

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 August 2026

K. Raza, Ed.
Cisco Systems

X. Liu
Individual

S. Easale
Juniper Networks

L. Andersson
Huawei Technologies

J. Tantsura
Microsoft Corporation

3 February 2026

YANG Data Model for MPLS mLDP
draft-ietf-mpls-mldp-yang-16

Abstract

This document describes a YANG data model for the Multiprotocol Label Switching (MPLS) Multipoint Label Distribution Protocol (mLDP). The mLDP YANG data model augments the MPLS LDP YANG 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 7 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Base and Extended	3
2. Specification of Requirements	4
3. Overview	4
3.1. Multipoint LSP FECs	6
3.2. YANG Model	8
4. The Complete Tree	8
5. Configuration	15
5.1. High-level Hierarchy	15
5.2. Base Parameters	16
5.3. Capabilities Parameters	17
5.4. Forwarding Parameters	17
6. Operational State	17
6.1. Root State	18
6.2. Bindings State	19
6.2.1. Recursive FEC	22
6.3. Capabilities State	22
7. Notifications	24
8. Actions	24
9. YANG Specification	24
9.1. Base	24
9.2. Extended	37
10. Security Considerations	56
10.1. YANG Data Model	56
10.1.1. Writable Nodes	57
10.1.2. Readable Nodes	57
10.1.3. Notifications	57
11. IANA Considerations	58
12. References	58
12.1. Normative References	58
12.2. Informative References	61
Appendix A. Data Tree Example	61
Appendix B. Acknowledgments	69
Appendix C. Contributors	69
Authors' Addresses	70

1. Introduction

This document introduces a YANG data model for the MPLS Multipoint Label Distribution Protocol (mLDP). The mLDP YANG data model being defined here is dependent on the LDP YANG data model [RFC9070]. This implies that an operator will need to use the LDP base YANG data model to configure and manage the control plane for mLDP. For example, an operator would enable LDP discovery on MPLS interface to establish LDP session for mLDP peering on which mLDP bindings could be exchanged. Similarly, an operator could query state information for an LDP peer to verify peering attributes, etc.

Moreover, it is important to note here that any assumptions made in the LDP YANG data 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

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] (and some important extensions like the targeted mLDP [RFC7060]) and constitute the minimum requirements for a typical base mLDP deployment, whereas the "extended" category contains all other non-base features. All the items in the base category are mandatory and, hence, no "if-feature" is allowed under the "base" category. The base and extended categories are defined in their own modules as described later.

The examples of base mLDP features include the enablement of mLDP and its capabilities, and static configuration of leaf IPv4 LSPs with generic LSP Id, whereas the examples of extended mLDP feature include enhanced mLDP capabilities, Recursive FEC [RFC6512], traffic protection [RFC7715] [RFC7431], static leaf IPv6 LSPs with generic LSP Id, and static leaf IPv4/IPv6 LSPs of other opaque types (transit, bidir). It is worth highlighting that any IPv6 related feature support is categorized as an extended feature.

While "base" model support will suffice for small deployments, it is expected that large deployments will require both the "base" and "extended" model support from the vendors.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

2. Specification of Requirements

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

3. Overview

This document defines two new modules for mLDP YANG support:

- * "ietf-mpls-mlbp" module that specifies the base mLDP features
- * "ietf-mpls-mlbp-extended" module that specifies the extended mLDP features

Both the modules augment the LDP module (/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ldp:mpls-ldp) as defined in [RFC9070]

There are four types of containers in our module(s):

- * Read-write parameters for configuration (Section 5)
- * Read-only parameters for operational state (Section 6)
- * Notifications for events (Section 7)

Currently, no RPCs for executing commands to perform some actions are defined in our mLDP modules.

In mLDP YANG modules, 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.

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

```

+-- rw routing
  +-- rw control-plane-protocols
    +-- rw control-plane-protocol
      +-- rw mpls-ldp
        +-- rw some_ldp_container
          |   +-- rw mldp
          |   |   +-- rw ...           // mldp base
          |   |   |   +-- rw ...
          |   |   |   +-- ro ...
          |   |   |   +--
          |   |   +-- rw mldp-ext:... // mldp extended
          |   |   |   +-- rw ...
          |   |   |   +-- ro ...
          |   |   |   +--
          +-- ro someother_ldp_container
            +-- ro mldp
              +-- ro ...           // mldp base
              |   +-- ro ...
              |   +--
              +-- ro mldp-ext:... // mldp extended
              |   +-- ro ...
              |   +--
notifications:
  +--- n mpls-mldp-some_event
  +--- n ...

```

Figure 1: mLDP YANG Tree Organization

Before going into data model details, following important points need to be noted:

- * This model aims to address the core mLDP constructs, parameters, and features as per RFC specification, as well as well-known and widely deployed manageability controls (such as timers, delays etc.). Any vendor-specific feature should be defined in a vendor-specific augmentation of this model.

- * This model does not cover any applications or clients running on top of mLDP, nor does it cover any Operations, Administration, and Maintenance (OAM) procedures for mLDP.
- * This mLDP model builds on top of LDP YANG model [RFC9070] which is a VRF-centric model. A "network-instance", as defined in [RFC8529], refers to a VRF instance (both default and non-default) within the scope of this model.
- * This model supports two address families, namely, "ipv4" and "ipv6".

The mLDP areas and features that are within the scope of this modeling effort are as follows:

- * Base:
 - mLDP Base Specification [RFC6388]
 - Targeted mLDP [RFC7060]
 - Static LSPs at Leaf node (manually provisioned)
- * Extended:
 - mLDP Recursive FEC [RFC6512]
 - mLDP Fast-Reroute (FRR):
 - o Node Protection [RFC7715]
 - o Multicast-only [RFC7431]
 - Hub-and-Spoke Multipoint LSPs [RFC7140]
 - In-band Signaling:
 - o mLDP In-band Signaling [RFC6826]
 - o mLDP In-band signaling in a VRF [RFC7246]
 - o mLDP In-band Signaling with Wildcards [RFC7438]

3.1. Multipoint LSP FECs

A Multipoint LSP is associated with a FEC which is represented as (root-address, opaque-element) tuple.

We can categorize the opaque types into following four base categories:

- * Generic LSP Identifier
- * Transit Source
- * Transit Bidir
- * Recursive

Furthermore, there are VPN variants that can be applied on top of the base categories.

The following table lists various type of opaque elements with their keys, as later used in the configuration and state model:

Opaque Type	Variant	Opaque specific Key	RFCs
Generic LSP Identifier	Base	LSP Id	[RFC6388]
Transit Source	Base (IPv4/IPv6)	Source, Group	[RFC6826]
	VPN	Source, Group, RD	[RFC7246]
Transit Bidir	Base (IPv4/IPv6)	Masklen, RP, Group	[RFC6826]
	VPN	Masklen, RP, Group, RD	[RFC7246]
Recursive	Base	Opaque Value Element Octets	[RFC6512]
	VPN	RD, Opaque Value Element Octets	[RFC6512]

Table 1: Opaque Types and Keys

It is to be noted that key for an MP LSP is a combination of the FEC's "root address" and FEC opaque value element specific keys (as listed above)

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

3.2. YANG Model

A simplified graphical tree representation of base and extended mLDP YANG data models is presented in Figure 2. The meaning of the symbols in these tree diagrams are defined in [RFC8340].

The actual YANG specification for base and extended modules is captured in Section 9.

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

4. The Complete Tree

The following is a complete tree representation of configuration, state, and notification items under mLDP base and extended modules. Since mLDP modules augment LDP module, the mLDP tree view is shown within the LDP tree. In this view, we only show mLDP specific leaves and containers along with their hierarchy; we do not show any leaves or containers that are LDP specific.

```
module: ietf-mpls-mlldp
```

```
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ldp:mpls-ldp/ldp:global
  /ldp:capability:
    +--rw mldp
      +--rw p2mp
        | +--rw enabled?    boolean
      +--rw mp2mp
        | +--rw enabled?    boolean
      +--rw make-before-break
        | +--rw enabled?    boolean
        | +--rw switchover-delay? uint16
        | +--rw timeout?    uint16
      +--rw mldp-ext:hub-and-spoke {capability-mldp-hsmp}?
        | +--rw mldp-ext:enabled? boolean
      +--rw mldp-ext:node-protection {capability-mldp-node-protection}?
        +--rw mldp-ext:plr?          boolean
        +--rw mldp-ext:merge-point
          +--rw mldp-ext:enabled?    boolean
          +--rw mldp-ext:targeted-session-teardown-delay? uint16
```



```

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 is-capable?    boolean
+--ro mp2mp
|  +--ro is-capable?    boolean
+--ro make-before-break
|  +--ro is-capable?    boolean
+--ro mldp-ext:hub-and-spoke
|  +--ro mldp-ext:is-capable?    boolean
+--ro mldp-ext:node-protection
  +--ro mldp-ext:is-plr-capable?    boolean
  +--ro mldp-ext:is-merge-point-capable?    boolean

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ldp:mpls-ldp/ldp:global:
+--rw mldp
+--rw enabled?                boolean
+--rw address-families
  +--rw ipv4!
  |  +--rw enabled?                boolean
  |  +--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?    leafref
  |  |  +--ro bindings
  |  |  |  +--ro opaque-value-element-lspid
  |  |  |  |  +--ro fec-label* [lsp-id]
  |  |  |  |  |  +--ro lsp-id                uint32
  |  |  |  |  |  +--ro multipoint-type?
  |  |  |  |  |  |  multipoint-type
  |  |  |  |  |  +--ro opaque-value-element-type?
  |  |  |  |  |  |  identityref
  |  |  |  +--ro peer*
  |  |  |  |  [direction peer advertisement-type]
  |  |  |  |  +--ro peer                leafref
  |  |  |  |  +--ro direction
  |  |  |  |  |  ldp:downstream-upstream
  |  |  |  |  +--ro advertisement-type
  |  |  |  |  |  ldp:advertised-received
  |  |  |  +--ro label?
  |  |  |  |  rt-types:mpls-label

```

```

        +--ro mbb-status?
        |   lsp-mbb-role
        +--ro mldp-ext:mofrr-status?
           lsp-mofrr-role
+--ro mldp-ext:opaque-value-element-transit-source
  +--ro mldp-ext:fec-label*
     [source-address group-address rd]
  +--ro mldp-ext:source-address
     |   inet:ipv4-address
  +--ro mldp-ext:group-address
     |   inet:ipv4-address
  +--ro mldp-ext:rd
     |   rt-types:route-distinguisher
  +--ro mldp-ext:multipoint-type?
     |   mldp:multipoint-type
  +--ro mldp-ext:opaque-value-element-type?
     |   identityref
  +--ro mldp-ext:peer*
     [direction peer advertisement-type]
    +--ro mldp-ext:peer
       |   leafref
    +--ro mldp-ext:direction
       |   ldp:downstream-upstream
    +--ro mldp-ext:advertisement-type
       |   ldp:advertised-received
    +--ro mldp-ext:label?
       |   rt-types:mpls-label
    +--ro mldp-ext:mbb-status?
       |   lsp-mbb-role
    +--ro mldp-ext:mofrr-status?
       |   lsp-mofrr-role
+--ro mldp-ext:opaque-value-element-transit-bidir
  +--ro mldp-ext:fec-label*
     [masklen rp group-address rd]
  +--ro mldp-ext:masklen
     |   uint8
  +--ro mldp-ext:rp
     |   inet:ipv4-address
  +--ro mldp-ext:group-address
     |   inet:ipv4-address
  +--ro mldp-ext:rd
     |   rt-types:route-distinguisher
  +--ro mldp-ext:multipoint-type?
     |   mldp:multipoint-type
  +--ro mldp-ext:opaque-value-element-type?
     |   identityref
  +--ro mldp-ext:peer*
     [direction peer advertisement-type]

```

```

|
|
|
|      +--ro mldp-ext:peer
|      |      leafref
|      +--ro mldp-ext:direction
|      |      ldp:downstream-upstream
|      +--ro mldp-ext:advertisement-type
|      |      ldp:advertised-received
|      +--ro mldp-ext:label?
|      |      rt-types:mpls-label
|      +--ro mldp-ext:mbb-status?
|      |      lsp-mbb-role
|      +--ro mldp-ext:mofrr-status?
|      |      lsp-mofrr-role
|      +--ro mldp-ext:opaque-value-element-recursive
|      |      +--ro mldp-ext:fec-label* [rd opaque-value]
|      |      +--ro mldp-ext:rd
|      |      |      rt-types:route-distinguisher
|      |      +--ro mldp-ext:opaque-value
|      |      |      binary
|      |      +--ro mldp-ext:multipoint-type?
|      |      |      mldp:multipoint-type
|      |      +--ro mldp-ext:opaque-value-element-type?
|      |      |      identityref
|      |      +--ro mldp-ext:peer*
|      |      |      [direction peer advertisement-type]
|      |      +--ro mldp-ext:peer
|      |      |      leafref
|      |      +--ro mldp-ext:direction
|      |      |      ldp:downstream-upstream
|      |      +--ro mldp-ext:advertisement-type
|      |      |      ldp:advertised-received
|      |      +--ro mldp-ext:label?
|      |      |      rt-types:mpls-label
|      |      +--ro mldp-ext:mbb-status?
|      |      |      lsp-mbb-role
|      |      +--ro mldp-ext:mofrr-status?
|      |      |      lsp-mofrr-role
|      +--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:enabled?                          boolean
+--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?      leafref
+--ro mldp-ext:bindings
|   +--ro mldp-ext:opaque-value-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:opaque-value-element-type?
|   |   |   |   identityref
|   |   +--ro mldp-ext:peer*
|   |   |   [direction peer advertisement-type]
|   |   |   +--ro mldp-ext:peer
|   |   |   |   leafref
|   |   |   +--ro mldp-ext:direction
|   |   |   |   ldp:downstream-upstream
|   |   |   +--ro mldp-ext:advertisement-type
|   |   |   |   ldp:advertised-received
|   |   |   +--ro mldp-ext:label?
|   |   |   |   rt-types:mpls-label
|   |   |   +--ro mldp-ext:mbb-status?
|   |   |   |   lsp-mbb-role
|   |   |   +--ro mldp-ext:mofrr-status?
|   |   |   |   lsp-mofrr-role
|   +--ro mldp-ext:opaque-value-element-transit-source
|   |   +--ro mldp-ext:fec-label*
|   |   |   [source-address group-address rd]
|   |   |   +--ro mldp-ext:source-address
|   |   |   |   inet:ipv6-address
|   |   |   +--ro mldp-ext:group-address
|   |   |   |   inet:ipv6-address
|   |   |   +--ro mldp-ext:rd
|   |   |   |   rt-types:route-distinguisher
|   |   |   +--ro mldp-ext:multipoint-type?
|   |   |   |   mldp:multipoint-type
|   |   |   +--ro mldp-ext:opaque-value-element-type?
|   |   |   |   identityref
|   |   +--ro mldp-ext:peer*
|   |   |   [direction peer advertisement-type]
|   |   |   +--ro mldp-ext:peer
|   |   |   |   leafref
|   |   |   +--ro mldp-ext:direction
|   |   |   |   ldp:downstream-upstream
|   |   |   +--ro mldp-ext:advertisement-type
|   |   |   |   ldp:advertised-received
|   |   |   +--ro mldp-ext:label?
|   |   |   |   rt-types:mpls-label

```

```

        +--ro mldp-ext:mbb-status?
        |   lsp-mbb-role
        +--ro mldp-ext:mofrr-status?
        |   lsp-mofrr-role
+--ro mldp-ext:opaque-value-element-transit-bidir
+--ro mldp-ext:fec-label*
    [masklen rp group-address rd]
+--ro mldp-ext:masklen
    |   uint8
+--ro mldp-ext:rp
    |   inet:ipv6-address
+--ro mldp-ext:group-address
    |   inet:ipv6-address
+--ro mldp-ext:rd
    |   rt-types:route-distinguisher
+--ro mldp-ext:multipoint-type?
    |   mldp:multipoint-type
+--ro mldp-ext:opaque-value-element-type?
    |   identityref
+--ro mldp-ext:peer*
    [direction peer advertisement-type]
+--ro mldp-ext:peer
    |   leafref
+--ro mldp-ext:direction
    |   ldp:downstream-upstream
+--ro mldp-ext:advertisement-type
    |   ldp:advertised-received
+--ro mldp-ext:label?
    |   rt-types:mpls-label
+--ro mldp-ext:mbb-status?
    |   lsp-mbb-role
+--ro mldp-ext:mofrr-status?
    |   lsp-mofrr-role
+--ro mldp-ext:opaque-value-element-recursive
+--ro mldp-ext:fec-label* [rd opaque-value]
+--ro mldp-ext:rd
    |   rt-types:route-distinguisher
+--ro mldp-ext:opaque-value
    |   binary
+--ro mldp-ext:multipoint-type?
    |   mldp:multipoint-type
+--ro mldp-ext:opaque-value-element-type?
    |   identityref
+--ro mldp-ext:peer*
    [direction peer advertisement-type]
+--ro mldp-ext:peer
    |   leafref
+--ro mldp-ext:direction

```

```

|
|
|          ldp:downstream-upstream
+--ro mldp-ext:advertisement-type
|          ldp:advertised-received
+--ro mldp-ext:label?
|          rt-types:mpls-label
+--ro mldp-ext:mbb-status?
|          lsp-mbb-role
+--ro mldp-ext:mofrr-status?
|          lsp-mofrr-role
+--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

```

```
module: ietf-mpls-mldp-extended
```

```

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

```

```
module: ietf-mpls-mldp
```

```
notifications:
```

```

+---n mpls-mldp-fec-event
+--ro event-type?
|   ldp:oper-status-event-type
+--ro (opaque-value-element)?
+--:(opaque-value-element-lspid)
|   +--ro opaque-value-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
+--:(mldp-ext:opaque-value-element-transit-source)
|   +--ro mldp-ext:opaque-value-element-transit-source
|   |   +--ro mldp-ext:root-address?          inet:ip-address
|   |   +--ro mldp-ext:source-address?        inet:ip-address
|   |   +--ro mldp-ext:group-address?         inet:ip-address
|   |   +--ro mldp-ext:rd?
|   |   |   rt-types:route-distinguisher
|   |   +--ro mldp-ext:recur-root-address?    inet:ip-address
|   |   +--ro mldp-ext:recur-rd?
|   |   |   rt-types:route-distinguisher
+--:(mldp-ext:opaque-value-element-transit-bidir)
|   +--ro mldp-ext:opaque-value-element-transit-bidir
|   |   +--ro mldp-ext:root-address?          inet:ip-address
|   |   +--ro mldp-ext:masklen?              uint8

```

```

+--ro mldp-ext:rp?                inet:ip-address
+--ro mldp-ext:group-address?     inet:ip-address
+--ro mldp-ext:rd?
|      rt-types:route-distinguisher
+--ro mldp-ext:recur-root-address? inet:ip-address
+--ro mldp-ext:recur-rd?
      rt-types:route-distinguisher

```

Figure 2: Complete Tree

5. Configuration

This specification defines the parameters for configuring the base and extended mLDP features. As stated earlier, mLDP configuration items augment relevant LDP configuration hierarchy.

5.1. High-level Hierarchy

The following presents a high-level view of configuration hierarchy for mLDP with respect to LDP:

```

augment /rt:routing/rt:control-plane-protocols/
  rt:control-plane-protocol:
    +-- mpls-ldp
      +-- global
        +-- ...
        +-- mldp
          |
          | +-- ...
          | +-- address-families
          |   +-- ipv4 (similar tree for ipv6)
          |     +-- roots
          |       +-- root [addr]
          |         +-- bindings (state)
          |           +-- ...
          |       ...
          |       +-- ...
          +-- capability
            |
            | +-- mldp
            |   +-- ...
            |   +-- mldp-ext: ...
            |   +-- ...
            +-- forwarding-nexthop
              +-- interfaces
                +-- interface* [name]
                  +-- mldp-ext: ...

```

Figure 3: Configuration Hierarchy

The above hierarchy illustrates that mLDP configuration parameters are distributed amongst the following three sections:

- * mldp base
- * mldp capabilities
- * mldp forwarding

The following subsections describe the above mLDP sub-tree along with their configuration items.

5.2. Base Parameters

mldp container falls directly under mpls-ldp:global 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. It is to remind that the LDP modules reside under a network-instance and the scope of any configuration defined under this tree is network-instance (VRF) scoped. Given that mLDP requires LDP signaling, it is not sensible to allow disabling the LDP control plane under a 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. Alternatively, the same can be achieved by advertising the capabilities and the procedures defined in [RFC7473].
- * Per address-family mLDP 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.

5.3. Capabilities Parameters

ml dp capabilities are enabled under mldp specific container under ldp:global:capability container. 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.

It is worth reminding that mLDP requires LDP discovery and peer procedures to form mLDP peering but a peer is considered as an mLDP peer only when either p2mp or mp2mp capabilities have been successfully exchanged with the peer.

5.4. Forwarding Parameters

The container ldp:global:forwarding-nexthop is used to hold configuration to controlling LDP/mLDP forwarding behavior. One example of a such a configuration is to allow a user in disabling the use of LDP interface(s) as an mLDP forwarding egress interface for MP LSPs(s). This example configuration makes sense only when there are more than one interface available for the selection.

6. Operational State

The mLDP modules conform to the NMDA where the operational state data is combined with the associated configuration data in the same hierarchy. This means that 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.

The following are the main areas for which mLDP operational state is defined:

- * Root
- * Bindings (FEC-label)
- * Capabilities

6.1. Root State

The root address is a fundamental construct for MP FEC bindings and LSPs. The root state provides information on all the known roots in a given address-family and their root reachability information (as learnt from RIB). In case of multi-path reachability to a root, the selection of the upstream path is done on per-LSP basis at the time of LSP setup. Similarly, when protection mechanisms like Make-before-break (MBB) or Multicast-only FRR (MoFRR) are in place, the path designation as active/standby or primary/backup is also done on per-LSP basis. It should be noted that a given root can be shared amongst multiple P2MP and/or MP2MP LSPs. Moreover, an LSP can be signaled to more than one root for Root Node Redundancy (RNR) purposes.

The following example 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 2001:db8:0:9::1:
  path1:
    RIB: GigEthernet 1/0, 2001:db8:0:1::1
    LDP: peer 192.0.2.1:0

root . . . .
```

Figure 4: Example Root database

A root entry on a root LSR itself will be presented as follows:

```

root 203.0.113.10:
  is-self

root 2001:db8:0:9::1:
  is-self

```

Figure 5: Example Root entry on a Root node

The following diagram captures the high-level tree hierarchy for state of an mLDP root. The tree is shown for ipv4 roots only; a similar tree exists for ipv6 roots as well.

```

+--rw mpls-ldp
  +--rw global
    +--rw mldp:mldp
      +--rw mldp:address-families
        +--rw mldp:ipv4!
          +--ro mldp:roots
            +--ro mldp:root* [root-address]
              +--ro mldp:root-address inet:ipv4-address
              +--ro mldp:is-self?      boolean
              +--ro mldp:reachability* [address interface]
                +--ro mldp:address inet:ipv4-address
                +--ro mldp:interface if:interface-ref
                +--ro mldp:peer?      leafref

```

Figure 6: Root state tree

6.2. Bindings State

Binding state provides information on mLDP FEC-label bindings for both the P2MP and MP2MP FEC types. The state includes both the inbound (received) and the outbound (advertised) bindings. Like LDP, the state of FEC-label binding is presented per address-family in a FEC-centric view per address-family. An mLDP FEC is presented as (root-address, opaque-value-element) tuple as described earlier in Section 3, along with its direction (upstream or downstream) as picked with respect to the root reachability. The role of a given peer binding is also provided in case of MBB (as active or standby) and/or MoFRR (as primary or backup).

The following example illustrates a FEC-label binding database on a branch/transit LSR for both P2MP and MP2MP FECs of Transit Source type:

Transit IPv4 Source:

```
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)
```

```
FEC (root 203.0.113.1, S=198.51.100.2, G=224.1.1.1):
  type: mp2mp
  upstream:
    advertised:
      peer 192.0.2.1:0, label 26000 (local)
    received:
      peer 192.0.2.1:0, label 27000 (remote)
  downstream:
    advertised:
      peer 192.0.2.2:0, label 26001 (local), MBB role=active
      peer 192.0.2.3:0, label 26002 (local), MBB role=standby
    received:
      peer 192.0.2.2:0, label 27001 (remote)
      peer 192.0.2.3:0, label 28001 (remote)
```

Transit IPv6 Source:

```
FEC (root 203.0.113.1, S=2001:db8:0:9::1, G=ff39:1::1):
  type: p2mp
  upstream:
    advertised:
      peer 192.0.2.1:0, label 16100 (local)
  downstream:
    received:
      peer 192.0.2.2:0, label 17100 (remote)
      peer 192.0.2.3:0, label 18100 (remote)
```

```
FEC (root 203.0.113.1, S=2001:db8:0:9::2, G=ff39:1::1):
  type: mp2mp
  upstream:
    advertised:
      peer 192.0.2.1:0, label 26100 (local)
  received:
```

```

    peer 192.0.2.1:0, label 27100 (remote)
  downstream:
    advertised:
      peer 192.0.2.2:0, label 26101 (local), MBB role=active
      peer 192.0.2.3:0, label 26102 (local), MBB role=standby
    received:
      peer 192.0.2.2:0, label 27101 (remote)
      peer 192.0.2.3:0, label 28101 (remote)

```

Figure 7: Example Binding database

The following captures the high-level tree hierarchy for mLDP bindings state. The tree shown below is for ipv4 root only; a similar tree exists for ipv6 root as well.

```

+--rw mpls-ldp
  +--rw global
    +--rw mldp:mldp
      +--rw mldp:address-families
        +--rw mldp:ipv4!
          +--ro roots
            +--ro mldp:root* [root-address]
              +--ro mldp:root-address inet:ipv4-address
              +--ro mldp:bindings

```

The bindings tree as follows:

```

+--ro mldp:opaque-value-element-xxx
|   +--ro mldp:fec-label* [key11 key12 ...]
|   |   +--ro mldp:key11
|   |   +--ro mldp:key12
|   |   +--ro mldp:...
|   |   +--ro mldp:multipoint-type? multipoint-type
|   |   +--ro mldp:opaque-value-element-type? identityref
|   |   +--ro mldp:peer* [direction peer advertisement-type]
|   |   |   +--ro mldp:peer leafref
|   |   |   +--ro mldp:direction ldp:downstream-upstream
|   |   |   +--ro mldp:advertisement-type ldp:advertised-received
|   |   |   +--ro mldp:label? rt-types:mpls-label
|   |   |   +--ro mldp:mbb-status? lsp-mbb-role
|   |   |   +--ro mldp-ext:mofrr-status? lsp-mofrr-role
+--ro mldp:opaque-value-element-yyy
  +--ro mldp:fec-label* [key21 key22 ...]
  +--ro mldp: . . .

```

Figure 8: Binding state tree (ipv4 root)

mLDP binding state is organized and presented per root address; hence, the bindings container is under a root node in the model. The bindings state is made available for FECs pertaining to different types of opaque elements, with some state available under the "base" tree and the rest under the "extended" tree.

In the above tree, the various opaque types along with their type specific key(s) refer to the Table 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 worth highlighting that the address-family ipv4/ipv6 applies to "root" address in the mLDP binding tree. The other addresses (source, group, Rendezvous-Point (RP) etc.) do not have to be of the same address family type as the root.

6.2.1. Recursive FEC

The binding state for an MP LSP with recursive opaque value element applies only at a backbone node for MP LSP that is set up over a BGP-free backbone. In this scenario, the backbone node needs to maintain the binding state for a "recursive" FEC, where the root address is the recursive root and the opaque value element encodes the original FEC. It is to be noted that the original FEC relates to LSP's original root and opaque type (such as lspid, transit-source, transit-bidir, etc.). The backbone node does not interpret the fields of the original FEC; instead, the node treats it as a opaque binary stream for keying, storage, and display purposes.

6.3. Capabilities State

Like LDP, mLDP capabilities state comprises two types of information:

- * global: augments ldp:global/ldp:capability
- * per-peer: augments ldp:peers/ldp:peer/ldp:capability

The following tree captures the high-level tree hierarchy for mLDP capabilities state.

```

+--rw mpls-ldp
+--rw global
|   +--rw capability
|   |   +--rw mldp:mldp
|   |   |   +--rw mldp:p2mp
|   |   |   |   +--rw mldp:enabled?    boolean
|   |   |   +--rw mldp:mp2mp
|   |   |   |   +--rw mldp:enabled?    boolean
|   |   |   +--rw mldp:make-before-break
|   |   |   |   +--rw mldp:enabled?    boolean
|   |   |   |   +--rw mldp:switchover-delay?  uint16
|   |   |   |   +--rw mldp:timeout?    uint16
|   |   +--rw mldp-ext:hub-and-spoke {capability-mldp-hsmp}?
|   |   |   +--rw mldp-ext:enabled?    boolean
|   |   +--rw mldp-ext:node-protection
|   |   |   {capability-mldp-node-protection}?
|   |   |   +--rw mldp-ext:plr?        boolean
|   |   |   +--rw mldp-ext:merge-point
|   |   |   |   +--rw mldp-ext:enabled? boolean
|   |   |   |   +--rw mldp-ext:targeted-session-teardown-delay?
|   |   |   |   |   uint16
|   |   . . .
|   |   . . .
+--rw peers
+--rw peer* [lsr-id label-space-id]
|   +--rw lsr-id                rt-types:router-id
|   +--rw label-space-id        uint16
|   +--rw address-families
|   +--ro received-peer-state
|   |   +--ro capability
|   |   |   +--ro mldp:mldp
|   |   |   |   +--ro mldp:p2mp
|   |   |   |   |   +--ro mldp:is-capable?    boolean
|   |   |   |   +--ro mldp:mp2mp
|   |   |   |   |   +--ro mldp:is-capable?    boolean
|   |   |   |   +--ro mldp:make-before-break
|   |   |   |   |   +--ro mldp:is-capable?    boolean
|   |   |   |   +--ro mldp-ext:hub-and-spoke
|   |   |   |   |   +--ro mldp-ext:is-capable?    boolean
|   |   |   |   +--ro mldp-ext:node-protection
|   |   |   |   |   +--ro mldp-ext:is-plr-capable?    boolean
|   |   |   |   |   +--ro mldp-ext:is-merge-point-capable?    boolean

```

Figure 9: Capabilities state tree

7. Notifications

The mLDP notification model consists of notifications related to changes in the operational state of an mLDP FEC. The "base" mLDP data model includes only the "Generic LSP Identifier" opaque FEC type (for ipv4), while rest of the FEC types are covered by the "extended" mLDP model.

A simplified graphical representation of the data model for mLDP notifications is illustrated earlier in Figure 2.

8. Actions

This document does not define any RPCs/actions for mLDP.

9. YANG Specification

The following sections specify the actual YANG (module) specification for mLDP constructs defined earlier in the document.

9.1. Base

This YANG module imports types defined in [RFC9911], [RFC8343], [RFC8349], [RFC9070], and [RFC8294].

```
<CODE BEGINS> file "ietf-mpls-mldep@2026-02-03.yang"
// RFC Editor: replace the above date with the date of
// publication and remove this note.
```

```
module ietf-mpls-mldep {
  yang-version 1.1;

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

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 9911: 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 9070: YANG Data Model for MPLS LDP";
  }
```



```
}

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/mppls/>
  WG List:  <mailto:mppls@ietf.org>

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

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

  Editor:    Santosh Easale
             <mailto:santosh_easale@berkeley.edu>

  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 Multiprotocol Label Switching (MPLS) Multipoint
  LDP (mLDP).

  The types and groupings defined in this module are intended
  exclusively for use within this module itself or any other
  module that augments this module.

  Copyright (c) 2026 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 Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself for full legal notices.";

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

```
revision 2026-02-03 {
  // RFC Editor: replace the above date with the date of
  // publication and remove this note.
  description
    "Fixing copytight and authors before publication";
  reference
    "RFC XXXX: YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

revision 2025-12-15 {
  // RFC Editor: replace the above date with the date of
  // publication and remove this note.
  description
    "Removed static-lsps-leaf from the model";
  reference
    "RFC XXXX: YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

revision 2025-10-20 {
  // RFC Editor: replace the above date with the date of
  // publication and remove this note.
  description
    "Addressing RtgDir review comments related to recursive FEC";
  reference
    "RFC XXXX: YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

revision 2025-07-05 {
```

```
// RFC Editor: replace the above date with the date of
// publication and remove this note.
description
  "Addressing RtgDir review comments";
reference
  "RFC XXXX: Base YANG Data Model for MPLS mLDP";
// RFC Editor: replace XXXX with actual RFC number and remove
// this note
}

revision 2024-12-01 {
  // RFC Editor: replace the above date 2024-12-01 with the date of
  // publication and remove this note.
  description
    "Addressing YANG doctors review comments";
  reference
    "RFC XXXX: Base YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

revision 2021-11-11 {
  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)";
}

typedef lsp-mbb-role {
```

```
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
  "This type represents the Make-before-break
  role of an LSP.";
}

/*
 * Identities
 */

// IANA: LDP MP Opaque Value Element basic type
identity mldp-opaque-value-element-basic-type {
  description
    "Base identity from which specific Opaque value element
    basic types are derived.";
  reference
    "LDP MP Opaque Value Element basic type IANA registry at
    https://www.iana.org/assignments/ldp-namespaces/
    ldp-namespaces.xhtml#ldp-namespaces-11";
}

identity lspid {
  base mldp-opaque-value-element-basic-type;
  description "Generic LSP identifier type";
  reference "RFC 6388";
}

identity transit-source-v4 {
  base mldp-opaque-value-element-basic-type;
  description "Transit IPv4 Source TLV type";
  reference "RFC 6826";
}

identity transit-source-v6 {
  base mldp-opaque-value-element-basic-type;
  description "Transit IPv6 Source TLV type";
  reference "RFC 6826";
}
```

```
identity transit-bidir-v4 {
  base mldp-opaque-value-element-basic-type;
  description "Transit IPv4 Bidir TLV type";
  reference "RFC 6826";
}

identity transit-bidir-v6 {
  base mldp-opaque-value-element-basic-type;
  description "Transit IPv6 Bidir TLV type";
  reference "RFC 6826";
}

identity recur-opaque {
  base mldp-opaque-value-element-basic-type;
  description "Recursive Opaque value";
  reference "RFC 6512";
}

identity transit-source-vpnv4 {
  base mldp-opaque-value-element-basic-type;
  description "Transit VPNv4 Source TLV type";
  reference "RFC 7246";
}

identity transit-source-vpnv6 {
  base mldp-opaque-value-element-basic-type;
  description "Transit VPNv6 Source TLV type";
  reference "RFC 7246";
}

identity transit-bidir-vpnv4 {
  base mldp-opaque-value-element-basic-type;
  description "Transit VPNv4 Bidir TLV type";
  reference "RFC 7246";
}

identity transit-bidir-vpnv6 {
  base mldp-opaque-value-element-basic-type;
  description "Transit VPNv6 Bidir TLV type";
  reference "RFC 7246";
}

identity recur-opaque-vpn {
  base mldp-opaque-value-element-basic-type;
  description "VPN-Recursive Opaque value";
  reference "RFC 6512";
}
```

```
/*
 * Groupings
 */
grouping mldp-capabilities-base {
  description
    "A grouping describing the protocol capabilities of mLDP as
    specified in RFC6388";

  container p2mp {
    description
      "Configuration and state information for the
      point-to-multipoint capability";
    leaf enabled {
      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 enabled {
      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 enabled {
      type boolean;
      description
        "'true' to enable the make-before-break capability";
    }
    leaf switchover-delay {
      type uint16;
      units seconds;
      description
        "Switchover delay (in seconds) for activating an inactive
        element";
    }
    leaf timeout {
      type uint16;
      units seconds;
      description
        "To prevent waiting indefinitely for the MBB Notification
        to arrive, a timeout should be applied. As soon as the
```

```

        timer expires, the MBB procedures, same as those applied
        at the receipt of an MBB notification, are applied on
        the inactive element.";
    }
}
} // mldp-capabilities-base

grouping mldp-binding-label-peer-state-attributes {
  description
    "mLDP label binding per peer attributes";
  leaf direction {
    type ldp:downstream-upstream;
    description
      "Downstream or upstream label binding";
  }
  leaf advertisement-type {
    type ldp:advertised-received;
    description
      "Advertised (outbound) or received label (inbound) binding";
  }
  leaf label {
    type rt-types:mpls-label;
    description
      "Advertised (outbound) or received (inbound) label";
  }
  leaf mbb-status {
    when "../direction = 'upstream'" {
      description
        "This leaf is used for MBB upstream direction only.";
    }
    type lsp-mbb-role;
    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 peers with bindings exchnaged (sent and/or
      received)";
    leaf peer {
      type leafref {
        path
          "../../../../../../../../../../../../ldp:peers/ldp:peer/"
          + "ldp:lsr-id";
      }
    }
  }
}

```

```
    }
    description
      "[m]LDP peer from which this binding is received, or
      to which this binding is advertised.";
  }
  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 configuration and state
      information for multipoint LDP capabilities.";
    uses mldp-capabilities-base;
  }
}

/*
 * 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 peer";

    container p2mp {
      description
        "Operational state information for the point-to-multipoint
        capability of a peer";
      leaf is-capable {
        type boolean;
        description
          "'true' when point-to-multipoint capability is received
          from peer";
      }
    }
  }
  container mp2mp {
```



```
    description
      "Operational state information for the
      multipoint-to-multipoint capability of a peer";
    leaf is-capable {
      type boolean;
      description
        "'true' when multipoint-to-multipoint capability is
        received from peer";
    }
  }
  container make-before-break {
    description
      "Operational state information for the make-before-break
      capability of a peer";
    leaf is-capable {
      type boolean;
      description
        "'true' when make-before-break capability is received
        from peer";
    }
  }
} // 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
      capability container";

    leaf enabled {
      type boolean;
      description
        "'true' to enable mLDP protocol";
    }

    container address-families {
      description
        "Per-address-family configuration and operational state.
        The address family can be either IPv4 or IPv6.";

      container ipv4 {
        presence "Enables IPv4 mLDP support unless the
```

```
        'enabled' leaf is set to 'false'.";
description
    "Containing data related to the IPv4 address family.";
leaf enabled {
    type boolean;
    default "true";
    description
        "'false' to disable the IPv4 address family.";
}

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
                "The node itself is the root node.";
        }
    }

    list reachability {
        key "address interface";
        description
            "Root reachability information in form of a
            next-hop";
        leaf address {
            type inet:ipv4-address;
            description
                "The next-hop address to reach root";
        }
        leaf interface {
            type if:interface-ref;
            description
                "The next-hop interface to reach root";
        }
        leaf peer {
            type leafref {
                path

```

```

        ".../.../.../.../.../.../.../ldp:peers/"
        + "ldp:peer/ldp:lsr-id";
    }
    description
        "LDP peer associated with the nexthop (address,
        interface)";
    }
}

container bindings {
    description
        "mLDP FEC-label bindings";
    container opaque-value-element-lspid {
        description
            "Generic LSP identifier opaque element FEC-label
            bindings container";
        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-label bindings";
            leaf lsp-id {
                type uint32;
                description "LSP identifier";
            }
            leaf multipoint-type {
                type multipoint-type;
                description
                    "The type of multipoint: p2mp or mp2mp";
            }
            leaf opaque-value-element-type {
                type identityref {
                    base mldp-opaque-value-element-basic-type; }
                description "Type of Opaque value element";
            }
        }

        uses mldp-binding-label-state-attributes;
    } // fec-label
    } // opaque-value-element-lspid
} // bindings

} // list root
} // roots
} // 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-value-element {
    description
      "The type of opaque value element";
    case opaque-value-element-lspid {
      container opaque-value-element-lspid {
        description
          "Generic LSP identifier opaque element";
        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 "LSP identifier";
        }
        leaf multipoint-type {
          type multipoint-type;
          description
            "The type of multipoint: p2mp or mp2mp";
        }
      } // container opaque-value-element-lspid
    }
  }
}
<CODE ENDS>

```

Figure 10: mLDP base module

9.2. Extended

This YANG module imports types defined in [RFC9911], [RFC8343], [RFC8349], [RFC9070], and [RFC8294].

```
<CODE BEGINS> file "ietf-mpls-mldp-extended@2026-02-03.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 9911: 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 9070: YANG Data Model for MPLS LDP";
  }

  import ietf-mpls-ldp-extended {
    prefix "ldp-ext";
    reference "RFC 9070: YANG Data Model for MPLS LDP";
  }
  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/mppls/>>

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

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

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

Editor: Santosh Easale
<mailto:santosh_easale@berkeley.edu>

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 Multiprotocol Label Switching (MPLS) Multipoint LDP (mLDP).

The types and groupings defined in this module are intended exclusively for use within this module itself or any other module that augments this module.

Copyright (c) 2026 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 Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself

```
    for full legal notices.";

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

revision 2026-02-03 {
    // RFC Editor: replace the above date with the date of
    // publication and remove this note.
    description
        "Fixing copyright and authors before publication";
    reference
        "RFC XXXX: YANG Data Model for MPLS mLDP";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}

revision 2025-12-15 {
    // RFC Editor: replace the above date with the date of
    // publication and remove this note.
    description
        "Removed static-lsps-leaf from the model";
    reference
        "RFC XXXX: YANG Data Model for MPLS mLDP";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}

revision 2025-10-20 {
    // RFC Editor: replace the above date with the date of
    // publication and remove this note.
    description
        "Addressing RtgDir review comments related to recursive FEC";
    reference
        "RFC XXXX: Base YANG Data Model for MPLS mLDP";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}

revision 2025-07-05 {
    // RFC Editor: replace the above date with the date of
    // publication and remove this note.
    description
        "Addressing RtgDir review comments";
    reference
        "RFC XXXX: Base YANG Data Model for MPLS mLDP";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}
```

```
revision 2024-12-01 {
  // RFC Editor: replace the above date 2024-12-01 with the date of
  // publication and remove this note.
  description
    "Addressing YANG doctors review comments";
  reference
    "RFC XXXX: Base YANG Data Model for MPLS mLDP";
  // RFC Editor: replace XXXX with actual RFC number and remove
  // this note
}

revision 2021-11-11 {
  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.";
  reference
    "RFC7140: LDP Extensions for Hub and Spoke Multipoint
    Label Switched Path";
}

feature capability-mldp-node-protection {
  description
    "This feature indicates that the system allows to configure
    mLDP node-protection capability.";
  reference
    "RFC7715: Multipoint LDP (mLDP) Node Protection";
}

feature mldp-mofrr {
  description
    "This feature indicates that the system supports mLDP
    Multicast only FRR (MoFRR).";
  reference
    "RFC7431: Multicast-Only Fast Reroute";
}
```



```
feature per-peer-capability {
  description
    "This feature indicates that the system allows to configure
    mLDP capabilities at the per peer level.";
}

/*
 * Typedefs
 */
typedef lsp-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 of an LSP.";
}

/*
 * Groupings
 */
grouping mldp-ext-binding-label-peer-state {
  description
    "mLDP label binding peer state.";

  list peer {
    key "direction peer advertisement-type";
    description
      "List of peers with bindings exchnaged (sent and/or
      received)";
    leaf peer {
      type leafref {
        path
          ".../ldp:peers/ldp:peer/"
          + "ldp:lsr-id";
      }
      description
        "[m]LDP peer from which this binding is received, or
        to which this binding is advertised.";
    }
  }
  uses mldp:mldp-binding-label-peer-state-attributes;
}
```

```
    leaf mofrr-status {
      when "../direction = 'upstream'" {
        description
          "This leaf is used for MoFRR upstream direction only.";
      }
      type lsp-mofrr-role;
      description
        "The MoFRR status of this LSP";
    }
  } // peer
} // mldp-ext-binding-label-peer-state

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 enabled {
      type boolean;
      description
        "Enable hub-and-spoke-multipoint";
    }
  }
  container node-protection {
    if-feature capability-mldp-node-protection;
    description
      "Configure node-protection capability.";
    reference
      "RFC7715: mLDP Node Protection.";
    leaf plr {
      type boolean;
      description
        "Point of Local Repair (PLR) capable for Multipoint LSP
          node protection";
    }
  }
  container merge-point {
    description
      "Merge Point capable for Multipoint LSP node protection";
    leaf enabled {
      type boolean;
      description
        "Enable merge point capability";
    }
  }
}
```

```
    leaf targeted-session-teardown-delay {
        type uint16;
        units seconds;
        description
            "Targeted session teardown delay";
    }
} // merge-point
} // node-protection
} // mldp-ext-capabilities

grouping mldp-ext-per-af-config-attributes {
    description
        "mLDP per address family configuration attributes";
    container multicast-only-frr {
        if-feature mldp-mofrr;
        description
            "Multicast-only FRR (MoFRR) policy";
        reference
            "RFC7431: Multicast-Only Fast Reroute";
        leaf prefix-list {
            type ldp-ext:prefix-list-ref;
            description
                "Enables Multicast-only FRR (MoFRR) for the specified
                prefix-list";
        }
    } // multicast-only-frr

    container recursive-fec {
        description
            "Recursive FEC policy";
        leaf prefix-list {
            type ldp-ext:prefix-list-ref;
            description
                "Enables recursive FEC for the specified prefix-list";
        }
    } // recursive-fec
} // mldp-ext-per-af-config-attributes

grouping opaque-value-element-recursive-container {
    description
        "Container mLDP recursive opaque elements.";
    reference
        "RFC6512: Using Multipoint LDP When the
        Backbone Has No Route to the Root";

    container opaque-value-element-recursive {
        config false;
        description
            "Recursive opaque type";
    }
}
```

```
reference
  "RFC6512: Using Multipoint LDP When the
  Backbone Has No Route to the Root";
list fec-label {
  key
    "rd opaque-value";
  description
    "List of FEC-label bindings";
  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.";
  }
  leaf opaque-value {
    type binary;
    description
      "The opaque value that is not interpreted at this device.";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description "The type of multipoint: p2mp or mp2mp";
  }
  leaf opaque-value-element-type {
    type identityref
      { base mldp:mldp-opaque-value-element-basic-type; }
    description "Type of Opaque value element";
  }

  uses mldp-ext-binding-label-peer-state;
} // fec-label
} // opaque-value-element-recursive
} // opaque-value-element-recursive-container

grouping recursive-fec-base-attributes {
  description
    "mLDP recursive FEC base attributes.";
  reference
    "RFC6512: Using Multipoint LDP When the
    Backbone Has No Route to the Root";
  leaf recur-root-address {
    type inet:ip-address;
    description
      "Recursive root address";
  }
}
```

```
    } // recursive-fec-base-attributes

    grouping recursive-fec-attributes {
        description
            "mLDP recursive FEC attributes.";
        reference
            "RFC6512: Using Multipoint LDP When the
            Backbone Has No Route to the Root";

        uses recursive-fec-base-attributes;

        leaf recur-rd {
            type rt-types:route-distinguisher;
            description
                "Route Distinguisher in the VPN-Recursive
                Opaque Value";
        }
    } // recursive-fec-attributes

    /*
     * 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 capabilities.";

        uses mldp-ext-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-attributes;
    }

    // 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 IPv6 configuration";
        uses mldp-ext-per-af-config-attributes;
    }

    // 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-value-element-lspid/mldp:fec-label/"
+ "mldp:peer" {
  description "Augmentation for mLDP IPv4 state";

  leaf mofrr-status {
    when "../mldp:direction = 'upstream'" {
      description
        "This leaf is used for upstream direction only.";
    }
    type lsp-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
      "Operational state information for the peer's
      hub-and-spoke-multipoint capability.";
    reference
      "RFC7140: LDP Extensions for Hub and Spoke Multipoint
      Label Switched Path";
  }
}
```

```
    leaf is-capable {
      type boolean;
      description
        "'true' when hub-and-spoke-multipoint capability is
        received from peer";
    }
  }
  container node-protection {
    description
      "Operational state information for the peer's node-protection
      capability";
    reference
      "RFC7715: mLDP Node Protection.";
    leaf is-plr-capable {
      type boolean;
      description
        "'true' when Point of Local Repair (PLR) node-protection
        capability is received from peer";
    }
    leaf is-merge-point-capable {
      type boolean;
      description
        "'true' when Merge-point node-protection capability is
        received from peer";
    } // merge-point
  } // node-protection
}

// IPv4 bindings state
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/mldp:ml dp/"
+ "ml dp:address-families/ml dp:ipv4/ml dp:roots/ml dp:root/"
+ "ml dp:bindings" {
  description "Augmentation for mLDP IPv4 bindings extended type.";

  container opaque-value-element-transit-source {
    description
      "Transit IPv4/VPNv4 Source opaque type";
    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";
      description
        "List of FEC-label bindings";
      leaf source-address {
```

```
    type inet:ipv4-address;
    description
      "Source address (0.0.0.0 refers to wildcard *)";
  }
  leaf group-address {
    type inet:ipv4-address;
    description
      "Group address (0.0.0.0 refers to wildcard *)";
  }
  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.";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description "The type of multipoint: p2mp or mp2mp";
  }
  leaf opaque-value-element-type {
    type identityref
      { base mldp:mldp-opaque-value-element-basic-type; }
    description "Type of Opaque value element";
  }

  uses mldp-ext-binding-label-peer-state;
} // fec-label
} // opaque-value-element-transit-source

container opaque-value-element-transit-bidir {
  description
    "Transit IPv4/VPNv4 Bidir opaque type";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched
    Paths.";
  list fec-label {
    key
      "masklen rp group-address rd";
    description
      "List of FEC-label bindings";
    leaf masklen {
      type uint8;
      description "Mask length for the subnet";
    }
  }
}
```



```
    }
    leaf rp {
      type inet:ipv4-address;
      description
        "Rendezvous Point (RP) address";
    }
    leaf group-address {
      type inet:ipv4-address;
      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.";
    }
    leaf multipoint-type {
      type mldp:multipoint-type;
      description "The type of multipoint: p2mp or mp2mp";
    }
    leaf opaque-value-element-type {
      type identityref
        { base mldp:mldp-opaque-value-element-basic-type; }
      description "Type of Opaque value element";
    }
  }

  uses mldp-ext-binding-label-peer-state;
} // fec-label
} // opaque-value-element-transit-bidir

uses opaque-value-element-recursive-container;
}

// 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-value-element-transit-source {
    config false;
    description
      "Transit IPv6/VPNv6 Source opaque type";
    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";
  description
    "List of FEC-label bindings";
  leaf source-address {
    type inet:ipv6-address;
    description
      "Source address (::0 refers to wildcard *)";
  }
  leaf group-address {
    type inet:ipv6-address;
    description
      "Group address (::0 refers to wildcard *)";
  }
  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.";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description "The type of multipoint: p2mp or mp2mp";
  }
  leaf opaque-value-element-type {
    type identityref
      { base mldp:mldp-opaque-value-element-basic-type; }
    description "Type of Opaque value element";
  }

  uses mldp-ext-binding-label-peer-state;
} // fec-label
} // opaque-value-element-transit-source

container opaque-value-element-transit-bidir {
  config false;
  description
    "Transit IPv6/VPNv6 Bidir opaque type";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
```

```

    Multipoint-to-Multipoint Label Switched
    Paths.";
list fec-label {
  key
    "masklen rp group-address rd";
  description
    "List of FEC-label bindings";
  leaf masklen {
    type uint8;
    description "Mask length for the subnet";
  }
  leaf rp {
    type inet:ipv6-address;
    description
      "Rendezvous Point (RP) address";
  }
  leaf group-address {
    type inet:ipv6-address;
    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.";
  }
  leaf multipoint-type {
    type mldp:multipoint-type;
    description "The type of multipoint: p2mp or mp2mp";
  }
  leaf opaque-value-element-type {
    type identityref
      { base mldp:mldp-opaque-value-element-basic-type; }
    description "Type of Opaque value element";
  }
  uses mldp-ext-binding-label-peer-state;
} // fec-label
} // opaque-value-element-transit-bidir

uses opaque-value-element-recursive-container;
}

/*

```

```
* Per AF augmentation
*/
// IPv6 augmentation
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/mldp:ml dp/"
+ "ml dp:address-families" {
  description "Augmentation for mLDP IPv6 address family.";
  container ipv6 {
    presence "Enables IPv6 mLDP support unless the
              'enabled' leaf is set to 'false'.";
    description
      "Containing data related to the IPv6 address family.";
    leaf enabled {
      type boolean;
      default "true";
      description
        "'false' to disable the IPv6 address family.";
    }
  }
  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
          "The node itself is the root node.";
      }
    }
    list reachability {
      key "address interface";
      description
        "Root reachability information in form of a next-hop";
      leaf address {
        type inet:ipv6-address;
        description
          "The next-hop address to reach root";
      }
    }
  }
}
```

```

    }
    leaf interface {
        type if:interface-ref;
        description
            "The next-hop interface to reach root";
    }
    leaf peer {
        type leafref {
            path
                ".../.../.../.../.../.../.../ldp:peers/"
                + "ldp:peer/ldp:lsr-id";
        }
        description
            "LDP peer associated with the nexthop (address,
            interface)";
    }
}

container bindings {
    description
        "mLDP FEC-label bindings";
    container opaque-value-element-lspid {
        description
            "Generic LSP identifier opaque element FEC-label
            bindings container";
        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-label bindings";
            leaf lsp-id {
                type uint32;
                description "LSP identifier";
            }
            leaf multipoint-type {
                type mldp:multipoint-type;
                description
                    "The type of multipoint: p2mp or mp2mp";
            }
            leaf opaque-value-element-type {
                type identityref {
                    base mldp:mldp-opaque-value-element-basic-type; }
                description "Type of Opaque value element";
            }
        }
    }
}

```

```

        uses mldp-ext-binding-label-peer-state;
    } // fec-label
} // opaque-value-element-lspid
} // bindings
} // list root
} // roots
} // ipv6
}

/*
 * Global augmentation
 */
/*
 * Notifications
 */
augment "/mldp:mpls-mldp-fec-event/mldp:opaque-value-element/"
+ "mldp:opaque-value-element-lspid/mldp:opaque-value-element-lspid" {
description
"Augmentation for mLDP notification for opaque-value-element-lspid.";
    container recursive-fec {
        description
            "Container of recursive opaque values";
        uses recursive-fec-base-attributes;
    } // recursive-fec
}

augment "/mldp:mpls-mldp-fec-event/mldp:opaque-value-element" {
description
"Augmentation for mLDP notification for opaque types other
than lspid.";
    case opaque-value-element-transit-source {
        container opaque-value-element-transit-source {
            description
                "Transit IP/VPN Source opaque type";
            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;
        description
            "Group address";
    }
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "Route Distinguisher";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    uses recursive-fec-attributes;
} // opaque-value-element-transit-source container
} // opaque-value-element-transit-source case

case opaque-value-element-transit-bidir {
    container opaque-value-element-transit-bidir {
        description
            "Transit IP/VPN Bidir opaque type";
        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 masklen {
            type uint8;
            description "Mask length for the subnet";
        }
        leaf rp {
            type inet:ip-address;
            description
                "Rendezvous Point (RP) address";
        }
        leaf group-address {
            type inet:ip-address;
            description
                "Group address";
        }
    }
    leaf rd {
        type rt-types:route-distinguisher;
```

```
    description
      "Route Distinguisher";
    reference
      "RFC7246: Multipoint Label Distribution
       Protocol In-Band Signaling in a Virtual
       Routing and Forwarding (VRF) Table
       Context.";
  }
  uses recursive-fec-attributes;
} // opaque-value-element-transit-bidir container
} // opaque-value-element-transit-bidir case
} // augment
} // module
<CODE ENDS>
```

Figure 11: mLDP extended module

10. Security Considerations

This specification also inherits the security considerations captured in [RFC5920] and mLDP 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].

10.1. YANG Data Model

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

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

The following sub sections capture the security considerations with respect to mLDP data nodes defined in mLDP YANG modules. This goes without saying that LDP data nodes security considerations, as captured in in LDP YANG specification [RFC9070], apply orthogonally to mLDP as well.

10.1.1. Writable Nodes

There are several 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.

For mLDP, the ability to modify MPLS mLDP configuration may allow the entire MPLS mLDP domain to be compromised including forming LDP adjacencies and/or peer sessions with unauthorized routers to mount a massive Denial-of-Service (DoS) attack.

In particular, the subtrees and data nodes that are sensitive and vulnerable are same as captured in [RFC9070] section 10.1.1

10.1.2. Readable Nodes

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. The following text lists the subtrees and data nodes along with their sensitivity/vulnerability:

The exposure of mLDP databases (such as mLDP peers, mLDP Roots, mLDP FEC-Label bindings) as well LDP databases (such as hello adjacencies, TCP sessions, and address bindings) beyond the scope of the mLDP admin domain may be undesirable. The relevant subtrees and data nodes for LDP are captured in [RFC9070] section 10.1.2, whereas the relevant subtrees and data nodes for mLDP are as follows:

- * /mpls-ldp/global/mldp/address-families/ipv4/roots
- * /mpls-ldp/global/mldp/address-families/ipv6/roots
- * /mpls-ldp/global/mldp/address-families/ipv4/roots/root/bindings
- * /mpls-ldp/global/mldp/address-families/ipv6/roots/root/bindings
- * /mpls-ldp/peers/peer/received-peer-state/capability/mldp

10.1.3. Notifications

The mLDP implementations must rate-limit the generation of mLDP notifications to avoid creating significant notification load and possible side effects on the system stability.

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

Table 2: URIs

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

Table 3: YANG Modules

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

12. References

12.1. Normative References

- [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>>.
- [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>>.

- [RFC7438] Wijnands, IJ., Ed., Rosen, E., Gulko, A., Joerde, 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>>.
- [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>>.

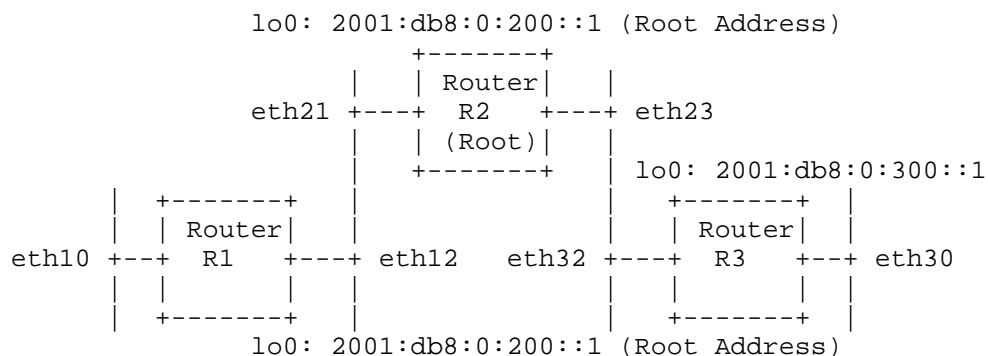
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.
- [RFC9070] Raza, K., Ed., Asati, R., Liu, X., Esale, S., Chen, X., and H. Shah, "YANG Data Model for MPLS LDP", RFC 9070, DOI 10.17487/RFC9070, March 2022, <<https://www.rfc-editor.org/info/rfc9070>>.
- [RFC9911] Schowalder, J., Ed., "Common YANG Data Types", RFC 9911, DOI 10.17487/RFC9911, December 2025, <<https://www.rfc-editor.org/info/rfc9911>>.

12.2. Informative References

- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<https://www.rfc-editor.org/info/rfc5920>>.
- [RFC7431] Karan, A., Filmsils, 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>>.
- [RFC7473] Raza, K. and S. Boutros, "Controlling State Advertisements of Non-negotiated LDP Applications", RFC 7473, DOI 10.17487/RFC7473, March 2015, <<https://www.rfc-editor.org/info/rfc7473>>.
- [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": {
            "roots": {
              "root": [ { "root-address": "2001:db8:0:200::1" } ]
            }
          }
        }
      }
    }
  },
  "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 corresponding 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": {
            "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-ml dp:ml dp": {
          "mp2mp": {
            "enable": true
          }
        }
      }
    }
  }
}

```

Appendix B. Acknowledgments

The authors would like to acknowledge Ladislav Lhotka, Acee Lindem, and Sowmya Krishnaswamy for their review and comments.

Appendix C. Contributors

Rajiv Asati
 Individual
 Email: Rajiv.ASATI@gmail.com

Xia Chen
 Huawei Technologies
 Email: xiachen002@outlook.com

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

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

Authors' Addresses

Kamran Raza (editor)
Cisco Systems
Email: skraza@cisco.com

Xufeng Liu
Individual
Email: xufeng.liu.ietf@gmail.com

Santosh Easale
Juniper Networks
Email: santosh_easale@berkeley.edu

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Jeff Tantsura
Microsoft Corporation
Email: jefftant.ietf@gmail.com