MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2021

K. Raza
Cisco Systems

X. Liu
Volta Networks

S. Esale
Juniper Networks

L. Andersson
Huawei Technologies

J. Tantsura
Nuage Networks

S. Krishnaswamy
Individual

March 7, 2021

YANG Data Model for MPLS mLDP
draft-ietf-mpls-mldp-yang-08

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 September 8, 2021.

Copyright Notice

Table of Contents

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:

- Configuration
- Operational State
- Executables (Actions)
- 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:

- Base
- 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-mldp and ietf-mpls-mldp-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-mldp" for the mLDP YANG base data model that augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ldp:mpls-ldp defined in

[I-D.ietf-mpls-ldp-yang]. The document also defines the "ietf-mpls-mldp-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
                   |       |   +-- rw ...
                   |       |   +-- 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-mldp-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)

- 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:

- Parameters that are mLDP specific

- 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:

- 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.

- 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:

- 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.

- 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.

- 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-mldp
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
ldp:mpls-ldp/ldp:global/ldp:capability:
    +--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/rt:control-plane-protocol/
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-mldp
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
ldp:mpls-ldp/ldp:global/ldp:capability:
    +--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/rt:control-plane-protocol/
```

```
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-
zone
          |  |  |     +--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
recur-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-
zone
          |  |        +--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-
distinguisher
             |  |        +--rw mldp-ext:multipoint-type?      mldp:multipoint-
type
             |  +--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-
zone
             |  |     +--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
recur-root-address recur-rd]
             |        +--rw mldp-ext:root-address          inet:ipv6-address
```

```
            |       +--rw mldp-ext:rp                    inet:ip-address
            |       +--rw mldp-ext:group-address         inet:ip-address-no-
zone
            |       +--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/rt:control-plane-protocol/
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 mdlp "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-mldp
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
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/rt:control-plane-protocol/
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/
peer/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-
upstream
                               +--ro peer                   -> /rt:routing/
control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                               +--ro advertisement-type     ldp:advertised-
received
                               +--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-mldp

  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
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/rt:control-plane-protocol/
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
recur-rd]
          |           |        +--ro mldp-ext:recur-root-address    inet:ip-
address
          |           |        +--ro mldp-ext:recur-rd              route-
distinguisher
          |           |        +--ro mldp-ext:multipoint-type?
mldp:multipoint-type
          |           |        +--ro mldp-ext:peer* [direction peer
advertisement-type]
          |           |           +--ro mldp-ext:direction
ldp:downstream-upstream
          |           |           +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
          |           |           +--ro mldp-ext:advertisement-type
ldp:advertised-received
          |           |           +--ro mldp-ext:label?               rt-
types:mpls-label
          |           |           +--ro mldp-ext:mbb-role?
enumeration
          |           |           +--ro mldp-ext:mofrr-role?          mofrr-
role
          |           +--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-
```

```
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
          |           |     +--ro mldp-ext:peer* [direction peer advertisement-
type]
          |           |        +--ro mldp-ext:direction
ldp:downstream-upstream
          |           |        +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
          |           |        +--ro mldp-ext:advertisement-type
ldp:advertised-received
          |           |        +--ro mldp-ext:label?               rt-
types:mpls-label
          |           |        +--ro mldp-ext:mbb-role?            enumeration
          |           |        +--ro mldp-ext:mofrr-role?          mofrr-role
          |           +--ro mldp-ext:opaque-element-bidir
          |              +--ro mldp-ext:fec-label* [rp group-address rd recur-
root-address recur-rd]
          |                 +--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
          |                +--ro mldp-ext:peer* [direction peer advertisement-
type]
          |                   +--ro mldp-ext:direction
ldp:downstream-upstream
          |                   +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
          |                   +--ro mldp-ext:advertisement-type
ldp:advertised-received
          |                   +--ro mldp-ext:label?                rt-
types:mpls-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-
type
                      |     +--ro mldp-ext:peer* [direction peer advertisement-
type]
                      |     |  +--ro mldp-ext:direction
ldp:downstream-upstream
                      |     |  +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                      |     |  +--ro mldp-ext:advertisement-type
ldp:advertised-received
                      |     |  +--ro mldp-ext:label?                rt-
types:mpls-label
                      |     |  +--ro mldp-ext:mbb-role?             enumeration
                      |     |  +--ro mldp-ext:mofrr-role?           mofrr-role
                      |     +--ro mldp-ext:recursive-fec* [recur-root-address
recur-rd]
```

```
                      |       +--ro mldp-ext:recur-root-address    inet:ip-
address
                      |       +--ro mldp-ext:recur-rd              route-
distinguisher
                      |       +--ro mldp-ext:multipoint-type?
mldp:multipoint-type
                      |       +--ro mldp-ext:peer* [direction peer
advertisement-type]
                      |          +--ro mldp-ext:direction
ldp:downstream-upstream
                      |          +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                      |          +--ro mldp-ext:advertisement-type
ldp:advertised-received
                      |          +--ro mldp-ext:label?                rt-
types:mpls-label
                      |          +--ro mldp-ext:mbb-role?
enumeration
                      |          +--ro mldp-ext:mofrr-role?           mofrr-
role
                      +--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-
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
                      |     +--ro mldp-ext:peer* [direction peer advertisement-
type]
                      |        +--ro mldp-ext:direction
ldp:downstream-upstream
                      |        +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                      |        +--ro mldp-ext:advertisement-type
ldp:advertised-received
                      |        +--ro mldp-ext:label?                rt-
types:mpls-label
                      |        +--ro mldp-ext:mbb-role?             enumeration
                      |        +--ro mldp-ext:mofrr-role?           mofrr-role
                      +--ro mldp-ext:opaque-element-bidir
                         +--ro mldp-ext:fec-label* [rp group-address rd recur-
root-address recur-rd]
                            +--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
                            +--ro mldp-ext:peer* [direction peer advertisement-
type]
                               +--ro mldp-ext:direction
ldp:downstream-upstream
                               +--ro mldp-ext:peer                  -> /
rt:routing/control-plane-protocols/ldp:mpls-ldp/peers/peer/lsr-id
                               +--ro mldp-ext:advertisement-type
ldp:advertised-received
                               +--ro mldp-ext:label?                rt-
types:mpls-label
                               +--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:

- Root
- Bindings (FEC-label)
- 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-familty 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-
upstream
                        |     |  +--ro peer                   leafref
                        |     |  +--ro advertisement-type     ldp:advertised-
received
                        |     |  +--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 in 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 avaiable 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:

- 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.

- 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].

- 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:

- global: augments ldp:global/ldp:state/ldp:capability.
- 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 includind 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@2021-03-07.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 2021-03-07 {
    // RFC Editor: replace the above date 2021-03-07 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/rt:control-plane-protocol/"
        + "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/rt:control-plane-protocol/"
    + "ldp:mpls-ldp/ldp:global/ldp:capability" {
    description "Augmentation for MLDP global capability";
    container mldp {
      description
        "This container contains the configruation and state
         information for multipoint LDP capabilities.";
      uses mldp-capabilities;
    }
  }

  /*
   * Operational state data nodes
   */
```

```
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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/rt:control-plane-protocol/"
```

```
    + "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-mldp-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@2021-03-07.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 2021-03-07 {
    // RFC Editor: replace the above date 2021-03-07 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 Multipoimt 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-attibutes {
    description
      "mLDP per address family configuration attibutes";
    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-attibutes

  grouping recursive-fec-attibutes {
    description
      "mLDP recursive FEC attibutes.";
    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-attibutes

  /*
```

```
   * Configuration data and operational state data nodes
   */
  // Global capability
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ldp:mpls-ldp/ldp:global/ldp:capability/mldp:mldp" {
    description "Augmentation for MLDP global capability.";

    uses mldp-ext-capabilities;
  }

  /* TODO: FIXME
  // Peer capability
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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/rt:control-plane-protocol/"
    + "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
    + "mldp:ipv4" {
    description "Augmentation for MLDP IPv4 configuration";
    uses mldp-ext-per-af-config-attibutes;
  }

  // IPv4 configured-leaf-lsps config
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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-attibutes;
    } // fec-label
  }
```

```
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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-attibutes;
      } // 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-attibutes;
      } // fec-label
    } // opaque-element-bidir
  }

  // IPv6 config
```

```
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "ldp:mpls-ldp/ldp:global/mldp:mldp/mldp:address-families/"
    + "ipv6" {
    description "Augmentation for MLDP IPv4 configuration";
    uses mldp-ext-per-af-config-attibutes;
  }

  // Global forwarding-nexthop
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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/rt:control-plane-protocol/"
    + "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/rt:control-plane-protocol/"
    + "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/rt:control-plane-protocol/"
    + "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-attibutes;
        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-attibutes;
        uses mldp-ext-binding-label-state-attributes;
      } // fec-label
    } // opaque-element-bidir
  }

  // IPv6 bindings state
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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-attibutes;
        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-attibutes;
        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/rt:control-plane-protocol/"
    + "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-attibutes;
      uses mldp-ext-binding-label-state-attributes;
    } // fec-label
  }

  // IPv6 bindings opaque-element-lspid state
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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-attibutes;
      uses mldp-ext-binding-label-state-attributes;
    } // fec-label
  }

  /*
   * Per AF augmentation
   */
  // IPv6 augmentation
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/"
    + "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-attibutes;
        } // 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-attibutes;
          } // 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-attibutes;
    } // 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-attibutes;
      } // opaque-element-transit
    } // 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-attibutes;
      } // 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-mldp	The IESG	N/A
urn:ietf:params:xml:ns:yang:ietf-mpls-mldp-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-mldp	urn:ietf:params:xml:ns:yang:ietf-mpls-mldp	mldp	This document
ietf-mpls-mldp-extended	urn:ietf:params:xml:ns:yang:ietf-mpls-mldp-extended	mldp-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-09 (work in progress), March 2020.

[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>.

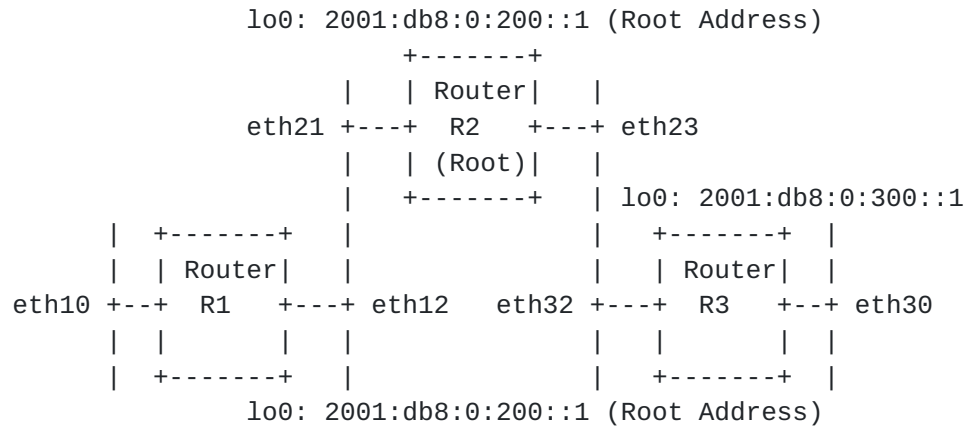
[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

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-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"
                  }
                ]
              }
            },
            "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, Inc.
Email: skraza@cisco.com

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

Santosh Esale
Juniper Networks
Email: santosh_easale@berkeley.edu

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