedures for TLVs and sub-TLVs	8
5.3. IANA assignments for TLVs and sub-TLVs	8
6. Acknowledgements	9
7. References	10
7.1. Normative References	10
7.2. Informative References	11
Authors' Addresses	12

1. Introduction

When RFC 8029 [RFC8029] was published it contained among other things updates to the "Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" IANA name space [IANA-LSP-PING].

RFC 8611 [RFC8611] updated the LSP Ping IANA registries to match RFC 8029, but the registrations can be further clarified and their definitions more precise.

This document updates RFC 8029 [[RFC8029] and RFC 8611 [RFC8611] by updating two groups of registries.

First the registries for Message Types [IANA-MT], Reply Modes [IANA-RM] and Return Codes [IANA-RC]. The changes to these registries are minor.

Second, this document updates the TLV and sub-TLV registries.

- o TLVs [IANA-TLV-reg]

- o Sub-TLVs for TLVs 1, 16 and 21 [IANA-Sub-1-16-21]
- o Sub-TLVs for TLV 6 [IANA-Sub-6]
- o Sub-TLVs for TLV 11 [IANA-Sub-11]
- o Sub-TLVs for TLV 20 [IANA-Sub-20]
- o Sub-TLVs for TLV 23 [IANA-Sub-23]
- o Sub-TLVs for TLV 27 [IANA-Sub-27]

The registry for sub-TLVs for TLV 9 [IANA-Sub-9] is not updated.

1.1. Requirement Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Updating the Message Types, Reply Mode and Return Codes Registries

The following changes are made to the Message Types, Reply Modes and Return Codes [IANA-MT] registries.

- o a small set of code points (4 code points) for experimental use is added, actually they are taken from the range for "Private Use".
- o the registration procedure "Specification Required" is changed to "RFC Required" and the note "Experimental RFC needed" is removed
- o In the listing of assignments the term "Vendor Private Use" is changed to "Private Use"
- o the registration procedures "Private Use" and "Experimental Use" are added to the table of registration procedures
- o A note "Not to be assigned" is added for the registration procedures "Private Use" and "Experimental Use"
- o In the list that captures the assignment status, the fields that are reserved, i.e. 0, Private Use and Experimental Use are clearly marked.

- * In the Return Codes [IANA-RC] registry the code point "0" already been assigned. This assignment is not changed and this registry will not have the "0" value "Reserved".

The new Registration Procedures layout and the new assignments for these registries will be found in Section 5.1.

3. Updating the TLV and sub-TLV registries

When a new LSP Ping sub-TLV registry were created by RFC 8611 [RFC8611] this registry "Sub-TLVs for TLV Type 6" [IANA-Sub-6] was set up following the intentions of RFC 8029.

The registry for "Sub-TLVs for TLV Type 6" will serve as a model to change/update the rest of the TLV and sub-TLV registries in this name space.

The registration procedures in the current registry for "Sub-TLVs for TLV Type 6" looks like this (2019-06-20). This will be used as a base-line and some additions/changes will be made as captured in the Appendixes:

Range	Registration Procedures	Note
0-16383	Standards Action	This range is for mandatory TLVs or for optional TLVs that require an error message if not recognized.
16384-31743	RFC Required	This range is for mandatory TLVs or for optional TLVs that require an error message if not recognized.
31744-32767	Private Use	Not to be assigned
32768-49161	Standards Action	This range is for optional TLVs that can be silently dropped if not recognized.
49162-64511	RFC Required	This range is for optional TLVs that can be silently dropped if not recognized.
64512-65535	Private Use	Not to be assigned

Sub-TLVs for TLV Type 6 Registration Procedures

This document adds small ranges of code points for Experimental Use to this registry and to registries listed in Section 5.2.

All registries will be changed to reflect the same model.

3.1. General principles the LSP Ping TLV and sub-TLV registries

The following principles are valid for all the LSP Ping TLV and sub-TLV IANA registries

- o all mandatory TLVs and sub-TLVs requires a response if the are not recognized
- o some optional TLVs and sub-TLVs requires a response if the are not recognized
- o some optional TLVs and sub-TLVs may be silently dropped if the are not recognized

The range of each TLV and sub-TLV registry is divided into to blocks, one with a range from 0 to 49161 for TLVs and sub-TLVs that require a response if not recognized. Another block in the range from 49161 to 65535, this block is for TLVs and sub-TLVs that may be silently dropped if not recognized.

Each of the blocks have code point spaces with the following registration procedures:

- o Standards Action
- o RFC Required
- o Experimental Use
- o Private Use

The exact defintion of registration procedures for IANA registries are found in [RFC8126]

3.1.1. Unrecognized Experimental and Private TLVs and sub-TLVs

Unrecognized TLVs and sub-TLVs for Expereimetal USe and Privagte Use are handled as any other unrecognised TLV or sub-TLV.

- o If the unrecognized TLV or sub-TLV is from the Experimental Use range (37144-37147) or from the Private Use range (31748-32767) a the Return Code of 2 ("One or more of the TLVs was not understood") will be sent in the echo response.

- o If the unrecognized TLV or sub-TLV is from the Experimental Use range (64512-64515) or from the Private Use range (64515-65535) the TLVs SHOULD be silently ignored.

IETF does not prescribe how recognized or unrecognized Experimental Use and Private Use TLVs and sub-TLVs are handled in experimental or private networks, that is up to the agency running the experiment or the private network. The statement above relates to how standard compliant implementations will treat the unrecognized TLVs and sub-TLVs from these ranges.

3.2. Changes to the LSP Ping registries

This section lists the changes to each MPLS LSP Ping Registry, in Section 5.1, Section 5.2 and Section 5.3 the changes are detailed and it is shown what the IANA registry version of the registration procedures and assignments would look like.

3.2.1. Common changes to the TLV and sub-TLV registries

The following changes are made to the TLV and sub-TLV registries.

- o two small set of code points (2 times 4 code points) for experimental use is added, actually they are take from the range for "Private Use".
- o the registration procedure "Specification Required" is changed to "RFC Required" and the note "Experimental RFC needed" is removed
- o In the listing of assignments the term "Vendor Private Use" is changed to "Private Use"
- o In the listing of assignments the range for "Experimental Use" is added
- o the registration procedures "Private Use" and "Experimental Use" are added to the table of registration procedures
- o A note "Not to be assigned" is added for the registration procedures "Experimental Use" and "Private Use"
- o In the list that capture assignment status, the fields that are reserved, i.e. 0, Experimental Use and Private Use are clearly marked.

The new Registration Procedures description and the new assignments for these registries will be found in Section 5.2 and Section 5.3.

4. Security Considerations

TBA

5. IANA Considerations

IANA is requested to update the LSP Ping name space as described in this document and documented in the Appendixies.

5.1. New Message Type, Reply Mode and Return Codes registries

This section details the updated registration procedures for Message Type, Reply Mode and Return Codes registries.

Range	Registration Procedures	Note
0-191	Standards Action	
192-247	RFC Required	
248-251	Experimental Use	Not to be assigned
252-255	Private Use	Not to be assigned

New common registration procedures

Value	Meaning	Reference
0	Reserved	This document
1-247	No changes to the existing assignments	
248-251	Reserved for Experimental Use	This document
252-255	Reserved for Private Use	[RFC8029]

Common Assignments for the Message Types, Reply Mode and Return Code registries

Note that for the Return Code registry the assignment for code point zero has been previously assigned, it is not changed but will remain:

Value	Meaning	Reference
0	No return code	[RFC8029]

Assignment for code point 0 in the Return Code registry

5.2. Common Registration Procedures for TLVs and sub-TLVs

This section describes the new registration procedures for the TLV and sub-TLV registries. The registry for sub-TLV 9 ([IANA-Sub-9] is not changed.

Range	Registration Procedures	Note
0-16383	Standards Action	This range is for mandatory TLVs or for optional TLVs that require an error message if not recognized.
16384-31743	RFC Required	This range is for mandatory TLVs or for optional TLVs that require an error message if not recognized.
37144-37147	Experimental Use	Not to be assigned
31748-32767	Private Use	Not to be assigned
32768-49161	Standards Action	This range is for optional TLVs that can be silently dropped if not recognized.
49162-64511	RFC Required	This range is for optional TLVs that can be silently dropped if not recognized.
64512-64515	Experimental Use	Not to be assigned
64515-65535	Private Use	Not to be assigned

TLV and sub-TLV Registration Procedures

5.3. IANA assignments for TLVs and sub-TLVs

The two tables in this section describes the updated IANA assignments for the TLV and sub-TLV registries. The registry for sub-TLV 9 ([IANA-Sub-9] is not changed.

Type	TLV name	Reference	sub-TLV registry
0	Reserved	This document	
1-31743	[any]	No changes to the current registry	[any]
37144-37147	Reserved for Experimental Use	This document	NA
31748-32767	Reserved for Private Use	This document	NA
32768-64511	[any]	No changes to the current registry.	[any]
64512-64515	Reserved for Experimental Use	This document	NA
64515-65535	Reserved for Private Use	This document	NA

TLV Assignments

Updated Sub-TLV assignments

Type	TLV name	Reference
0	Reserved	This document
1-31743	[any]	No changes to the current registry
37144-37147	Reserved for Experimental Use	This document
31748-32767	Reserved for Private Use	This document
32768-64511	[any]	No changes to the current registry.
64512-64515	Reserved for Experimental Use	This document
64515-65535	Reserved for Private Use	This document

Sub-TLV Assignments

6. Acknowledgements

TBA

7. References

7.1. Normative References

- [IANA-LSP-PING] "Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml>>.
- [IANA-MT] "Message Types", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#message-types>>.
- [IANA-RC] "Return Codes", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/#return-codes>>.
- [IANA-RM] "Reply Modes", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/#reply-modes>>.
- [IANA-Sub-1-16-21] "Sub-TLVs for TLV Types 1, 16, and 21", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-1-16-21>>.
- [IANA-Sub-11] "Sub-TLVs for TLV Type 11", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-11>>.
- [IANA-Sub-20] "Sub-TLVs for TLV Type 20", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-20>>.
- [IANA-Sub-23] "Sub-TLVs for TLV Type 23", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-23>>.

- [IANA-Sub-27]
"Sub-TLVs for TLV Type 27",
<<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-27>>.
- [IANA-Sub-6]
"Sub-TLVs for TLV Type 6",
<<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-6>>.
- [IANA-TLV-reg]
"TLVs", <<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#tlvs>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8611] Akiya, N., Swallow, G., Litkowski, S., Decraene, B., Drake, J., and M. Chen, "Label Switched Path (LSP) Ping and Traceroute Multipath Support for Link Aggregation Group (LAG) Interfaces", RFC 8611, DOI 10.17487/RFC8611, June 2019, <<https://www.rfc-editor.org/info/rfc8611>>.

7.2. Informative References

- [IANA-Sub-9]
"Sub-TLVs for TLV Type 9",
<<https://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml#sub-tlv-9>>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu

Tarek Saad
Juniper Networks

Email: tsaad.net@gmail.com

Mach Chen
Huawei Technologies

Email: mach.chen@huawei.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 2, 2019

K. Raza
Cisco Systems

X. Liu
Volta Networks

S. Esale
Juniper Networks

L. Andersson
Huawei Technologies

J. Tantsura
Nuage Networks

S. Krishnaswamy
Individual

May 31, 2019

YANG Data Model for MPLS mLDP
draft-ietf-mpls-mldp-yang-06

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) Multipoint Label Distribution Protocol (mLDP). The mLDP data model augments the LDP data model.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

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 <https://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 December 2, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Base and Extended	4
2. Specification of Requirements	4
3. Overview	4
3.1. Scope	5
3.2. FEC Types	6
4. Configuration	7
4.1. Configuration Hierarchy	7
4.2. mldp global container	9
4.3. Leveraging LDP containers	9
4.4. Configuration Tree	10
4.4.1. Base	10
4.4.2. Extended	11
5. Operational State	13
5.1. Base	13
5.2. Extended	14
5.3. Derived states	17
5.3.1. Root state	17
5.3.2. Bindings state	18
5.3.3. Capabilities state	21
6. Notifications	21
6.1. Base	21
6.2. Extended	22
7. Actions	23
8. Open Items	23
9. YANG Specification	23
9.1. Base	23
9.2. Extended	34
10. Security Considerations	55

11. IANA Considerations	56
12. Acknowledgments	57
13. References	57
13.1. Normative References	57
13.2. Informative References	60
Appendix A. Data Tree Example	60
Appendix B. Additional Contributors	68
Authors' Addresses	68

1. Introduction

This document introduces a YANG data model for MPLS Multipoint Label Distribution Protocol (mLDP). The mLDP model being defined here is dependent on the LDP YANG data model [I-D.ietf-mpls-ldp-yang]. This implies that an operator will need to use the base LDP module to configure and manage the control plane for mLDP. For example, an operator would enable LDP discovery on MPLS interface to establish LDP/mLDP peering on which mLDP bindings could be exchanged. Similarly, an operator could query state information for an LDP peer in order to verify peering attributes, etc.

Moreover, it is important to note here that any assumptions made in the LDP model also hold true in this document, unless otherwise explicitly stated.

Like its parent LDP data model, this mLDP model also defines the following constructs for managing the mLDP protocol:

- o Configuration
- o Operational State
- o Executables (Actions)
- o Notifications

The modeling in this document complies with the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

This document is organized to define the data model for each of the above constructs in the sequence as listed above.

1.1. Base and Extended

Like the LDP model, the configuration and state items are divided into the following two broad categories:

- o Base
- o Extended

The "base" category contains the basic and fundamental features that are covered in the mLDP base specification [RFC6388] alongwith few significant extension like targeted mLDP [RFC7060], constituting the minumum requirements for an mLDP deployment. Whereas, the "extended" category contains all other non-base features (such as recursive FEC support, protection etc.). All the items in the base category are mandatory and hence no "if-feature" is allowed under the "base" category. While "base" model support will suffice for small deployments, large deployments will require not only the "base" module support but also "extended" support for some selected and required features.

The base and extended categories are defined in their own modules `ietf-mpls-ldp` and `ietf-mpls-ldp-extended` respectively, each of which augments the LDP base model as defined within the `ietf-mpls-ldp` module [I-D.ietf-mpls-ldp-yang].

Like LDP, the mLDP "base" model configuration and state covers ipv4 address-family only, with ipv6 address-family related configuration and state be covered in the "extended" model.

In this document, when a simplified graphical representation of YANG model is presented in a tree diagrams, the meaning of the symbols in these tree diagrams is defined in [RFC8340].

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview

This document defines a YANG module named "ietf-mpls-ldp" for the mLDP YANG base data model that augments `/rt:routing/rt:control-plane-protocols/ldp:mpls-ldp` defined in [I-D.ietf-mpls-ldp-yang]. The

document also defines the "ietf-mpls-ldp-extended" YANG module that models the extended mLDP features.

The following diagram depicts high level mLDP yang tree organization and hierarchy with respect to LDP:

```

+-- rw routing
  +-- rw control-plane-protocols
    +-- rw mpls-ldp
      +-- rw some_ldp_container
        +-- rw mldp
          +-- rw ... // mldp base
          +-- ro ...
          +--
          +-- rw mldp-ext:... // mldp extended
          +-- rw ...
          +-- ro ...
          +--
      +-- ro some_ldp_container
        +-- ro mldp
          +-- ro ... // mldp base
          +-- ro ...
          +--
          +-- ro mldp-ext:... // mldp extended
          +-- ro ...
          +--

notifications:
+--- n mpls-ldp-some_event
+--- n ...

```

Figure 1

3.1. Scope

The main mLDP areas and features that are within the scope of this model are as follows:

o Base:

- * mLDP Base Specification [RFC6388]
- * Targeted mLDP [RFC7060]
- * Configured Leaf LSPs (manually provisioned)

- o Extended:

- * mLDP Recursive FEC [RFC6512]
- * mLDP Fast-Reroute (FRR):
 - + Node Protection [RFC7715]
 - + Multicast-only [RFC7431]
- * In-band Signaling:
 - + mLDP In-band Signaling [RFC6826]
 - + mLDP In-band signaling in a VRF [RFC7246]
 - + mLDP In-band Signaling with Wildcards [RFC7438]
- * Hub-and-Spoke Multipoint LSPs [RFC7140]

[Ed Note: Some of the topics in the above list are to be addressed/extended in a later revision of this document].

3.2. FEC Types

The FEC for Multipoint LSP is presented as (root-address, opaque-element). The following table lists various type of MP opaque elements with their keys, as covered in the configuration and state model:

Opaque Type	Key	RFC
Generic LSP Identifier	LSP Id	[RFC6388]
Transit IPv4 Source	Source, Group	[RFC6826]
Transit IPv6 Source	Source, Group	[RFC6826]
Transit IPv4 Bidir	RP, Group	[RFC6826]
Transit IPv6 Bidir	RP, Group	[RFC6826]
Transit VPNv4 Source	Source, Group, RD	[RFC7246]
Transit VPNv6 Source	Source, Group, RD	[RFC7246]
Transit VPNv4 Bidir	RP, Group, RD	[RFC7246]
Transit VPNv6 Bidir	RP, Group, RD	[RFC7246]
Recursive Opaque	Root	[RFC6512]
VPN-Recursive Opaque	Root, RD	[RFC6512]

Table 1: MP Opaque Types and keys

It should be noted that there are three basic types (LSP Id, Source, and Bidir) and then there are variants (VPN, recursive, VPN-recursive) on top of these basic types.

The "base" model includes only the "Generic LSP Identifier" opaque type (for ipv4), while rest of the above types are covered by the "extended" model.

4. Configuration

4.1. Configuration Hierarchy

The high-level configuration organization for the base and extended mLDP follows:

```

augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol
:
  +-- mpls-ldp
    +-- global
      +-- ...
      +-- ...
      +-- mldp
        |
        +-- ...
        +-- ...
        +-- address-families
          +-- ipv4
            |
            +-- ...
            +-- mldp-ext: ...
            +-- ...
            +-- configured-leaf-lsps
              +-- ...
              +-- ...
              +-- mldp-ext: ...
              +-- ...
          +-- mldp-ext: ipv6
            +-- ...
            +-- ...
            +-- configured-leaf-lsps
              +-- ...
              +-- ...
        +-- capability
          +-- mldp
            +-- ...
            +-- mldp-ext: ...
            +-- ...
        +-- forwarding-nexthop
          +--- interfaces
            +--- interface* [name]
              +--- mldp-ext: ...

```

Figure 2

From above hierarchy, we can categorize mLDP configuration parameters into two types:

- o Parameters that are mLDP specific
- o Parameters that leverage/extend LDP containers and parameters

The following subsections first describe the mLDP specific configuration parameters, followed by those leveraging LDP. It should be noted that these parameters are defined under their respective base or extended module as per their categorization.

4.2. mldp global container

mldp container is an augmentation of LDP global container and holds the configuration related to items that are mLDP specific. The main items under this container are:

- o mLDP enablement: To enable mLDP under a (VRF) routing instance, mldp is enabled in the mldp container under LDP. Given that mLDP requires LDP signaling, it is not sensible to allow disabling the LDP control plane under a (VRF) network-instance while requiring mLDP to be enabled for the same. However, if a user wants to only allow signaling for multipoint FECs on an LDP/mLDP enabled VRF instance, he/she can use LDP label-policies to disable unicast FECs under the VRF.
- o mLDP per-AF features: mLDP manages its own list of IP address-families and the features enabled underneath. The per-AF mLDP configuration items include:
 - * Multicast-only FRR: This enables Multicast-only FRR functionality for a given AF under mLDP. The feature allows route-policy to be configured for finer control/applicability of the feature.
 - * Recursive FEC: The recursive-fec feature [RFC6512] can be enabled per-AF with a route-policy.
 - * Configured Leaf LSPs: To provision multipoint leaf LSPs manually, a per-AF container is provided under LDP. The configuration is flexible and allows a user to specify MP LSPs of type p2mp or mp2mp with IPv4 or IPv6 root address(es) by using either LSP-Id or (S,G).

Targeted mLDP feature specification [RFC7060] does not require any mLDP specific configuration. It, however, requires LDP upstream-label-assignment capability [RFC6389] to be enabled.

4.3. Leveraging LDP containers

The mLDP configuration model leverages following configuration areas and containers that are already defined for LDP:

- o Capabilities: A new container "mldp" is defined that augments LDP's capabilities container. This new container specifies any mLDP specific capabilities and their parameters. Moreover, a new container "mldp" is also added by augmenting LDP per-peer capability container to override/control mLDP specific capabilities on a peer level. In the scope of this document, the

most important capabilities related to mLDP are p2mp, mp2mp, make-before-break, hub-and-spoke, and node-protection.

- o Discovery and Peering: mLDP requires LDP discovery and peer procedures to form mLDP peering. A peer is treated as an mLDP peer only when either P2MP or MP2MP capabilities have been successfully exchanged with the peer. If a user wish to selectively enable or disable mLDP with a LDP-enabled peer, he/she may use per-peer mLDP capabilities configuration. In most common deployments, it is desirable to disable mLDP (capabilities announcements) on a targeted-only LDP peering, where targeted-only peer is the one whose discovery sources are the targeted type only.
- o Forwarding: By default, mLDP is allowed to select any of the LDP enabled interface as a downstream interface towards a next-hop (LDP/mLDP peer) for MP LSP programming. However, a configuration option is provided to allow mLDP to exclude a given interface from such a selection. Note that such a configuration option will be useful only when there are more than one interface available for the downstream selection.

4.4. Configuration Tree

4.4.1. Base

A simplified graphical representation of the data model for mLDP base configuration follows:


```

module: ietf-mpls-ldp
  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global/ldp:cap
  ability:
    +--rw mldp
      +--rw p2mp
        | +--rw enable?    boolean
      +--rw mp2mp
        | +--rw enable?    boolean
      +--rw make-before-break
        +--rw enable?      boolean
        +--rw switchover-delay? uint16
        +--rw timeout?     uint16

  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global:
    +--rw mldp
      +--rw enable?          boolean
      +--rw address-families
        +--rw ipv4
          +--rw configured-leaf-lsps
            +--rw opaque-element-lspid
              +--rw fec-label* [root-address lsp-id]
                +--rw root-address          inet:ipv4-address
                +--rw lsp-id                uint32
                +--rw multipoint-type?      multipoint-type

```

Figure 3

4.4.2. Extended

A simplified graphical representation of the data model for mLDP extended configuration follows:

```

module: ietf-mpls-ldp
  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global/ldp:cap
  ability:
    +--rw mldp
      +--rw mldp-ext:hub-and-spoke {capability-mldp-hsmp}?
        | +--rw mldp-ext:enable?    boolean
      +--rw mldp-ext:node-protection {capability-mldp-node-protection}?
        +--rw mldp-ext:plr?          boolean
        +--rw mldp-ext:merge-point
          +--rw mldp-ext:enable?      boolean
          +--rw mldp-ext:targeted-session-teardown-delay? uint16

  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global:
    +--rw mldp
      +--rw enable?          boolean

```

```

    +--rw address-families
      +--rw ipv4
        +--rw configured-leaf-lsps
          +--rw mldp-ext:opaque-element-transit
            +--rw mldp-ext:fec-label* [root-address source-address group-
address rd recur-root-address recur-rd]
              +--rw mldp-ext:root-address          inet:ipv4-address
              +--rw mldp-ext:source-address         inet:ip-address
              +--rw mldp-ext:group-address          inet:ip-address-no-zo
ne
              +--rw mldp-ext:rd                    route-distinguisher
              +--rw mldp-ext:recur-root-address     inet:ip-address
              +--rw mldp-ext:recur-rd              route-distinguisher
              +--rw mldp-ext:multipoint-type?       mldp:multipoint-type
            +--rw mldp-ext:opaque-element-bidir
              +--rw mldp-ext:fec-label* [root-address rp group-address rd r
ecur-root-address recur-rd]
                +--rw mldp-ext:root-address         inet:ipv4-address
                +--rw mldp-ext:rp                    inet:ip-address
                +--rw mldp-ext:group-address         inet:ip-address-no-zo
ne
                +--rw mldp-ext:rd                    route-distinguisher
                +--rw mldp-ext:recur-root-address     inet:ip-address
                +--rw mldp-ext:recur-rd              route-distinguisher
                +--rw mldp-ext:multipoint-type?       mldp:multipoint-type
              +--rw mldp-ext:multicast-only-frr {mldp-mofrr}?
                +--rw mldp-ext:prefix-list?         ldp-ext:prefix-list-ref
              +--rw mldp-ext:recursive-fec
                +--rw mldp-ext:prefix-list?         ldp-ext:prefix-list-ref
          +--rw mldp-ext:ipv6
            +--rw mldp-ext:configured-leaf-lsps
              +--rw mldp-ext:opaque-element-lspid
                +--rw mldp-ext:fec-label* [root-address lsp-id]
                  +--rw mldp-ext:root-address       inet:ipv6-address
                  +--rw mldp-ext:lsp-id             uint32
                  +--rw mldp-ext:multipoint-type?   mldp:multipoint-type
                  +--rw mldp-ext:recursive-fec* [recur-root-address recur-rd
]
                    +--rw mldp-ext:recur-root-address inet:ip-address
                    +--rw mldp-ext:recur-rd          route-distinguish
r
                    +--rw mldp-ext:multipoint-type? mldp:multipoint-ty
pe
              +--rw mldp-ext:opaque-element-transit
                +--rw mldp-ext:fec-label* [root-address source-address group-
address rd recur-root-address recur-rd]
                  +--rw mldp-ext:root-address       inet:ipv6-address
                  +--rw mldp-ext:source-address     inet:ip-address
                  +--rw mldp-ext:group-address       inet:ip-address-no-zo
ne
                  +--rw mldp-ext:rd                route-distinguisher
                  +--rw mldp-ext:recur-root-address inet:ip-address
                  +--rw mldp-ext:recur-rd          route-distinguisher
                  +--rw mldp-ext:multipoint-type?   mldp:multipoint-type
                +--rw mldp-ext:opaque-element-bidir
                  +--rw mldp-ext:fec-label* [root-address rp group-address rd r
ecur-root-address recur-rd]
                    +--rw mldp-ext:root-address     inet:ipv6-address

```

```

ne
|
|      +---rw mldp-ext:rp                      inet:ip-address
|      +---rw mldp-ext:group-address           inet:ip-address-no-zo
|
|      +---rw mldp-ext:rd                      route-distinguisher
|      +---rw mldp-ext:recur-root-address      inet:ip-address
|      +---rw mldp-ext:recur-rd               route-distinguisher
|      +---rw mldp-ext:multipoint-type?       mldp:multipoint-type
+---rw mldp-ext:multicast-only-frr {mldp-mofrr}?
|   +---rw mldp-ext:prefix-list?    ldp-ext:prefix-list-ref
+---rw mldp-ext:recursive-fec
|   +---rw mldp-ext:prefix-list?    ldp-ext:prefix-list-ref

augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:peers/ldp:peer
/ldp:capability:
+---rw mldp {per-peer-capability}?
+---rw p2mp
|   +---rw enable?    boolean
+---rw mp2mp
|   +---rw enable?    boolean
+---rw make-before-break
|   +---rw enable?          boolean
|   +---rw switchover-delay? uint16
|   +---rw timeout?        uint16

augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global/ldp-ext
:forwarding-nexthop/ldp-ext:interfaces/ldp-ext:interface/ldp-ext:address-family:
+---rw mldp-disable?    boolean

```

Figure 4

5. Operational State

The operational state of mLDP can be queried and obtained from various read-only mldp "state" containers that augment ldp containers.

5.1. Base

A simplified graphical representation of the data model for mLDP base operational state follows:

```

module: ietf-mpls-ldp
  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:peers/ldp:peer
/ldp:received-peer-state/ldp:capability:
    +--ro mldp
      +--ro p2mp
        |   +--ro enable?    boolean
      +--ro mp2mp
        |   +--ro enable?    boolean
      +--ro make-before-break
        +--ro enable?    boolean

  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global:
    +--rw mldp
      +--rw enable?          boolean
      +--rw address-families
        +--rw ipv4
          +--ro roots
            +--ro root* [root-address]
              +--ro root-address    inet:ipv4-address
              +--ro is-self?        boolean
              +--ro reachability* [address interface]
                |   +--ro address    inet:ipv4-address
                |   +--ro interface  if:interface-ref
                |   +--ro peer?      -> ../../../../../../ldp:peers/pe
er/lsr-id
              +--ro bindings
                +--ro opaque-element-lspid
                  +--ro fec-label* [lsp-id]
                    +--ro lsp-id          uint32
                    +--ro multipoint-type? multipoint-type
                    +--ro peer* [direction peer advertisement-type]
                      +--ro direction      ldp:downstream-upstr
eam
                      +--ro peer            -> /rt:routing/contr
ol-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                      +--ro advertisement-type  ldp:advertised-recei
ved
                      +--ro label?          rt-types:mpls-label
                      +--ro mbb-role?        enumeration
                      +--ro mldp-ext:mofrr-role? mofrr-role

```

Figure 5

5.2. Extended

A simplified graphical representation of the data model for mLDP extended operational state follows:

```

module: ietf-mpls-ldp

  augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:peers/ldp:peer
/ldp:received-peer-state/ldp:capability:

```

```

    +--ro mldp
      +--ro mldp-ext:hub-and-spoke
        |   +--ro mldp-ext:enable?    boolean
      +--ro mldp-ext:node-protection
        +--ro mldp-ext:plr?            boolean
        +--ro mldp-ext:merge-point?    boolean

augment /rt:routing/rt:control-plane-protocols/ldp:mpls-ldp/ldp:global:
  +--rw mldp
    +--rw enable?                      boolean
    +--rw address-families
      +--rw ipv4
        +--ro roots
          +--ro root* [root-address]
            +--ro root-address          inet:ipv4-address
          +--ro bindings
            +--ro opaque-element-lspid
              |   +--ro mldp-ext:recursive-fec* [recur-root-address re
cur-rd]
              |   |   +--ro mldp-ext:recur-root-address    inet:ip-addr
ess
              |   |   +--ro mldp-ext:recur-rd              route-distin
guisher
              |   |   +--ro mldp-ext:multipoint-type?      mldp:multipo
int-type
              |   |   +--ro mldp-ext:peer* [direction peer advertisemen
t-type]
              |   |   +--ro mldp-ext:direction            ldp:downs
tream-upstream
              |   |   +--ro mldp-ext:peer                  -> /rt:ro
uting/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
              |   |   +--ro mldp-ext:advertisement-type    ldp:adver
tised-received
              |   |   +--ro mldp-ext:label?                rt-types:
mpls-label
              |   |   +--ro mldp-ext:mbb-role?              enumerati
on
              |   |   +--ro mldp-ext:mofrr-role?            mofrr-rol
e
              |   +--ro mldp-ext:opaque-element-transit
                |   +--ro mldp-ext:fec-label* [source-address group-address
rd recur-root-address recur-rd]
                |   |   +--ro mldp-ext:source-address      inet:ip-address
                |   |   +--ro mldp-ext:group-address        inet:ip-address
-no-zone
                |   |   +--ro mldp-ext:rd                  route-distingui
sher
                |   |   +--ro mldp-ext:recur-root-address  inet:ip-address
                |   |   +--ro mldp-ext:recur-rd            route-distingui
sher
                |   |   +--ro mldp-ext:multipoint-type?    mldp:multipoint
-type
                |   |   +--ro mldp-ext:peer* [direction peer advertisement-t
ype]
                |   |   +--ro mldp-ext:direction            ldp:downstre
am-upstream
                |   |   +--ro mldp-ext:peer                  -> /rt:routi
ng/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                |   |   +--ro mldp-ext:advertisement-type  ldp:advertis
ed-received
                |   |   +--ro mldp-ext:label?              rt-types:mpl
s-label
                |   |   +--ro mldp-ext:mbb-role?            enumeration

```

			+++ro mldp-ext:mofrr-role?	mofrr-role
			+++ro mldp-ext:opaque-element-bidir	
ot-address			+++ro mldp-ext:fec-label* [rp group-address rd recur-ro	
			+++ro mldp-ext:rp	inet:ip-address
-no-zone			+++ro mldp-ext:group-address	inet:ip-address
sher			+++ro mldp-ext:rd	route-distingui

```

|
|      +--ro mldp-ext:recur-root-address    inet:ip-address
|      +--ro mldp-ext:recur-rd              route-distingui
sher
|
|      +--ro mldp-ext:multipoint-type?       mldp:multipoint
-type
|
|      +--ro mldp-ext:peer* [direction peer advertisement-t
ype]
|
|      +--ro mldp-ext:direction              ldp:downstre
am-upstream
|
|      +--ro mldp-ext:peer                    -> /rt:routi
ng/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
|
|      +--ro mldp-ext:advertisement-type     ldp:advertis
ed-received
|
|      +--ro mldp-ext:label?                 rt-types:mpl
s-label
|
|      +--ro mldp-ext:mbb-role?               enumeration
|      +--ro mldp-ext:mofrr-role?            mofrr-role
+--rw mldp-ext:ipv6
  +--ro mldp-ext:roots
    +--ro mldp-ext:root* [root-address]
      +--ro mldp-ext:root-address    inet:ipv6-address
      +--ro mldp-ext:is-self?        boolean
      +--ro mldp-ext:reachability* [address interface]
        +--ro mldp-ext:address        inet:ipv6-address
        +--ro mldp-ext:interface      if:interface-ref
        +--ro mldp-ext:peer?          -> ../../../../../../ldp
:peers/peer/lsr-id
  +--ro mldp-ext:bindings
    +--ro mldp-ext:opaque-element-lspid
      +--ro mldp-ext:fec-label* [lsp-id]
        +--ro mldp-ext:lsp-id          uint32
        +--ro mldp-ext:multipoint-type? mldp:multipoint-ty
pe
      |
      | +--ro mldp-ext:peer* [direction peer advertisement-t
ype]
      |
      | | +--ro mldp-ext:direction          ldp:downstre
am-upstream
      | |
      | | +--ro mldp-ext:peer                -> /rt:routi
ng/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
      | |
      | | +--ro mldp-ext:advertisement-type  ldp:advertis
ed-received
      | |
      | | +--ro mldp-ext:label?              rt-types:mpl
s-label
      | |
      | | +--ro mldp-ext:mbb-role?            enumeration
      | | +--ro mldp-ext:mofrr-role?          mofrr-role
      | +--ro mldp-ext:recursive-fec* [recur-root-address re
cur-rd]
      |
      | +--ro mldp-ext:recur-root-address    inet:ip-addr
ess
      |
      | +--ro mldp-ext:recur-rd              route-distin
guisher
      |
      | +--ro mldp-ext:multipoint-type?       mldp:multipo
int-type
      |
      | +--ro mldp-ext:peer* [direction peer advertisemen
t-type]
      |
      | +--ro mldp-ext:direction              ldp:downs
tream-upstream
      |
      | +--ro mldp-ext:peer                    -> /rt:ro
uting/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
      |
      | +--ro mldp-ext:advertisement-type     ldp:adver
tised-received
      |
      | +--ro mldp-ext:label?                 rt-types:
mpls-label

```

on		+++ro mldp-ext:mbb-role?	enumerati
e		+++ro mldp-ext:mofrr-role?	mofrr-rol
		+++ro mldp-ext:opaque-element-transit	
rd recur-root-address		+++ro mldp-ext:fec-label* [source-address group-address	
		+++ro mldp-ext:source-address	inet:ip-address
-no-zone		+++ro mldp-ext:group-address	inet:ip-address
sher		+++ro mldp-ext:rd	route-distingui
		+++ro mldp-ext:recur-root-address	inet:ip-address

sher		+++ro mldp-ext:recur-rd	route-distingui
-type		+++ro mldp-ext:multipoint-type?	mldp:multipoint
ype]		+++ro mldp-ext:peer* [direction peer advertisement-t	
am-upstream		+++ro mldp-ext:direction	ldp:downstre
ng/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id		+++ro mldp-ext:peer	-> /rt:routi
ed-received		+++ro mldp-ext:advertisement-type	ldp:advertis
s-label		+++ro mldp-ext:label?	rt-types:mpl
		+++ro mldp-ext:mbb-role?	enumeration
		+++ro mldp-ext:mofrr-role?	mofrr-role
		+++ro mldp-ext:opaque-element-bidir	
ot-address recur-rd]		+++ro mldp-ext:fec-label* [rp group-address rd recur-ro	
		+++ro mldp-ext:rp	inet:ip-address
-no-zone		+++ro mldp-ext:group-address	inet:ip-address
sher		+++ro mldp-ext:rd	route-distingui
		+++ro mldp-ext:recur-root-address	inet:ip-address
sher		+++ro mldp-ext:recur-rd	route-distingui
-type		+++ro mldp-ext:multipoint-type?	mldp:multipoint
ype]		+++ro mldp-ext:peer* [direction peer advertisement-t	
am-upstream		+++ro mldp-ext:direction	ldp:downstre
ng/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id		+++ro mldp-ext:peer	-> /rt:routi
ed-received		+++ro mldp-ext:advertisement-type	ldp:advertis
s-label		+++ro mldp-ext:label?	rt-types:mpl
		+++ro mldp-ext:mbb-role?	enumeration
		+++ro mldp-ext:mofrr-role?	mofrr-role

Figure 6

5.3. Derived states

The main areas for which mLDP operational derived state is defined are:

- o Root
- o Bindings (FEC-label)
- o Capabilities

5.3.1. Root state

The root address is a fundamental construct for MP FEC bindings and LSPs. The root state provides information on all the known roots in a given address-family and their root reachability information (as learnt from RIB). In case of multi-path reachability to a root, the

selection of the upstream path is done on per-LSP basis at the time of LSP setup. Similarly, when protection mechanisms like Make-

before-break (MBB) or Multicast-only FRR (MoFRR) are in place, the path designation as active/standby or primary/backup is also done on per-LSP basis. It should be noted that a given root can be shared amongst multiple P2MP and/or MP2MP LSPs. Moreover, an LSP can be signaled to more than one root for Root Node Redundancy (RNR) purposes.

The following diagram illustrates a root database on a branch/transit LSR:

```
root 203.0.113.1:
  path1:
    RIB: GigEthernet 1/0, 198.51.100.1;
    LDP: peer 192.0.2.1:0
  path2:
    RIB: GigEthernet 2/0, 198.51.100.16;
    LDP: peer 192.0.2.2:0

root 203.0.113.2:
  path1:
    RIB: 198.51.100.100;          (NOTE: This is a recursive path)
    LDP: peer 192.0.2.100:0      (NOTE: T-mLDP peer)

root . . . .
```

Figure 7

A root entry on a root LSR itself will be presented as follows:

```
root 203.0.113.10:
  is-self
```

Figure 8

5.3.2. Bindings state

Binding state provides information on mLDP FEC-label bindings for both the P2MP and MP2MP FEC types. Like LDP, the FEC-label binding derived state is presented in a FEC-centric view per address-family, and provides information on both inbound (received) and outbound (advertised) bindings. The FEC is presented as (root-address, opaque-element-data) as described earlier in section Section 3.2, and the direction (upstream or downstream) is picked with respect to root reachability. In case of MBB or/and MoFRR, the role of a given peer

binding is also provided with respect to MBB (active or standby) or/ and MoFRR (primary or backup).

A high-level tree hierarchy for mLDP bindings state follows:

```

+--rw mpls-ldp!
+--rw global
+--rw mldp
+--rw address-families
+--rw ipv4 (or ipv6)
+--ro state
+--ro roots
+--ro root* [root-address]
+--ro ....
+--ro bindings
+--ro opaque-element-xxx
|   +--ro fec-label* [type-specific-key]
|   |   +--ro some_key_1 ...
|   |   +--ro some_key_2 ...
|   |   +--ro multipoint-type?          multipoint-type
|   |   +--ro peer* [direction peer advertisement-type]
|   |   |   +--ro direction          ldp:downstream-ups
|   |   |   +--ro peer                leafref
|   |   |   +--ro advertisement-type  ldp:advertised-rec
|   |   +--ro label?                  mpls:mpls-label
|   |   +--ro mbb-role?                enumeration
|   |   +--ro mldp-ext:mofrr-role?     mofrr-role
+--ro opaque-element-yyy
|   +--ro fec-label* [type-specific-key]
|   +--ro some_key_1 ...
...

```

Figure 9

mLDP binding state is organized and presented per root address, and hence the bindings container is under a root node in the model. The bindings state is made available for FECs pertaining to different types of opaque elements, with some state available under the "base" tree and the rest under the "extended" tree.

In the above tree, the various opaque types along with their type specific key(s) refer to the table Table 1, as captured earlier in the document. For example, if the opaque type is a Generic LSP Identifier, then the type-specific-key will be a uint32 LSP-Id key. Please see the complete model for all other types.

It is important to note the following:

- o The address-family ipv4/ipv6 applies to "root" address in the mLDP binding tree. The other addresses (source, group, Rendezvous-Point etc.) do not have to be of the same address family type as the root.
- o The "recur-root-address" field applies to the Recursive opaque type, and the (recur-root-address, recur-rd) fields applies to the VPN-Recursive opaque types as defined in [RFC6512].
- o In case of a recursive FEC, the address-family of the recur-root-address could be different than the address-family of the root address of the original encapsulated MP FEC.

The following diagram illustrates the FEC-label binding information structure for a P2MP (Transit IPv4 Source type) LSP on a branch/transit LSR:

```
FEC (root 203.0.113.1, S=198.51.100.1, G=224.1.1.1):
  type: p2mp
  upstream:
    advertised:
      peer 192.0.2.1:0, label 16000 (local)
  downstream:
    received:
      peer 192.0.2.2:0, label 17000 (remote)
      peer 192.0.2.3:0, label 18000 (remote)
```

Figure 10

The following diagram illustrates the FEC-label binding information structure for a similar MP2MP LSP on a branch/transit LSR:

```
FEC (root 203.0.113.2, RP=198.51.100.2, G=224.1.1.1):
  type: mp2mp
  upstream:
    advertised:
      peer 192.0.2.1:0, label 16000 (local)
    received:
      peer 192.0.2.1:0, label 17000 (remote)
  downstream:
    advertised:
      peer 192.0.2.2:0, label 16001 (local), MBB role=active
      peer 192.0.2.3:0, label 16002 (local), MBB role=standby
    received:
      peer 192.0.2.2:0, label 17001 (remote)
      peer 192.0.2.3:0, label 18001 (remote)
```

Figure 11

5.3.3. Capabilities state

Like LDP, mLDP capabilities state comprise two types of information:

- o global: augments `ldp:global/ldp:state/ldp:capability`.
- o per-peer: augments `ldp:peers/ldp:peer/ldp:state/ldp:capability`

6. Notifications

The mLDP notification module consists of notifications related to changes in the operational state of an mLDP FEC.

6.1. Base

A simplified graphical representation of the base data model for mLDP notifications follows:

```

module: ietf-mpls-mldp
notifications:
  +---n mpls-mldp-fec-event
    +--ro event-type?                                ldp:oper-status-event-type
    +--ro (opaque-element)?
      +---:(opaque-element-lspid)
        +--ro opaque-element-lspid
          +--ro root-address?                        inet:ip-address
          +--ro lsp-id?                              uint32
          +--ro multipoint-type?                    multipoint-type
          +--ro mldp-ext:recursive-fec
            +--ro mldp-ext:recur-root-address?      inet:ip-address
            +--ro mldp-ext:recur-rd?                route-distinguisher
            +--ro mldp-ext:multipoint-type?         mldp:multipoint-type

```

Figure 12

6.2. Extended

A simplified graphical representation of the extended data model for mLDP notifications follows:

```

module: ietf-mpls-mldp
notifications:
  +---n mpls-mldp-fec-event
    +--ro event-type?                               ldp:oper-status-event-type
    +--ro (opaque-element)?
      +---:(mldp-ext:opaque-element-transit)
        +--ro mldp-ext:opaque-element-transit
          +--ro mldp-ext:root-address?             inet:ip-address
          +--ro mldp-ext:source-address?            inet:ip-address
          +--ro mldp-ext:group-address?             inet:ip-address-no-zone
          +--ro mldp-ext:rd?                        route-distinguisher
          +--ro mldp-ext:recur-root-address?        inet:ip-address
          +--ro mldp-ext:recur-rd?                  route-distinguisher
          +--ro mldp-ext:multipoint-type?           mldp:multipoint-type
      +---:(mldp-ext:opaque-element-bidir)
        +--ro mldp-ext:opaque-element-bidir
          +--ro mldp-ext:root-address?             inet:ip-address
          +--ro mldp-ext:rp?                       inet:ip-address
          +--ro mldp-ext:group-address?             inet:ip-address-no-zone
          +--ro mldp-ext:rd?                        route-distinguisher
          +--ro mldp-ext:recur-root-address?        inet:ip-address
          +--ro mldp-ext:recur-rd?                  route-distinguisher
          +--ro mldp-ext:multipoint-type?           mldp:multipoint-type

```

Figure 13

7. Actions

Currently, no RPCs/actions are defined for mLDP.

8. Open Items

A list of open items that are to be addressed in future revisions of this document follows:

- o Specify default values for configuration parameters

9. YANG Specification

The YANG definition, i.e., the modules, for mLDP constructs defined earlier in this document are included in the subsections below.

9.1. Base

This YANG module imports types defined in [RFC6991], [RFC8343], [RFC8349], [I-D.ietf-mpls-ldp-yang], and [RFC8294].


```
<CODE BEGINS> file "ietf-mpls-mldp@2018-10-22.yang"
// RFC Editor: replace the above date with the date of
// publication and remove this note.

module ietf-mpls-mldp {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-mldp";
  prefix "mldp";

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-mpls-ldp {
    prefix "ldp";
    reference "RFC XXXX: A YANG Data Model for MPLS LDP";
  }
  // RFC Editor: replace the XXXX with actual LDP YANG RFC number at
  // time of publication and remove this note.

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management (NMDA
      version)";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  organization
    "IETF MPLS Working Group";
  contact
    "WG Web:  <http://tools.ietf.org/wg/mpls/>
    WG List:  <mailto:mpls@ietf.org>

    Editor:   Kamran Raza
              <mailto:skraza@cisco.com>
```

Editor: Sowmya Krishnaswamy
<mailto:krishnaswamy.sowmya@gmail.com>

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Santosh Esale
<mailto:sesale@juniper.net>

Editor: Loa Andersson
<mailto:loa@pi.nu>

Editor: Jeff Tantsura
<mailto:jefftant.ietf@gmail.com>;

description

"This YANG module defines the essential components for the management of Multi-Protocol Label Switching (MPLS) Multipoint LDP (mLDP).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

revision 2018-10-22 {
// RFC Editor: replace the above date 2018-10-22 with the date of
// publication and remove this note.

description

"Initial revision.";

reference

"RFC XXXX: Base YANG Data Model for MPLS mLDP";
// RFC Editor: replace XXXX with actual RFC number and remove
// this note

}

```
/*
 * Typedefs
 */
typedef multipoint-type {
  type enumeration {
    enum p2mp {
      description "Point to multipoint";
    }
    enum mp2mp {
      description "Multipoint to multipoint";
    }
  }
  description
    "The type of a multipoint LSP: either Point to multipoint
    (p2mp) or Multipoint to multipoint (mp2mp)";
}

/*
 * Groupings
 */
grouping mldp-capabilities {
  description
    "A grouping describing the protocol capabilities of mLDP";
  container p2mp {
    description
      "Configuration and state information for the
      point-to-multipoint capability";
    leaf enable {
      type boolean;
      description
        "'true' to enable the point-to-multipoint capability";
    }
  }
  container mp2mp {
    description
      "Configuration and state information for the
      multipoint-to-multipoint capability";
    leaf enable {
      type boolean;
      description
        "'true' to enable the multipoint-to-multipoint capability";
    }
  }
  container make-before-break {
    description
      "Configuration and state information for the
      make-before-break capability.";
    leaf enable {
```

```
        type boolean;
        description
            "'true' to enable the make-before-break capability";
    }
    leaf switchover-delay {
        type uint16;
        units seconds;
        description
            "Switchover delay in seconds";
    }
    leaf timeout {
        type uint16;
        units seconds;
        description
            "Timeout in seconds";
    }
} // mldp-capabilities

grouping mldp-binding-label-peer-state-attributes {
    description
        "mLDP label binding per peer attributes";
    leaf direction {
        type ldp:downstream-upstream;
        description
            "Downstream or upstream";
    }
    leaf peer {
        type leafref {
            path
                "/rt:routing/rt:control-plane-protocols/"
                + "ldp:mpls-ldp/ldp:peers/ldp:peer/ldp:lsr-id";
        }
        description
            "LDP peer from which this binding is received,
            or to which this binding is advertised.";
    }
    leaf advertisement-type {
        type ldp:advertised-received;
        description
            "Advertised or received";
    }
    leaf label {
        type rt-types:mpls-label;
        description
            "Advertised (outbound) or received (inbound) label";
    }
    leaf mbb-role {
```

```
    when "../direction = 'upstream'" {
        description
            "This leaf is used for upstream only.";
    }
    type enumeration {
        enum none {
            description "Make-Before-Break (MBB) is not enabled";
        }
        enum active {
            description "This LSP is active.";
        }
        enum inactive {
            description "This LSP is inactive.";
        }
    }
    description
        "The MBB status of this LSP";
}
} // mldp-binding-label-peer-state-attributes

grouping mldp-binding-label-state-attributes {
    description
        "mLDP label binding attributes";
    list peer {
        key "direction peer advertisement-type";
        description
            "List of advertised and received peers";
        uses mldp-binding-label-peer-state-attributes;
    } // peer
} // mldp-binding-label-state-attributes

/*
 * Configuration data and operational state data nodes
 */
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/global/ldp:capability" {
    description "Augmentation for mLDP global capability";
    container mldp {
        description
            "This container contains the configuration and state
            information for multipoint LDP capabilities.";
        uses mldp-capabilities;
    }
}

/*
 * Operational state data nodes
 */
```

```
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:peers/ldp:peer/ldp:received-peer-state/"
+ "ldp:capability" {
  description
    "Augmentation for MLDP received peer state capability";
  container mldp {
    description
      "Operational state information for the protocol capabilities
      of mLDP";

    container p2mp {
      description
        "Operational state information for the point-to-multipoint
        capability";
      leaf enable {
        type boolean;
        description
          "'true' to enable the point-to-multipoint capability";
      }
    }
    container mp2mp {
      description
        "Operational state information for the
        multipoint-to-multipoint capability";
      leaf enable {
        type boolean;
        description
          "'true' to enable the multipoint-to-multipoint
          capability";
      }
    }
    container make-before-break {
      description
        "Operational state information for the make-before-break
        capability";
      leaf enable {
        type boolean;
        description
          "'true' to enable the make-before-break capability";
      }
    }
  } // mldp
}

/*
 * Global augmentation
 */
augment "/rt:routing/rt:control-plane-protocols/"
```

```
+ "ldp:mpls-ldp/ldp:global" {
  description "MLDP global augmentation.";
  container mldp {
    description
      "mLDP attributes at per instance level. Defining
       attributes here does not enable any MP capabilities.
       MP capabilities need to be explicitly enabled under
       container capability.";

    leaf enable {
      type boolean;
      description
        "'true' to enable mLDP";
    }

    container address-families {
      description
        "Per address family parameters";

      container ipv4 {
        description
          "IPv4 information";
        container roots {
          config false;
          description
            "IPv4 multicast LSP roots";
          list root {
            key "root-address";
            description
              "List of roots for configured multicast LSPs";

            leaf root-address {
              type inet:ipv4-address;
              description
                "Root address.";
            }

            leaf is-self {
              type boolean;
              description
                "I am the root node.";
            }
          }

          list reachability {
            key "address interface";
            description
              "A next-hop for reachability to root,
               as a RIB view";
          }
        }
      }
    }
  }
}
```

```
leaf address {
  type inet:ipv4-address;
  description
    "The next-hop address to reach root";
}
leaf interface {
  type if:interface-ref;
  description
    "Interface connecting to next-hop";
}
leaf peer {
  type leafref {
    path
      "../.../.../.../.../.../ldp:peers/"
      + "ldp:peer/ldp:lsr-id";
  }
  description
    "LDP peer from which this next-hop can be
    reached";
}
}

container bindings {
  description
    "mLDP FEC to label bindings";
  container opaque-element-lspid {
    description
      "The type of opaque value element is the generic
      LSP identifier";
    reference
      "RFC6388: Label Distribution Protocol
      Extensions for Point-to-Multipoint and
      Multipoint-to-Multipoint Label Switched
      Paths.";
    list fec-label {
      key
        "lsp-id";
      description
        "List of FEC to label bindings";
      leaf lsp-id {
        type uint32;
        description "ID to identify the LSP";
      }
      leaf multipoint-type {
        type multipoint-type;
        description
          "The type of mutipoint: p2mp or mp2mp";
      }
    }
  }
}
```



```

        uses mldp-binding-label-state-attributes;
    } // fec-label
    } // opaque-element-lspid
    } // bindings
    } // list root
} // roots

container configured-leaf-lsps {
  description
    "Configured multicast LSPs.";
  container opaque-element-lspid {
    description
      "The type of opaque value element is
       the generic LSP identifier";
    reference
      "RFC6388: Label Distribution Protocol
       Extensions for Point-to-Multipoint and
       Multipoint-to-Multipoint Label Switched
       Paths.";
    list fec-label {
      key
        "root-address lsp-id";
      description
        "List of FEC to label bindings.";
      leaf root-address {
        type inet:ipv4-address;
        description
          "Root address";
      }
      leaf lsp-id {
        type uint32;
        description "ID to identify the LSP";
      }
      leaf multipoint-type {
        type multipoint-type;
        description
          "The type of mutipoint: p2mp or mp2mp";
      }
    } // fec-label
  } // opaque-element-lspid
} // configured-leaf-lsps
} // ipv4
} // list address-family
} // mldp
}

/*
 * Notifications

```

```

    */
notification mpls-mlldp-fec-event {
  description
    "Notification event for a change of FEC status";
  leaf event-type {
    type ldp:oper-status-event-type;
    description "Event type";
  }
  choice opaque-element {
    description
      "The type of opaque value element";
    case opaque-element-lspid {
      container opaque-element-lspid {
        description
          "The type of opaque value element is
            the generic LSP identifier";
        reference
          "RFC6388: Label Distribution Protocol
            Extensions for Point-to-Multipoint and
            Multipoint-to-Multipoint Label Switched
            Paths.";
        leaf root-address {
          type inet:ip-address;
          description
            "Root address.";
        }
        leaf lsp-id {
          type uint32;
          description "ID to identify the LSP";
        }
        leaf multipoint-type {
          type multipoint-type;
          description
            "The type of mutipoint: p2mp or mp2mp";
        }
      } // container opaque-element-lspid
    }
  }
}
}
<CODE ENDS>

```

Figure 14

9.2. Extended

This YANG module imports types defined in [RFC6991], [RFC8343], [RFC8349], [I-D.ietf-mpls-ldp-yang], and [RFC8294].

```
<CODE BEGINS> file "ietf-mpls-mldp-extended@2018-10-22.yang"
// RFC Editor: replace the above date with the date of
// publication and remove this note.

module ietf-mpls-mldp-extended {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-mldp-extended";
  prefix "mldp-ext";

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management (NMDA
      version)";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  import ietf-mpls-ldp {
    prefix "ldp";
    reference "RFC XXXX: A YANG Data Model for MPLS LDP";
  }
  // RFC Editor: replace the XXXX with actual LDP YANG RFC number at
  // time of publication and remove this note.
  }

  import ietf-mpls-ldp-extended {
```

```
    prefix "ldp-ext";
    reference "RFC XXXX: A YANG Data Model for MPLS LDP";
// RFC Editor: replace the XXXX with actual LDP YANG RFC number at
// time of publication and remove this note.
}
import ietf-mpls-mldp {
    prefix "mldp";
    reference "RFC XXXX: Base YANG Data Model for MPLS mLDP";
// RFC Editor: replace the XXXX with actual mLDP YANG RFC number at
// time of publication and remove this note.
}
```

organization

"IETF MPLS Working Group";

contact

"WG Web: <<http://tools.ietf.org/wg/mpls/>>

WG List: <<mailto:mpls@ietf.org>>

Editor: Kamran Raza
<<mailto:skraza@cisco.com>>

Editor: Sowmya Krishnaswamy
<<mailto:krishnaswamy.sowmya@gmail.com>>

Editor: Xufeng Liu
<<mailto:xufeng.liu.ietf@gmail.com>>

Editor: Santosh Esale
<<mailto:sesale@juniper.net>>

Editor: Loa Andersson
<<mailto:loa@pi.nu>>

Editor: Jeff Tantsura
<<mailto:jefftant.ietf@gmail.com>>;

description

"This YANG module defines the extended components for the management of Multi-Protocol Label Switching (MPLS) Multipoint LDP (mLDP).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions

```
Relating to IETF Documents
(http://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

revision 2018-10-22 {
  // RFC Editor: replace the above date 2018-10-22 with the date of
  // publication and remove this note.

  description
    "Initial revision.";
  reference
    "RFC XXXX: Extended YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

/*
 * Features
 */
feature capability-mldp-hsmp {
  description
    "This feature indicates that the system allows to configure
    mLDP hub-and-spoke-multipoint capability.";
}

feature capability-mldp-node-protection {
  description
    "This feature indicates that the system allows to configure
    mLDP node-protection capability.";
}

feature mldp-mofrr {
  description
    "This feature indicates that the system supports mLDP
    Multicast only FRR (MoFRR).";
}

feature per-peer-capability {
  description
    "This feature indicates that the system allows to configure
    mLDP capabilities at the per peer level.";
}
```

```
/*
 * Typedefs
 */
typedef mofrr-role {
  type enumeration {
    enum none {
      description "MOFRR is not enabled.";
    }
    enum primary {
      description "This LSP is primary.";
    }
    enum backup {
      description "This LSP is backup.";
    }
  }
  description
    "This type represents the MOFRR (Multicast only FRR) role
    status of a LSP.";
}

/*
 * Groupings
 */
grouping mldp-ext-binding-label-state-attributes {
  description
    "mLDP label binding attributes";

  list peer {
    key "direction peer advertisement-type";
    description
      "List of advertised and received peers";
    uses mldp:mldp-binding-label-peer-state-attributes;

    leaf mofrr-role {
      when "../direction = 'upstream'" {
        description
          "For upstream.";
      }
      type mofrr-role;
      description
        "The MOFRR status of this LSP";
    }
  } // peer
} // mldp-ext-binding-label-state-attributes

grouping mldp-ext-capabilities {
  description
    "mLDP extended capabilities";
```

```
container hub-and-spoke {
  if-feature capability-mldp-hsmp;
  description
    "Configure hub-and-spoke-multipoint capability";
  reference
    "RFC7140: LDP Extensions for Hub and Spoke Multipoint
    Label Switched Path";
  leaf enable {
    type boolean;
    description
      "Enable hub-and-spoke-multipoint";
  }
}
container node-protection {
  if-feature capability-mldp-node-protection;
  description
    "Configure node-protection capability.";
  reference
    "RFC7715: mLDP Node Protection.";
  leaf plr {
    type boolean;
    description
      "Point of Local Repair (PLR) capable for Multipoint LSP
      node protection";
  }
}
container merge-point {
  description
    "Merge Point capable for Multipoint LSP node protection";
  leaf enable {
    type boolean;
    description
      "Enable merge point capability";
  }
  leaf targeted-session-teardown-delay {
    type uint16;
    units seconds;
    description
      "Targeted session teardown delay";
  }
} // merge-point
} // mldp-ext-capabilities

grouping mldp-ext-per-af-config-attributes {
  description
    "mLDP per address family configuration attributes";
  container multicast-only-frr {
    if-feature mldp-mofrr;
```

```
    description
      "Multicast-only FRR (MoFRR) policy";
    leaf prefix-list {
      type ldp-ext:prefix-list-ref;
      description
        "Enables Multicast-only FRR (MoFRR) for the specified
        access list";
    }
  } // multicast-only-frr
  container recursive-fec {
    description
      "Recursive FEC policy";
    leaf prefix-list {
      type ldp-ext:prefix-list-ref;
      description
        "Enables recursive FEC for the specified prefix-list";
    }
  } // recursive-fec
} // mldp-ext-per-af-config-attributes

grouping recursive-fec-attributes {
  description
    "mLDP recursive FEC attributes.";
  leaf recur-root-address {
    type inet:ip-address;
    description
      "Recursive root address";
    reference
      "RFC6512: Using Multipoint LDP When the
      Backbone Has No Route to the Root";
  }
  leaf recur-rd {
    type rt-types:route-distinguisher;
    description
      "Route Distinguisher in the VPN-Recursive
      Opaque Value";
    reference
      "RFC6512: Using Multipoint LDP When the
      Backbone Has No Route to the Root";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description
      "The type of mutipoint: p2mp or mp2mp";
  }
} // recursive-fec-attributes

/*
```



```
* Configuration data and operational state data nodes
*/
// Global capability
augment "/rt:routing/rt:control-plane-protocols/"
  + "ldp:mpls-ldp/ldp:global/ldp:capability/mldp:mldp" {
    description "Augmentation for MLDP global capability.";

    uses mldp-ext-capabilities;
  }

// Peer capability
augment "/rt:routing/rt:control-plane-protocols/"
  + "ldp:mpls-ldp/ldp:peers/ldp:peer/ldp:capability" {
    description "Augmentation for MLDP peer capability.";
    container mldp {
      if-feature per-peer-capability;
      description
        "mLDP capabilities";
      uses mldp:mldp-capabilities;
    }
  }

// IPv4 config
augment "/rt:routing/rt:control-plane-protocols/"
  + "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
  + "mldp:ipv4" {
    description "Augmentation for MLDP IPv4 configuration";
    uses mldp-ext-per-af-config-attributes;
  }

// IPv4 configured-leaf-lsps config
augment "/rt:routing/rt:control-plane-protocols/"
  + "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
  + "mldp:ipv4/mldp:configured-leaf-lsps/"
  + "mldp:opaque-element-lspid/mldp:fec-label" {
    description
      "Augmentation for MLDP IPv4 configured-leaf-lsps
      configuration for opaque-element-lspid";
    list recursive-fec {
      key
        "recur-root-address recur-rd";
      description
        "List of recursive opaque values";
      uses recursive-fec-attributes;
    } // fec-label
  }
}
```

```
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "mldp:ipv4/mldp:configured-leaf-lsps" {
  description
    "Augmentation for MLDP IPv4 configured-leaf-lsps
    configuration";

  container opaque-element-transit {
    description
      "The type of opaque value element is the transit IPv4
      source.";
    reference
      "RFC6826: Multipoint LDP In-Band Signaling for
      Point-to-Multipoint and
      Multipoint-to-Multipoint Label Switched Paths.";
    list fec-label {
      key
        "root-address source-address group-address " +
        "rd recur-root-address recur-rd";
      description
        "List of FEC to label bindings";
      leaf root-address {
        type inet:ipv4-address;
        description
          "Root address";
      }
      leaf source-address {
        type inet:ip-address;
        description
          "Source address";
      }
      leaf group-address {
        type inet:ip-address-no-zone;
        description
          "Group address";
      }
      leaf rd {
        type rt-types:route-distinguisher;
        description
          "Route Distinguisher";
        reference
          "RFC7246: Multipoint Label Distribution
          Protocol In-Band Signaling in a Virtual
          Routing and Forwarding (VRF) Table
          Context.";
      }
      uses recursive-fec-attributes;
    } // fec-label
  }
}
```

```
    } // opaque-element-transit

    container opaque-element-bidir {
      description
        "The type of opaque value element is
         the generic LSP identifier";
      reference
        "RFC6826: Multipoint LDP In-Band Signaling for
         Point-to-Multipoint and
         Multipoint-to-Multipoint Label Switched
         Paths.";
      list fec-label {
        key
          "root-address rp group-address rd recur-root-address "
          + "recur-rd";
        description
          "List of FEC to label bindings";
        leaf root-address {
          type inet:ipv4-address;
          description
            "Root address";
        }
        leaf rp {
          type inet:ip-address;
          description
            "Rendezvous-Point (RP) address";
        }
        leaf group-address {
          type inet:ip-address-no-zone;
          description
            "Group address";
        }
        leaf rd {
          type rt-types:route-distinguisher;
          description
            "Route Distinguisher";
          reference
            "RFC7246: Multipoint Label Distribution
             Protocol In-Band Signaling in a Virtual
             Routing and Forwarding (VRF) Table
             Context.";
        }
        uses recursive-fec-atttributes;
      } // fec-label
    } // opaque-element-bidir
  }

  // IPv6 config
```

```
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "ipv6" {
  description "Augmentation for MLDP IPv4 configuration";
  uses mldp-ext-per-af-config-attributes;
}

// Global forwarding-nexthop
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/ldp-ext:forwarding-nexthop/"
+ "ldp-ext:interfaces/ldp-ext:interface/ldp-ext:address-family" {
  description
    "Augmentation for MLDP nexthop forwarding interface";
  leaf mldp-disable {
    type boolean;
    description
      "Disable mLDp forwarding on this interface";
  }
}

/*
 * Operational state data nodes
 */
// IPv4 state for per peer bindings
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "mldp:ipv4/mldp:roots/mldp:root/mldp:bindings/"
+ "mldp:opaque-element-lspid/mldp:fec-label/mldp:peer" {
  description "Augmentation for MLDP IPv4 state";

  leaf mofrr-role {
    when "../mldp:direction = 'upstream'" {
      description
        "For upstream";
    }
    type mofrr-role;
    description
      "The MOFRR status of this LSP";
  }
}

// Peer capability state
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:peers/ldp:peer/ldp:received-peer-state/"
+ "ldp:capability/mldp:mldp" {
  description
    "Augmentation for MLDP received peer state capability.";
  container hub-and-spoke {
```

```
description
  "Configure hub-and-spoke-multipoint capability.";
reference
  "RFC7140: LDP Extensions for Hub and Spoke Multipoint
  Label Switched Path";
leaf enable {
  type boolean;
  description
    "Enable hub-and-spoke-multipoint";
}
}
container node-protection {
  description
    "Configure node-protection capability";
  reference
    "RFC7715: mLDP Node Protection.";
  leaf plr {
    type boolean;
    description
      "Point of Local Repair (PLR) capable for Multipoint LSP
      node protection";
  }
  leaf merge-point {
    type boolean;
    description
      "Merge Point capable for Multipoint LSP node protection";
  } // merge-point
} // node-protection
}

// IPv4 bindings state
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "mldp:ipv4/mldp:roots/mldp:root/mldp:bindings" {
  description "Augmentation for MLDP IPv4 bindings.";
  container opaque-element-transit {
    description
      "The type of opaque value element is the transit IPv4
      source.";
    reference
      "RFC6826: Multipoint LDP In-Band Signaling for
      Point-to-Multipoint and
      Multipoint-to-Multipoint Label Switched Paths.";
    list fec-label {
      key
        "source-address group-address "
        + "rd recur-root-address recur-rd";
      description
```

```
        "List of FEC to label bindings";
    leaf source-address {
        type inet:ip-address;
        description
            "Source address";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
    uses mldp-ext-binding-label-state-attributes;
} // fec-label
} // opaque-element-transit

container opaque-element-bidir {
    description
        "The type of opaque value element is
        the generic LSP identifier.";
    reference
        "RFC6826: Multipoint LDP In-Band Signaling for
        Point-to-Multipoint and
        Multipoint-to-Multipoint Label Switched
        Paths.";
    list fec-label {
        key
            "rp group-address rd recur-root-address recur-rd";
        description
            "List of FEC to label bindings";
        leaf rp {
            type inet:ip-address;
            description
                "Rendezvous Point (RP) address";
        }
        leaf group-address {
            type inet:ip-address-no-zone;
            description

```

```

        "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
    uses mldp-ext-binding-label-state-attributes;
} // fec-label
} // opaque-element-bidir
}

// IPv6 bindings state
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "ipv6/roots/root/bindings" {
    description "Augmentation for MLDP IPv6 bindings.";
    container opaque-element-transit {
        config false;
        description
            "The type of opaque value element is the transit IPv6
            source.";
        reference
            "RFC6826: Multipoint LDP In-Band Signaling for
            Point-to-Multipoint and
            Multipoint-to-Multipoint Label Switched
            Paths.";
        list fec-label {
            key
                "source-address group-address "
            + "rd recur-root-address recur-rd";
            description
                "List of FEC to label bindings";
            leaf source-address {
                type inet:ip-address;
                description
                    "Source address";
            }
            leaf group-address {
                type inet:ip-address-no-zone;
                description
                    "Group address";
            }
        }
    }
}

```

```
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
    uses mldp-ext-binding-label-state-attributes;
} // fec-label
} // opaque-element-transit

container opaque-element-bidir {
    config false;
    description
        "The type of opaque value element is
        the generic LSP identifier";
    reference
        "RFC6826: Multipoint LDP In-Band Signaling for
        Point-to-Multipoint and
        Multipoint-to-Multipoint Label Switched
        Paths.";
    list fec-label {
        key
            "rp group-address rd recur-root-address recur-rd";
        description
            "List of FEC to label bindings";
        leaf rp {
            type inet:ip-address;
            description
                "Rendezvous Point (RP) address";
        }
        leaf group-address {
            type inet:ip-address-no-zone;
            description
                "Group address";
        }
        leaf rd {
            type rt-types:route-distinguisher;
            description
                "Route Distinguisher";
            reference
                "RFC7246: Multipoint Label Distribution
                Protocol In-Band Signaling in a Virtual
```



```

        Routing and Forwarding (VRF) Table
        Context.";
    }
    uses recursive-fec-attributes;
    uses mldp-ext-binding-label-state-attributes;
} // fec-label
} // opaque-element-bidir
}

// IPv4 bindings opaque-element-lspid state
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "mldp:ipv4/mldp:roots/mldp:root/mldp:bindings/"
+ "mldp:opaque-element-lspid/mldp:fec-label" {
    description
        "Augmentation for MLDP IPv4 bindings with opaque type LSP ID.";
    list recursive-fec {
        key
            "recur-root-address recur-rd";
        description
            "List of recursive opaque values";
        uses recursive-fec-attributes;
        uses mldp-ext-binding-label-state-attributes;
    } // fec-label
}

// IPv6 bindings opaque-element-lspid state
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
+ "ipv6/roots/root/bindings/opaque-element-lspid/fec-label" {
    description
        "Augmentation for MLDP IPv6 bindings with opaque type LSP ID.";
    list recursive-fec {
        key "recur-root-address recur-rd";
        config false;
        description
            "List of recursive opaque values";
        uses recursive-fec-attributes;
        uses mldp-ext-binding-label-state-attributes;
    } // fec-label
}

/*
 * Per AF augmentation
 */
// IPv6 augmentation
augment "/rt:routing/rt:control-plane-protocols/"
+ "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families" {

```

```
description "Augmentation for MLDP IPv6 address family.";
container ipv6 {
  description
    "IPv6 information";

  container roots {
    config false;
    description
      "IPv6 multicast LSP roots";
    list root {
      key "root-address";
      description
        "List of roots for configured multicast LSPs";

      leaf root-address {
        type inet:ipv6-address;
        description
          "Root address";
      }

      leaf is-self {
        type boolean;
        description
          "This is the root";
      }

      list reachability {
        key "address interface";
        description
          "A next-hop for reachability to root,
          as a RIB view";
        leaf address {
          type inet:ipv6-address;
          description
            "The next-hop address to reach root";
        }
        leaf interface {
          type if:interface-ref;
          description
            "Interface connecting to next-hop";
        }
        leaf peer {
          type leafref {
            path
              "../.../.../.../.../.../ldp:peers/"
              + "ldp:peer/ldp:lsr-id";
          }
          description

```

```

        "LDP peer from which this next-hop can be
        reached";
    }
}

container bindings {
    description
        "mLDP FEC to label bindings";
    container opaque-element-lspid {
        description
            "The type of opaque value element is
            the generic LSP identifier";
        reference
            "RFC6388: Label Distribution Protocol
            Extensions for Point-to-Multipoint and
            Multipoint-to-Multipoint Label Switched
            Paths.";
        list fec-label {
            key
                "lsp-id";
            description
                "List of FEC to label bindings";
            leaf lsp-id {
                type uint32;
                description "ID to identify the LSP";
            }
            leaf multipoint-type {
                type mldp:multipoint-type;
                description
                    "The type of mutipoint: p2mp or mp2mp";
            }
        }

        uses mldp-ext-binding-label-state-attributes;
    } // fec-label
    } // opaque-element-lspid
    } // bindings
    } // list root
} // roots

container configured-leaf-lsps {
    description
        "Configured multicast LSPs";

    container opaque-element-lspid {
        description
            "The type of opaque value element is
            the generic LSP identifier";
        reference

```

```
"RFC6388: Label Distribution Protocol
Extensions for Point-to-Multipoint and
Multipoint-to-Multipoint Label Switched
Paths.";
list fec-label {
  key
    "root-address lsp-id";
  description
    "List of FEC to label bindings";
  leaf root-address {
    type inet:ipv6-address;
    description
      "Root address";
  }
  leaf lsp-id {
    type uint32;
    description "ID to identify the LSP";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description
      "The type of mutipoint: p2mp or mp2mp";
  }
  list recursive-fec {
    key
      "recur-root-address recur-rd";
    description
      "List of recursive opaque values";
    uses recursive-fec-attibutes;
  } // fec-label
} // fec-label
} // opaque-element-lspid

container opaque-element-transit {
  description
    "The type of opaque value element is the transit IPv4
    source.";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched Paths.";
  list fec-label {
    key
      "root-address source-address group-address "
      + "rd recur-root-address recur-rd";
    description
      "List of FEC to label bindings";
    leaf root-address {
```

```
        type inet:ipv6-address;
        description
            "Root address";
    }
    leaf source-address {
        type inet:ip-address;
        description
            "Source address";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
} // fec-label
} // opaque-element-transit

container opaque-element-bidir {
    description
        "The type of opaque value element is
        the generic LSP identifier";
    reference
        "RFC6826: Multipoint LDP In-Band Signaling for
        Point-to-Multipoint and
        Multipoint-to-Multipoint Label Switched
        Paths.";
    list fec-label {
        key
            "root-address rp group-address rd recur-root-address "
            + "recur-rd";
        description
            "List of FEC to label bindings.";
        leaf root-address {
            type inet:ipv6-address;
            description
                "Root address";
        }
    }
}
```

```

    leaf rp {
        type inet:ip-address;
        description
            "Rendezvous Point (RP) address";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
    } // fec-label
    } // opaque-element-bidir
    } // configured-leaf-lsps
    } // ipv6
}

/*
 * Global augmentation
 */
/*
 * Notifications
 */
augment "/mldp:mpls-mldp-fec-event/mldp:opaque-element/"
+ "mldp:opaque-element-lspid/mldp:opaque-element-lspid" {
    description
        "Augmentation for MLDP notification for opaque-element-lspid.";
    container recursive-fec {
        description
            "Container of recursive opaque values";
        uses recursive-fec-attributes;
    } // fec-label
}

augment "/mldp:mpls-mldp-fec-event/mldp:opaque-element" {
    description
        "Augmentation for MLDP notification.";
    case opaque-element-transit {

```

```
container opaque-element-transit {
  description
    "The type of opaque value element is the transit IPv4
    source.";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched Paths.";
  leaf root-address {
    type inet:ip-address;
    description
      "Root address";
  }
  leaf source-address {
    type inet:ip-address;
    description
      "Source address";
  }
  leaf group-address {
    type inet:ip-address-no-zone;
    description
      "Group address";
  }
  leaf rd {
    type rt-types:route-distinguisher;
    description
      "Route Distinguisher";
    reference
      "RFC7246: Multipoint Label Distribution
      Protocol In-Band Signaling in a Virtual
      Routing and Forwarding (VRF) Table
      Context.";
  }
  uses recursive-fec-attributes;
} // opaque-element-transit

case opaque-element-bidir {
  container opaque-element-bidir {
    description
      "The type of opaque value element is
      the generic LSP identifier";
    reference
      "RFC6826: Multipoint LDP In-Band Signaling for
      Point-to-Multipoint and
      Multipoint-to-Multipoint Label Switched
      Paths.";
    leaf root-address {
```

```

        type inet:ip-address;
        description
            "Root address";
    }
    leaf rp {
        type inet:ip-address;
        description
            "Rendezvous Point (RP) address";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
} // opaque-element-bidir
} // opaque-element-bidir
}
}
<CODE ENDS>

```

Figure 15

10. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes.

It goes without saying that this specification also inherits the security considerations captured in the actual protocol specification documents, namely base mLDP [RFC6388], targeted mLDP [RFC7060], mLDP Recursive FEC [RFC6512], Multicast-only FRR [RFC7431], mLDP Node Protection [RFC7715], mLDP In-band Signaling [RFC6826] [RFC7246] [RFC7438], and Hub-and-Spoke Multipoint LSPs [RFC7140].

11. IANA Considerations

This document requests the registration of the following URIs in the IETF "XML registry" [RFC3688]:

URI	Registrant	XML
urn:ietf:params:xml:ns:yang:ietf-mpls-mlldp	The IESG	N/A
urn:ietf:params:xml:ns:yang:ietf-mpls-mlldp-extended	The IESG	N/A

This document requests the registration of the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name	Namespace	Prefix	Reference
ietf-mpls-mlldp	urn:ietf:params:xml:ns:yang:ietf-mpls-mlldp	mlldp	This document
ietf-mpls-mlldp-extended	urn:ietf:params:xml:ns:yang:ietf-mpls-mlldp-extended	mlldp-ext	This document

12. Acknowledgments

The authors would like to acknowledge Ladislav Lhotka and Acee Lindem for their review and comments.

13. References

13.1. Normative References

- [I-D.ietf-mpls-ldp-yang]
Raza, K., Asati, R., Liu, X., Esale, S., Chen, X., and H. Shah, "YANG Data Model for MPLS LDP", draft-ietf-mpls-ldp-yang-06 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <<https://www.rfc-editor.org/info/rfc6388>>.
- [RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, DOI 10.17487/RFC6389, November 2011, <<https://www.rfc-editor.org/info/rfc6389>>.

- [RFC6512] Wijnands, IJ., Rosen, E., Napierala, M., and N. Leymann, "Using Multipoint LDP When the Backbone Has No Route to the Root", RFC 6512, DOI 10.17487/RFC6512, February 2012, <<https://www.rfc-editor.org/info/rfc6512>>.
- [RFC6826] Wijnands, IJ., Ed., 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, DOI 10.17487/RFC6826, January 2013, <<https://www.rfc-editor.org/info/rfc6826>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7060] Napierala, M., Rosen, E., and IJ. Wijnands, "Using LDP Multipoint Extensions on Targeted LDP Sessions", RFC 7060, DOI 10.17487/RFC7060, November 2013, <<https://www.rfc-editor.org/info/rfc7060>>.
- [RFC7140] Jin, L., Jounay, F., Wijnands, IJ., and N. Leymann, "LDP Extensions for Hub and Spoke Multipoint Label Switched Path", RFC 7140, DOI 10.17487/RFC7140, March 2014, <<https://www.rfc-editor.org/info/rfc7140>>.
- [RFC7246] Wijnands, IJ., Ed., Hitchen, P., Leymann, N., Henderickx, W., Gulko, A., and J. Tantsura, "Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context", RFC 7246, DOI 10.17487/RFC7246, June 2014, <<https://www.rfc-editor.org/info/rfc7246>>.
- [RFC7431] Karan, A., Filsfils, C., Wijnands, IJ., Ed., and B. Decraene, "Multicast-Only Fast Reroute", RFC 7431, DOI 10.17487/RFC7431, August 2015, <<https://www.rfc-editor.org/info/rfc7431>>.
- [RFC7438] Wijnands, IJ., Ed., Rosen, E., Gulko, A., Joorde, U., and J. Tantsura, "Multipoint LDP (mLDP) In-Band Signaling with Wildcards", RFC 7438, DOI 10.17487/RFC7438, January 2015, <<https://www.rfc-editor.org/info/rfc7438>>.
- [RFC7715] Wijnands, IJ., Ed., Raza, K., Atlas, A., Tantsura, J., and Q. Zhao, "Multipoint LDP (mLDP) Node Protection", RFC 7715, DOI 10.17487/RFC7715, January 2016, <<https://www.rfc-editor.org/info/rfc7715>>.

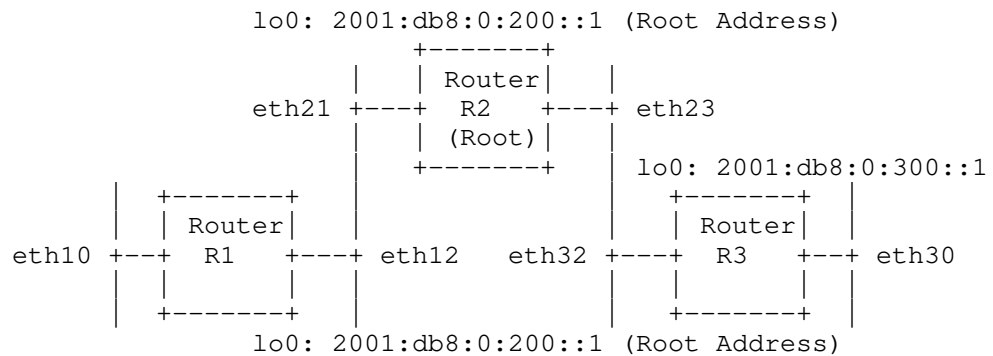
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

13.2. Informative References

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951], containing both configuration and state data.



The configuration instance data tree for Router R3 in the above figure could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "lo0",
        "description": "R3 loopback interface.",
        "type": "iana-if-type:softwareLoopback",
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:300::1",
              "prefix-length": 64
            }
          ]
        }
      },
      {
        "name": "eth30",
        "description": "An interface connected to client routers.",
      }
    ]
  }
}

```

```

        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv6": {
            "forwarding": true
        }
    },
    {
        "name": "eth32",
        "description": "An interface connected to root (R2).",
        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv6": {
            "forwarding": true
        }
    }
]
},
"ietf-routing:routing": {
    "router-id": "203.0.113.3",
    "control-plane-protocols": {
        "ietf-mpls-ldp:mpls-ldp": {
            "global": {
                "address-families": {
                    "ietf-mpls-ldp-extended:ipv6": {
                        "enable": true
                    }
                }
            },
            "capability": {
                "ietf-mpls-mldp:mldp": {
                    "mp2mp": {
                        "enable": true
                    }
                }
            }
        },
        "ietf-mpls-mldp:mldp": {
            "enable": true,
            "address-families": {
                "ietf-mpls-mldp-extended:ipv6": {
                    "configured-leaf-lsps": {
                        "opaque-element-lspid": {
                            "fec-label": [
                                {
                                    "root-address": "2001:db8:0:200::1",
                                    "lsp-id": 201,
                                    "multipoint-type": "mp2mp"
                                }
                            ]
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
  },
  "discovery": {
    "interfaces": {
      "interface": [
        {
          "name": "eth30",
          "address-families": {
            "ietf-mpls-ldp-extended:ipv6": {
              "enable": true
            }
          }
        },
        {
          "name": "eth32",
          "address-families": {
            "ietf-mpls-ldp-extended:ipv6": {
              "enable": true
            }
          }
        }
      ]
    }
  }
}

```

The cooresponding operational state data for Router R3 could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "lo0",
        "description": "R3 loopback interface.",
        "type": "iana-if-type:softwareLoopback",
        "phys-address": "00:00:5e:00:53:03",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2018-10-15T12:34:56-05:00"
        },
        "ietf-ip:ipv6": {
          "mtu": 1500,
          "address": [

```

```

        {
            "ip": "2001:db8:0:300::1",
            "prefix-length": 64,
            "origin": "static",
            "status": "preferred"
        },
        {
            "ip": "fe80::200:5eff:fe00:5303",
            "prefix-length": 64,
            "origin": "link-layer",
            "status": "preferred"
        }
    ],
    "neighbor": [
    ]
}
},
{
    "name": "eth30",
    "description": "An interface connected to client routers.",
    "type": "iana-if-type:ethernetCsmacd",
    "phys-address": "00:00:5e:00:53:30",
    "oper-status": "up",
    "statistics": {
        "discontinuity-time": "2018-10-15T12:34:56-05:00"
    },
    "ietf-ip:ipv6": {
        "forwarding": true,
        "mtu": 1500,
        "address": [
            {
                "ip": "fe80::200:5eff:fe00:5330",
                "prefix-length": 64,
                "origin": "link-layer",
                "status": "preferred"
            }
        ],
        "neighbor": [
        ]
    }
}
},
{
    "name": "eth32",
    "description": "An interface connected to root (R2).",
    "type": "iana-if-type:ethernetCsmacd",
    "phys-address": "00:00:5e:00:53:32",
    "oper-status": "up",
    "statistics": {

```



```
        "discontinuity-time": "2018-10-15T12:34:56-05:00"
    },
    "ietf-ip:ipv6": {
        "forwarding": true,
        "mtu": 1500,
        "address": [
            {
                "ip": "fe80::200:5eff:fe00:5332",
                "prefix-length": 64,
                "origin": "link-layer",
                "status": "preferred"
            }
        ],
        "neighbor": [
            {
                "ip": "fe80::200:5eff:fe00:5323",
                "link-layer-address": "00:00:5e:00:53:23",
                "origin": "dynamic",
                "is-router": [null],
                "state": "reachable"
            }
        ]
    }
},
    "ietf-routing:routing": {
        "router-id": "203.0.113.3",
        "interfaces": {
            "interface": [
                "lo0",
                "eth30",
                "eth32"
            ]
        },
        "control-plane-protocols": {
            "ietf-mpls-ldp:mpls-ldp": {
                "global": {
                    "address-families": {
                        "ietf-mpls-ldp-extended:ipv6": {
                            "enable": true
                        }
                    }
                },
                "capability": {
                    "ietf-mpls-mldp:mldp": {
                        "mp2mp": {
                            "enable": true
                        }
                    }
                }
            }
        }
    }
}
```

```
    }
  },
  "ietf-mpls-mlldp:mlldp": {
    "enable": true,
    "address-families": {
      "ietf-mpls-mlldp-extended:ipv6": {
        "configured-leaf-lsps": {
          "opaque-element-lspid": {
            "fec-label": [
              {
                "root-address": "2001:db8:0:200::1",
                "lsp-id": 201,
                "multipoint-type": "mp2mp"
              }
            ]
          }
        }
      }
    },
    "roots": {
      "root": [
        {
          "root-address": "2001:db8:0:200::1",
          "is-self": false,
          "reachability": [
            {
              "address": "fe80::200:5eff:fe00:5323",
              "interface": "eth32",
              "peer": "203.0.113.2"
            }
          ]
        }
      ],
      "bindings": {
        "opaque-element-lspid": {
          "fec-label": [
            {
              "lsp-id": 201,
              "multipoint-type": "mp2mp",
              "peer": [
                {
                  "direction": "upstream",
                  "peer": "203.0.113.2",
                  "advertisement-type": "advertised",
                  "label": 3201
                },
                {
                  "direction": "upstream",
                  "peer": "203.0.113.2",
                  "advertisement-type": "received",
                  "label": 2301
                }
              ]
            }
          ]
        }
      }
    }
  }
}
```

```

    ]
  }
]
}
}
}
}
}
},
"discovery": {
  "interfaces": {
    "interface": [
      {
        "name": "eth30",
        "address-families": {
          "ietf-mpls-ldp-extended:ipv6": {
            "enable": true,
            "hello-adjacencies": {
              "hello-adjacency": [
            ]
          }
        }
      }
    ]
  },
  {
    "name": "eth32",
    "address-families": {
      "ietf-mpls-ldp-extended:ipv6": {
        "enable": true,
        "hello-adjacencies": {
          "hello-adjacency": [
            {
              "adjacent-address":
                "fe80::200:5eff:fe00:5323",
              "flag": ["adjacency-flag-active"],
              "hello-holdtime": {
                "adjacent": 15,
                "negotiated": 15,
                "remaining": 9
              },
              "next-hello": 3,
              "statistics": {
                "discontinuity-time":
                  "2018-10-15T12:34:56-05:00"
              }
            }
          ]
        }
      }
    ]
  },

```

```

        "peer": {
            "lsr-id": "203.0.113.2",
            "label-space-id": 0
        }
    ]
}
}
}
}
}
}
},
"peers": {
    "peer": [
        {
            "lsr-id": "203.0.113.2",
            "label-space-id": 0,
            "label-advertisement-mode": {
                "local": "downstream-unsolicited",
                "peer": "downstream-unsolicited",
                "negotiated": "downstream-unsolicited"
            },
            "next-keep-alive": 5,
            "session-holdtime": {
                "peer": 180,
                "negotiated": 180,
                "remaining": 78
            },
            "session-state": "operational",
            "tcp-connection": {
                "local-address": "fe80::200:5eff:fe00:5332",
                "local-port": 646,
                "remote-address": "fe80::200:5eff:fe00:5323",
                "remote-port": 646
            },
            "up-time": "P2H33M5S",
            "statistics": {
                "discontinuity-time": "2018-10-15T12:34:56-05:00"
            },
            "received-peer-state": {
                "capability": {
                    "ietf-mpls-mldp:mldp": {
                        "mp2mp": {
                            "enable": true
                        }
                    }
                }
            }
        }
    ]
}

```

```
    }  
  }  
] }  
}  
}  
}  
}
```

Appendix B. Additional Contributors

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Authors' Addresses

Kamran Raza
Cisco Systems
Email: skraza@cisco.com

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Jeff Tantsura
Nuage Networks
Email: jefftant.ietf@gmail.com

Sowmya Krishnaswamy
Individual
Email: krishnaswamy.sowmya@gmail.com

Rajiv Asati
Cisco Systems, Inc.
Email: rajiva@cisco.com

Xia Chen
Huawei Technologies
Email: jescia.chenxia@huawei.com

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 15, 2020

T. Saad
Juniper Networks
R. Gandhi
Cisco Systems, Inc.
X. Liu
Volta Networks
V. Beeram
Juniper Networks
I. Bryskin
Huawei Technologies
September 12, 2019

A YANG Data Model for MPLS Static LSPs
draft-ietf-mpls-static-yang-10

Abstract

This document contains the specification for the MPLS Static Label Switched Paths (LSPs) YANG model. The model allows for the provisioning of static LSP(s) on Label Edge Router(s) LER(s) and Label Switched Router(s) LSR(s) devices along a LSP path without the dependency on any signaling protocol. The MPLS Static LSP model augments the MPLS base YANG model with specific data to configure and manage MPLS Static LSP(s).

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 <https://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 15, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Terminology	3
1.2. Acronyms and Abbreviations	3
2. MPLS Static LSP Model	3
2.1. Model Organization	4
2.2. Model Tree Diagram	4
2.3. Model Overview	6
2.4. Model YANG Module(s)	7
3. IANA Considerations	14
4. Security Considerations	15
5. Contributors	15
6. References	16
6.1. Normative References	16
6.2. Informative References	17
Authors' Addresses	18

1. Introduction

This document describes a YANG [RFC7950] data model for configuring and managing the Multiprotocol Label Switching (MPLS) [RFC3031] Static LSPs. The model allows the configuration of LER and LSR devices with the necessary MPLS cross-connects or bindings to realize an end-to-end LSP service.

A static LSP is established by manually specifying incoming and outgoing MPLS label(s) and necessary forwarding information on each of the traversed LER and LSR devices (ingress, transit, or egress nodes) of the forwarding path.

For example, on an ingress LER device, the model is used to associate a specific Forwarding Equivalence Class (FEC) of packets- e.g. matching a specific IP prefix in a Virtual Routing or Forwarding (VRF) instance- to an MPLS outgoing label imposition, next-hop(s) and respective outgoing interface(s) to forward the packet. On an LSR device, the model is used to create a binding that swaps the incoming label with an outgoing label and forwards the packet on one or

multiple egress path(s). On an egress LER, it is used to create a binding that decapsulates the incoming MPLS label and performs forwarding based on the inner MPLS label (if present) or IP forwarding in the packet.

The MPLS Static LSP YANG model is broken into two modules "ietf-mpls-static" and "ietf-mpls-static-extended". The "ietf-mpls-static" module covers basic features for the configuration and management of unidirectional Static LSP(s), while "ietf-mpls-static-extended" covers extended features like the configuration and management of bidirectional Static LSP(s) and LSP admission control.

The module "ietf-mpls-static" augments the MPLS Base YANG model defined in module "ietf-mpls" in [I-D.ietf-mpls-base-yang].

1.1. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology for describing YANG data models is found in [RFC7950].

1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

IILM: Incoming Label Map

2. MPLS Static LSP Model

2.1. Model Organization

The base MPLS Static LSP model covers the core features with the minimal set of configuration parameters needed to manage and operate MPLS Static LSPs.

Additional MPLS Static LSP parameters as well as optional feature(s) are grouped in a separate MPLS Static LSP extended model. The relationship between the MPLS base and other MPLS modules are shown in Figure 1.

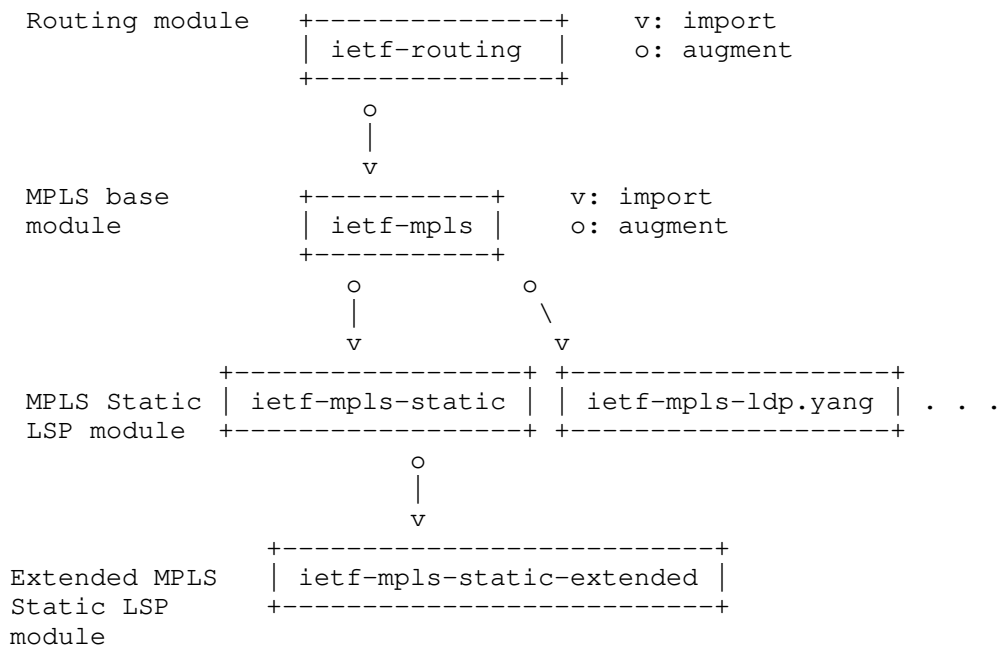


Figure 1: Relationship between MPLS modules

2.2. Model Tree Diagram

The MPLS Static and extended LSP tree diagram as per [RFC8340] is shown in Figure 2.

```

module: ietf-mpls-static
  augment /rt:routing/mpls:mpls:
    +--rw static-lsps
      +--rw static-lsp* [name]
        +--rw name          string
        +--rw operation?    mpls:mpls-operations-type

```

```

+--rw in-segment
|   +--rw fec
|   |   +--rw (type)?
|   |   |   +--:(ip-prefix)
|   |   |   |   +--rw ip-prefix?          inet:ip-prefix
|   |   |   +--:(mpls-label)
|   |   |   |   +--rw incoming-label?      rt-types:mpls-label
|   |   +--rw incoming-interface?        if:interface-ref
+--rw out-segment
|   +--rw (out-segment)?
|   +--:(nhlfe-single)
|   |   +--rw nhlfe-single
|   |   |   +--rw mpls-label-stack
|   |   |   |   +--rw entry* [id]
|   |   |   |   |   +--rw id                uint8
|   |   |   |   |   +--rw label?            rt-types:mpls-label
|   |   |   |   |   +--rw ttl?              uint8
|   |   |   |   |   +--rw traffic-class?    uint8
|   |   |   +--rw outgoing-interface?      if:interface-ref
|   +--:(nhlfe-multiple)
|   |   +--rw nhlfe-multiple
|   |   |   +--rw nhlfe* [index]
|   |   |   |   +--rw index                string
|   |   |   |   +--rw backup-index?         string
|   |   |   |   +--rw loadshare?            uint16
|   |   |   |   +--rw role?                 nhlfe-role
|   |   |   |   +--rw mpls-label-stack
|   |   |   |   |   +--rw entry* [id]
|   |   |   |   |   |   +--rw id                uint8
|   |   |   |   |   |   +--rw label?            |
|   |   |   |   |   |   |   rt-types:mpls-label
|   |   |   |   |   |   +--rw ttl?              uint8
|   |   |   |   |   |   +--rw traffic-class?    uint8
|   |   |   +--rw outgoing-interface?      if:interface-ref
+--rw mpls-static-ext:bandwidth?            uint32
+--rw mpls-static-ext:lsp-priority-setup?   uint8
+--rw mpls-static-ext:lsp-priority-hold?    uint8

module: ietf-mpls-static-extended
augment /rt:routing/mpls:mpls:
+--rw bidir-static-lsps
|   +--rw bidir-static-lsp* [name]
|   |   +--rw name                string
|   |   +--rw forward-lsp?        mpls-static:static-lsp-ref
|   |   +--rw reverse-lsp?        mpls-static:static-lsp-ref

```

Figure 2: MPLS Static LSP tree diagram

2.3. Model Overview

This document defines two YANG modules for MPLS Static LSP(s) configuration and management: `ietf-mpls-static.yang` and `ietf-mpls-static-extended.yang`.

The `ietf-mpls-static` module contains the following high-level types and groupings:

`static-lsp-ref`:

A YANG reference type for a static LSP that can be used by data models to reference a configured static LSP.

`in-segment`:

A YANG grouping that describes parameters of an incoming class of FEC associated with a specific LSP as described in the MPLS architecture document [RFC3031]. The model allows the following types of traffic to be mapped onto the static LSP on an ingress LER:

- o Unlabeled traffic destined to a specific prefix
- o Labeled traffic arriving with a specific label

`out-segment`:

A YANG grouping that describes parameters for the forwarding path(s) and their associated attributes for an LSP. The model allows for the following cases:

- o single forwarding path or NHLFE
- o multiple forwarding path(s) or NHLFE(s), each of which can serve a primary, backup or both role(s).

The `ietf-mpls-static-extended` module contains the following high-level types and groupings:

`bidir-static-lsp`:

A YANG grouping that describes list of static bidirectional LSPs

The `ietf-mpls-static-extended` augments the `ietf-mpls-static` model with additional parameters to configure and manage:

- o Bidirectional Static LSP(s)
- o Defining Static LSP bandwidth allocation

- o Defining Static LSP preemption priorities

2.4. Model YANG Module(s)

Configuring LSPs through an LSR/LER involves the following steps:

- o Enabling MPLS on MPLS capable interfaces.
- o Configuring in-segments and out-segments on LER(s) and LSR(s) traversed by the LSP.
- o Setting up the cross-connect per LSP to associate segments and/or to indicate connection origination and termination.
- o Optionally specifying label stack actions.
- o Optionally specifying segment traffic parameters.

The objects covered by this model are derived from the Incoming Label Map (ILM) and Next Hop Label Forwarding Entry (NHLFE) as specified in the MPLS architecture document [RFC3031].

The ietf-mpls-static module imports the following modules:

- o ietf-inet-types defined in [RFC6991]
- o ietf-routing defined in [RFC8349]
- o ietf-routing-types defined in [RFC8294]
- o ietf-interfaces defined in [RFC8343]
- o ietf-mpls defined in [I-D.ietf-mpls-base-yang]
- o ietf-te defined in [I-D.ietf-teas-yang-te]

The ietf-mpls-static module is shown below:

```
<CODE BEGINS> file "ietf-mpls-static@2019-09-12.yang"
module ietf-mpls-static {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-static";
  prefix "mpls-static";

  import ietf-mpls {
    prefix "mpls";
    reference "draft-ietf-mpls-base-yang: MPLS Base YANG Data Model";
  }
}
```

```
import ietf-routing {
  prefix "rt";
  reference "RFC8349: A YANG Data Model for Routing Management";
}

import ietf-routing-types {
  prefix "rt-types";
  reference "RFC8294: Common YANG Data Types for the Routing Area";
}

import ietf-inet-types {
  prefix inet;
  reference "RFC6991: Common YANG Data Types";
}

import ietf-interfaces {
  prefix "if";
  reference "RFC7223: A YANG Data Model for Interface Management";
}

organization "IETF MPLS Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/mpls/>

  WG List:    <mailto:mpls@ietf.org>

  Editor:     Tarek Saad
              <mailto:tsaad@juniper.net>

  Editor:     Rakesh Gandhi
              <mailto:rgandhi@cisco.com>

  Editor:     Xufeng Liu
              <mailto:xufeng.liu.ietf@gmail.com>

  Editor:     Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>

  Editor:     Igor Bryskin
              <mailto:Igor.Bryskin@huawei.com>";

description
  "This YANG module augments the 'ietf-routing' module with basic
  configuration and operational state data for MPLS static
  The model fully conforms to the Network Management Datastore
  Architecture (NMDA)."
```

Copyright (c) 2018 IETF Trust and the persons
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject
to the license terms contained in, the Simplified BSD License
set forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents

(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see
the RFC itself for full legal notices.";

```
// RFC Ed.: replace XXXX with actual RFC number and remove this  
// note.
```

```
// RFC Ed.: update the date below with the date of RFC publication  
// and remove this note.
```

```
revision "2019-09-12" {  
  description  
    "Latest revision of MPLS Static LSP YANG module";  
  reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";  
}
```

```
typedef static-lsp-ref {  
  type leafref {  
    path "/rt:routing/mpls:mpls/mpls-static:static-lsps/" +  
      "mpls-static:static-lsp/mpls-static:name";  
  }  
  description  
    "This type is used by data models that need to reference  
    configured static LSP.";  
}
```

```
grouping in-segment {  
  description "In-segment grouping";  
  container in-segment {  
    description "MPLS incoming segment";  
    container fec {  
      description "Forwarding Equivalence Class grouping";  
      choice type {  
        description "FEC type choices";  
        case ip-prefix {  
          leaf ip-prefix {  
            type inet:ip-prefix;  
            description "An IP prefix";  
          }  
        }  
      }  
    }  
  }  
}
```

```
    case mpls-label {
      leaf incoming-label {
        type rt-types:mpls-label;
        description "label value on the incoming packet";
      }
    }
  }
  leaf incoming-interface {
    type if:interface-ref;
    description
      "Optional incoming interface if FEC is restricted
       to traffic incoming on a specific interface";
  }
}

grouping out-segment {
  description "Out-segment grouping";
  container out-segment {
    description "MPLS outgoing segment";
    choice out-segment {
      description "The MPLS out-segment type choice";
      case nhlfe-single {
        container nhlfe-single {
          description "Container for single NHLFE entry";
          uses mpls:nhlfe-single-contents;
          leaf outgoing-interface {
            type if:interface-ref;
            description
              "The outgoing interface";
          }
        }
      }
      case nhlfe-multiple {
        container nhlfe-multiple {
          description "Container for multiple NHLFE entries";
          list nhlfe {
            key index;
            description "MPLS NHLFE entry";
            uses mpls:nhlfe-multiple-contents;
            leaf outgoing-interface {
              type if:interface-ref;
              description
                "The outgoing interface";
            }
          }
        }
      }
    }
  }
}
```



```

    }
  }
}

augment "/rt:routing/mpls:mpls" {
  description "Augmentations for MPLS Static LSPs";
  container static-lsps {
    description
      "Statically configured LSPs, without dynamic signaling";
    list static-lsp {
      key name;
      description "list of defined static LSPs";
      leaf name {
        type string;
        description "name to identify the LSP";
      }
      leaf operation {
        type mpls:mpls-operations-type;
        description
          "The MPLS operation to be executed on the incoming packet";
      }
      uses in-segment;
      uses out-segment;
    }
  }
}
}
<CODE ENDS>

```

The ietf-mpls-static-extended module imports the following modules:

- o ietf-mpls defined in [I-D.ietf-mpls-base-yang]
- o ietf-mpls-static defined in this document
- o ietf-routing defined in [RFC8349]

The ietf-mpls-static-extended module is shown below:

```

<CODE BEGINS> file "ietf-mpls-static-extended@2019-09-12.yang"
module ietf-mpls-static-extended {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended";
  prefix "mpls-static-ext";

  import ietf-mpls {
    prefix "mpls";
  }
}

```

```
    reference "draft-ietf-mpls-base-yang: MPLS Base YANG Data Model";
  }

  import ietf-routing {
    prefix "rt";
    reference "RFC8349: A YANG Data Model for Routing Management";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference "RFC8294: Common YANG Data Types for the Routing Area";
  }

  import ietf-mpls-static {
    prefix "mpls-static";
    reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";
  }

  organization "IETF MPLS Working Group";

  contact
    "WG Web:    <http://tools.ietf.org/wg/mpls/>

    WG List:    <mailto:mpls@ietf.org>

    Editor:     Tarek Saad
                <mailto:tsaad@juniper.net>

    Editor:     Rakesh Gandhi
                <mailto:rgandhi@cisco.com>

    Editor:     Xufeng Liu
                <mailto:xufeng.liu.ietf@gmail.com>

    Editor:     Vishnu Pavan Beeram
                <mailto:vbeeram@juniper.net>

    Editor:     Igor Bryskin
                <mailto:Igor.Bryskin@huawei.com>";

  description
    "This YANG module contains the Extended MPLS Static LSP YANG
    data model. The model fully conforms to the Network Management
    Datastore Architecture (NMDA).

    Copyright (c) 2018 IETF Trust and the persons
    identified as authors of the code. All rights reserved."
```

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.

// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision "2019-09-12" {
  description
    "Latest revision of MPLS Static LSP Extended YANG module";
  reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";
}

grouping bidir-static-lsp {
  description
    "grouping for top level list of static bidirectional LSPs";
  leaf forward-lsp {
    type mpls-static:static-lsp-ref;
    description
      "Reference to a configured static forward LSP";
  }
  leaf reverse-lsp {
    type mpls-static:static-lsp-ref;
    description
      "Reference to a configured static reverse LSP";
  }
}

augment "/rt:routing/mppls:mpls/mppls-static:static-lsps" {
  description
    "Augmentation for static MPLS LSPs";

  leaf bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "Bytes per second";
    description
      "Bandwidth using offline calculation";
  }
  leaf lsp-priority-setup {
    type uint8 {
```

```
        range "0..7";
    }
    description "LSP setup priority";
}
leaf lsp-priority-hold {
    type uint8 {
        range "0..7";
    }
    description "LSP hold priority";
}
}

augment "/rt:routing/mpls:mpls" {
    description "Augmentations for MPLS Static LSPs";
    container bidir-static-lsps {
        description
            "Statically configured bidirectional LSPs";
        list bidir-static-lsp {
            key name;
            description "List of static bidirectional LSPs";

            leaf name {
                type string;
                description "Name that identifies the bidirectional LSP";
            }
            uses bidir-static-lsp;
        }
    }
}
}
}
<CODE ENDS>
```

3. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls-static
Registrant Contact: The MPLS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended
Registrant Contact: The MPLS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

```
name:      ietf-mpls-static
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls-static
prefix:    ietf-mpls-static
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:  RFCXXXX

name:      ietf-mpls-static-extended
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended
prefix:    ietf-mpls-static-extended
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:  RFCXXXX
```

4. Security Considerations

The YANG modules specified in this document define schemas for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

All nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default) may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /ietf-routing:routing/ietf-mpls:mpls:/ietf-mpls:static-lsps: This entire subtree is related to security.

An administrator needs to restrict write access to all configurable objects within this data model.

5. Contributors

Himanshu Shah
Ciena
email: hshah@ciena.com

Kamran Raza
Cisco Systems, Inc.
email: skraza@cisco.com

6. References

6.1. Normative References

- [I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-10 (work in progress), February 2019.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-21 (work in progress), April 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

6.2. Informative References

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad.net@gmail.com

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Network Work group
Internet-Draft
Updates: 8287 (if approved)
Intended status: Standards Track
Expires: May 7, 2020

N. Nainar
C. Pignataro
Cisco Systems, Inc.
M. Aissaoui
Nokia
November 4, 2019

OSPFv3 CodePoint for MPLS LSP Ping
draft-nainar-mpls-lsp-ping-ospfv3-codepoint-00

Abstract

IANA has created "Protocol in the Segment IS Sub-TLV" registry and "Protocol in the Label Stack Sub-TLV of the Downstream Detailed Mapping TLV" under the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry. RFC8287 defines the code point for different Interior Gateway Protocol (IGP).

This document proposes the code point to be used in the Segment ID Sub-TLV and Downstream Detailed Mapping TLV when the IGP protocol is OSPFv3.

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 <https://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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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	2
3. Requirements notation	3
4. OSPFv3 protocol in Segment ID Sub-TLVs	3
5. OSPFv3 protocol in Downstream Detailed Mapping TLV	3
6. IANA Considerations	3
6.1. Protocol in the Segment ID sub-TLV	3
6.2. Protocol in Label Stack Sub-TLV of Downstream Detailed Mapping TLV	3
7. Security Considerations	4
8. Acknowledgement	4
9. Normative References	4
Authors' Addresses	5

1. Introduction

IANA has created "Protocol in the Segment IS Sub-TLV" registry and "Protocol in the Label Stack Sub-TLV of the Downstream Detailed Mapping TLV" under the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry [IANA-MPLS-LSP-PING]. [RFC8287] defines the code point for different Interior Gateway Protocol (IGP).

[RFC5340] describes OSPF version 3 (OSPFv3) protocol to support IPv6. [RFC5838] describes the mechanism to support multiple address families (AFs) in OSPFv3. Accordingly OSPFv3 may be used to advertise IPv6 and IPv4 prefixes.

This document proposes the code point to be used in the Segment ID Sub-TLV (Type 34, 35 and 36) and Downstream Detailed Mapping (DDMAP) TLV when the IGP protocol is OSPFv3.

2. Terminology

This document uses the terminologies defined in [RFC8402], [RFC8029], [RFC8287] and so the readers are expected to be familiar with the same.

3. Requirements notation

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. OSPFv3 protocol in Segment ID Sub-TLVs

When the protocol field of the Segment ID Sub-TLV Type 34, 35 and 36 is set to TBD1, the responder MUST perform the FEC validation using OSPFv3 as the IGP protocol.

5. OSPFv3 protocol in Downstream Detailed Mapping TLV

The protocol field of the Downstream Detailed Mapping (DDMAP) TLV in an echo reply is set to TBD2 when OSPFv3 is used to distribute the label carried in the Downstream Label field.

6. IANA Considerations

6.1. Protocol in the Segment ID sub-TLV

IANA is requested to assign one new code point of OSPFv3 from "Protocol in the Segment ID sub-TLV" registry under the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry:

Value	Meaning	Reference
-----	-----	-----
TBD1	OSPFv3	This document

6.2. Protocol in Label Stack Sub-TLV of Downstream Detailed Mapping TLV

IANA is requested to assign one new code point for OSPFv3 from "Protocol in Label Stack Sub-TLV of Downstream Detailed Mapping TLV" registry under the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry:

Value	Meaning	Reference
-----	-----	-----
TBD2	OSPFv3	This document

7. Security Considerations

This document updates [RFC8287] and does not introduce any additional security considerations.

8. Acknowledgement

To be Updated.

9. Normative References

[IANA-MPLS-LSP-PING]

IANA, "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters",
<<http://www.iana.org/assignments/mpls-lsp-ping-parameters/mpls-lsp-ping-parameters.xhtml>>.

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

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.

[RFC5838] Lindem, A., Ed., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, DOI 10.17487/RFC5838, April 2010, <<https://www.rfc-editor.org/info/rfc5838>>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,
Decraene, B., Litkowski, S., and R. Shakir, "Segment
Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,
July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

Authors' Addresses

Nagendra Kumar Nainar
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: cpignata@cisco.com

Mustapha Aissaoui
Nokia
Canada

Email: mustapha.aissaoui@nokia.com

Network Work group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

N. Nainar, Ed.
C. Pignataro, Ed.
Z. Ali
C. Filsfils
Cisco
July 8, 2019

Segment Routing Generic TLV for MPLS Label Switched Path (LSP) Ping/
Traceroute
draft-nainar-mpls-spring-lsp-ping-sr-generic-sid-00

Abstract

RFC8402 introduces Segment Routing architecture that leverages source routing and tunneling paradigms and can be directly applied to the Multi Protocol Label Switching (MPLS) data plane. A node steers a packet through a controlled set of instructions called segments, by prepending the packet with Segment Routing header. SR architecture defines different types of segments with different forwarding semantics associated. SR can be applied to the MPLS directly and to IPv6 dataplane using a new routing header.

RFC8287 defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifier (SIDs) with an MPLS data plane. Various SIDs are proposed as part of SR architecture with different associated instructions that raises a need to come up with new Target FEC Stack Sub-TLV for each such SIDs.

This document defines a new Target FEC Stack Sub-TLV that is used to validate the instruction associated with any SID.

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 <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Challenges with Existing Mechanism	3
2. Requirements notation	3
3. Terminology	3
4. Target FEC Stack sub-TLV for Segment Routing SID	4
4.1. Segment Routing Generic Label	4
4.2. FEC for Path validation	4
5. Procedures	5
5.1. SID to Interface Mapping	5
5.2. Initiator behavior	6
5.2.1. SRGL in Target FEC Stack TLV	6
5.3. Responder behavior	7
5.4. PHP flag behavior	8
6. IANA Considerations	8
7. Security Considerations	8
8. Acknowledgement	8
9. Contributors	8
10. References	8
10.1. Normative References	8
10.2. Informative References	9
Authors' Addresses	9

1. Introduction

[RFC8402] introduces and describes a Segment Routing architecture that leverages the source routing and tunneling paradigms. A node steers a packet through a controlled set of instructions called segments, by prepending the packet with Segment Routing header. A detailed definition of the Segment Routing architecture is available in [RFC8402]

As described in [RFC8402] and [I-D.ietf-spring-segment-routing-mpls], the Segment Routing architecture can be directly applied to an MPLS data plane, the Segment identifier (Segment ID) will be of 20-bits size and the Segment Routing header is the label stack.

1.1. Challenges with Existing Mechanism

[RFC8287] defines the mechanism to perform LSP Ping and Traceroute for Segment Routing with MPLS data plane. [RFC8287] defines the Target FEC Stack Sub-TLVs for IGP-Prefix Segment ID and IGP-Adjacency Segment ID.

There are various other Segment IDs proposed by different documents that are applicable for SR architecture.

[I-D.ietf-idr-bgp-prefix-sid] defines BGP Prefix Segment ID, [I-D.ietf-idr-bgppls-segment-routing-epe] defines BGP Peering Segment ID such as Peer Node SID, Peer Adj SID and Peer Set SID.

[I-D.sivabalan-pce-binding-label-sid] defines Path Binding Segment ID. As SR evolves for different usecases, we may see more types of SIDs defined in the future. This raises a need to propose new Target FEC Stack Sub-TLV for each such Segment ID that may need specific or network wide software upgrade to support such new Target FEC Stack Sub-TLVs.

So instead of proposing different Target FEC Stack Sub-TLV for each SID, this document attempt to propose a SR Generic Label Sub-TLV for Target FEC Stack TLV with the procedure to validate the associated instruction.

This document describes the new Target FEC Stack Sub-TLV that carries the SID and the assigner node information and the procedure to use LSP Ping and Traceroute using the new sub-tlv to support path validation and fault isolation for any SR Segment IDs.

2. Requirements notation

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 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when and only when, they appear in all capitals, as shown here.

3. Terminology

This document uses the terminologies defined in [RFC8402], [RFC8029], readers are expected to be familiar with it.

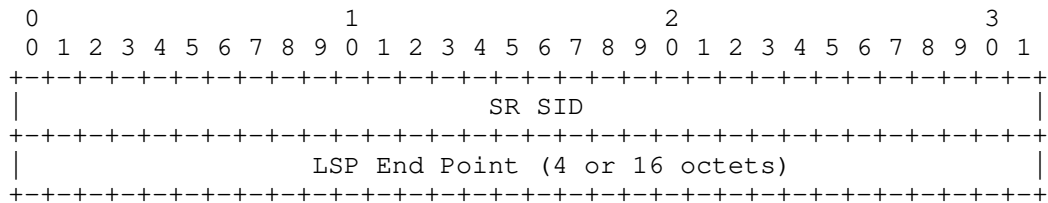
4. Target FEC Stack sub-TLV for Segment Routing SID

Following the procedure defined in [RFC8029], below defined Target FEC Stack Sub-TLV will be included for each labels in the stack. The below Sub-TLV is defined for Target FEC Stack TLV (Type 1), the Reverse-Path Target FEC Stack TLV (Type 16), and the Reply Path TLV (Type 21).

sub-Type	Value Field
TBD1	Segment Routing Generic Label (SRGL)

4.1. Segment Routing Generic Label

The format of the Sub-TLV is as specified below:



SR SID

Carries 20 bits of Segment ID that is used for validating the instruction.

LSP End Point

This field carries the node address of the end point that terminates the LSP.

4.2. FEC for Path validation

In SR architecture, any SID is associated with topology or service instruction. While the topology instruction steers the packet over best path or specific path, the service instruction instructs the type of service to be applied on the packet.

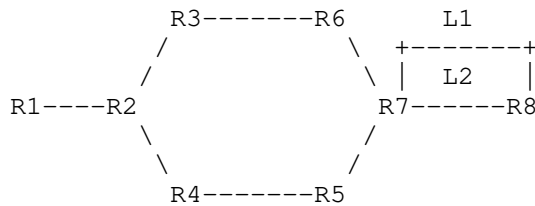


Figure 1: Segment Routing network

The Node Segment IDs for Rx for Algo 0 is 16000x. (Ex: For R1, it is 160001)
 The Node Segment IDs for Rx for Algo 128 is 16128x. (Ex: For R1, it is 161281)

9178 --> Adjacency Segment ID from R7 to R8 over link L1.
 9278 --> Adjacency Segment ID from R7 to R8 over link L2.
 9378 --> Parallel Adjacency Segment ID from R7 to R8 over Link L1 or L2.
 9187 --> Adjacency Segment ID from R8 to R7 over link L1.
 9287 --> Adjacency Segment ID from R8 to R7 over link L2.
 9387 --> Parallel Adjacency Segment ID from R8 to R7 over Link L1 or L2.

The instruction associated with any SID can be validated by verifying if the segment is terminated on the correct node and optionally received over the correct incoming interface. In Figure 1, inorder to validate the SID 9178, R1 can use {(SID=9178);(EndPoint=R8)} as FEC in Target FEC Stack Sub-TLV.

5. Procedures

This section describes the procedure to validate SR Generic Label Sub-TLV.

5.1. SID to Interface Mapping

Any End point MAY maintain a SID to Interface mapping table that maintains the below:

- o All the local Prefix/Node SID with any SR enabled interface as incoming interface.
- o All the Adj-SIDs assigned by directly connected remote nodes with the relevant interface incoming interface.

In Figure 1, R8 maintains 160008 and 161288 with Incoming interface as any SR enabled interface. Similarly, R8 maintains 9178 with Link L1 as incoming interface, 9278 with Link L2 as incoming interface and 9378 with Link L1 or L2 as incoming interface.

How this mapping is populated and maintained is a local implementation matter. It can be populated based on the IGP database or can be based on a query to Path Computation Element (PCE) controller. The mapping can be persistent or on-demand triggered by receiving LSP Ping Request.

5.2. Initiator behavior

This section defines the Target FEC Stack TLV construction mechanism by an initiator when using SR Generic Label Sub-TLV.

Ping

Initiator MUST include FEC(s) corresponding to the destination segment.

Initiator MAY include FECs corresponding to some or all of segments imposed in the label stack by the initiator to communicate the segments traversed.

Traceroute

Initiator MUST initially include FECs corresponding to all of segments imposed in the label stack.

When a received echo reply contains FEC Stack Change TLV with one or more of original segment(s) being popped, initiator MAY remove corresponding FEC(s) from Target FEC Stack TLV in the next (TTL+1) traceroute request as defined in section 4.6 of [RFC8029].

When a received echo reply does not contain FEC Stack Change TLV, initiator MUST NOT attempt to remove FEC(s) from Target FEC Stack TLV in the next (TTL+1) traceroute request.

5.2.1. SRGL in Target FEC Stack TLV

When the last segment ID in the label stack is IGP Prefix SID, Binding SID or BGP Prefix SID, set the LSP End Point field to the address of the Node that assigns the Prefix SID. The SR SID field is set to the value derived based on the index and the SRGB advertised by the LSP End Point.

When the last segment ID in the label stack is IGP Adj-SID or BGP Peering SID, set the LSP End Point field to the address of the adjacency node for which the SID is assigned to. The SR field is set to the Segment ID value.

How the above values are derived is a local implementation matter. It can be manually defined using CLI knob while triggering the LSP Ping Request or can use other mechanisms like querying the local database.

5.3. Responder behavior

Step 4a defined in Section 7.4 of [RFC8287] is updated as below:

If the Label-stack-depth is 0 and Target FEC Stack Sub-TLV at FEC-stack-depth is TBD1 (SRGL) {

- * Set the Best-return-code to 10 when LSP End Point Address does not match the local node address.
 - * Set the Best-return-code to 35, if Interface-I does not match the SID to Interface mapping for the received SR SID.
 - * set FEC-Status to 1, and return.
- }

If the Label-stack-depth is greater than 0 and Target FEC Stack Sub-TLV at FEC-stack-depth is TBD1 (SRGL), {

- * If the Label at Label-stack-depth is Imp-null {
 - + Set the Best-return-code to 10 when LSP End Point Address does not match the local node address.
 - + Set the Best-return-code to 35, if Interface-I does not match the SID to Interface mapping for the received SR SID.
 - + set FEC-Status to 1, and return.
- }
- * Else:
 - + Set the Best-return-code to 10 when the index derived from the label at Label-stack-depth is not advertised by LSP End Point.
 - + set FEC-Status to 1, and return.
- }

5.4. PHP flag behavior

To be Updated

6. IANA Considerations

To be Updated.

7. Security Considerations

To be Updated

8. Acknowledgement

TBD

9. Contributors

Danial Johari, Cisco Systems

10. References

10.1. Normative References

[I-D.ietf-idr-bgp-prefix-sid]

Previdi, S., Filsfils, C., Lindem, A., Sreekantiah, A.,
and H. Gredler, "Segment Routing Prefix SID extensions for
BGP", draft-ietf-idr-bgp-prefix-sid-27 (work in progress),
June 2018.

[I-D.ietf-idr-bgpls-segment-routing-epe]

Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray,
S., and J. Dong, "BGP-LS extensions for Segment Routing
BGP Egress Peer Engineering", draft-ietf-idr-bgpls-
segment-routing-epe-19 (work in progress), May 2019.

[I-D.sivabalan-pce-binding-label-sid]

Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J.,
Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID
in PCE-based Networks.", draft-sivabalan-pce-binding-
label-sid-07 (work in progress), July 2019.

[RFC2119]

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

- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [RFC8402] Filts, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

10.2. Informative References

- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filts, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

Authors' Addresses

Nagendra Kumar Nainar (editor)
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: naikumar@cisco.com

Carlos Pignataro (editor)
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Routing area
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2020

S. Hegde
K. Arora
S. Ninan
M. Srivastava
Juniper Networks Inc.
N. Kumar
Cisco Systems, Inc.
November 3, 2019

PMS/Head-end based MPLS Ping and Traceroute in Inter-AS SR Networks
draft-ninan-spring-mpls-inter-as-oam-02

Abstract

Segment Routing (SR) architecture leverages source routing and tunneling paradigms and can be directly applied to the use of a Multiprotocol Label Switching (MPLS) data plane. Segment Routing also provides an easy and efficient way to provide inter connectivity in a large scale network as described in [RFC8604]. [RFC8287] illustrates the problem and defines extensions to perform LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with an MPLS data plane. It is useful to have the LSP Ping and traceroute procedures when an SR end-to-end path spans across multiple ASes. This document describes mechanisms to facilitate LSP ping and traceroute in inter-AS SR networks in an efficient manner with simple OAM protocol extension which uses dataplane forwarding alone for sending Echo-Reply.

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 <https://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 May 6, 2020.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Reverse Path Segment List TLV	4
2.1. Reverse Path Segment List TLV definition	5
2.1.1. Segment sub-TLV	5
2.2. SRv6 Dataplane	9
3. Detailed Procedures	10
3.1. Sending an Echo-Request	10
3.2. Receiving an Echo-Request	10
3.3. Sending an Echo-Reply	10
4. Detailed Example	10
4.1. Procedures for Segment Routing LSP ping	11
4.2. Procedures for Segment Routing LSP Traceroute	12
5. Building Reverse Path Segment List TLV dynamically	12
5.1. The procedures to build the reverse path	12
5.2. Details with example	13
6. Security Considerations	13
7. IANA Considerations	13
8. Contributors	13
9. Acknowledgments	14
10. References	14
10.1. Normative References	14
10.2. Informative References	14
Authors' Addresses	16

1. Introduction

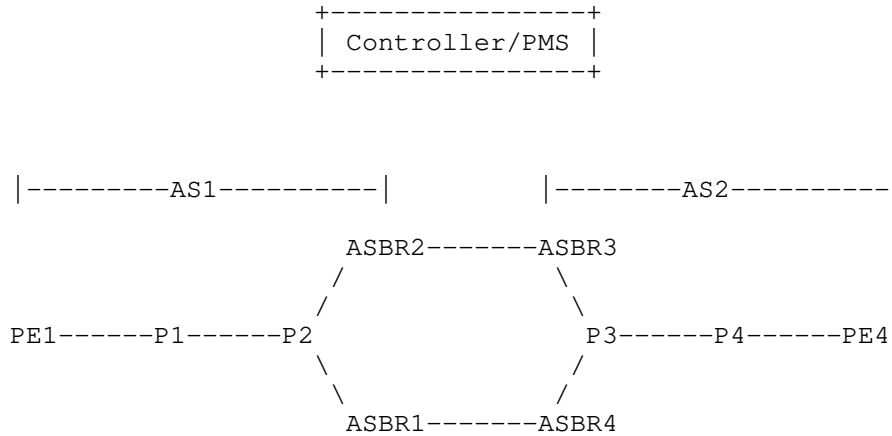


Figure 1: Inter-AS Segment Routing topology

Many network deployments have built their networks consisting of multiple Autonomous Systems either for ease of operations or as a result of network mergers and acquisitions. Segment Routing can be deployed in such scenarios to provide end to end paths, traversing multiple Autonomous systems(AS). These paths consist of Segment Identifiers(SID) of different type as per [RFC8402].

[I-D.ietf-spring-segment-routing-mpls] specifies the forwarding plane behaviour to allow Segment Routing to operate on top of MPLS data plane. [I-D.ietf-spring-segment-routing-central-epe] describes BGP peering SIDs, which will help in steering packet from one Autonomous system to another. Using above SR capabilities, paths which span across multiple Autonomous systems can be created.

For example Figure 1 describes an inter-AS network scenario consisting of ASes AS1 and AS2. Both AS1 and AS2 are Segment Routing enabled and the EPE links have EPE labels configured and advertised via [I-D.ietf-idr-bgpls-segment-routing-epe]. Controller or head-end can build end-to-end Traffic-Engineered path Node-SIDs, Adjacency-SIDs and EPE-SIDs. It is advantageous for operations to be able to perform LSP ping and traceroute procedures on these inter-AS SR paths. LSP ping/traceroute procedures use ip connectivity for Echo-reply to reach the head-end. In inter-AS networks, ip connectivity may not be there from each router in the path. For example in Figure 1 P3 and P4 may not have ip connectivity for PE1.

[RFC8403] describes mechanisms to carry out the MPLS ping/traceroute from a PMS. It is possible to build GRE tunnels or static routes to each router in the network to get IP connectivity for the reverse path. This mechanism is operationally very heavy and requires PMS to be capable of building huge number of GRE tunnels, which may not be feasible.

It is not possible to carry out LSP ping and Traceroute functionality on these paths to verify basic connectivity and fault isolation using existing LSP ping and Traceroute mechanism([RFC8287] and [RFC8029]). This is because, there exists no IP connectivity to source address of ping packet, which is in a different AS, from the destination of Ping/Traceroute.

[RFC7743] describes a Echo-relay based solution based on advertising a new Relay Node Address Stack TLV containing stack of Echo-relay ip addresses. This mechanism requires the return ping packet to reach the control plane on every relay node.

This document describes a mechanism which is efficient and simple and can be easily deployed in SR networks. This mechanism uses a new Reverse Path Segment List TLV to convey the reverse path. The TLV can either be derived by a smart application/controller which has a full topology view or by the help of intermediate nodes.

2. Reverse Path Segment List TLV

Segment Routing networks statically assign the labels to nodes and PMS/Head-end may know the entire database. The reverse path can be built from PMS/Head-end by stacking segments for the reverse path. A new TLV "Reverse Path Segment List TLV" is defined. Each TLV contains a list of segment sub-TLVs which may be a prefix/adjacency/binding SID/EPE SID. MPLS Echo -request should contain this TLV, which defines reverse path to reach source from the destination.

The new Reverse Path Segment List TLV is an optional TLV. This TLV is carried in the Echo-Request message. This optional TLV MAY appear in the Echo-request message in any order before or after Target FEC Stack TLV. The Reverse Path Segment List TLV is defined as below. Each MPLS Echo-request SHOULD contain this TLV in inter-AS cases, which will enable remote end(egress/transit routers) to send the reply to source.

In some cases, the head-end may not have complete visibility. In such cases, it can rely on downstream routers to build the reverse path. For this purpose, the TLV is carried in the Echo-Reply message. Section 5 describes one basic idea in this direction.

2.1. Reverse Path Segment List TLV definition

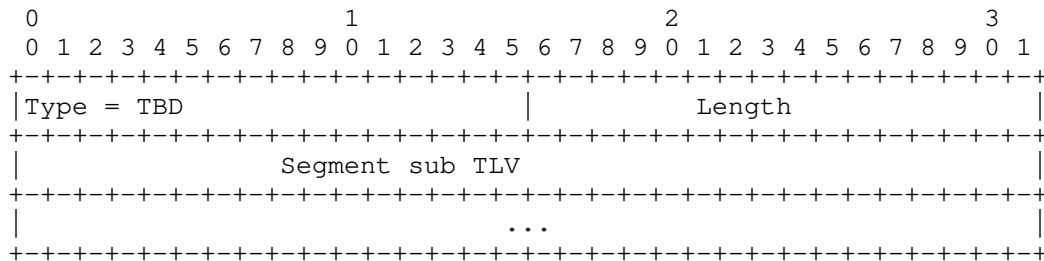


Figure 2: Reverse Path Segment List TLV

Type: TBD

Length: Length of TLV including TLV header and length of sub TLV.

There can be one or more segment sub-TLVs in a Reverse Path Segment List TLV. The applicable segment types are described in Section 2.1.1. The Segment type in a Reverse Path Segment List TLV MAY be same or different.

2.1.1. Segment sub-TLV

[I-D.ietf-spring-segment-routing-policy] defines various types of segments. These segment types are applicable here. One or more segment sub-TLV can be included. The segment sub-TLVs included MAY be of different types.

Below types of segment sub-TLVs are applicable for the Reverse Path Segment List Tlv.

Type 1: SID only, in the form of MPLS Label

Type 3: IPv4 Node Address with optional SID

Type 4: IPv6 Node Address with optional SID for SR MPLS

2.1.1.1. Type 1: SID only, in the form of MPLS Label

The Type-1 Segment Sub-TLV encodes a single SID in the form of an MPLS label. The format is as follows:

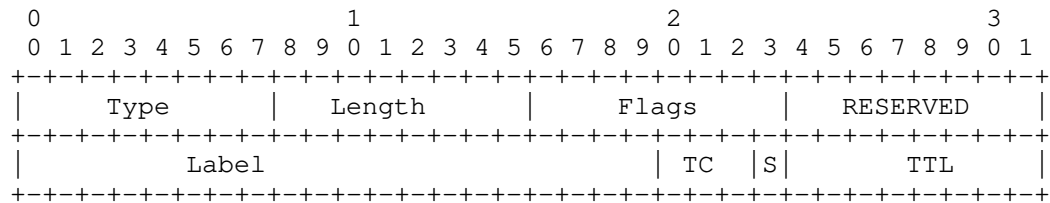


Figure 3: Type 1 Segment sub-TLV

where:

Type: 1 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in [I-D.ietf-idr-segment-routing-te-policy]).

Length is 6.

Flags: 1 octet of flags as defined in Section Section 2.1.1.4.

RESERVED: 1 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

Label: 20 bits of label value.

TC: 3 bits of traffic class

S: 1 bit of bottom-of-stack.

TTL: 1 octet of TTL.

The following applies to the Type-1 Segment sub-TLV:

The S bit SHOULD be zero upon transmission, and MUST be ignored upon reception.

If the originator wants the receiver to choose the TC value, it sets the TC field to zero.

If the originator wants the receiver to choose the TTL value, it sets the TTL field to 255.

If the originator wants to recommend a value for these fields, it puts those values in the TC and/or TTL fields.

The receiver MAY override the originator's values for these fields. This would be determined by local policy at the receiver. One

possible policy would be to override the fields only if the fields have the default values specified above.

2.1.1.2. Type 3: IPv4 Node Address with optional SID for SR-MPLS

The Type-3 Segment Sub-TLV encodes an IPv4 node address, SR Algorithm and an optional SID in the form of an MPLS label. The format is as follows:

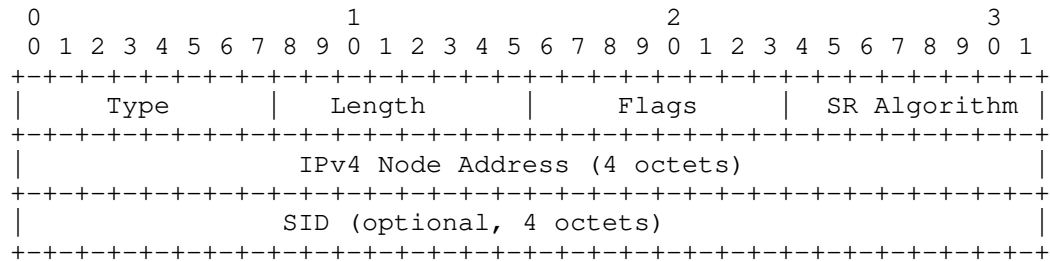


Figure 4: Type 3 Segment sub-TLV

where:

Type: 3 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in [I-D.ietf-idr-segment-routing-te-policy]).

Length is 6 or 10.

Flags: 1 octet of flags as defined in Section Section 2.1.1.4.

SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section Section 2.1.1.4 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be unset on transmission and MUST be ignored on receipt.

IPv4 Node Address: a 4 octet IPv4 address representing a node.

SID: 4 octet MPLS label.

The following applies to the Type-3 Segment sub-TLV:

The IPv4 Node Address MUST be present.

The SID is optional and specifies a 4 octet MPLS SID containing label, TC, S and TTL as defined in Section Section 2.1.1.1.

If length is 6, then only the IPv4 Node Address is present.

If length is 10, then the IPv4 Node Address and the MPLS SID are present.

2.1.1.3. Type 4: IPv6 Node Address with optional SID for SR MPLS

The Type-4 Segment Sub-TLV encodes an IPv6 node address, SR Algorithm and an optional SID in the form of an MPLS label. The format is as follows:

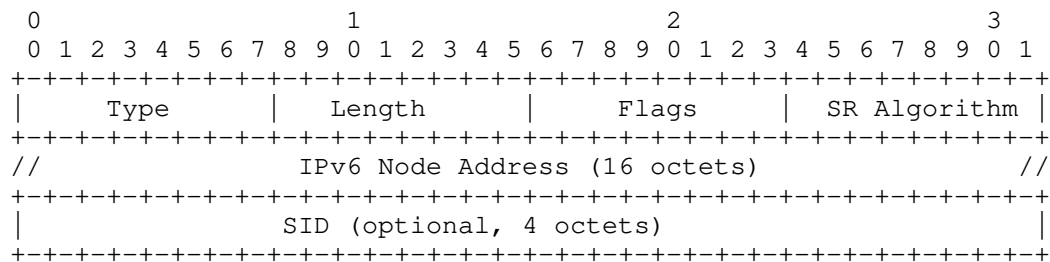


Figure 5: Type 4 Segment sub-TLV

where:

Type: 4 (to be assigned by IANA from the registry "SR Policy List Sub-TLVs" defined in [I-D.ietf-idr-segment-routing-te-policy]).

Length is 18 or 22.

Flags: 1 octet of flags as defined in Section Section 2.1.1.4.

SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section Section 2.1.1.4 is present. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy]. When A-Flag is not encoded, this field SHOULD be unset on transmission and MUST be ignored on receipt.

IPv6 Node Address: a 16 octet IPv6 address representing a node.

SID: 4 octet MPLS label.

The following applies to the Type-4 Segment sub-TLV:

The IPv6 Node Address MUST be present.

The SID is optional and specifies a 4 octet MPLS SID containing label, TC, S and TTL as defined in Section 2.1.1.1 .

If length is 18, then only the IPv6 Node Address is present.

If length is 22, then the IPv6 Node Address and the MPLS SID are present.

2.1.1.4. Segment Flags

The Segment Types described above MAY contain following flags in the "Flags" field (codes to be assigned by IANA from the registry "SR Policy Segment Flags" defined in [I-D.ietf-idr-segment-routing-te-policy])

```

  0 1 2 3 4 5 6 7
+---+---+---+---+
|V|A|           |
+---+---+---+---+
```

Figure 6: Flags

where:

V-Flag: This flag is used by SRPM for the purpose of "SID verification" as described in Section 5.1 in [I-D.ietf-spring-segment-routing-policy].

A-Flag: This flag indicates the presence of SR Algorithm id in the "SR Algorithm" field applicable to various Segment Types. SR Algorithm is used by SRPM as described in section 4 in [I-D.ietf-spring-segment-routing-policy].

Unused bits in the Flag octet SHOULD be set to zero upon transmission and MUST be ignored upon receipt.

The following applies to the Segment Flags:

V-Flag is applicable to all Segment Types.

A-Flag is applicable to Segment Types 3, 4 and 9. If A-Flag appears with any other Segment Type, it MUST be ignored.

2.2. SRv6 Dataplane

SRv6 dataplane is not in the scope of this document and will be addressed in a separate document.

3. Detailed Procedures

3.1. Sending an Echo-Request

In the inter-AS scenario when there is no reverse path connectivity, LSP ping initiator MUST add a Reverse Path Segment List TLV in the Echo-request message. The reverse Segment List MUST correspond to the return path from the egress. The Reverse Path Segment List TLV is an ordered list of Segments. The first Segment corresponds to the top Segment in MPLS header that the responder MUST use while sending the Echo-reply.

3.2. Receiving an Echo-Request

When a receiver does not understand the Reverse Path Segment List TLV, it SHOULD silently ignore the TLV and proceed with normal processing as described in [RFC8029]. When a Reverse Path Segment List TLV is received, and the responder supports processing it, it MUST use the Segments in Reverse Path Segment List TLV to build the echo-reply. The responder MUST follow the normal FEC validation procedures as described in [RFC8029] and [RFC8287] and this document does not suggest any change to those procedures. When the Echo-reply has to be sent out the Reverse Path Segment List TLV is used to construct the MPLS packet to send out.

3.3. Sending an Echo-Reply

The Echo-Reply message is sent as MPLS packet with a MPLS label stack. The Echo-Reply message MUST be constructed as described in the [RFC8029]. An MPLS packet is constructed with Echo-reply in the payload. The top label MUST be constructed from the first Segment from the Reverse Path Segment List TLV. The remaining labels MUST follow the order from the Reverse Path Segment List TLV. The responder MAY check the reachability of the top label in its own LFIB before sending the Echo-Reply.

4. Detailed Example

An example topology is given in Figure 1 . This will be used in below sections to explain LSP Ping and Traceroute procedures. The PMS/Head-end has complete view of topology. PE1, P1, P2, ASBR1 and ASBR2 are in AS1. Similarly ASBR3, ASBR4, P3, P4 and PE4 are in AS2.

AS1 and AS2 have Segment Routing enabled. IGPs like OSPF/ISIS are used to flood SIDs in each Autonomous System. The ASBR1, ASBR2, ASBR3, ASBR4 advertise BGP EPE SIDs for the inter-AS links. Topology of AS1 and AS2 are advertised via BGP-LS to the controller/PMS or

Head-end node. The EPE-SIDs are also advertised via BGP-LS as described in [I-D.ietf-idr-bgpls-segment-routing-epe]

The description in the document uses below notations for Segment Identifiers(SIDs).

Node SIDs : N-PE1, N-P1, N-ASBR1 etc.

Adjacency SIDs : Adj-PE1-P1, Adj-P1-P2 etc.

EPE SIDS : EPE-ASBR2-ASBR3, EPE-ASBR1-ASBR4, EPE-ASBR3-ASBR2 etc.

Let us consider a traffic engineered path built from PE1 to PE4 with Segment List stack as below. N-P1, N-ASBR1, EPE-ASBR1-ASBR4, N-PE4 for following procedures. This stack may be programmed by controller/PMS or Head-end router PE1 may have imported the whole topology information from BGP-LS and computed the inter-AS path.

4.1. Procedures for Segment Routing LSP ping

To perform LSP ping procedure on an SR-Path from PE1 to PE4 consisting of label stacks [N-P1,N-ASBR1,EPE-ASBR1-ASBR4, N-PE4], The remote end(PE4) needs IP connectivity to head end(PE1) for the Segment Routing ping to succeed, because Echo-reply needs to travel back to PE1 from PE4. But in typical deployment scenario there will be no ip route from PE4 to PE1 as they belong to different ASes.

PE1 adds Reverse Path from PE4 to PE1 in the MPLS Echo-request using multiple Segments in "Reverse Path Segment List TLV" as defined above. An example reverse path Segment List for PE1 to PE4 for LSP ping is [N-ASBR4, EPE-ASBR4-ASBR1, N-PE1]. An implementation may also build a Reverse Path Segment List consisting of labels to reach its own AS. Once the label stack is popped-off the Echo-reply message will be exposed. The further packet forwarding will be based on ip lookup. An example Reverse Path Segment List for this case could be [N-ASBR4, EPE-ASBR4-ASBR1].

On receiving MPLS Echo-request PE4 first validates FEC in the Echo-request. PE4 then builds label stack to send the response from PE4 to PE1 by copying the labels from "Reverse Path Segment List TLV". PE4 builds the Echo-reply packet with the MPLS label stack constructed and imposes MPLS headers on top of Echo-reply packet and sends out the packet towards PE1. This Segment List stack can successfully steer reply back to Head-end node(PE1).

4.2. Procedures for Segment Routing LSP Traceroute

As described in the procedures for LSP ping, the reverse Segment List may be sent from head-end in which case the LSP Traceroute procedures are similar to LSP ping. The head-end constructs the Reverse Path Segment List TLV and the egress node uses the Reverse Path Segment List to construct the Echo-reply packet header. Head-end/PMS is aware of the reverse path from every node visited in the network and builds the Reverse Path Segment List for every visited node accordingly.

For Example:

For the same traffic engineered path PE1 to PE4 mentioned in above sections, let us assume there is no reverse path available from the nodes ASBR4 to PE1. During the Traceroute procedure, when PE1 has to visit ASBR4, it builds reverse Path Label Stack TLV and includes label to the border-node which has the route to, PE1. In this example the Reverse Path Segment List TLV will contain [EPE-ASBR4-ASBR1]. Further down the traceroute procedure when P3 or P4 node is being visited, PE1 build the Reverse Path Segment List TLV containing [N-ASBR4, EPE-ASBR4-ASBR1]. The Echo-reply will be an MPLS packet with this label stack and will be forwarded to PE1.

5. Building Reverse Path Segment List TLV dynamically

In some cases, the head-end may not have complete visibility of Inter-AS topology. In such cases, it can rely on downstream routers to build the reverse path for mpls traceroute procedures. For this purpose, the Reverse Path Segment List TLV is carried in the Echo-Reply.

5.1. The procedures to build the reverse path

When an ASBR receives an echo-request from another AS, and ASBR is configured to build the Reverse Path dynamically, ASBR MUST build a Reverse Path Segmnet List TLV and add it in echo-reply. ASBR MUST locally decide the outgoing interface for the echo-reply packet. Generally, remote ASBR will choose interface on which the incoming OAM packet was receieved to send the echo-reply out. Reverse Path Segment List TLV is built by adding two segment sub TLVs. The top segment sub TLV consists of the ASBR's Node SID and second segment consists of the EPE SID in the reverse direction to reach the AS from which the OAM packet was received. The type of segment chosen to build Reverse Path Segment List TLV is implementation dependent. In cases where the AS is configured with different SRGBs, the Node SID of the ASBR should be represented using type 3 segment so that all the nodes inside the AS can correctly translate the Node-SID to a label.

Irrespective of which type of segment is included in the Reverse Path Segment List TLV, the responder of echo-request always translates the Reverse Path Segment List TLV to a label stack and builds MPLS header for the the echo-reply packet.

5.2. Details with example

Let us consider a traffic engineered path built from PE1 to PE4 with a label stack as below. N-P1, N-ASBR1, EPE-ASBR1-ASBR4, N-PE4 for the following procedures. This traceroute doesn't need any Reverse Path Segment List TLV till it leaves AS1, because IP connectivity will be there to send echo-reply. But this traceroute requires Reverse Path Segment List TLV once it starts probing AS2 routers. According to this procedure, ASBR4 should add Reverse Path Segment List TLV in its echo-reply. ASBR4 should form this Reverse Path Segment List TLV using its own Node SID(N-ASBR4) and EPE SID (EPE-ASRB4-ASBR1) labels. Then PE1 should use this Reverse Path Segment List TLV in subsequent echo-requests. In this example, when the subsequent echo-request reaches P3, it should use this Reverse Path Segment List TLV for sending the echo-reply. The same Reverse Path Segment List TLV is enough for any router in AS2 to send the reply. Because the first label(N-ASBR4) can direct echo-reply to ASBR4 and second one (EPE-ASBR4-ASBR1) to direct echo-reply to AS1. Once echo reply reaches AS1, normal IP forwarding helps it to reach PE1 or the head-end.

6. Security Considerations

TBD

7. IANA Considerations

Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters TLVs Registry

Reverse Path Segment List TLV : TBD

8. Contributors

1.Carlos Pignataro

Cisco Systems, Inc.

cpignata@cisco.com

2. Zafar Ali

Cisco Systems, Inc.

zali@cisco.com

9. Acknowledgments

Thanks to Bruno Decreane for suggesting use of generic Segment sub-TLV.

10. References

10.1. Normative References

- [I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-07 (work in progress), July 2019.
- [I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", draft-ietf-spring-segment-routing-central-epe-10 (work in progress), December 2017.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.

10.2. Informative References

- [I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-19 (work in progress), May 2019.
- [I-D.ietf-mpls-interas-lspping]
Nadeau, T. and G. Swallow, "Detecting MPLS Data Plane Failures in Inter-AS and inter-provider Scenarios", draft-ietf-mpls-interas-lspping-00 (work in progress), March 2007.

- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-03 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7743] Luo, J., Ed., Jin, L., Ed., Nadeau, T., Ed., and G. Swallow, Ed., "Relayed Echo Reply Mechanism for Label Switched Path (LSP) Ping", RFC 7743, DOI 10.17487/RFC7743, January 2016, <<https://www.rfc-editor.org/info/rfc7743>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <<https://www.rfc-editor.org/info/rfc8403>>.
- [RFC8604] Filsfils, C., Ed., Previdi, S., Dawra, G., Ed., Henderickx, W., and D. Cooper, "Interconnecting Millions of Endpoints with Segment Routing", RFC 8604, DOI 10.17487/RFC8604, June 2019, <<https://www.rfc-editor.org/info/rfc8604>>.

Authors' Addresses

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Kapil Arora
Juniper Networks Inc.

Email: kapilaro@juniper.net

Samson Ninan
Juniper Networks Inc.

Email: samsonn@juniper.net

Mukul Srivastava
Juniper Networks Inc.

Email: msri@juniper.net

Nagendra Kumar
Cisco Systems, Inc.

Email: naikumar@cisco.com

MPLS
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

Q. Xiong
G. Mirsky
ZTE Corporation
W. Cheng
China Mobile
July 13, 2020

The Use of Path Segment in SR-MPLS and MPLS Interworking
draft-xiong-mpls-path-segment-sr-mpls-interworking-02

Abstract

This document illustrates the SR-MPLS and MPLS interworking scenarios to support end-to-end bidirectional tunnel across multiple domains with the use of Path Segments.

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 <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	4
3. SR-MPLS Interworking with MPLS	4
3.1. Stitching of Path Segments	5
3.2. Nesting of Path Segments	6
4. Security Considerations	7
5. Acknowledgements	7
6. IANA Considerations	7
7. Normative References	8
Authors' Addresses	8

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an SR Policy instantiated as an ordered list of instructions called "segments". SR supports a per-flow explicit routing while maintaining per-flow state only at the ingress nodes of the SR domain. Segment Routing can be instantiated on MPLS data plane which is referred to as SR-MPLS [RFC8660]. SR-MPLS leverages the MPLS label stack to construct the SR path.

IP/MPLS technology can be deployed in domains, which may serve as an access, aggregation, or core network. Further, using SR architecture, the IP/MPLS network may be upgraded to support the SR-MPLS technology. As such transformation is performed incrementally, by one domain at the time, operators are faced with a requirement to support the interworking between MPLS and SR-MPLS networks at the boundaries to provide the end-to-end bidirectional service. As defined in [RFC8402], the headend of an SR Policy binds a Binding Segment ID (B-SID) to its policy. The B-SID could be bound to a SID List or selected path and used to stitch the SR list and the SR Label Switched Paths (LSP) across multiple domains. The use of the B-SID is recommended to reduce the size of the label stack and stitch the SR LSPs.

In some scenarios, for example, a mobile backhaul transport network, it is required to provide end-to-end bidirectional path across SR and MPLS networks. The Path Segment as defined in [I-D.ietf-spring-mpls-path-segment] can be used to support bidirectional tunnel scenarios such as SR path Performance Measurement (PM), end-to-end 1+1 SR path protection and bidirectional SR paths correlation.

This document illustrates the SR-MPLS and MPLS interworking scenarios to support end-to-end bidirectional tunnel across multiple domains with the use of Path Segments.

2. Conventions used in this document

2.1. Terminology

ABR: Area Border Routers. Routers used to connect two IGP areas (areas in OSPF or levels in IS-IS).

AS: Autonomous System. An Autonomous System is composed by one or more IGP areas.

ASBR: Autonomous System Border Router. A router used to connect together ASes of the same or different service providers via one or more inter-AS links.

Border Node: An ABR that interconnects two or more IGP areas.

Border Link: Two ASes are interconnected with ASBRs.

B-SID: Binding Segment ID.

Domains: Autonomous System (AS) or IGP Area. An Autonomous System is composed of one or more IGP areas.

e-PSID: end-to-end Path Segment.

IGP: Interior Gateway Protocol.

N-PSID: Nesting of Path Segments.

PM: Performance Measurement.

SID: Segment ID.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

S-PSID: Stitching of Path Segments.

VPN: Virtual Private Network.

2.2. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. SR-MPLS Interworking with MPLS

It is required to establish the end-to-end Virtual Private Network (VPN) service across the access network, aggregation network, and core network. For example, SR-MPLS may be deployed in access and core network, and MPLS may be deployed in the aggregation network. The network interworking should be taken into account in deployment are the following:

- o Border Node or Border Link
- o Stitching of Path Segments or Nesting of Path Segments
- o End-to-end Path Monitoring

The domains of the networks may be IGP Areas or ASes. The SR-MPLS and MPLS networks can be interconnected with a border node between IGP areas or border links between ASes. MPLS domain can be deployed between two SR-MPLS domains, as Figure 1 shows. The packets being transmitted along the SR path in SR-MPLS domains by using the SID list at the ingress node. And the path in MPLS domains can be pre-configuration either via NMS or via the MPLS control plane signaling. This document takes border node scenarios across IGP Areas domains for example. The border link scenarios are in future discussion.

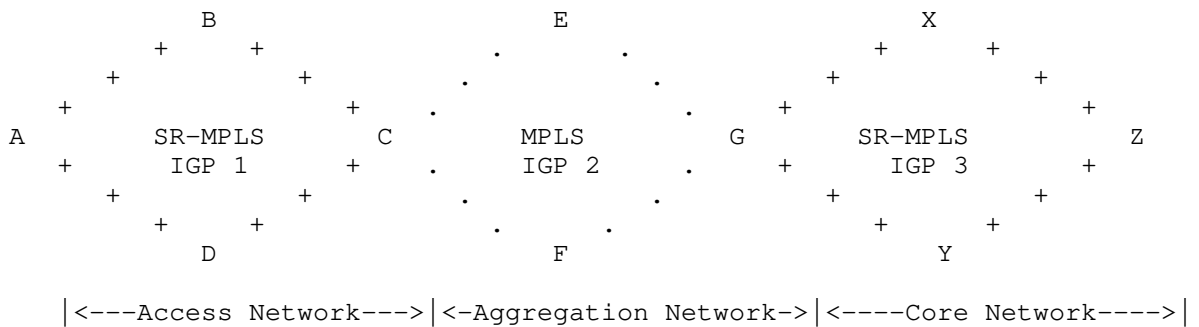


Figure 1: SR-MPLS and MPLS interworking Scenario

The VPN service across the SR-MPLS and MPLS domains is an end-to-end bidirectional path. In the SR-MPLS network, a Path Segment uniquely identifies an SR path and can be used for the end-to-end bidirectional path. This document illustrates the end-to-end Path Segment used in the interworking scenario including the stitching and nesting models. As described in [I-D.ietf-spring-mpls-path-segment], an end-to-end path segment or PSID (e-PSID), is also referred to as Nesting of Path SID (N-PSID) in nesting model or Stitching of Path SID (S-PSID) in stitching model.

3.1. Stitching of Path Segments

It is a common requirement that SR-MPLS needs to interwork with MPLS when SR is incrementally deployed in the MPLS domain. Figure 2 shows the stitching of Path Segments in SR-MPLS interworking with MPLS. The SR-LSPs and IP/MPLS LSPs are established independently in each domain which consist of SID list or MPLS label. The end-to-end bidirectional path acrossing the SR-MPLS and MPLS networks is split into multiple segments which can be identified by the S-PSID. The end-to-end path is terminated at the egress node in egress domain. The S-PSID will be popped out at the border node in each domain and correlated to the S-PSID of next domain.

The correlation of S-PSIDs can bind the segments of end-to-end path. The S-PSIDs are valid in the corresponding domain and the border nodes maintain the forwarding entries of that S-PSID segment that maps to the next S-PSID and the related path segments. In the headend node, the S-PSID can correlate the inter-domain path of reverse direction and bind the two unidirectional paths. The stitching of Path Segments can support the end-to-end path stitching and monitoring.

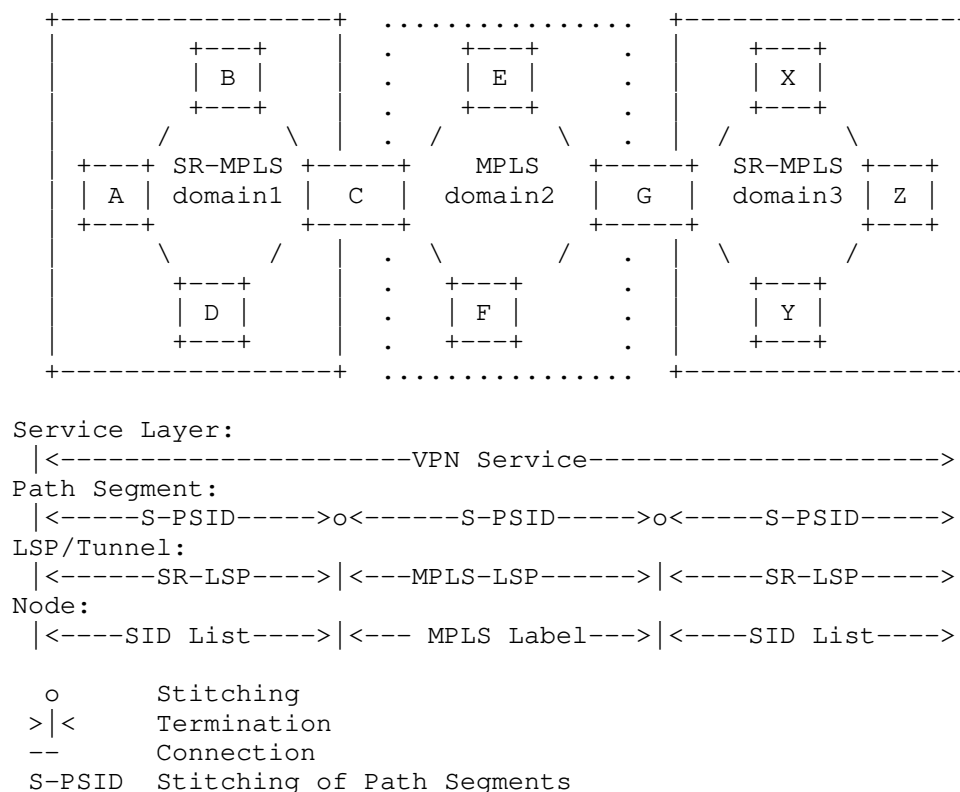


Figure 2: Stitching of Path Segments in SR-MPLS and MPLS interworking

3.2. Nesting of Path Segments

Figure 3 displays the nesting of Path Segments in SR-MPLS and MPLS interworking. The SR-LSPs and IP/MPLS LSPs are established in respective domain which consist of SID list or MPLS label. The SR-LSPs and IP/MPLS LSPs may be stitched across domains with B-SID. Comparing with S-PSID in the stitching model, the N-PSID presents end-to-end encapsulation in the packet from an SR-MPLS domain to an MPLS domain which is encapsulated at the ingress nodes and decapsulated at the egress nodes. The transit nodes, even the border nodes of domains, are not aware of the N-PSID.

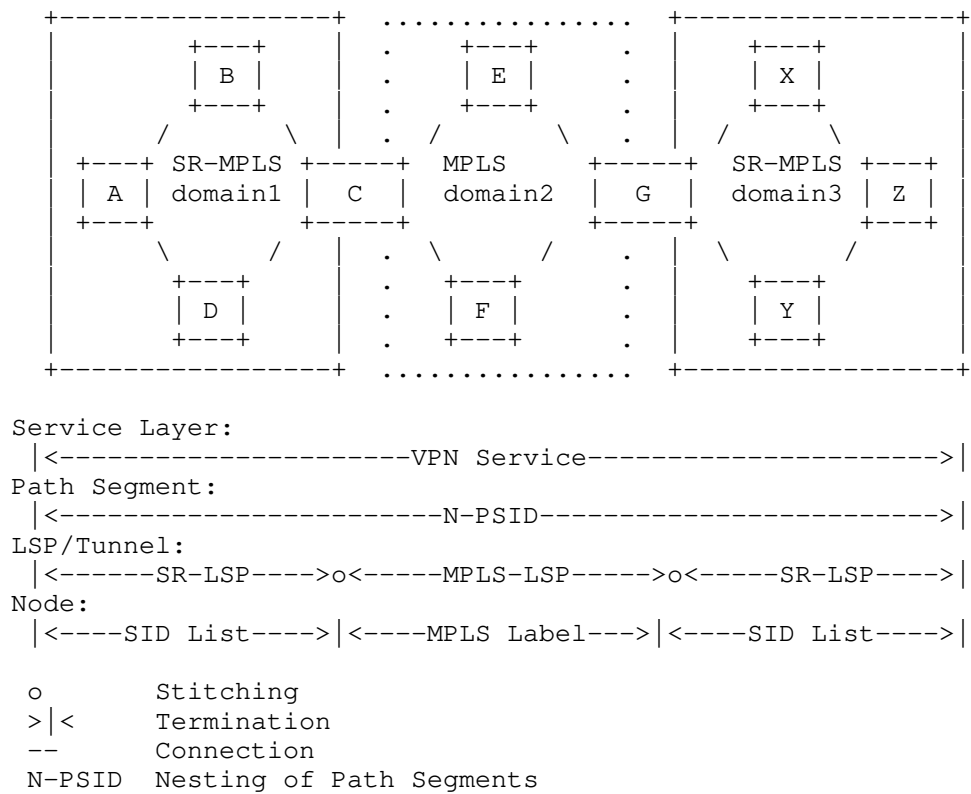


Figure 3: Nesting of Path Segments in SR-MPLS and MPLS interworking

4. Security Considerations

TBA

5. Acknowledgements

TBA

6. IANA Considerations

TBA

7. Normative References

- [I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler,
"Path Segment in MPLS Based Segment Routing Network",
draft-ietf-spring-mpls-path-segment-02 (work in progress),
February 2020.
- [I-D.xiong-spring-path-segment-sr-inter-domain]
Xiong, Q., Mirsky, G., and W. Cheng, "The Use of Path
Segment in SR Inter-domain Scenarios", draft-xiong-spring-
path-segment-sr-inter-domain-01 (work in progress),
October 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,
Decraene, B., Litkowski, S., and R. Shakir, "Segment
Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,
July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S.,
Decraene, B., Litkowski, S., and R. Shakir, "Segment
Routing with the MPLS Data Plane", RFC 8660,
DOI 10.17487/RFC8660, December 2019,
<<https://www.rfc-editor.org/info/rfc8660>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 13, 2019

L. Zheng
G. Zheng
Huawei Technologies
G. Mirsky
ZTE Corp.
R. Rahman
F. Iqbal
Cisco Systems
January 9, 2019

YANG Data Model for LSP-Ping
draft-zheng-mpls-lsp-ping-yang-cfg-10

Abstract

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. RFC 8029 defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in RFC 6020 and RFC 7950, is a data modeling language used to specify the contents of a conceptual data stores that allows networked devices to be managed using NETCONF, as specified in RFC 6241. This document defines a YANG data model that can be used to configure and manage LSP-Ping.

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 <https://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 July 13, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://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. Requirements Language	3
1.2. Support of Long Running Command with NETCONF	3
2. Scope	3
3. Design of the Data Model	4
3.1. The Configuration of Control Information	4
3.2. The Configuration of Schedule Parameters	5
3.3. Display of Result Information	6
4. Data Hierarchy	7
5. Interaction with other MPLS OAM Tools Models	9
6. LSP-Ping YANG Module	10
7. Examples	21
7.1. Configuration of Control Information	21
7.2. The Configuration of Schedule Parameters	22
7.3. Display of Result Information	23
8. Security Considerations	25
9. IANA Considerations	26
Contributors	26
Acknowledgments	27
12. References	27
12.1. Normative References	27
12.2. Informative References	27
Authors' Addresses	28

1. Introduction

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. [RFC8029] defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in [RFC6020] and [RFC7950], is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. This document defines a YANG data model that can be used to configure and manage LSP-Ping [RFC8029].

The rest of this document is organized as follows. Section 2 presents the scope of this document. Section 3 provides the design of the LSP-Ping configuration data model in details by containers. Section 4 presents the complete data hierarchy of LSP-Ping YANG model. Section 5 discusses the interaction between LSP-Ping data model and other MPLS tools data models. Section 6 specifies the YANG module and section 7 lists examples which conform to the YANG module specified in this document. Finally, security considerations are discussed in Section 8.

This version of the LSP Ping data model conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

1.1. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Support of Long Running Command with NETCONF

LSP Ping is one of the examples of what can be described as "long-running operation". Unlike most of the configuration operations that result in single response execution of an LSP Ping triggers multiple responses from a node under control. The question of implementing the long-running operation in NETCONF is still open and possible solutions being discussed:

1. Consecutive Remote Processing Calls (RPC) to poll for results.
2. Model presented in [RFC4560].
3. The one outlined in [I-D.mahesh-netconf-persistent].

The problem of long-running operation as well can be considered as a case of controlling and obtaining results from a Measurement Agent (MA) as defined in [RFC7594].

2. Scope

The fundamental mechanism of LSP-Ping is defined in [RFC8029]. Extensions of LSP-Ping has been developed over the years. There are extensions for performing LSP ping, for example, over P2MP MPLS LSPs [RFC6425] or for Segment Routing IGP Prefix and Adjacency SIDs with an MPLS data plane [RFC8287]. These extensions will be considered in a later update of this document.

3. Design of the Data Model

This YANG data model is defined to be used to configure and manage LSP-Ping and it provides the following features:

1. The configuration of control information of an LSP-Ping test.
2. The configuration of schedule parameters of an LSP-Ping test.
3. Display of result information of an LSP-Ping test.

The top-level container `lsp-pings` holds the configuration of the control information, schedule parameters and result information for multiple instances of LSP-Ping test.

3.1. The Configuration of Control Information

Container `lsp-pings:lsp-ping:control-parameters` defines the configuration parameters which control an LSP-Ping test. Examples are the `target-fec-type/target-fec` of the echo request packet and the `reply mode` of the echo reply packet. Values of some parameters may be auto-assigned by the system, but in several cases, there is a requirement for configuration of these parameters. Examples of such parameters are source address and outgoing interface.

The data hierarchy for control information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
        +--rw target-fec-type?      target-fec-type
        +--rw (target-fec)?
          +--:(ip-prefix)
            +--rw ip-address?        inet:ip-address
          +--:(bgp)
            +--rw bgp?                inet:ip-address
          +--:(rsvp)
            +--rw tunnel-interface?  string
          +--:(vpn)
            +--rw vrf-name?           uint32
            +--rw vpn-ip-address?     inet:ip-address
          +--:(pw)
            +--rw vcid?               uint32
          +--:(vpls)
            +--rw vsi-name?           string
        +--rw traffic-class?        uint8
        +--rw reply-mode?            reply-mode
        +--rw timeout?               uint32
        +--rw timeout-units?         units
        +--rw interval?              uint32
        +--rw interval-units?        units
        +--rw probe-count?           uint32
        +--rw data-size?              uint32
        +--rw data-fill?              string
        +--rw description?            string
        +--rw source-address?         inet:ip-address
        +--rw ttl?                    uint8
        +--rw (outbound)?
          +--:(interface)
            +--rw interface-name?     string
          +--:(nexthop)
            +--rw nexthop?             inet:ip-address

```

3.2. The Configuration of Schedule Parameters

Container `lsp-pings:lsp-ping:scheduling-parameters` defines the schedule parameters of an LSP-Ping test, which describes when to start and when to end the test. Four start modes and three end modes are defined respectively. To be noted that, the configuration of "interval" and "probe-count" parameter defined in container `lsp-pings:lsp-ping:control-parameters` could also determine when the test ends implicitly. All these three parameters are optional. If the user

does not configure either "interval" or "probe-count" parameter, then the default values will be used by the system. If the user configures "end-test", then the actual end time of the LSP-Ping test is the smaller one between the configuration value of "end-test" and the time implicitly determined by the configuration value of "interval"/"probe-count".

The data hierarchy for schedule information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
        +--rw (start-test)?
          +--:(now)
            | +--rw start-test-now?          empty
          +--:(at)
            | +--rw start-test-at?           yang:date-and-time
          +--:(delay)
            | +--rw start-test-delay?        uint32
            | +--rw start-test-delay-units?  units
          +--:(daily)
            | +--rw start-test-daily?        yang:date-and-time
        +--rw (end-test)?
          +--:(at)
            | +--rw end-test-at?             yang:date-and-time
          +--:(delay)
            | +--rw end-test-delay?          uint32
            | +--rw end-test-delay-units?    units
          +--:(lifetime)
            | +--rw end-test-lifetime?       uint32
            | +--rw lifetime-units?         units

```

3.3. Display of Result Information

Container `lsp-pings:lsp-ping:result-info` shows the result of the current LSP-Ping test. Both the statistical result e.g. `min-rtt`, `max-rtt`, and per test probe result e.g. `return code`, `return subcode`, are shown.

The data hierarchy for display of result information is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
      ...
      +--ro result-info
        +--ro operational-status?    operational-status
        +--ro source-address?        inet:ip-address
        +--ro target-fec-type?       target-fec-type
        +--ro (target-fec)?
          +--:(ip-prefix)
            | +--ro ip-address?       inet:ip-address
          +--:(bgp)
            | +--ro bgp?              inet:ip-address
          +--:(rsvp)
            | +--ro tunnel-interface? string
          +--:(vpn)
            | +--ro vrf-name?         uint32
            | +--ro vpn-ip-address?   inet:ip-address
          +--:(pw)
            | +--ro vcid?              uint32
          +--:(vpls)
            | +--ro vsi-name?         string
        +--ro min-rtt?               uint32
        +--ro max-rtt?               uint32
        +--ro average-rtt?           uint32
        +--ro probe-responses?       uint32
        +--ro sent-probes?           uint32
        +--ro sum-of-squares?        uint32
        +--ro last-good-probe?       yang:date-and-time
        +--ro probe-results
          +--ro probe-result* [probe-index]
            +--ro probe-index         uint32
            +--ro return-code?        uint8
            +--ro return-sub-code?     uint8
            +--ro rtt?                uint32
            +--ro result-type?        result-type

```

4. Data Hierarchy

The complete data hierarchy of LSP-Ping YANG model is presented below.

```

module: ietf-lsp-ping

```

```

+--rw lsp-pings
  +--rw lsp-ping* [lsp-ping-name]
    +--rw lsp-ping-name          string
    +--rw control-parameters
      +--rw target-fec-type?      target-fec-type
      +--rw (target-fec)?
        +--:(ip-prefix)
          +--rw ip-address?       inet:ip-address
        +--:(bgp)
          +--rw bgp?              inet:ip-address
        +--:(rsvp)
          +--rw tunnel-interface? string
        +--:(vpn)
          +--rw vrf-name?         uint32
          +--rw vpn-ip-address?   inet:ip-address
        +--:(pw)
          +--rw vcid?             uint32
        +--:(vpls)
          +--rw vsi-name?         string
      +--rw traffic-class?        uint8
      +--rw reply-mode?           reply-mode
      +--rw timeout?              uint32
      +--rw timeout-units?        units
      +--rw interval?            uint32
      +--rw interval-units?       units
      +--rw probe-count?          uint32
      +--rw data-size?            uint32
      +--rw data-fill?            string
      +--rw description?          string
      +--rw source-address?       inet:ip-address
      +--rw ttl?                  uint8
      +--rw (outbound)?
        +--:(interface)
          +--rw interface-name?   string
        +--:(nexthop)
          +--rw nexthop?          inet:ip-address
    +--rw scheduling-parameters
      +--rw (start-test)?
        +--:(now)
          +--rw start-test-now?    empty
        +--:(at)
          +--rw start-test-at?     yang:date-and-time
        +--:(delay)
          +--rw start-test-delay?   uint32
          +--rw start-test-delay-units? units
        +--:(daily)
          +--rw start-test-daily?   yang:date-and-time
      +--rw (end-test)?

```



```

    +---:(at)
    |   +---rw end-test-at?                yang:date-and-time
    +---:(delay)
    |   +---rw end-test-delay?             uint32
    |   +---rw end-test-delay-units?       units
    +---:(lifetime)
    |   +---rw end-test-lifetime?          uint32
    |   +---rw lifetime-units?            units
+---ro result-info
+---ro operational-status?                operational-status
+---ro source-address?                   inet:ip-address
+---ro target-fec-type?                  target-fec-type
+---ro (target-fec)?
|   +---:(ip-prefix)
|   |   +---ro ip-address?                inet:ip-address
+---:(bgp)
|   +---ro bgp?                          inet:ip-address
+---:(rsvp)
|   +---ro tunnel-interface?             string
+---:(vpn)
|   +---ro vrf-name?                     uint32
|   +---ro vpn-ip-address?               inet:ip-address
+---:(pw)
|   +---ro vcid?                         uint32
+---:(vpls)
|   +---ro vsi-name?                     string
+---ro min-rtt?                          uint32
+---ro max-rtt?                          uint32
+---ro average-rtt?                      uint32
+---ro probe-responses?                  uint32
+---ro sent-probes?                      uint32
+---ro sum-of-squares?                   uint32
+---ro last-good-probe?                  yang:date-and-time
+---ro probe-results
    +---ro probe-result* [probe-index]
        +---ro probe-index                uint32
        +---ro return-code?               uint8
        +---ro return-sub-code?           uint8
        +---ro rtt?                       uint32
        +---ro result-type?               result-type

```

5. Interaction with other MPLS OAM Tools Models

TBA

6. LSP-Ping YANG Module

```
<CODE BEGINS> file "ietf-lsp-ping@2018-11-29.yang"
module ietf-lsp-ping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lsp-ping";
  //namespace need to be assigned by IANA
  prefix "lsp-ping";

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types{
    prefix yang;
    reference "RFC 6991: Common YANG Types.";
  }

  organization "IETF Multiprotocol Label Switching Working Group";

  contact
    "WG Web: http://tools.ietf.org/wg/mppls/
    WG List: mppls@ietf.org

    Editor: Greg Mirsky
      gregimirsky@gmail.com
    Editor: Lianshu Zheng
      vero.zheng@huawei.com
    Editor: Guangying Zheng
      zhengguangying@huawei.com
    Editor: Reshad Rahman
      rrahman@cisco.com
    Editor: Faisal Iqbal
      faiqbal@cisco.com";

  description
    "This YANG module specifies a vendor-independent model
    for the LSP Ping.

    This YANG data model is defined to be used to configure and manage
    LSP-Ping and it provides the following features:
    1. The configuration of control information of an LSP-Ping test.
    2. The configuration of schedule parameters of an LSP-Ping test.
    3. Display of result information of an LSP-Ping test.

    Copyright (c) 2018 IETF Trust and the persons identified as
    the document authors. All rights reserved.
    Redistribution and use in source and binary forms, with or
```

without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

```
reference "draft-zheng-mpls-lsp-ping-yang-cfg";

revision "2018-11-29" {
  description
    "10 version, refine the target fec type,
    as per RFC8029 and update Security Considerations section.";
  reference "draft-zheng-mpls-lsp-ping-yang-cfg";
}

typedef target-fec-type {
  type enumeration {
    enum ip-prefix {
      value "0";
      description "IPv4/IPv6 prefix";
    }
    enum bgp {
      value "1";
      description "BGP IPv4/IPv6 prefix";
    }
    enum rsvp {
      value "2";
      description "Tunnel interface";
    }
    enum vpn {
      value "3";
      description "VPN IPv4/IPv6 prefix";
    }
    enum pw {
      value "4";
      description "FEC 128 pseudowire IPv4/IPv6";
    }
    enum vpls {
      value "5";
      description "FEC 129 pseudowire IPv4/IPv6";
    }
  }
  description "Target FEC type, as defined in RFC 8029";
}
```

```
typedef reply-mode {
  type enumeration {
    enum do-not-reply {
      value "1";
      description "Do not reply";
    }
    enum reply-via-udp {
      value "2";
      description "Reply via an IPv4/IPv6 UDP packet";
    }
    enum reply-via-udp-router-alert {
      value "3";
      description
        "Reply via an IPv4/IPv6 UDP packet with Router Alert";
    }
    enum reply-via-control-channel {
      value "4";
      description
        "Reply via application level control channel";
    }
  }
  description "Reply mode";
}

typedef units {
  type enumeration {
    enum seconds {
      description "Seconds";
    }
    enum milliseconds {
      description "Milliseconds";
    }
    enum microseconds {
      description "Microseconds";
    }
    enum nanoseconds {
      description "Nanoseconds";
    }
  }
  description "Time units";
}

typedef operational-status {
  type enumeration {
    enum enabled {
      value "1";
      description "The Test is active";
    }
  }
}
```

```
    enum disabled {
        value "2";
        description "The test has stopped";
    }
    enum completed {
        value "3";
        description "The test is completed";
    }
}
description "Operational state of an LSP Ping test";
}

typedef result-type {
    type enumeration {
        enum success {
            value "1";
            description "The test probe is successful";
        }
        enum fail {
            value "2";
            description "The test probe has failed";
        }
        enum timeout {
            value "3";
            description "The time of the test probe has expired";
        }
    }
    description "Result of each LSP Ping test probe";
}

container lsp-pings {
    description "Multi-instance of the LSP Ping test";
    list lsp-ping {
        key "lsp-ping-name";
        description "LSP Ping test";
        leaf lsp-ping-name {
            type string {
                length "1..31";
            }
            mandatory "true";
            description "LSP Ping test name";
        }
        container control-parameters {
            description "Control information of the LSP Ping test";
            leaf target-fec-type {
                type target-fec-type;
                description "Specifies the address type of the Target FEC";
            }
        }
    }
}
```

```
choice target-fec {
  case ip-prefix {
    leaf ip-address {
      type inet:ip-address;
      description "IPv4/IPv6 Prefix";
    }
  }
  case bgp {
    leaf bgp {
      type inet:ip-address;
      description "BGP IPv4/IPv6 Prefix";
    }
  }
  case rsvp {
    leaf tunnel-interface {
      type string;
      description "Tunnel interface";
    }
  }
  case vpn {
    leaf vrf-name {
      type uint32;
      description "Layer3 VPN Name";
    }
    leaf vpn-ip-address {
      type inet:ip-address;
      description "Layer3 VPN IPv4 Prefix";
    }
  }
  case pw {
    leaf vcid {
      type uint32;
      description "VC ID";
    }
  }
  case vpls {
    leaf vsi-name {
      type string;
      description "VPLS VSI";
    }
  }
  description "Specifies the type of the Target FEC";
}
leaf traffic-class {
  type uint8;
  description "Specifies the Traffic Class";
}
leaf reply-mode {
```

```
    type reply-mode;
    description "Specifies the Reply Mode";
  }
  leaf timeout {
    type uint32;
    description
      "Specifies the time-out value for a LSP Ping operation.";
  }
  leaf timeout-units {
    type units;
    description "Time-out units";
  }
  leaf interval {
    type uint32;
    default 1;
    description
      "Specifies the interval between transmissions
       of LSP Ping echo request packets (probes)
       as part of the LSP Ping test.";
  }
  leaf interval-units {
    type units;
    default seconds;
    description "Interval units";
  }
  leaf probe-count {
    type uint32;
    default 5;
    description
      "Specifies the number of probes sent in the LSP Ping test.";
  }
  leaf data-size {
    type uint32;
    description
      "Specifies the size of the data portion to
       be transmitted in an LSP Ping operation, in octets.";
  }
  leaf data-fill {
    type string{
      length "0..1564";
    }
    description
      "Used together with the corresponding
       data-size value to determine how to fill the data
       portion of a probe packet.";
  }
  leaf description {
    type string{
```

```
        length "1..31";
    }
    description "A descriptive name of the LSP Ping test";
}
leaf source-address {
    type inet:ip-address;
    description "Specifies the source address";
}
leaf ttl {
    type uint8;
    default 255;
    description "Time to live";
}
choice outbound {
    case interface {
        leaf interface-name{
            type string{
                length "1..255";
            }
            description "Specifies the outgoing interface";
        }
    }
    case nexthop{
        leaf nexthop {
            type inet:ip-address;
            description "Specifies the nexthop";
        }
    }
    description "Specifies the out interface or nexthop";
}
}

container scheduling-parameters {
    description "LSP Ping test schedule parameter";
    choice start-test{
        case now {
            leaf start-test-now {
                type empty;
                description "Start test now";
            }
        }
        case at {
            leaf start-test-at {
                type yang:date-and-time;
                description "Start test at a specific time";
            }
        }
        case delay {
```



```
    leaf start-test-delay {
        type uint32;
        description "Start after a specific delay";
    }
    leaf start-test-delay-units {
        type units;
        default seconds;
        description "Delay units";
    }
}
case daily {
    leaf start-test-daily {
        type yang:date-and-time;
        description "Start test daily";
    }
}
description
    "Specifies when the test begins to start,
    include 4 schedule method: start now(1), start at(2),
    start delay(3), start daily(4).";
}

choice end-test{
    case at {
        leaf end-test-at{
            type yang:date-and-time;
            description "End test at a specific time";
        }
    }
    case delay {
        leaf end-test-delay {
            type uint32;
            description "End after a specific delay";
        }
        leaf end-test-delay-units {
            type units;
            default seconds;
            description "Delay units";
        }
    }
}
case lifetime {
    leaf end-test-lifetime {
        type uint32;
        description "Set the test lifetime";
    }
    leaf lifetime-units {
        type units;
        default seconds;
    }
}
```

```
        description "Lifetime units";
    }
}
description
    "Specifies when the test ends, include 3
    schedule method: end at(1), end delay(2),
    end lifetime(3).";
}
}

container result-info {
    config "false";
    description "LSP Ping test result information";
    leaf operational-status {
        type operational-status;
        description "Operational state of a LSP Ping test";
    }
    leaf source-address {
        type inet:ip-address;
        description "The source address of the test";
    }
    leaf target-fec-type {
        type target-fec-type;
        description "The Target FEC address type";
    }
    choice target-fec {
        case ip-prefix {
            leaf ip-address {
                type inet:ip-address;
                description "IPv4/IPv6 Prefix";
            }
        }
        case bgp {
            leaf bgp {
                type inet:ip-address;
                description "BGP IPv4/IPv6 Prefix";
            }
        }
        case rsvp {
            leaf tunnel-interface {
                type string;
                description "Tunnel interface";
            }
        }
        case vpn {
            leaf vrf-name {
                type uint32;
                description "Layer3 VPN Name";
            }
        }
    }
}
```

```
    }
    leaf vpn-ip-address {
        type inet:ip-address;
        description "Layer3 VPN IPv4 Prefix";
    }
}
case pw {
    leaf vcid {
        type uint32;
        description "VC ID";
    }
}
case vpls {
    leaf vsi-name {
        type string;
        description "VPLS VSI";
    }
}
description "The Target FEC address";
}
leaf min-rtt {
    type uint32;
    description
        "The minimum LSP Ping round-trip-time (RTT)
        received measured in usec.";
}
leaf max-rtt {
    type uint32;
    description
        "The maximum LSP Ping round-trip-time (RTT)
        received measured in usec.";
}
leaf average-rtt {
    type uint32;
    description
        "The current average LSP Ping round-trip-time
        (RTT) measured in usec.";
}
leaf probe-responses {
    type uint32;
    description
        "Number of responses received for the
        corresponding LSP Ping test.";
}
leaf sent-probes {
    type uint32;
    description
        "Number of probes sent for the
```

```
        corresponding LSP Ping test.";
    }
    leaf sum-of-squares {
        type uint32;
        description
            "The sum of the squares of RTT,
            calculated as the sum of the squared
            differences between each RTT and the overall
            mean RTT, for all replies received.";
    }
    leaf last-good-probe {
        type yang:date-and-time;
        description
            "Date and time when the last response
            was received for a probe.";
    }
}

container probe-results {
    description "Result info of test probes";
    list probe-result {
        key "probe-index";
        description "Result info of each test probe";
        leaf probe-index {
            type uint32;
            config false;
            description "Probe index";
        }
        leaf return-code {
            type uint8;
            config false;
            description "The Return Code set in the echo reply";
        }
        leaf return-sub-code {
            type uint8;
            config false;
            description
                "The Return Sub-code set in the echo reply.";
        }
        leaf rtt {
            type uint32;
            config false;
            description "The round-trip-time (RTT) received";
        }
        leaf result-type {
            type result-type;
            config false;
            description "The probe result type";
        }
    }
}
```

```
    }  
  }  
}  
}  
}  
<CODE ENDS>
```

7. Examples

The following examples show the netconf RPC communication between client and server for one LSP-Ping test case.

7.1. Configuration of Control Information

Configure the control-parameters for sample-test-case.

Request from netconf client:

```
<rpc
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <control-parameters>
            <target-fec-type>ip-prefix</target-fec-type>
            <ip-prefix>2001:db8::1:100/64</ip-prefix>
            <reply-mode>reply-via-udp</reply-mode>
            <timeout>1</timeout>
            <timeout-units>seconds</timeout-units>
            <interval>1</interval>
            <interval-units>seconds</interval-units>
            <probe-count>6</probe-count>
            <admin-status>enabled</admin-status>
            <data-size>64</data-size>
            <data-fill>this is a lsp ping test</data-fill>
            <source-address>2001:db8::4</source-address>
            <ttl>56</ttl>
          </control-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.2. The Configuration of Schedule Parameters

Set the scheduling-parameters for sample-test-case to start the test.

Request from netconf client:

```
<rpc
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <scheduling-parameters>
            <start-test-now/>
          </scheduling-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.3. Display of Result Information

Get the result-info of sample-test-case.

Request from netconf client:

```
<rpc
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <result-info/>
        </lsp-ping>
      </lsp-pings>
    </filter>
  </get>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
<data>
  <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
    <lsp-ping>
      <lsp-ping-name>sample-test-case</lsp-ping-name>
      <result-info>
        <operational-status>completed</operational-status>
        <source-address>2001:db8::4</source-address>
        <target-fec-type>ip-prefix</target-fec-type>
        <ip-prefix>2001:db8::1:100/64</ip-prefix>
        <min-rtt>10</min-rtt>
        <max-rtt>56</max-rtt>
        <average-rtt>36</average-rtt>
        <probe-responses>6</probe-responses>
        <sent-probes>6</sent-probes>
        <sum-of-squares>8882</sum-of-squares>
        <last-good-probe>2015-07-01T10:36:56</last-good-probe>
        <probe-results>
          <probe-result>
            <probe-index>0</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>10</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>1</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>56</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>2</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>35</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>3</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>38</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>4</probe-index>
            <return-code>0</return-code>
          </probe-result>
        </probe-results>
      </result-info>
    </lsp-ping>
  </lsp-pings>
```



```
        <return-sub-code>3</return-sub-code>
        <rtt>36</rtt>
        <result-type>success</result-type>
    </probe-result>
    <probe-result>
        <probe-index>5</probe-index>
        <return-code>0</return-code>
        <return-sub-code>3</return-sub-code>
        <rtt>41</rtt>
        <result-type>success</result-type>
    </probe-result>
</probe-results>
</result-info>
</lsp-ping>
</lsp-pings>
</data>
</rpc-reply>
```

8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have an adverse effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can adversely affect the routing subsystem of both the local device and the network. This may lead to corruption of the measurement that may result in false corrective action, e.g., false negative or false positive. That could be, for example, prolonged and undetected

deterioration of the quality of service or actions to improve the quality unwarranted by the real network conditions.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can disclose the operational state information of VRRP on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

TBD

The LSP ping YANG module inherits all security consideration of [RFC8029].

9. IANA Considerations

The IANA is requested to assign a new namespace URI from the IETF XML registry.

URI:TBA

Contributors

Yanfeng Zhang

Huawei Technologies

zhangyanfeng@huawei.com

Sam Aldrin

Google

aldrin.ietf@gmail.com

Acknowledgments

TBD

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informative References

- [I-D.mahesh-netconf-persistent] Jethanandani, M., "NETCONF and persistent responses", draft-mahesh-netconf-persistent-00 (work in progress), October 2014.
- [RFC4560] Quittek, J., Ed. and K. White, Ed., "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations", RFC 4560, DOI 10.17487/RFC4560, June 2006, <<https://www.rfc-editor.org/info/rfc4560>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

Authors' Addresses

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Guangying Zheng
Huawei Technologies
China

Email: zhengguangying@huawei.com

Greg Mirsky
ZTE Corp.
USA

Email: gregimirsky@gmail.com

Reshad Rahman
Cisco Systems
Canada

Email: rrahman@cisco.com

Faisal Iqbal
Cisco Systems

Email: faiqbal@cisco.com