Internet Engineering Task Force (IETF)
Request for Comments: 8960
Category: Standards Track
ISSN: 2070-1721

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

December 2020

# A YANG Data Model for MPLS Base

## Abstract

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

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8960.



## Table of Contents

## 1. Introduction

A core routing YANG data model is defined in [RFC8349]; it provides a basis for the development of routing data models for specific Address Families (AFs). Specifically, [RFC8349] defines a model for a generic Routing Information Base (RIB) that is AF agnostic. [RFC8349] also defines two instances of RIBs based on the generic RIB model for IPv4 and IPv6 AFs.

The MPLS base model defined in this document augments the generic RIB model defined in [RFC8349] with additional data that enables MPLS forwarding for one or more specific destination prefixes present in one or more AF RIBs, as described in the MPLS architecture document [RFC3031].

The MPLS base model also defines a new instance of the generic RIB YANG data model as defined in [RFC8349] to store native MPLS routes. The native MPLS RIB instance stores one or more routes that are not associated with other AF instance RIBs (such as IPv4 or IPv6 instance RIBs) but are enabled for MPLS forwarding. Examples of such native MPLS routes are routes programmed by RSVP on one or more transit MPLS routers along the path of a Label Switched Path (LSP). Other examples are MPLS routes that cross-connect to specific Layer 2 adjacencies, such as Layer 2 Attachment Circuits (ACs); or Layer 3 adjacencies, such as Segment Routing (SR) Adjacency Segments (Adj-SIDs) as described in [RFC8402].

The MPLS base YANG data model serves as a basis for future development of MPLS YANG data models covering MPLS features and subsystems that are more sophisticated. The main purpose is to provide essential building blocks for other YANG data models involving different control-plane protocols and MPLS functions.

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

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

### 1.1. Terminology

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

### 1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

RIB: Routing Information Base

LSP: Label Switched Path

LSR: Label Switching Router

NHLFE: Next Hop Label Forwarding Entry

## 2. MPLS Base Model

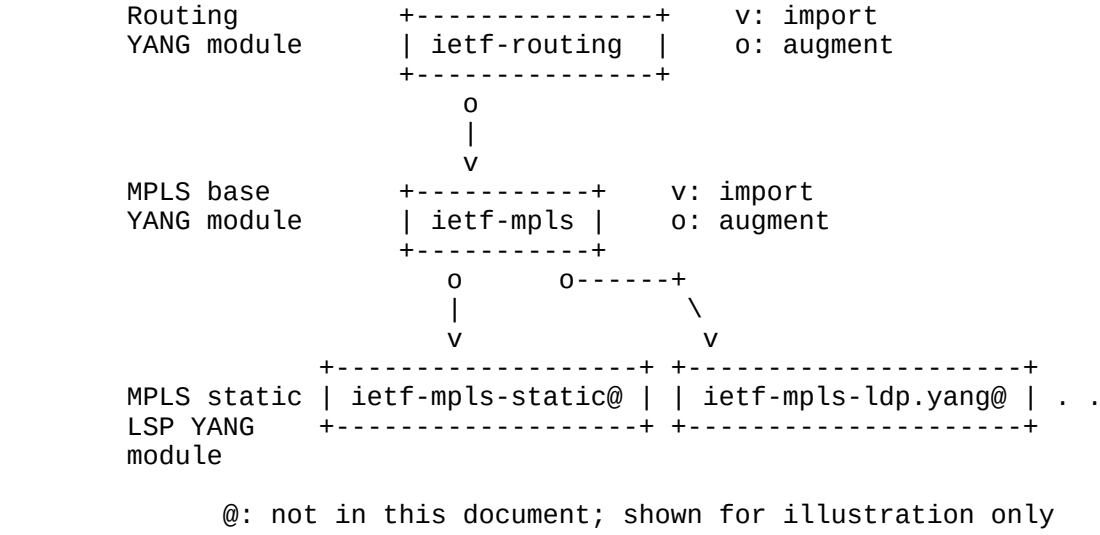
This document describes the "ietf-mpls" YANG module, which provides base components of the MPLS data model. It is expected that other MPLS YANG modules will augment the "ietf-mpls" YANG module for other MPLS extensions to provision LSPs (e.g., MPLS static, MPLS LDP, or MPLS RSVP-TE LSPs).

### 2.1. Model Overview

This document models MPLS-labeled routes as an augmentation of the generic routing RIB data model as defined in [RFC8349]. For example, IP prefix routes (e.g., routes stored in IPv4 or IPv6 RIBs) are augmented to carry additional data to enable them for MPLS forwarding.

This document also defines a new instance of the generic RIB model defined in [RFC8349] to store one or more native MPLS routes (described further in Section 2.3) by extending the identity "address-family" defined in [RFC8349] with a new "mpls" identity; see Section 3 of [RFC8349].

### 2.2. Model Organization




@: not in this document; shown for illustration only

Figure 1: Relationship between MPLS Modules

The "ietf-mpls" YANG module defines the following identities:

mpls:
: Identity that extends the "address-family" identity of RIB instances, as defined in [RFC8349], to represent the native MPLS RIB instance.

label-block-alloc-mode:
: A base YANG identity for one or more supported label-block allocation modes.

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

mpls-operations-type:
: An enumeration type that represents support for possible MPLS operation types (impose-and-forward, pop-and-forward, pop-impose-and-forward, and pop-and-lookup).

nhlfe-role:
   An enumeration type that represents the role of the Next Hop Label Forwarding Entry (NHLFE).

nhlfe-single-contents:
   A YANG grouping that describes a single NHLFE and its associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is specific to the case when a single next hop is associated with the route.

The NHLFE is used when forwarding a labeled packet. It contains the following information:

1. The packet's next hop. For "nhlfe-single-contents", only a single next hop is expected, while for "nhlfe-multiple-contents", multiple next hops are possible.

2. The operation to perform on the packet's label stack. This can be one of the following operations:

   a. Replace the label at the top of the label stack with one or more specified new labels.

   b. Pop the label stack.

   c. Replace the label at the top of the label stack with a specified new label, and then push one or more specified new labels onto the label stack.

   d. Push one or more labels onto an unlabeled packet.

The NHLFE may also contain:

1. The data-link encapsulation to use when transmitting the packet.

2. The way to encode the label stack when transmitting the packet.

3. Any other information needed in order to properly dispose of the packet.

nhlfe-multiple-contents:
   A YANG grouping that describes a set of NHLFEs and their associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is used when multiple next hops are associated with the route.

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

label-blocks:
   A YANG grouping that describes the list of assigned MPLS label blocks and their properties.

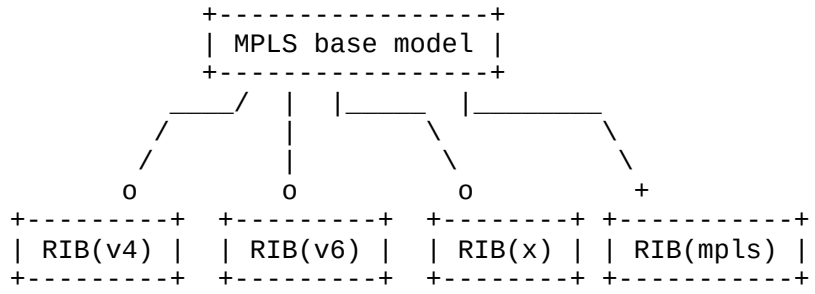
rib-mpls-properties:
   A YANG grouping for the augmentation of the generic RIB with MPLS label forwarding data as defined in [RFC3031].

rib-active-route-mpls-input:
   A YANG grouping for the augmentation to the "active-route" RPC that is specific to the MPLS RIB instance.

## 2.3. Model Design

The MPLS routing model is based on the core routing data model defined in [RFC8349]. Figure 2 shows the extensions introduced by

the MPLS base model on defined RIBs.




+: created by the MPLS base model
o: augmented by the MPLS base model

Figure 2: Relationship between MPLS Model and RIB Instances

As shown in Figure 2, the MPLS base YANG data model augments defined instances of AF RIBs with additional data that enables MPLS forwarding for destination prefixes stored in such RIBs. For example, an IPv4 prefix stored in RIB(v4) is augmented to carry an MPLS local label and one or more per-next-hop remote labels to enable MPLS forwarding for such a prefix.

The MPLS base model also creates a separate instance of the generic RIB model defined in [RFC8349] to store one or more MPLS native routes that are enabled for MPLS forwarding but are not stored in one or more other AF RIBs.

Some examples of such native MPLS routes are:

* Routes programmed by RSVP on Label Switching Routers (LSRs) along the path of an LSP,

* Routes that cross-connect an MPLS local label to a Layer 2 or Layer 3 Virtual Routing and Forwarding (VRF) entity,

* Routes that cross-connect an MPLS local label to a specific Layer 2 adjacency or interface, such as Layer 2 Attachment Circuits (ACs), or

* Routes that cross-connect an MPLS local label to a Layer 3 adjacency or interface, such as MPLS Segment Routing (SR) Adjacency Segments (Adj-SIDs) or SR MPLS Binding SIDs as defined in [RFC8402].

## 2.4. Model Tree Diagram

The MPLS base tree diagram, which follows the notation defined in [RFC8340], is shown in Figure 3.

```
module: ietf-mpls
  augment /rt:routing:
    +--rw mpls
       +--rw ttl-propagate?       boolean
       +--rw mpls-label-blocks
       |  +--rw mpls-label-block* [index]
       |     +--rw index                     string
       |     +--rw start-label?              rt-types:mpls-label
       |     +--rw end-label?                rt-types:mpls-label
       |     +--rw block-allocation-mode?    identityref
       |     +--ro inuse-labels-count?       yang:gauge32
       +--rw interfaces
          +--rw interface* [name]
             +--rw name                       if:interface-ref
```

```
            +--rw mpls-enabled?             boolean
            +--rw maximum-labeled-packet?   uint32
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
    +--ro mpls-enabled?         boolean
    +--ro mpls-local-label?     rt-types:mpls-label
    +--ro destination-prefix?   -> ../mpls-local-label
    +--ro route-context?        string
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
            /rt:next-hop-options/rt:simple-next-hop:
    +--ro mpls-label-stack
       +--ro entry* [id]
          +--ro id               uint8
          +--ro label?           rt-types:mpls-label
          +--ro ttl?             uint8
          +--ro traffic-class?   uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
            /rt:next-hop-options/rt:next-hop-list/rt:next-hop-list
            /rt:next-hop:
    +--ro index?              string
    +--ro backup-index?       string
    +--ro loadshare?          uint16
    +--ro role?               nhlfe-role
    +--ro mpls-label-stack
       +--ro entry* [id]
          +--ro id               uint8
          +--ro label?           rt-types:mpls-label
          +--ro ttl?             uint8
          +--ro traffic-class?   uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input:
    +---w destination-address?   -> ../mpls-local-label
    +---w mpls-local-label?      rt-types:mpls-label
  augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
            /rt:route/rt:next-hop/rt:next-hop-options
            /rt:simple-next-hop:
    +-- mpls-label-stack
       +-- entry* [id]
          +-- id               uint8
          +-- label?           rt-types:mpls-label
          +-- ttl?             uint8
          +-- traffic-class?   uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
            /rt:route/rt:next-hop/rt:next-hop-options
            /rt:next-hop-list/rt:next-hop-list/rt:next-hop:
    +-- index?              string
    +-- backup-index?       string
    +-- loadshare?          uint16
    +-- role?               nhlfe-role
    +-- mpls-label-stack
       +-- entry* [id]
          +-- id               uint8
          +-- label?           rt-types:mpls-label
          +-- ttl?             uint8
          +-- traffic-class?   uint8
```

Figure 3: MPLS Base Tree Diagram

## 2.5. MPLS Base YANG Module

This section describes the "ietf-mpls" YANG module, which provides base components of the MPLS data model. Other YANG modules may import and augment the MPLS base module to add feature-specific data.

The "ietf-mpls" YANG module imports the following YANG modules:

* "ietf-routing" as defined in [RFC8349]

```
*  "ietf-routing-types" as defined in [RFC8294]

*  "ietf-yang-types" as defined in [RFC6991]

*  "ietf-interfaces" as defined in [RFC8343]

This YANG module also references the following RFCs in defining the
types, YANG groupings, and other features of the YANG module:
[RFC3031], [RFC3032], [RFC4090], [RFC5714], and [RFC7424].

<CODE BEGINS> file "ietf-mpls@2020-12-18.yang"
module ietf-mpls {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls";

  prefix mpls;

  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-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

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

     Editor:   Tarek Saad
               <mailto:tsaad@juniper.net>

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

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

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

     Editor:   Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>";
  description
    "This YANG module defines the essential components for the
     management of the MPLS subsystem.  The model fully conforms
     to the Network Management Datastore Architecture (NMDA).

     Copyright (c) 2020 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.
```

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

   This version of this YANG module is part of RFC 8960; see the
   RFC itself for full legal notices.";

revision 2020-12-18 {
  description
    "Initial revision.";
  reference
    "RFC 8960: A YANG Data Model for MPLS Base";
}

/* Identities */

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

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

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

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

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

/**
 * Typedefs
 */

typedef mpls-operations-type {
  type enumeration {
    enum impose-and-forward {
      description
        "Operation to impose one or more outgoing labels and
         forward to the next hop.";
    }
    enum pop-and-forward {
      description
        "Operation to pop the incoming label and forward to the
         next hop.";
```

```
    }
    enum pop-impose-and-forward {
      description
        "Operation to pop the incoming label, impose one or more
         outgoing labels, and forward to the next hop.";
    }
    enum swap-and-forward {
      description
        "Operation to swap the incoming label with the outgoing
         label and forward to the next hop.";
    }
    enum pop-and-lookup {
      description
        "Operation to pop the incoming label and perform
         a lookup.";
    }
  }
  description
    "Types of MPLS operations.";
}

typedef nhlfe-role {
  type enumeration {
    enum primary {
      description
        "The next hop acts as the primary for carrying traffic.";
    }
    enum backup {
      description
        "The next hop acts as the backup.";
    }
    enum primary-and-backup {
      description
        "The next hop simultaneously acts as both the primary and
         the backup for carrying traffic.";
    }
  }
  description
    "Role of the next hop.";
}

grouping nhlfe-single-contents {
  description
    "A grouping that describes a single Next Hop Label Forwarding
     Entry (NHLFE) and its associated parameters as described in
     the MPLS architecture.  This grouping is specific to the case
     when a single next hop is associated with the route.";
  uses rt-types:mpls-label-stack;
}

grouping nhlfe-multiple-contents {
  description
    "A grouping that describes a set of NHLFEs and their
     associated parameters as described in the MPLS
     architecture.  This grouping is used when multiple next hops
     are associated with the route.";
  leaf index {
    type string;
    description
      "A user-specified identifier utilized to uniquely
       reference the next-hop entry in the next-hop list.
       The value of this index has no semantic meaning
       other than for referencing the entry.";
  }
  leaf backup-index {
    type string;
```

```
    description
      "A user-specified identifier utilized to uniquely
       reference the backup next-hop entry in the NHLFE list.
       The value of this index has no semantic meaning
       other than for referencing the entry.";
    reference
      "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
       RFC 5714: IP Fast Reroute Framework";
  }
  leaf loadshare {
    type uint16;
    default "1";
    description
      "This value is used to compute a load share to perform
       unequal load balancing when multiple outgoing next hops are
       specified.  A share is computed as a ratio of this number to
       the total under all next hops.";
    reference
      "RFC 3031: Multiprotocol Label Switching Architecture,
       Sections 3.11 and 3.12
       RFC 7424: Mechanisms for Optimizing Link Aggregation Group
       (LAG) and Equal-Cost Multipath (ECMP) Component Link
       Utilization in Networks, Section 5.4";
  }
  leaf role {
    type nhlfe-role;
    description
      "Role of the NHLFE.";
  }
  uses nhlfe-single-contents;
}

grouping interfaces-mpls {
  description
    "List of MPLS interfaces.";
  container interfaces {
    description
      "List of MPLS-enabled interfaces.";
    list interface {
      key "name";
      description
        "MPLS-enabled interface entry.";
      leaf name {
        type if:interface-ref;
        description
          "A reference to the name of an interface in the system
           that is to be enabled for MPLS.";
      }
      leaf mpls-enabled {
        type boolean;
        default "false";
        description
          "'true' if MPLS encapsulation is enabled on the
           interface.
           'false' if MPLS encapsulation is disabled on the
           interface.";
      }
      leaf maximum-labeled-packet {
        type uint32;
        units "octets";
        description
          "Maximum labeled packet size.";
        reference
          "RFC 3032: MPLS Label Stack Encoding, Section 3.2";
      }
    }
```

```
  }
}

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

grouping label-blocks {
  description
    "Label-block allocation grouping.";
  container mpls-label-blocks {
    description
      "Label-block allocation container.";
    list mpls-label-block {
      key "index";
      description
        "List of MPLS label blocks.";
      leaf index {
        type string;
        description
          "A user-specified identifier utilized to uniquely
           reference an MPLS label block.";
      }
      leaf start-label {
        type rt-types:mpls-label;
        must '. <= ../end-label' {
          error-message "'start-label' must be less than or equal "
                      + "to 'end-label'";
        }
        description
          "Label-block start.";
      }
      leaf end-label {
        type rt-types:mpls-label;
        must '. >= ../start-label' {
          error-message "'end-label' must be greater than or "
                      + "equal to 'start-label'";
        }
        description
          "Label-block end.";
      }
      leaf block-allocation-mode {
        type identityref {
          base label-block-alloc-mode;
        }
        description
          "Label-block allocation mode.";
      }
      leaf inuse-labels-count {
        when "derived-from-or-self(../block-allocation-mode, "
           + "'mpls:label-block-alloc-mode-manager')";
        type yang:gauge32;
        config false;
        description
          "Number of labels in use in the label block.";
      }
    }
  }
}
```

```
grouping rib-mpls-properties {
  description
    "A grouping of native MPLS RIB properties.";
  leaf destination-prefix {
    type leafref {
      path "../mpls-local-label";
    }
    description
      "MPLS destination prefix.";
  }
  leaf route-context {
    type string;
    description
      "A context associated with the native MPLS route.";
  }
}

grouping rib-active-route-mpls-input {
  description
    "A grouping applicable to native MPLS RIB 'active-route'
     RPC input augmentation.";
  leaf destination-address {
    type leafref {
      path "../mpls-local-label";
    }
    description
      "MPLS native 'active-route' destination.";
  }
  leaf mpls-local-label {
    type rt-types:mpls-label;
    description
      "MPLS local label.";
  }
}

augment "/rt:routing" {
  description
    "MPLS augmentation.";
  container mpls {
    description
      "MPLS container to be used as an augmentation target node
       for the configuration of other MPLS sub-features, e.g.,
       MPLS static Label Switched Paths (LSPs), MPLS LDP LSPs,
       and Traffic Engineering MPLS LSP Tunnels.";
    uses globals;
    uses label-blocks;
    uses interfaces-mpls;
  }
}

/* Augmentation of MPLS routes */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
  description
    "This augmentation is applicable to all MPLS routes.";
  leaf mpls-enabled {
    type boolean;
    default "false";
    description
      "Indicates whether MPLS is enabled for this route.";
  }
  leaf mpls-local-label {
    when "../mpls-enabled = 'true'";
    type rt-types:mpls-label;
    description
```

```
      "MPLS local label associated with the route.";
  }
  uses rib-mpls-properties {
    /* MPLS Address Family (AF) augmentation to the
       native MPLS RIB */
    when "derived-from-or-self(../../rt:address-family, "
       + "'mpls:mpls')" {
      description
        "This augment is valid only for routes of the native MPLS
         RIB.";
    }
  }
}

/* MPLS simple-next-hop augmentation */

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

/* MPLS next-hop-list augmentation */

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

/* MPLS RPC input augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" {
  description
    "Input MPLS augmentation for the 'active-route' action
     statement.";
  uses rib-active-route-mpls-input {
    /* MPLS AF augmentation to the native MPLS RIB */
    when "derived-from-or-self(../rt:address-family, "
       + "'mpls:mpls')" {
      description
        "This augment is valid only for routes of the native MPLS
         RIB.";
    }
  }
}

/* MPLS RPC output augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
      + "rt:output/rt:route/"
      + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  description
    "Output MPLS augmentation for the 'active-route' action
     statement.";
  uses nhlfe-single-contents;
```

```
  }

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

Figure 4: MPLS Base YANG Module

## 3. IANA Considerations

This document registers the following URI in the "ns" subregistry of the "IETF XML Registry" [RFC3688].

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

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

Name: ietf-mpls
Namespace: urn:ietf:params:xml:ns:yang:ietf-mpls
Prefix: mpls
Reference: RFC 8960

## 4. Security Considerations

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

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

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

"/rt:routing/mpls:mpls/mpls:label-blocks":
   There are data nodes under this path that are writable, such as "start-label" and "end-label". Write operations to those data nodes may result in disruption to existing traffic.

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

"/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop/
rt:next-hop-options/rt:next-hop-list/rt:next-hop-list/rt:next-hop"
and "/rt:routing/rt:ribs/rt:rib/rt:active-
route/rt:output/rt:route/rt:next-hop/rt:next-hop-options/
rt:simple-next-hop":
   These two paths are augmented by additional MPLS leafs defined in
   this model.  Access to this information may disclose the next-hop
   information for the prefix route and/or other information.

Some of the RPC operations in this YANG module may be considered
sensitive or vulnerable in some network environments.  It is thus
important to control access to these operations.  These are the
operations and their sensitivity/vulnerability:

"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" and
"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route":
   These two paths are augmented by additional MPLS data nodes that
   are defined in this model.  Access to those paths may disclose
   information about per-prefix routes and/or other information; such
   disclosure may be used for further attacks.

The security considerations spelled out in [RFC3031] and [RFC3032]
apply for this document as well.

# 5. References

## 5.1. Normative References


[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y.,
Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack
Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001,
<https://www.rfc-editor.org/info/rfc3032>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,
<https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,
<https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
<https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
<https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types",
RFC 6991, DOI 10.17487/RFC6991, July 2013,
<https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
RFC 7950, DOI 10.17487/RFC7950, August 2016,
<https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
<https://www.rfc-editor.org/info/rfc8040>.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[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>.

## 5.2. Informative References

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

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <https://www.rfc-editor.org/info/rfc5714>.

[RFC7424] Krishnan, R., Yong, L., Ghanwani, A., So, N., and B. Khasnabish, "Mechanisms for Optimizing Link Aggregation Group (LAG) and Equal-Cost Multipath (ECMP) Component Link Utilization in Networks", RFC 7424, DOI 10.17487/RFC7424, January 2015, <https://www.rfc-editor.org/info/rfc7424>.

[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 Instance Example

A simple network setup is shown in Figure 5. R1 runs the IS-IS routing protocol and learns about the reachability of two IPv4 prefixes (P1: 198.51.100.1/32 and P2: 198.51.100.2/32) and two IPv6

prefixes (P3: 2001:db8:0:10::1/128 and P4: 2001:db8:0:10::2/128). We also assume that R1 learns about local and remote MPLS label bindings for each prefix using IS-IS (e.g., using Segment Routing (SR) extensions).

```
State on R1:
============
    IPv4 Prefix           MPLS Label
P1: 198.51.100.1/32       16001
P2: 198.51.100.2/32       16002

    IPv6 Prefix           MPLS Label
P3: 2001:db8:0:10::1/128  16003
P4: 2001:db8:0:10::2/128  16004

RSVP MPLS LSPv4-Tunnel:
 Source:        198.51.100.3
 Destination:   198.51.100.4
 Tunnel-ID:     10
 LSP-ID:        1
                                192.0.2.5/30
                                2001:db8:0:1::1/64
                               eth0
                               +---
                              /
                        +-----+
                        | R1  |
                        +-----+
                              \
                               +---
                               eth1
                                192.0.2.13/30
                                2001:db8:0:2::1/64
```

Figure 5: Example of Network Configuration

The instance data tree could then be illustrated as shown in Figure 6, using JSON format [RFC7951]:

```
{
  "ietf-routing:routing":{
    "ribs":{
      "rib":[
        {
          "name":"RIB-V4",
          "address-family":
          "ietf-ipv4-unicast-routing:v4ur:ipv4-unicast",
          "routes":{
            "route":[
              {
                "next-hop":{
                  "outgoing-interface":"eth0",
                  "ietf-mpls:mpls-label-stack":{
                    "entry":[
                      {
                        "id":1,
                        "label":16001,
                        "ttl":255
                      }
                    ]
                  },
                  "ietf-ipv4-unicast-routing:next-hop-address":
                  "192.0.2.5"
                },
                "source-protocol":"ietf-isis:isis",
                "ietf-mpls:mpls-enabled":true,
```

```
        "ietf-mpls:mpls-local-label":16001,
        "ietf-ipv4-unicast-routing:destination-prefix":
        "198.51.100.1/32",
        "ietf-mpls:route-context":"SID-IDX:1"
      },
      {
        "next-hop":{
          "next-hop-list":{
            "next-hop":[
              {
                "outgoing-interface":"eth0",
                "ietf-mpls:index":"1",
                "ietf-mpls:backup-index":"2",
                "ietf-mpls:role":"primary-and-backup",
                "ietf-mpls:mpls-label-stack":{
                  "entry":[
                    {
                      "id":1,
                      "label":16002,
                      "ttl":255
                    }
                  ]
                },
                "ietf-ipv4-unicast-routing:address":
                "192.0.2.5"
              },
              {
                "outgoing-interface":"eth1",
                "ietf-mpls:index":"2",
                "ietf-mpls:backup-index":"1",
                "ietf-mpls:role":"primary-and-backup",
                "ietf-mpls:mpls-label-stack":{
                  "entry":[
                    {
                      "id":1,
                      "label":16002,
                      "ttl":255
                    }
                  ]
                },
                "ietf-ipv4-unicast-routing:address":
                "192.0.2.13"
              }
            ]
          }
        },
        "source-protocol":"ietf-isis:isis",
        "ietf-mpls:mpls-enabled":true,
        "ietf-mpls:mpls-local-label":16002,
        "ietf-ipv4-unicast-routing:destination-prefix":
        "198.51.100.2/32",
        "ietf-mpls:route-context":"SID-IDX:2"
      }
    ]
  }
},
{
  "name":"RIB-V6",
  "address-family":
  "ietf-ipv6-unicast-routing:v6ur:ipv6-unicast",
  "routes":{
    "route":[
      {
        "next-hop":{
          "outgoing-interface":"eth0",
          "ietf-mpls:mpls-label-stack":{
```

```
      "entry":[
        {
          "id":1,
          "label":16003,
          "ttl":255
        }
      ]
    },
    "ietf-ipv6-unicast-routing:next-hop-address":
    "2001:db8:0:1::1"
  },
  "source-protocol":"ietf-isis:isis",
  "ietf-mpls:mpls-enabled":true,
  "ietf-mpls:mpls-local-label":16003,
  "ietf-ipv6-unicast-routing:destination-prefix":
  "2001:db8:0:10::1/128",
  "ietf-mpls:route-context":"SID-IDX:3"
},
{
  "next-hop":{
    "next-hop-list":{
      "next-hop":[
        {
          "outgoing-interface":"eth0",
          "ietf-mpls:index":"1",
          "ietf-mpls:backup-index":"2",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16004,
                "ttl":255
              }
            ]
          },
          "ietf-ipv6-unicast-routing:address":
          "2001:db8:0:1::1"
        },
        {
          "outgoing-interface":"eth1",
          "ietf-mpls:index":"2",
          "ietf-mpls:backup-index":"1",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16004,
                "ttl":255
              }
            ]
          },
          "ietf-ipv6-unicast-routing:address":
          "2001:db8:0:2::1"
        }
      ]
    }
  },
  "source-protocol":"ietf-isis:isis",
  "ietf-mpls:mpls-enabled":true,
  "ietf-mpls:mpls-local-label":16004,
  "ietf-ipv6-unicast-routing:destination-prefix":
  "2001:db8:0:10::2/128",
  "ietf-mpls:route-context":"SID-IDX:4"
}
```

```
            ]
          }
        },
        {
          "name":"RIB-MPLS",
          "address-family":"ietf-mpls:mpls:mpls",
          "routes":{
            "route":[
              {
                "next-hop":{
                  "outgoing-interface":"eth0",
                  "ietf-mpls:mpls-label-stack":{
                    "entry":[
                      {
                        "id":1,
                        "label":24002,
                        "ttl":255
                      }
                    ]
                  },
                  "ietf-ipv4-unicast-routing:next-hop-address":
                  "192.0.2.5"
                },
                "source-protocol":"ietf-rsvp:rsvp",
                "ietf-mpls:mpls-enabled":true,
                "ietf-mpls:mpls-local-label":24001,
                "ietf-mpls:destination-prefix":"24001",
                "ietf-mpls:route-context":
                "RSVP Src:198.51.100.3,Dst:198.51.100.4,T:10,L:1"
              }
            ]
          }
        }
      ]
    },
    "ietf-mpls:mpls":{
      "mpls-label-blocks":{
        "mpls-label-block":[
          {
           "index":"mpls-srgb-label-block",
           "start-label":16000,
           "end-label":16500,
           "block-allocation-mode":
           "ietf-mpls:label-block-alloc-mode-manager"
          }
        ]
      },
      "interfaces":{
        "interface":[
          {
            "name":"eth0",
            "mpls-enabled":true,
            "maximum-labeled-packet":1488
          },
          {
            "name":"eth1",
            "mpls-enabled":true,
            "maximum-labeled-packet":1488
          }
        ]
      }
    }
  }
}
```


Figure 6: Instance Data Tree Example

## Acknowledgments

The authors would like to thank Xia Chen for her contributions to the early draft revisions of this document.

## Contributors

Igor Bryskin
Huawei Technologies

Email: i_bryskin@yahoo.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

## Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Kamran Raza
Cisco Systems, Inc.

Email: skraza@cisco.com

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net