MBONED WG
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2020

Z. Zhang
ZTE Corporation
C. Wang
Individual
Y. Cheng
China Unicom
X. Liu
Volta Networks
M. Sivakumar
Juniper networks

March 7, 2020

Multicast YANG Data Model
draft-ietf-mboned-multicast-yang-model-03

Abstract

This document provides a general multicast YANG data model, which takes full advantages of existed multicast protocol models to control the multicast network, and guides the deployment of multicast service.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 8, 2020.

Table of Contents

1. Introduction

Currently, there are many multicast protocol YANG models, such as PIM, MLD, and BIER and so on. But all these models are distributed in different working groups as separate files and focus on the protocol itself. Furthermore, they cannot describe a high-level multicast service required by network operators.

This document provides a general and all-round multicast model, which stands at a high level to take full advantages of these aforementioned models to control the multicast network, and guide the deployment of multicast service.

This model is designed to be used along with other multicast YANG models such as PIM [I-D.ietf-pim-yang], which are not covered in this document.

1.1. Terminology

The terminology for describing YANG data models is found in [RFC6020] and [RFC7950], including:

- augment
- data model
- data node
- identity
- module

The following abbreviations are used in this document and the defined model:

BIER: Bit Index Explicit Replication [RFC8279].

MLD: Multicast Listener Discovery [I-D.ietf-bier-mld].

PIM: Protocol Independent Multicast [RFC7761].

BGP: Border Gateway Protocol [RFC4271].

MVPN: Multicast in MPLS/BGP IP VPNs [RFC6513].

MLDP: Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths [RFC6388].

OSPF: Open Shortest Path First [RFC2328].

ISIS: Intermediate System to Intermediate System Routeing Exchange Protocol [RFC1195].

BABEL: [I-D.ietf-babel-rfc6126bis].

P2MP-TE: Point-to-Multipoint Traffic Engineering [RFC4875].

BIER-TE: Traffic Engineering for Bit Index Explicit Replication [I-D.ietf-bier-te-arch].

1.2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

1.4. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
rt	ietf-routing	[RFC8349]
ospf	ietf-ospf	[I-D.ietf-ospf-yang]

Table 1

1.5. Usage of Multicast Model

This multicast YANG data model is mainly used by the management tools run by the network operators, in order to manage, monitor and debug the network resources which are used to deliver multicast service. This model is used for gathering data from the network as well.

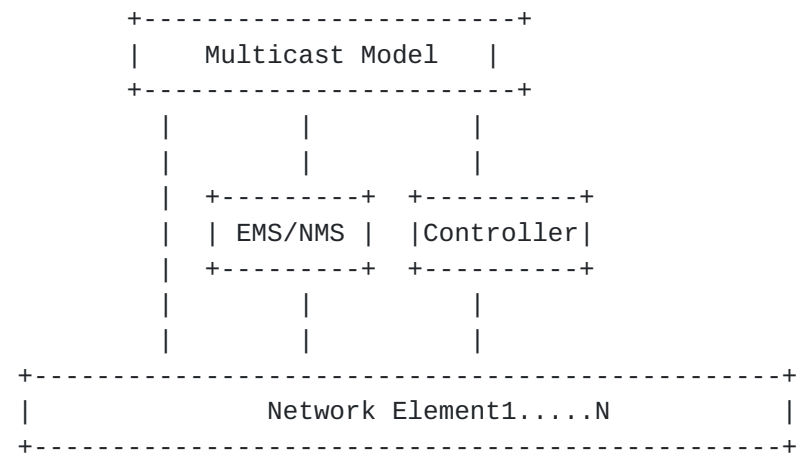

Figure 1: Usage of Multicast Model

Detailly, in figure 1, there is an example of usage of this multicast model. Network operators can use this model in a controller which is responsible to implement specific multicast flows with specific protocols and invoke the corresponding protocols' model to configure the network elements through NETCONF/RESTCONF/CLI. Or network operators can use this model to the EMS/NMS to manage the network elements or configure the network elements directly.

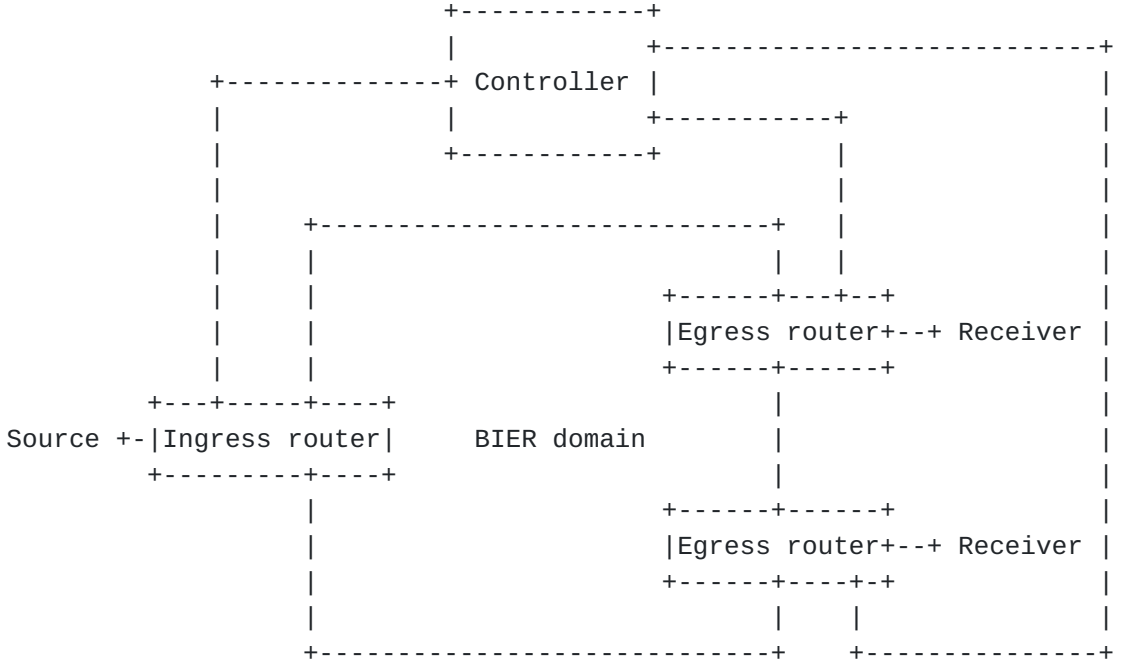

Figure 2: Example

The network administrator can use the multicast model and associated models to deploy the multicast service. For example, suppose that the flow for a multicast service is 233.252.0.0/16, the flow should be forwarded by BIER [RFC8279] with MPLS encapsulation [RFC8296]. Correspoding IGP protocol which is used to build BIER transport layer is OSPF [RFC2328].

In this model, the correspond key is set to 233.252.0.0/16, the transport technology is set to BIER. The BIER underlay protocol is set to OSPF. The model is sent to every egde router from the controller. If the BIER transport layer which depends on OSPF has not been built in the network, the multicast YANG model will invoke the BIER YANG model which is defined in [I-D.ietf-bier-bier-yang] generation in the controller. After the BIER transport layer is built, the ingress router encapsulates the multicast flow with BIER header and sends it into the network. Intermediate routers forward the flows to all the egress nodes by BIER forwarding.

On the other hand, when the network elements detect failure or some other changes, the network devices can send the affected multicast flows and the associated overlay/ transport/ underlay information to the controller. Then the controller/ EMS/NMS can response immediately due to the failure and distribute new model for the flows to the network nodes quickly. Such as the changing of the failure overlay protocol to another one, as well as transport and underlay protocol.

Specifically, in section 3, it provides a human readability of the whole multicast network through UML like class diagram, which frames different multicast components and correlates them in a readable fashion. Then, based on this UML like class diagram, there is instantiated and detailed YANG model in Section 5.

In other words, this document does not define any specific protocol model, instead, it depends on many existed multicast protocol models and relates several multicast information together to fulfill multicast service.

2. Design of the multicast model

2.1. Scope of Model

This model can be used to configure and manage Multicast service. The operational state data can be retrieved by this model. The subscription and push mechanism defined in [RFC8639] and [RFC8641] can be implemented by the user to subscribe to notifications on the data nodes in this model.

The model contains all the basic configuration parameters to operate the model. Depending on the implementation choices, some systems may not allow some of the advanced parameters to be configurable. The occasionally implemented parameters are modeled as optional features in this model. This model can be extended, and it has been structured in a way that such extensions can be conveniently made.

2.2. Specification

The configuration data nodes cover configurations. The container "multicast-model" is the top level container in this data model. The presence of this container is expected to enable Multicast service functionality. The notification includes the error reason and the associated data nodes.

3. Module Structure

This model imports and augments the ietf-routing YANG model defined in [RFC8349]. Both configuration data nodes and state data nodes of [RFC8349] are augmented.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407].

3.1. UML like Class Diagram for Multicast YANG data Model

The following is a UML like diagram for Multicast YANG data Model.

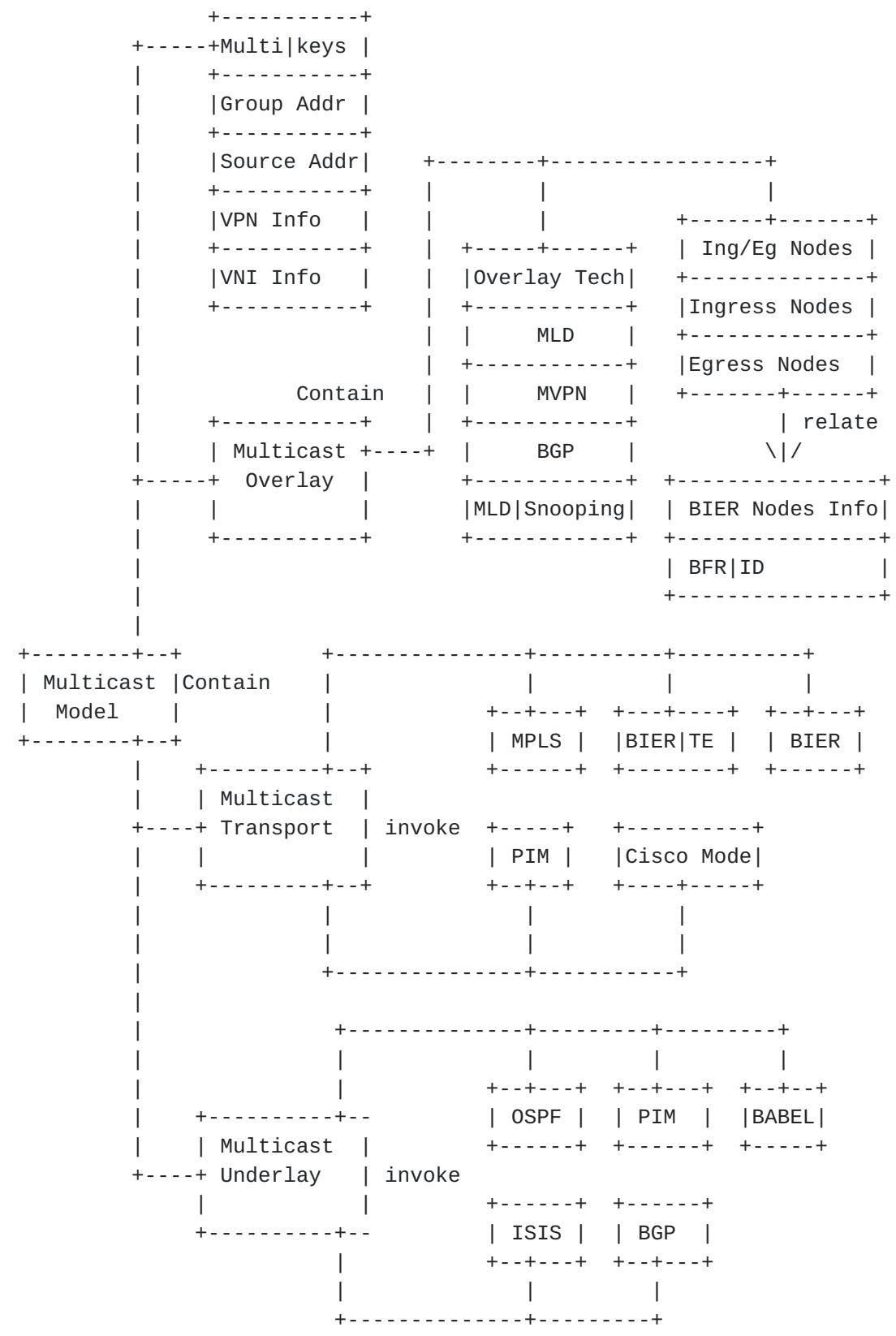

Figure 3: UML like Class Diagram for Multicast YANG data Model

3.2. Model Structure

```
 module: ietf-multicast-model
   +--rw multicast-model
      +--rw multicast-keys* [vpn-rd source-address group-address
                             vni-type vni-value]
         +--rw vpn-rd                 rt-types:route-distinguisher
         +--rw source-address         ip-multicast-source-address
         +--rw group-address
                 rt-types:ip-multicast-group-address
         +--rw vni-type               virtual-type
         +--rw vni-value              uint32
         +--rw multicast-overlay
         |  +--rw ingress-egress
         |  |  +--rw ingress-node?   inet:ip-address
         |  |  +--rw egress-nodes* [egress-node]
         |  |     +--rw egress-node    inet:ip-address
         |  +--rw bier-ids
         |  |  +--rw sub-domain?     uint16
         |  |  +--rw ingress-node?   uint16
         |  |  +--rw egress-nodes* [egress-node]
         |  |     +--rw egress-node    uint16
         |  +--rw (overlay-tech-type)?
         |     +--:(bgp)
         |     +--:(evpn)
         |     +--:(mld)
         |     |  +--rw mld-instance-group?
                          rt-types:ip-multicast-group-address
         |     +--:(mld-snooping)
         |     +--:(mvpn)
         |     +--:(pim)
         +--rw multicast-transport
         |  +--rw (transport)?
         |     +--:(bier)
         |     |  +--rw bier
         |     |     +--rw sub-domain?        uint16
         |     |     +--rw bitstringlength?   uint16
         |     |     +--rw set-identifier?    uint16
         |     |     +--rw (encap-type)?
         |     |        +--:(mpls)
         |     |        +--:(eth)
         |     |        +--:(ipv6)
         |     +--:(bier-te)
         |     |  +--rw bier-te
         |     |     +--rw sub-domain?        uint16
         |     |     +--rw bitstringlength?   uint16
         |     |     +--rw set-identifier?    uint16
         |     |     +--rw (encap-type)?
```

```
         |     |     |  +--:(mpls)
         |     |     |  +--:(eth)
         |     |     |  +--:(ipv6)
         |     |     +--rw bier-te-adj*       uint16
         |     +--:(cisco-mode)
         |     |  +--rw cisco-mode
         |     |     +--rw p-group?
                           rt-types:ip-multicast-group-address
         |     +--:(mpls)
         |     |  +--rw mpls
         |     |     +--rw (mpls-tunnel-type)?
         |     |        +--:(mldp)
         |     |        |  +--rw mldp-tunnel-id?       uint32
         |     |        |  +--rw mldp-backup-tunnel?   boolean
         |     |        +--:(p2mp-te)
         |     |           +--rw te-tunnel-id?         uint32
         |     |           +--rw te-backup-tunnel?     boolean
         |     +--:(pim)
         |        +--rw pim
        +--rw multicast-underlay
           +--rw (underlay)?
              +--:(bgp)
              +--:(ospf)
              |  +--rw ospf
              |     +--rw topology?
                          -> /rt:routing/control-plane-protocols
                             /control-plane-protocol/ospf:ospf
                             /topologies/topology/name
              +--:(isis)
              +--:(babel)

   notifications:
     +---n head-end-event
        +--ro event-type?                enumeration
        +--ro multicast-key
        |  +--ro vpn-rd?           rt-types:route-distinguisher
        |  +--ro source-address?   ip-multicast-source-address
        |  +--ro group-address?    rt-types:ip-multicast-group-address
        |  +--ro vni-type?         virtual-type
        |  +--ro vni-value?        uint32
        +--ro (overlay-tech-type)?
        |  +--:(bgp)
        |  +--:(evpn)
        |  +--:(mld)
        |  |  +--ro mld-instance-group?
                     rt-types:ip-multicast-group-address
        |  +--:(mld-snooping)
        |  +--:(mvpn)
```

```
        |  +--:(pim)
        +--ro transport-tech
        |  +--ro (transport)?
        |     +--:(bier)
        |     |  +--ro bier
        |     |     +--ro sub-domain?        uint16
        |     |     +--ro bitstringlength?   uint16
        |     |     +--ro set-identifier?    uint16
        |     |     +--ro (encap-type)?
        |     |        +--:(mpls)
        |     |        +--:(eth)
        |     |        +--:(ipv6)
        |     +--:(bier-te)
        |     |  +--ro bier-te
        |     |     +--ro sub-domain?        uint16
        |     |     +--ro bitstringlength?   uint16
        |     |     +--ro set-identifier?    uint16
        |     |     +--ro (encap-type)?
        |     |     |  +--:(mpls)
        |     |     |  +--:(eth)
        |     |     |  +--:(ipv6)
        |     |     +--ro bier-te-adj*       uint16
        |     +--:(cisco-mode)
        |     |  +--ro cisco-mode
        |     |     +--ro p-group?   rt-types:ip-multicast-group-address
        |     +--:(mpls)
        |     |  +--ro mpls
        |     |     +--ro (mpls-tunnel-type)?
        |     |        +--:(mldp)
        |     |        |  +--ro mldp-tunnel-id?       uint32
        |     |        |  +--ro mldp-backup-tunnel?   boolean
        |     |        +--:(p2mp-te)
        |     |           +--ro te-tunnel-id?         uint32
        |     |           +--ro te-backup-tunnel?     boolean
        |     +--:(pim)
        |        +--ro pim
        +--ro underlay-tech
           +--ro (underlay)?
              +--:(bgp)
              +--:(ospf)
              |  +--ro ospf
              |     +--ro topology?
                           -> /rt:routing/control-plane-protocols
                              /control-plane-protocol/ospf:ospf
                              /topologies/topology/name
              +--:(isis)
              +--:(babel)
```

3.3. Multicast YANG data model Configuration

This model is used with other protocol data model to provide multicast service.

This model includes multicast service keys and three layers: the multicast overlay, the transport layer and the multicast underlay information. Multicast keys include the features of multicast flow, such as(vpnid, multicast source and multicast group) information. In data center network, for fine-grained to gather the nodes belonging to the same virtual network, there may need VNI-related information to assist.

Multicast overlay defines (ingress-node, egress-nodes) nodes information. If the transport layer is BIER, there may define BIER information including (Subdomain, ingress-node BFR-id, egress-nodes BFR-id). If no (ingress-node, egress-nodes) information are defined directly, there may need overlay multicast signaling technology, such as MLD or MVPN, to collect these nodes information.

Multicast transport layer defines the type of transport technologies that can be used to forward multicast flow, including BIER forwarding type, MPLS forwarding type, or PIM forwarding type and so on. One or several transport technologies could be defined at the same time. As for the detailed parameters for each transport technology, this multicast YANG data model can invoke the corresponding protocol model to define them.

Multicast underlay defines the type of underlay technologies, such as OSPF, ISIS, BGP, PIM or BABEL and so on. One or several underlay technologies could be defined at the same time if there is protective requirement. As for the specific parameters for each underlay technology, this multicast YANG data model can depend the corresponding protocol model to configure them as well.

The configuration modeling branch is composed of the keys, overlay layer, transport layer and underlay layer.

3.4. Multicast YANG data model State

Multicast model states are the same with the configuration.

3.5. Multicast YANG data model Notification

The defined Notifications include the events of head end nodes. Like head node failer, overlay/ transport/ underlay module loading/ unloading. And the potential failer about some multicast flows and associated overlay/ transport/ underlay technologies.

4. Multicast YANG data Model

This module references [RFC1195], [RFC2328], [RFC4271], [RFC4541], [RFC4875], [RFC5340], [RFC6037], [RFC6388], [RFC6513], [RFC6991], [RFC7348], [RFC7432], [RFC7637], [RFC7716], [RFC7761], [RFC8279], [RFC8294], [RFC8296], [RFC8343], [RFC8344], [RFC8349], [RFC8639], [RFC8641], [I-D.ietf-pim-yang], [I-D.ietf-bier-bier-yang], [I-D.ietf-bier-te-arch], [I-D.ietf-nvo3-geneve], [I-D.ietf-bier-mld], [I-D.ietf-bess-evpn-bum-procedure-updates], [I-D.ietf-bier-evpn], [I-D.zhang-bier-bierin6], [I-D.ietf-babel-rfc6126bis], [I-D.ietf-bier-pim-signaling].

```
<CODE BEGINS> file "ietf-multicast-model@2020-03-06.yang"
module ietf-multicast-model {

  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-multicast-model";
  prefix multicast-model;

  import ietf-inet-types {
    prefix "inet";
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
                 (NMDA Version)";
  }
  import ietf-ospf {
    prefix "ospf";
    reference
      "I-D.ietf-ospf-yang: YANG Data Model for OSPF Protocol";
  }

  organization " IETF MBONED (MBONE Deployment) Working Group";
  contact
    "WG List:  <mailto:mboned@ietf.org>

     Editor:   Zheng Zhang
               <mailto:zzhang_ietf@hotmail.com>
```

```
       Editor:   Cui Wang
                 <mailto:lindawangjoy@gmail.com>
       Editor:   Ying Cheng
                 <mailto:chengying10@chinaunicom.cn>
       Editor:   Xufeng Liu
                 <mailto:xufeng.liu.ietf@gmail.com>
       Editor:   Mahesh Sivakumar
                 <mailto:sivakumar.mahesh@gmail.com>
     ";

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

   description
     "The module defines the YANG definitions for multicast service
      management.

      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 XXXX
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC
      itself for full legal notices.

      The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
      'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
      'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document
      are to be interpreted as described in BCP 14 (RFC 2119)
      (RFC 8174) when, and only when, they appear in all
      capitals, as shown here.";

   revision 2020-03-06 {
     description
       "Initial revision.";
     reference
       "RFC XXXX: A YANG Data Model for multicast YANG.";
   }

   /*
    *typedef
    */
```

```
    typedef ip-multicast-source-address {
      type union {
        type rt-types:ipv4-multicast-source-address;
        type rt-types:ipv6-multicast-source-address;
      }
      description
        "This type represents a version-neutral IP multicast
         source address.  The format of the textual
         representation implies the IP version.";
      reference
        "RFC8294: Common YANG Data Types for the Routing Area.";
    }

    typedef virtual-type {
      type enumeration {
        enum vxlan {
          description
            "The VXLAN encapsulation is used for flow encapsulation.";
          reference
            "RFC 7348: Virtual eXtensible Local Area Network (VXLAN):
             A Framework for Overlaying Virtualized Layer 2 Networks
             over Layer 3 Networks.";
        }
        enum nvgre {
          description
            "The NVGRE encapsulation is used for flow encapsulation.";
          reference
            "RFC 7637: NVGRE: Network Virtualization Using Generic
             Routing Encapsulation.";
        }
        enum geneve {
          description
            "The GENEVE encapsulation is used for flow encapsulation.";
          reference
            "I-D.ietf-nvo3-geneve: Geneve: Generic Network
             Virtualization Encapsulation.";
        }
      }
      description
        "The encapsulation type used for the flow. In case the virtual
         type is set, the associated vni-value should also be defined.";
    } // virtual-type

    /*
     * Identities
     */

    identity multicast-model {
```

```
      base rt:control-plane-protocol;
      description "Identity for the Multicast model.";
    }

    grouping general-multicast-key {
      description
        "The general multicast keys. They are used to distinguish
         different multicast service.";
      leaf vpn-rd {
        type rt-types:route-distinguisher;
        description
          "A Route Distinguisher used to distinguish
           routes from different MVPNs.";
        reference
          "RFC 8294: Common YANG Data Types for the Routing Area.
           RFC 6513: Multicast in MPLS/BGP IP VPNs.";
      }
      leaf source-address {
        type ip-multicast-source-address;
        description
          "The IPv4/IPv6 source address of the multicast flow. The
           value set to zero means that the receiver interests
           in all source that relevant to one given group.";
      }
      leaf group-address {
        type rt-types:ip-multicast-group-address;
        description
          "The IPv4/IPv6 group address of multicast flow. This
           type represents a version-neutral IP multicast group
           address. The format of the textual representation
           implies the IP version.";
        reference
          "RFC8294: Common YANG Data Types for the Routing Area.";
      }
      leaf vni-type {
        type virtual-type;
        description
          "The type of virtual network identifier. Includes the
           Vxlan, NVGRE and Geneve. This value and vni-value is
           used to indicate a specific virtual multicast service.";
      }
      leaf vni-value {
        type uint32;
        description
          "The value of Vxlan network identifier, virtual subnet ID
           or virtual net identifier. This value and vni-type is used
           to indicate a specific virtual multicast service.";
      }
```

```
    } // general-multicast-key

    grouping encap-type {
      description
        "The encapsulation type used for flow forwarding.";
      choice encap-type {
        case mpls {
          description "The BIER forwarding depends on mpls.";
          reference
            "RFC 8296: Encapsulation for Bit Index Explicit
             Replication (BIER) in MPLS and Non-MPLS Networks.";
        }
        case eth {
          description "The BIER forwarding depends on ethernet.";
          reference
            "RFC 8296: Encapsulation for Bit Index Explicit
             Replication (BIER) in MPLS and Non-MPLS Networks.";
        }
        case ipv6 {
          description "The BIER forwarding depends on IPv6.";
          reference
            "I-D.zhang-bier-bierin6: BIER in IPv6 (BIERin6)";
        }
        description "The encapsulation type in BIER.";
      }
    } // encap-type

    grouping bier-key {
      description
        "The key parameters set for BIER/BIER TE forwarding.";
      reference
        "RFC 8279: Multicast Using Bit Index Explicit Replication
         (BIER).";

      leaf sub-domain {
        type uint16;
        description
          "The subdomain id that the multicast flow belongs to.";
      }
      leaf bitstringlength {
        type uint16;
        description
          "The bitstringlength used by BIER forwarding.";
      }
      leaf set-identifier {
        type uint16;
        description
          "The set identifier used by the multicast flow.";
```

```
      }
      uses encap-type;
    }

    grouping transport-tech {
      choice transport {
        description "The selected transport technology.";
        container bier {
          description
            "The transport technology is BIER. The BIER technology
             is introduced in RFC8279. The parameter is consistent
             with the definition in BIER YANG data model.";
          reference
            "RFC 8279: Multicast Using Bit Index Explicit
             Replication (BIER).
             I-D.ietf-bier-bier-yang: YANG Data Model for BIER
             Protocol.";

          uses bier-key;
        }

        container bier-te {
          description
            "The transport technology is BIER-TE.";
          reference
            "I-D.ietf-bier-te-arch: Traffic Engineering for Bit Index
             Explicit Replication (BIER-TE)";

          uses bier-key;

          leaf-list bier-te-adj {
            type uint16;
            description
              "The adjacencies ID used in BIER TE forwarding
               encapsulation.";
          }
        }

        container cisco-mode {
          description
            "The transport technology is cisco-mode: Cisco MDT.";
          reference
            "RFC 6037: Cisco Systems' Solution for Multicast in
             BGP/MPLS IP VPNs";

          leaf p-group {
            type rt-types:ip-multicast-group-address;
            description
```

```
            "The address of p-group. It is used to encapsulate
             and forward flow according to multicast tree from
             ingress node to egress nodes.";
          }
          uses transport-pim;
        }

        container mpls {
          description
            "The transport technology is mpls. MVPN overlay can use
             mpls tunnel technologies to build transport layer.";
          reference
            "RFC 6513: Multicast in MPLS/BGP IP VPNs.";

          choice mpls-tunnel-type {
            case mldp {
              description "The mldp tunnel.";
              reference
                "RFC 6388: Label Distribution Protocol Extensions
                 for Point-to-Multipoint and Multipoint-to-Multipoint
                 Label Switched Paths.";

              leaf mldp-tunnel-id {
                type uint32;
                description
                  "The tunnel id that correspond this flow.";
              }
              leaf mldp-backup-tunnel {
                type boolean;
                description
                  "If the backup tunnel function should be
                   supported.";
              }
            }
            case p2mp-te {
              description
                "The p2mp te tunnel.";
              reference
                "RFC 4875: Extensions to Resource Reservation Protocol
                 - Traffic Engineering (RSVP-TE) for Point-to-Multipoint
                 TE Label Switched Paths (LSPs).";

              leaf te-tunnel-id {
                type uint32;
                description
                  "The tunnel id that correspond this flow.";
              }
              leaf te-backup-tunnel {
```

```
                type boolean;
                description
                  "If the backup tunnel function should be
                   supported.";
              }
            }
            description "The collection types of mpls tunnels";
          }
        } // mpls

        container pim {
          description
            "The transport technology is PIM. PIM is used
             commonly in traditional network.";
          reference
            "RFC 7761: Protocol Independent Multicast - Sparse Mode
                 (PIM-SM): Protocol Specification (Revised).";
          uses transport-pim;
        }
      } // choice
    } // transport-tech

    grouping underlay-tech {
      choice underlay {
        case bgp {
          description
            "The underlay technology is BGP. BGP protocol
             should be used to run if BGP is used as
             underlay protocol.";
          reference
            "RFC 4271: A Border Gateway Protocol 4 (BGP-4)";
        }
        container ospf {
          description
            "The underlay technology is OSPF. OSPF protocol
             should be triggered to run if OSPF is used as underlay
             protocol.";
          reference
            "RFC 2328: OSPF Version 2.
             RFC 5340: OSPF for IPv6.
             I-D.ietf-ospf-yang: YANG Data Model for OSPF Protocol.";

          leaf topology {
            type leafref {
              path "/rt:routing/rt:control-plane-protocols/"
                 + "rt:control-plane-protocol/ospf:ospf/"
                 + "ospf:topologies/ospf:topology/ospf:name";
            }
```

```
            description
              "The designed topology name of ospf protocol.";
          }
        }
        case isis {
          description
            "The underlay technology is ISIS. ISIS protocol should
             be triggered to run if ISIS is used as underlay protocol.
             And the associated extensions can be used.";
          reference
            "RFC 1195: Use of OSI IS-IS for Routing in TCP/IP and
             Dual Environments";
        }
        case babel {
          description
            "The underlay technology is Babel. Babel protocol
             should be triggered to run if Babel is used as
             underlay protocol.";
          reference
            "I-D.ietf-babel-rfc6126bis: The Babel Routing Protocol.";
        }
      } // choice
    } // underlay-tech

    /*overlay*/

    grouping overlay-tech {
      choice overlay-tech-type {
        case bgp {
          description
            "BGP technology is used for multicast overlay.";
          reference
            "RFC 7716: Global Table Multicast with BGP Multicast
                       VPN (BGP-MVPN) Procedures.";
        }
        case evpn {
          description
            "EVPN technology is used for multicast overlay.";
          reference
            "RFC 7432: BGP MPLS-Based Ethernet VPN.
             I-D.ietf-bess-evpn-bum-procedure-updates: Updates on
               EVPN BUM Procedures.
             I-D.ietf-bier-evpn: EVPN BUM Using BIER.";
        }
        case mld {
          description
            "MLD technology is used for multicast overlay.";
          reference
```

```
            "I-D.ietf-bier-mld: BIER Ingress Multicast Flow Overlay
             using Multicast Listener Discovery Protocols.";
          leaf mld-instance-group {
            type rt-types:ip-multicast-group-address;
            description
              "The multicast address used for multiple MLD instance
               support.";
          }
        }
        case mld-snooping {
          description
            "MLD snooping technology is used for multicast overlay.";
          reference
            "RFC 4541: Considerations for Internet Group Management
                       Protocol (IGMP) and Multicast Listener
                       Discovery (MLD) Snooping Switches.";
        }
        case mvpn {
          description
            "MVPN technology is used for multicast overlay.";
          reference
            "RFC 6513: Multicast in MPLS/BGP IP VPNs.";
        }
        case pim {
          description
            "PIM technology is used for multicast overlay.";
          reference
            "I-D.ietf-bier-pim-signaling: PIM Signaling
             Through BIER Core.";
        }
        description
          "The overlay technology used for multicast service.";
      }
      description "The overlay technology used for multicast service.";
    } // overlay-tech

    /*transport*/

    grouping transport-pim {
      description
        "The requirement information of pim transportion.";
      reference
        "RFC 7761: Protocol Independent Multicast - Sparse Mode
                   (PIM-SM): Protocol Specification (Revised).";
    } //transport-pim

    /*underlay*/
```

```
    container multicast-model {
      description
        "The model of multicast YANG data. Include keys, overlay,
         transport and underlay.";

      list multicast-keys{
        key "vpn-rd source-address group-address vni-type vni-value";
        uses general-multicast-key;

        container multicast-overlay {
          description
            "The overlay information of multicast service.
             Overlay technology is used to exchange multicast
             flows information. Overlay technology may not be
             used in SDN controlled completely situation, but
             it can be used in partial SDN controlled situation
             or non-SDN controlled situation. Different overlay
             technologies can be choosed according to different
             deploy consideration.";

          container ingress-egress {
            description
              "The ingress and egress nodes address collection.
               The ingress node may use the egress nodes set
               directly to encapsulate the multicast flow by
               transport technology.";

            leaf ingress-node {
              type inet:ip-address;
              description
                "The ip address of ingress node for one or more
                 multicast flow. Or the ingress node of MVPN and
                 BIER. In MVPN, this is the address of ingress
                 PE; in BIER, this is the BFR-prefix of ingress
                 nodes.";
            }
            list egress-nodes {
              key "egress-node";
              description
                "The egress multicast nodes of the multicast flow.
                 Or the egress node of MVPN and BIER. In MVPN, this
                 is the address of egress PE; in BIER, this is the
                 BFR-prefix of ingress nodes.";

              leaf egress-node {
                type inet:ip-address;
                description
                  "The ip-address set of egress multicast nodes.";
```

```
            }
          }
        }

        container bier-ids {
          description
            "The BFR-ids of ingress and egress BIER nodes for
             one or more multicast flows. This overlay is used
             with BIER transport technology. The egress nodes
             set can be used to encapsulate the multicast flow
             directly in the ingress node.";
          reference
            "RFC 8279: Multicast Using Bit Index Explicit
             Replication (BIER)";

          leaf sub-domain {
            type uint16;
            description
              "The sub-domain that this multicast flow belongs to.";
          }
          leaf ingress-node {
            type uint16;
            description
              "The ingress node of multicast flow. This is the
               BFR-id of ingress nodes.";
          }
          list egress-nodes {
            key "egress-node";
            description
              "The egress nodes of multicast flow.";

            leaf egress-node {
              type uint16;
              description
                "The BFR-ids of egress multicast BIER nodes.";
            }
          }
        }
        uses overlay-tech;
      }

      container multicast-transport {
        description
          "The transportion of multicast service. Transport
           protocol is responsible for delivering multicast
           flows from ingress nodes to egress nodes with or
           without specific encapsulation. Different transport
           technology can be choosed according to different
```

```
            deploy consideration. Once a transport technology
            is choosed, associated protocol should be triggered
            to run.";

          uses transport-tech;
        }
        container multicast-underlay {
          description
            "The underlay of multicast service. Underlay protocol
             is used to build transport layer. Underlay protocol
             need not be assigned in ordinary network since
             existed underlay protocol fits well, but it can be
             assigned in particular networks for better
             controll. Once a underlay technology is choosed,
             associated protocol should be triggered to run.";

          uses underlay-tech;
        }
        description
          "The model of multicast YANG data. Include keys,
           overlay, transport and underlay.";
      }
    }

    /*Notifications*/

    notification head-end-event {
      leaf event-type {
        type enumeration {
          enum down {
            description
              "There is something wrong with head end node,
               and head end node can't work properlay.";
          }
          enum module-loaded {
            description
              "The new modules that can be used by multicast
               flows have been loaded.";
          }
          enum module-unloaded {
            description
              "The new modules that can be used by multicast
               flows have been unloaded.";
          }
        }
        description "Event type.";
      }
      container multicast-key {
```

```
        uses general-multicast-key;
        description
          "The associated multicast keys that are influenced by
           head end node failer.";
      }
      uses overlay-tech;

      container transport-tech {
        description
          "The modules can be used to forward multicast flows.";
        uses transport-tech;
      }

      container underlay-tech {
        description
          "There is something wrong with the module which is
           used to build multicast transport layer.";
        uses underlay-tech;
      }
      description
        "Notification events for the head end nodes. Like head
         node failer, overlay/ transport/ underlay module
         loading/ unloading. And the potential failer about some
         multicast flows and associated
         overlay/ transport/ underlay technologies.";
    }
  }
  <CODE ENDS>
```

5. 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 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 data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/multicast-model,

multicast-model

These data nodes in this model specifies the configuration for the multicast service at the top level. Modifying the configuration can cause multicast service to be deleted or reconstructed.

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 data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/multicast-model,

Unauthorized access to any data node of the above tree can disclose the operational state information of multicast service on this device.

6. IANA Considerations

RFC Ed.: Please replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

The IANA is requested to assign one new URI from the IETF XML registry [RFC3688]. Authors are suggesting the following URI:

URI: urn:ietf:params:xml:ns:yang:ietf-multicast-model

Registrant Contact: The IESG

XML: N/A, the requested URI is an XML namespace

This document also requests one new YANG module name in the YANG Module Names registry [RFC6020] with the following suggestion:

name: ietf-multicast-model

namespace: urn:ietf:params:xml:ns:yang:ietf-multicast-model

prefix: multicast-model

reference: RFC XXXX

7. Acknowledgements

The authors would like to thank Stig Venaas, Jake Holland, Min Gu for their valuable comments and suggestions.

8. References

8.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <https://www.rfc-editor.org/info/rfc4875>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

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

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

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

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/ BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

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

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K., and D. Pacella, "Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures", RFC 7716, DOI 10.17487/RFC7716, December 2015, <https://www.rfc-editor.org/info/rfc7716>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[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>.

[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>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[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>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[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>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <https://www.rfc-editor.org/info/rfc8344>.

[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>.

[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

[I-D.ietf-babel-rfc6126bis]
Chroboczek, J. and D. Schinazi, "The Babel Routing Protocol", draft-ietf-babel-rfc6126bis-17 (work in progress), February 2020.

[I-D.ietf-bess-evpn-bum-procedure-updates]
Zhang, Z., Lin, W., Rabadan, J., Patel, K., and A. Sajassi, "Updates on EVPN BUM Procedures", draft-ietf-bess-evpn-bum-procedure-updates-08 (work in progress), November 2019.

[I-D.ietf-bier-bier-yang]
Chen, R., hu, f., Zhang, Z., dai.xianxian@zte.com.cn, d., and M. Sivakumar, "YANG Data Model for BIER Protocol", draft-ietf-bier-bier-yang-06 (work in progress), February 2020.

[I-D.ietf-bier-evpn]
Zhang, Z., Przygienda, T., Sajassi, A., and J. Rabadan, "EVPN BUM Using BIER", draft-ietf-bier-evpn-02 (work in progress), November 2019.

[I-D.ietf-bier-mld]
Pfister, P., Wijnands, I., Venaas, S., Wang, C., Zhang, Z., and M. Stenberg, "BIER Ingress Multicast Flow Overlay using Multicast Listener Discovery Protocols", draft-ietf-bier-mld-04 (work in progress), March 2020.

[I-D.ietf-bier-pim-signaling]
Bidgoli, H., Kotalwar, J., Xu, F., mishra, m., Zhang, Z., and A. Dolganow, "PIM Signaling Through BIER Core", draft-ietf-bier-pim-signaling-08 (work in progress), November 2019.

[I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., and M. Menth, "Path Engineering for Bit Index Explicit Replication (BIER-TE)", draft-ietf-bier-te-arch-06 (work in progress), February 2020.

[I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-14 (work in progress), September 2019.

[I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", draft-ietf-ospf-yang-29 (work in progress), October 2019.

[I-D.ietf-pim-yang]
Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and f. hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (work in progress), May 2018.

[I-D.zhang-bier-bierin6]
Zhang, Z., Przygienda, T., Wijnands, I., Bidgoli, H., and M. McBride, "BIER in IPv6 (BIERin6)", draft-zhang-bier-bierin6-04 (work in progress), January 2020.

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

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <https://www.rfc-editor.org/info/rfc4541>.

[RFC6037] Rosen, E., Ed., Cai, Y., Ed., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, DOI 10.17487/RFC6037, October 2010, <https://www.rfc-editor.org/info/rfc6037>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in JSON encoding [RFC7951], containing configuration data.

The configuration example:

```
{
    "ietf-multicast-model:multicast-model":{
        "multicast-keys":[
            {
                "vpn-rd":"0:65532:4294967292",
                "source-address":"*",
                "group-address":"234.232.203.84",
                "vni-type":"nvgre",
                "vni-value":0,
                "multicast-overlay":{
                    "ingress-egress":{
                        "ingress-node":"146.150.100.0",
                        "egress-nodes":[
                            {
                                "egress-node":"110.141.168.0"
                            }
                        ]
                    },
                },
                "multicast-transport":{
                    "bier":{
                        "sub-domain":0,
                        "bitstringlength":256,
                        "set-identifier":0
                    }
                },
                "multicast-underlay":{
                    "ospf":{
                        "topology":"2"
                    }
                }
            }
        ]
    }
}
```


Authors' Addresses

Zheng Zhang
ZTE Corporation
China

Email: zzhang_ietf@hotmail.com

Cui(Linda) Wang
Individual
Australia

Email: lindawangjoy@gmail.com

Ying Cheng
China Unicom
Beijing
China

Email: chengying10@chinaunicom.cn

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Mahesh Sivakumar
Juniper networks
1133 Innovation Way
Sunnyvale, CALIFORNIA 94089
USA

Email: sivakumar.mahesh@gmail.com