MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2018

T. Saad
K. Raza
R. Gandhi
Cisco Systems Inc
X. Liu
Jabil
V. Beeram
Juniper Networks
H. Shah
Ciena
I. Bryskin
X. Chen
Huawei Technologies
R. Jones
Brocade
B. Wen
Comcast

July 02, 2017

A YANG Data Model for MPLS Base
draft-ietf-mpls-base-yang-05

Abstract

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

Table of Contents

1. Introduction

A core routing data model is defined in [RFC8022], and it provides a basis for the development of data models for routing protocols. The MPLS base model augments this model with additional data specific to MPLS switching [RFC3031]. The interface data model is defined in [RFC7223] and is used for referencing interface from the MPLS base model.

The MPLS base YANG module augments the "routing" read-write (rw) and "routing-state" read-only (ro) branches of the ietf-routing module defined in [RFC8022]. The approach described in [I-D.openconfig-netmod-opstate] is adopted to represent data pertaining to configuration intended, applied state and derived state data elements. Each container in the model holds a "config" and "state" sub-container. The "config" sub-container contains the intended configuration data, and the state sub-container contains both the applied configuration and any derived state, such as counters or statistical information.

This document defines the specification for the "ietf-mpls" YANG module that provides base components of the MPLS data model. It is expected that other MPLS YANG modules will augment the "ietf-mpls" base model to define data models for other MPLS technologies (e.g. MPLS LDP or MPLS RSVP-TE).

This document also defines a way to model MPLS labelled routes as an augmentation of the the routing RIB model defined in [RFC8022] for IP prefix routes that are MPLS labelled. Other MPLS non-IP prefix routes are also modelled by introducing a new "mpls" address-family RIB.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 RFC2119}}.

1.1.1. Keywords

The following terms are defined in [RFC6020]:

o augment,

o configuration data,

o data model,

o data node,

o feature,

o mandatory node,

o module,

o schema tree,

o state data,

o RPC operation.

1.2. MPLS Base Tree Diagram

The MPLS base tree diagram is shown in Figure 1.

module: ietf-mpls

```
augment /rt:routing:
  +--rw mpls
     +--rw config
     |  +--rw ttl-propagate?   boolean
     +--ro state
     |  +--ro ttl-propagate?   boolean
     +--rw label-blocks
     |  +--rw label-block* [index]
     |     +--rw index          -> ../config/index
     |     +--rw start-label?   -> ../config/start-label
     |     +--rw end-label?     -> ../config/end-label
     |     +--rw config
     |     |  +--rw index?                   string
     |     |  +--rw start-label?             rt-types:mpls-label
     |     |  +--rw end-label?               rt-types:mpls-label
     |     |  +--rw block-allocation-mode?   identityref
     |     +--ro state
     |        +--ro index?                   string
     |        +--ro start-label?             rt-types:mpls-label
     |        +--ro end-label?               rt-types:mpls-label
     |        +--ro block-allocation-mode?   identityref
     |        +--ro free-labels-count?       uint32
     |        +--ro inuse-labels-count?      uint32
     +--rw interface* [name]
        +--rw name      if:interface-ref
        +--rw config
        |  +--rw enabled?   boolean
        |  +--rw mtu?       uint32
        +--ro state
           +--ro enabled?   boolean
           +--ro mtu?       uint32
augment /rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route:
  +--ro local-label?   rt-types:mpls-label
augment /rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route/
rt:next-hop/rt:next-hop-options/rt:simple-next-hop:
  +--ro remote-labels* [index]
     +--ro index    uint8
     +--ro label?   rt-types:mpls-label
augment /rt:routing-state/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 remote-labels* [index]
     +--ro index    uint8
     +--ro label?   rt-types:mpls-label
```

```
    augment /rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input:
      +---- index?           string
      +---- backup-index?    string
      +---- loadshare?       uint16
      +---- role?            nhlfe-role
      +---- remote-labels* [index]
         +---- index    uint8
         +---- label?   rt-types:mpls-label
    augment /rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:output/
    rt:route:
      +---- index?           string
      +---- backup-index?    string
      +---- loadshare?       uint16
      +---- role?            nhlfe-role
      +---- remote-labels* [index]
         +---- index    uint8
         +---- label?   rt-types:mpls-label
```

Figure 1: MPLS Base tree diagram

1.3. MPLS Base Module

```
<CODE BEGINS> file "ietf-mpls@2017-07-02.yang"
module ietf-mpls {

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

  prefix "mpls";

  import ietf-routing {
    prefix "rt";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

  organization "IETF MPLS Working Group";

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

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

```
     WG Chair: Loa Andersson
               <mailto:loa@pi.nu>

     WG Chair: Ross Callon
               <mailto:rcallon@juniper.net>

     WG Chair: George Swallow
               <mailto:swallow.ietf@gmail.com>

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

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

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

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

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

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

     Editor:   Igor Bryskin
               <mailto: Igor.Bryskin@huawei.com>

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

     Editor:   Raqib Jones
               <mailto:raqib@Brocade.com>

     Editor:   Bin Wen
               <mailto:Bin_Wen@cable.comcast.com>";

  description
    "This YANG module defines the essential components for the
     management of the MPLS subsystem.";

  revision "2017-07-02" {
    description
      "Latest revision:
        - Addressed MPLS-RT review comments";
    reference "RFC 3031: A YANG Data Model for base MPLS";
```

```
  }

  /* Identities */

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

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

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

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

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

  grouping nhlfe-simple-contents {
```

```
    description
      "MPLS simple NHLFE contents";
    list remote-labels {
      key "index";
      description "Remote label list";
      leaf index {
        type uint8 {
          range "0..255";
        }
        description
          "Index of the label. Index 0 indicates
           top of the label stack";
      }
      leaf label {
        type rt-types:mpls-label;
        description
          "The outgoing MPLS labels to impose";
      }
    }
  }

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

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

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

```
        total under all next-hops(s).";
    }

    leaf role {
      type nhlfe-role;
      description "NHLFE role";
    }
    uses nhlfe-simple-contents;
  }

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

  grouping interfaces-mpls {
    description "List of MPLS interfaces";
    list interface {
      key "name";
      description "List of MPLS interfaces";
      leaf name {
        type if:interface-ref;
        description
          "The name of a configured MPLS interface";
      }
      container config {
        description "Holds intended configuration";
        uses interface-mpls_config;
      }
      container state {
        config false;
        description "Holds inuse configuration";
        uses interface-mpls_config;
      }
    }
  }
```

```
  grouping label-block_config {
    description "Label-block configuration items";
    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 start-label {
      type rt-types:mpls-label;
      description "Label-block start";
    }
    leaf end-label {
      type rt-types:mpls-label;
      description "Label-block end";
    }
    leaf block-allocation-mode {
      type identityref {
        base label-block-alloc-mode;
      }
      description "Label-block allocation mode";
    }
  }

  grouping label-block_state {
    description "Label-block state items";
    leaf free-labels-count {
      when
      "../block-allocation-mode = 'label-block-alloc-mode-manager'";
      type uint32;
      description "Label-block free labels count";
    }
    leaf inuse-labels-count {
      when
      "../block-allocation-mode = 'label-block-alloc-mode-manager'";
      type uint32;
      description "Label-block inuse labels count";
    }
  }

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

```
    }
  }

  grouping globals {
    description "MPLS global configuration grouping";
    container config {
      description "Holds intended configuration";
      uses globals_config;
    }
    container state {
      config false;
      description "Holds inuse configuration";
      uses globals_config;
    }
  }
  grouping label-blocks {
    description "Label-block allocation grouping";
    container label-blocks {
      description "Label-block allocation container";
      list label-block {
        must "start-label >= end-label" {
          error-message "start-label can not be less than end-label";
        }
        key index;
        unique "start-label end-label";
        leaf index {
          type leafref {
            path "../config/index";
          }
          description "Label-block index";
        }
        leaf start-label {
          type leafref {
            path "../config/start-label";
          }
          description
            "Label-block start label reference";
        }
        leaf end-label {
          type leafref {
            path "../config/end-label";
          }
          description
            "Label-block end label reference";
        }
        description "List of MPLS label-blocks";
        container config {
          description "Holds intended configuration";
```

```
          uses label-block_config;
        }
        container state {
          config false;
          description "Holds inuse configuration";
          uses label-block_config;
          uses label-block_state;
        }
      }
    }
  }

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

  /* State data */
  augment "/rt:routing-state/rt:ribs/rt:rib/rt:routes/rt:route" {
    description
      "This leaf augments an IPv4 unicast route.";
    leaf local-label {
      type rt-types:mpls-label;
      description
        "MPLS local label.";
    }
  }

  augment "/rt:routing-state/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 IPv4 unicast routes.";
    uses nhlfe-simple-contents;
  }

  augment "/rt:routing-state/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 IPv4 unicast
```

```
       routes.";
    uses nhlfe-contents;
  }

  augment
    "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/rt:input" {
    description
      "This augment adds the input parameter of the 'active-route'
       action.";
    uses nhlfe-contents;
  }

  augment "/rt:routing-state/rt:ribs/rt:rib/rt:active-route/"
        + "rt:output/rt:route" {
    description
      "This augment adds the destination prefix to the reply of the
       'active-route' action.";
    uses nhlfe-contents;
  }
}
<CODE ENDS>
```

Figure 2: MPLS base YANG module

2. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-mpls XML: N/A, the requested URI is an XML namespace.

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

name: ietf-mpls namespace: urn:ietf:params:xml:ns:yang:ietf-mpls prefix: ietf-mpls reference: RFC3031

3. Security Considerations

The YANG module defined in this document is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which 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.

4. References

4.1. Normative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <http://www.rfc-editor.org/info/rfc8022>.

4.2. Informative References

[I-D.openconfig-netmod-opstate]
Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling of Operational State Data in YANG", draft-openconfig-netmod-opstate-01 (work in progress), July 2015.

Authors' Addresses

Tarek Saad
Cisco Systems Inc

Email: tsaad@cisco.com

Kamran Raza
Cisco Systems Inc

Email: skraza@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Jabil

Email: Xufeng_Liu@jabil.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Himanshu Shah
Ciena

Email: tsaad@cisco.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com