MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2021

T. Saad
Juniper Networks
K. Raza
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
October 26, 2020

A YANG Data Model for MPLS Base
draft-ietf-mpls-base-yang-17

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 models) will augment the MPLS base YANG data model.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 29, 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction 2
  1.1. Terminology 3
  1.2. Acronyms and Abbreviations 3
2. MPLS Base Model 4
  2.1. Model Overview 4
  2.2. Model Organization 4
  2.3. Model Design 6
  2.4. Model Tree Diagram 8
  2.5. Model YANG Module 9
3. IANA Considerations 19
4. Security Considerations 20
5. Acknowledgement 21
6. Appendix A. Data Tree Instance Example 21
7. Contributors 27
8. References 27
  8.1. Normative References 27
  8.2. Informative References 29
Authors' Addresses 29

1. Introduction

A core routing YANG data model is defined in [RFC8349], and 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 Address-Family (AF) agnostic. [RFC8349] also defines two instances of RIBs based on the generic RIB model for IPv4 and IPv6 AFs.

The MPLS base model that is defined in this document augments the generic RIB model defined in [RFC8349] with additional data that enables MPLS forwarding for the specific destination prefix(es) present in the AF RIB(s) 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 route(s) that are not associated with other AF instance RIBs (such as IPv4, or IPv6 instance RIB(s)),

but are enabled for MPLS forwarding. Examples of such native MPLS routes are routes programmed by RSVP on transit MPLS router(s) along the path of a Label Switched Path (LSP). Other example(s) are MPLS routes that cross-connect to specific Layer-2 adjacencies, such as Layer-2 Attachment Circuit(s) (ACs)), or Layer-3 adjacencies, such as Segment-Routing (SR) Adjacency Segments (Adj-SIDs) described in [RFC8402].

The MPLS base YANG data model serves as a basis for future development of MPLS YANG data models covering more-sophisticated MPLS feature(s) and sub-system(s). 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 at IETF (e.g. by TEAS and MPLS working groups).

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

1.1. Terminology

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

1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

RIB: Routing Information Base

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

ILM: Incoming Label Map

2. MPLS Base Model

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

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 it for MPLS forwarding.

This document also defines a new instance of the generic RIB defined in [RFC8349] to store native MPLS route(s) (described further in Section 2.3) by extending the identity 'address-family' defined in [RFC8349] with a new "mpls" identity as suggested in Section 3 of [RFC8349].

2.2. Model Organization

```
  Routing          +---------------+    v: import
  YANG module      | ietf-routing  |    o: augment
                   +---------------+
                       o
                       |
                       v
  MPLS base        +-----------+    v: import
  YANG module      | ietf-mpls |    o: augment
                   +-----------+
                      o      o------+
                      |              \
                      v               v
             +-------------------+ +---------------------+
  MPLS Static | ietf-mpls-static@ | | ietf-mpls-ldp.yang@ | . .
  LSP YANG   +-------------------+ +---------------------+
  module
        @: not in this document, shown for illustration only
```

Figure 1: Relationship between MPLS modules

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

mpls:

This identity extends the 'address-family' identity for RIB instance(s) identity as defined in [RFC8349] to represent the native MPLS RIB instance.

label-block-alloc-mode:

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

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 NHLFE entry.

nhlfe-single-contents:

A YANG grouping that describes single Next Hop Label Forwarding Entry (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 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 label 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 label(s) on an unlabeled packet

It may also contain:

d) the data link encapsulation to use when transmitting the packet

e) the way to encode the label stack when transmitting the packet

f) any other information needed in order to properly dispose of the packet.

nhlfe-multiple-contents:

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

```
                              +-----------------+
                              | MPLS base model |
                              +-----------------+
                            ____/  |  |_____  |________
                           /       |        \          \
                          /        |         \          \
                         o         o          o          +
                   +---------+  +---------+  +--------+ +-----------+
                   | RIB(v4) |  | RIB(v6) |  | RIB(x) | | RIB(mpls) |
                   +---------+  +---------+  +--------+ +-----------+


           +: 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 instance(s) of AF RIB(s) with additional data that enables MPLS forwarding for destination prefix(es) store in such RIB(s). For example, an IPv4 prefix stored in RIB(v4) is augmented to carry a MPLS local label and per next-hop remote label(s) to enable MPLS forwarding for such prefix.

The MPLS base model also creates a separate instance of the generic RIB model defined in [RFC8349] to store MPLS native route(s) that are enabled for MPLS forwarding, but not stored in other AF RIB(s).

Some examples of such native MPLS routes are:

- o routes programmed by RSVP on Label Switched Router(s) (LSRs) along the path of a Label Switched Path (LSP),

- o routes that cross-connect an MPLS local label to a Layer-2, or Layer-3 VRF,

- o routes that cross-connect an MPLS local label to a specific Layer-2 adjacency or interface, such as Layer-2 Attachment Circuit(s) (ACs), or

- o 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), SR MPLS Binding SIDs, etc. as defined in [RFC8402].

2.4. Model Tree Diagram

The MPLS base tree diagram that 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. Model YANG Module

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

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

o ietf-routing defined in [RFC8349]

o ietf-routing-types defined in [RFC8294]

o ietf-interfaces defined in [RFC8343]

This YANG module also references the following RFCs in defining the types and YANG grouping of the YANG module: [RFC3032], [RFC3031], and [RFC7424].

```
<CODE BEGINS> file "ietf-mpls@2020-10-26.yang"
module ietf-mpls {
  yang-version 1.1;
```

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

  /* Replace with IANA when assigned */

  prefix mpls;

  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC8294:Common YANG Data Types for the Routing Area";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC6991: Common YANG Data Types";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC8343: A YANG Data Model for Interface Management";
  }

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

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

     Editor:   Tarek Saad
               <mailto:tsaad@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) 2018 IETF Trust and the persons
     identified as authors of the code.  All rights reserved.

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

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

  revision 2020-10-26 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for base MPLS";
  }

  /* Identities */

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

  identity 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 reserved block
       is managed by label manager.";
  }

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

  /**
   * Typedefs
   */

  typedef mpls-operations-type {
    type enumeration {
      enum impose-and-forward {
        description
          "Operation impose outgoing label(s) and forward to
           next-hop.";
      }
      enum pop-and-forward {
        description
          "Operation pop incoming label and forward to next-hop.";
      }
      enum pop-impose-and-forward {
        description
          "Operation pop incoming label, impose one or more
           outgoing label(s) and forward to next-hop.";
      }
      enum swap-and-forward {
        description
          "Operation swap incoming label, with outgoing label and
           forward to next-hop.";
      }
      enum pop-and-lookup {
        description
          "Operation pop incoming label and perform a lookup.";
      }
    }
    description
      "MPLS operations types.";
  }
```

```
  typedef nhlfe-role {
    type enumeration {
      enum primary {
        description
          "Next-hop acts as primary for carrying traffic.";
      }
      enum backup {
        description
          "Next-hop acts as backup.";
      }
      enum primary-and-backup {
        description
          "Next-hop acts as primary and backup simultaneously
           for carry traffic.";
      }
    }
    description
      "The next-hop role.";
  }

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

```
        The value of this index has no semantic meaning
        other than for referencing the entry.";
      reference
        "RFC4090 and RFC5714";
    }
    leaf loadshare {
      type uint16;
      default "1";
      description
        "This value is used to compute a loadshare to perform un-equal
         load balancing when multiple outgoing next-hop(s) are
         specified. A share is computed as a ratio of this number to the
         total under all next-hops(s).";
      reference
        "RFC7424, section 5.4,
         RFC3031, section 3.11 and 3.12.";
    }
    leaf role {
      type nhlfe-role;
      description
        "NHLFE role.";
    }
    uses nhlfe-single-contents;
  }

  grouping interfaces-mpls {
    description
      "List of MPLS interfaces.";
    container interfaces {
      description
        "List of MPLS enabled interaces.";
      list interface {
        key "name";
        description
          "MPLS enabled interface entry.";
        leaf name {
          type if:interface-ref;
          description
            "A reference to the name of a 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
            "RFC3032, 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 utilised to uniquely
             reference an MPLS label block.";
        }
        leaf start-label {
          type rt-types:mpls-label;
          must '. <= ../end-label' {
            error-message
              "The start-label must be less than or equal "
            + "to end-label";
          }
          description
```

```
            "Label-block start.";
        }
        leaf end-label {
          type rt-types:mpls-label;
          must '. >= ../start-label' {
            error-message
              "The end-label must be greater than or equal "
            + "to start-label";
          }
          description
            "Label-block end.";
        }
        leaf block-allocation-mode {
          type identityref {
            base label-block-alloc-mode;
          }
          description
            "Label-block allocation mode.";
        }
        leaf inuse-labels-count {
          when "derived-from-or-self(../block-allocation-mode, "
             + "'mpls:label-block-alloc-mode-manager')";
          type yang:gauge32;
          config false;
          description
            "Label-block inuse labels count.";
        }
      }
    }
  }

  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
         other MPLS sub-features config, e.g. MPLS static LSP, MPLS
         LDP LSPs, and Trafic Engineering MPLS LSP Tunnels, etc.";
      uses globals;
      uses label-blocks;
      uses interfaces-mpls;
    }
  }

  /* MPLS routes augmentation */

  augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
    description
      "This 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 AF augmentation to native MPLS RIB */
      when "derived-from-or-self(../../rt:address-family, "
         + "'mpls:mpls')" {
        description
          "This augment is valid only for routes of 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
      "Augment 'simple-next-hop' case in IP unicast routes.";
    uses nhlfe-single-contents {
      when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route"
         + "/mpls:mpls-enabled = 'true'";
    }
  }

  /* MPLS next-hop-list augmentation */

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

  /* MPLS RPC input augmentation */

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

```
        description
          "This augment is valid only for routes of 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 URIs in the 'ns' sub-registry of the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

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

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

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

4. Security Considerations

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

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

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 writeable such as 'start-label' and 'end-label'. Write operations to those data npdes may cause disruptive action to existing traffic.

Some of the readable data nodes in these 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 leaf(s) defined in this model. Access to this information may disclose the next-hop or path per prefix 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 node(s) that are defined in this model. Access to those path(s) may may disclose information about per prefix route and/or other information and that may be further used for further attack(s).

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

5. Acknowledgement

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

6. Appendix A. Data Tree Instance Example

A simple network setup is shown in Figure 5. R1 runs the ISIS routing protocol, and learns reachability about two IPv4 prefixes: P1: 198.51.100.1/32 and P2: 198.51.100.1/32, and two IPv6 prefixes P3: 2001:db8:0:10::1/64 and P4: 2001:db8:0:10::1/64. We also assume that R1 learns about local and remote MPLS label bindings for each prefix using ISIS (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::1/64 16003
P4: 2001:db8:0:10::2/64 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 as follows:


```
{
  "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":"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":"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":"isis:isis",
            "ietf-mpls:mpls-enabled":true,
            "ietf-mpls:mpls-local-label":16001,
            "ietf-ipv6-unicast-routing:destination-prefix":
            "2001:db8:0:10::1/6",
            "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":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":"isis:isis",
                "ietf-mpls:mpls-enabled":true,
                "ietf-mpls:mpls-local-label":16004,
                "ietf-ipv6-unicast-routing:destination-prefix":
                "2001:db8:0:10::2/64",
                "ietf-mpls:route-context":"SID-IDX:2"
              }
            ]
```

```
          }
        },
        {
          "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":"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":"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: Foo bar.

7. Contributors

Igor Bryskin
Huawei Technologies
email: i_bryskin@yahoo.com

Himanshu Shah
Ciena
email: hshah@ciena.com

8. References

8.1. 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>.

[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>.

8.2. 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>.

[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>.

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