Workgroup: MBONED WG
Internet-Draft:
draft-ietf-mboned-multicast-yang-model-05
Published: 25 August 2021
Intended Status: Standards Track
Expires: 26 February 2022
Authors: Z. Zhang C. Wang Y. Cheng
ZTE Corporation Individual China Unicom
X. Liu M. Sivakumar
Volta Networks Juniper networks

Multicast YANG Data Model

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 26 February 2022.

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

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

This model can 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:

BABEL: [RFC8966].

BGP: Border Gateway Protocol [RFC4271].

BIER: Bit Index Explicit Replication [RFC8279].

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

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

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

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

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

OSPF: Open Shortest Path First [RFC2328].

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

PIM: Protocol Independent Multicast [RFC7761].

SR-P2MP: Segment Routing Point-to-Multipoint [I-D.ietf-pim-sr-p2mp-policy].

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]
isis	ietf-isis	[I-D.ietf-isis-yang-isis-cfg]
ospf	ietf-ospf	[I-D.ietf-ospf-yang]
rt-types	ietf-routing-types	[RFC8294]
rt	ietf-routing	[RFC8349]
yang	ietf-yang-types	[RFC6991]

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 that are used to deliver multicast service. This model is used for gathering data from the network as well.

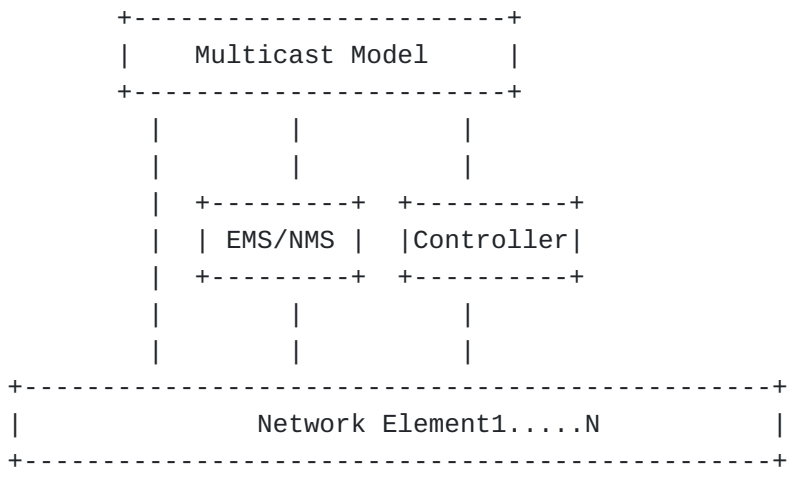

Figure 1: Usage of Multicast Model

Figure 1 illustrates example use cases for this multicast model. Network operators can use this model in a controller which is responsible to implement specific multicast flows with specific protocols and work with the corresponding protocols' model to configure the network elements through NETCONF/RESTCONF/CLI. Or network operators can use this model to the EMS (Element Management System)/ NMS (Network Management System) to manage or configure the network elements directly.

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 respond 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 4.

The usage of this model is flexible. The multicast-keys indicate the flow characters. The flow can be L3 multicast flow, or L2 flow which is also called BUM (Broadcast, Unknown unicast, Multicast) flow in EVPN ([RFC7432]) deployment.

Among the multicast-keys, the group-address of L3 multicast flow and the mac-address of BUM flow are the most important keys. The other keys are optional, and need not be all set. For example, only group-address is set, this is (*,G) analogous. If source-address and

group-address are both set, this is (S,G) analogous. In addition to the source-address and group-address, when vpn-rd is also set, this is MVPN use case. If mac-address and vpn-rd are set, this is EVPN use case. In case vni-value is set with associated group-address, etc., this is NVO3 multicast use case.

*When the controller manages all the ingress and egress routers for the flow, it sends the model that is set with flow characters, ingress and egress nodes information to the ingress and egress nodes. Then the ingress and egress nodes can work without any other dynamic overlay protocols.

*When the controller manages the ingress nodes only for the flow, it sends the model that is set with the flow characters to the ingress nodes. The dynamic overlay protocol can be set or not. If the overlay protocol is set, the nodes use the protocol to signal the flow information with other nodes. If the overlay protocol is not set, the nodes use the local running overlay protocol to signal the flow information.

*When the transport protocol is set in the model, the nodes encapsulate the flow according to the transport protocol. When the transport protocol is not set in the model, the nodes use the local configured transport protocol for encapsulation.

*When the transport protocol is set in the model, the underlay protocol may be set in the model also. In case the underlay protocol is set, the nodes use the underlay protocol to signal and build the transport/forwarding layer. In case the underlay protocol is not set, the nodes use the local configured underlay protocol to signal and build the transport/forwarding layer.

*More than one ingress node for a multicast flow can be set in the model. In this situation, two or more ingress nodes can used for a multicast flow forwarding, the ingress routers can be backup for each other. More information can be found in [I-D.szcl-mboned-redundant-ingress-failover].

1.5.1. Example

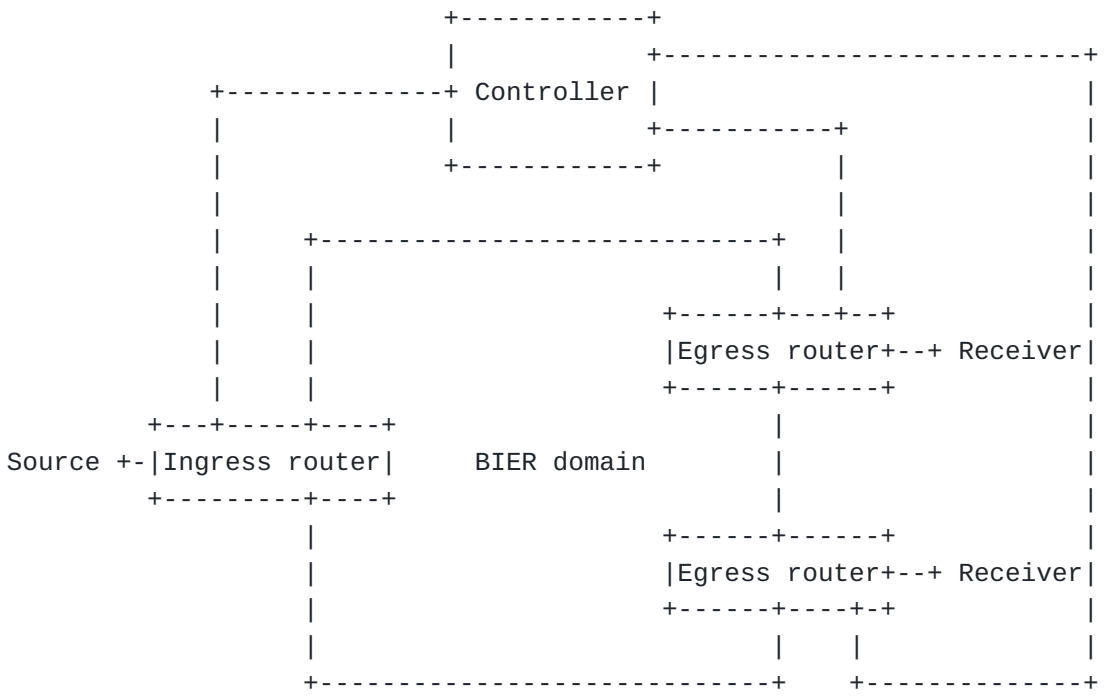

Figure 2: Example

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]. Corresponding IGP protocol which is used to build BIER transport layer is OSPF [RFC2328].

In this model, the corresponding group-address that is in multicast-keys 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 edge router from the controller. If the BIER transport layer which depends on OSPF has not been built in the network, the multicast YANG model may invoke the BIER YANG model that 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.

Another example for this figure is, the controller can act as the BIER overlay only. The routers in the domain build BIER forwarding plane beforehand. The controller sends the multicast group-address and/or the source-address to the edge routers in BIER domain only, without transport and underlay set in the model. Then the ingres router can encapsulate the multicast flow with BIER encapsulation automatically.

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 is used to notify the controller that there is error and the error reason.

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.

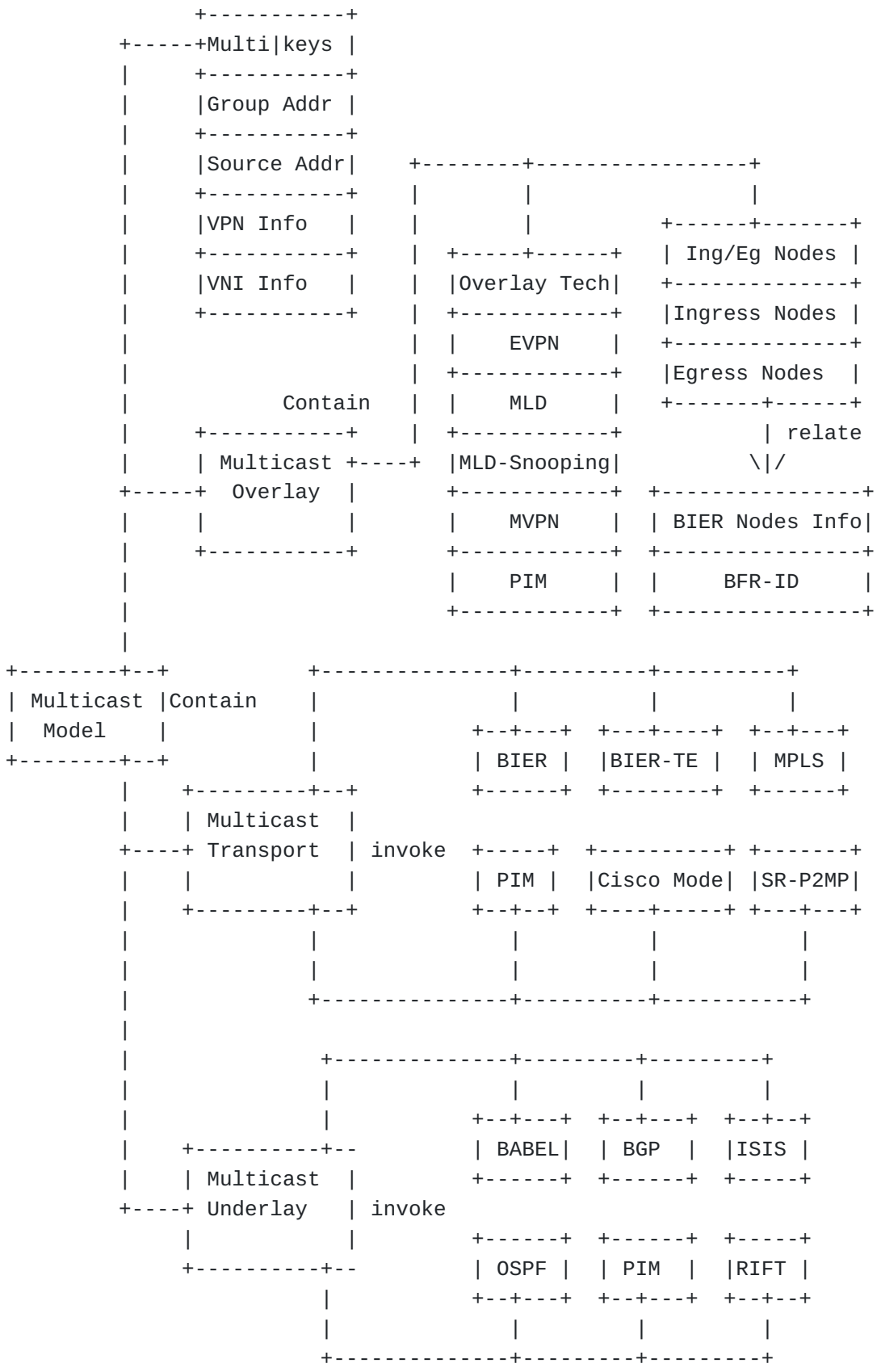

Figure 3: UML like Class Diagram for Multicast YANG data Model

3.2. Model Structure

```
module: ietf-multicast-model
  +--rw multicast-model
     +--rw multicast-keys*
             [vpn-rd source-address group-address mac-address 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 mac-address            yang:mac-address
        +--rw vni-value              uint32
        +--rw multicast-overlay
        |  +--rw vni-type?          virtual-type
        |  +--rw ingress-egress
        |  |  +--rw ingress-nodes* [ingress-node]
        |  |  |  +--rw ingress-node    inet:ip-address
        |  |  +--rw egress-nodes* [egress-node]
        |  |     +--rw egress-node    inet:ip-address
        |  +--rw bier-ids {bier}?
        |  |  +--rw sub-domain?      uint16
        |  |  +--rw ingress-nodes* [ingress-node]
        |  |  |  +--rw ingress-node    uint16
        |  |  +--rw egress-nodes* [egress-node]
        |  |     +--rw egress-node    uint16
        |  +--rw dynamic-overlay
        |     +--rw type?   identityref
        |     +--rw mld
        |        +--rw mld-instance-group?
        |                rt-types:ip-multicast-group-address
        +--rw multicast-transport
        |  +--rw type?           identityref
        |  +--rw bier
        |  |  +--rw sub-domain?        uint16
        |  |  +--rw bitstringlength?   uint16
        |  |  +--rw set-identifier?    uint16
        |  |  +--rw (encap-type)?
        |  |     +--:(mpls)
        |  |     +--:(eth)
        |  |     +--:(ipv6)
        |  +--rw bier-te
        |  |  +--rw sub-domain?        uint16
        |  |  +--rw bitstringlength?   uint16
        |  |  +--rw set-identifier?    uint16
        |  |  +--rw (encap-type)?
        |  |  |  +--:(mpls)
        |  |  |  +--:(eth)
        |  |  |  +--:(ipv6)
        |  |  +--rw bitstring* [name]
        |  |     +--rw name           string
        |  |     +--rw bier-te-adj* [adj-id]
```

```
      |  |        +--rw adj-id    uint16
      |  +--rw cisco-mdt
      |  |  +--rw p-group?   rt-types:ip-multicast-group-address
      |  +--rw rsvp-te-p2mp
      |  |  +--rw template-name?   string
      |  +--rw pim
      |  |  +--rw source-address?   ip-multicast-source-address
      |  |  +--rw group-address
      |  |          rt-types:ip-multicast-group-address
      |  +--rw sr-p2mp
      |     +--rw ir-segment-lists* [name]
      |     |  +--rw name    string
      |     +--rw replication-segment* [replication-id node-id]
      |        +--rw replication-id    tree-sid
      |        +--rw node-id           inet:ip-address
      +--rw multicast-underlay
         +--rw type?   identityref
         +--rw ospf
         |  +--rw topology?   string
         +--rw isis
         |  +--rw topology?   string
         +--rw pim
            +--rw source-address?   ip-multicast-source-address
            +--rw group-address
                    rt-types:ip-multicast-group-address

notifications:
  +---n ingress-egress-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 mac-address?      yang:mac-address
     |  +--ro vni-value?        uint32
     +--ro dynamic-overlay
     |  +--ro type?   identityref
     |  +--ro mld
     |     +--ro mld-instance-group?
     |              rt-types:ip-multicast-group-address
     +--ro transport-tech
     |  +--ro type?           identityref
     |  +--ro bier
     |  |  +--ro sub-domain?        uint16
     |  |  +--ro bitstringlength?   uint16
     |  |  +--ro set-identifier?    uint16
     |  |  +--ro (encap-type)?
     |  |     +--:(mpls)
     |  |     +--:(eth)
```

```
|  |     +--:(ipv6)
|  +--ro bier-te
|  |  +--ro sub-domain?        uint16
|  |  +--ro bitstringlength?   uint16
|  |  +--ro set-identifier?    uint16
|  |  +--ro (encap-type)?
|  |  |  +--:(mpls)
|  |  |  +--:(eth)
|  |  |  +--:(ipv6)
|  |  +--ro bitstring* [name]
|  |     +--ro name           string
|  |     +--ro bier-te-adj* [adj-id]
|  |        +--ro adj-id    uint16
|  +--ro cisco-mdt
|  |  +--ro p-group?   rt-types:ip-multicast-group-address
|  +--ro rsvp-te-p2mp
|  |  +--ro template-name?   string
|  +--ro pim
|  |  +--ro source-address?   ip-multicast-source-address
|  |  +--ro group-address
|  |          rt-types:ip-multicast-group-address
|  +--ro sr-p2mp
|     +--ro ir-segment-lists* [name]
|     |  +--ro name    string
|     +--ro replication-segment* [replication-id node-id]
|        +--ro replication-id    tree-sid
|        +--ro node-id           inet:ip-address
+--ro underlay-tech
   +--ro type?   identityref
   +--ro ospf
   |  +--ro topology?   string
   +--ro isis
   |  +--ro topology?   string
   +--ro pim
      +--ro source-address?   ip-multicast-source-address
      +--ro group-address
              rt-types:ip-multicast-group-address
```

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 may 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 ingress or egress nodes. Like ingress node failure, overlay/ transport/ underlay module loading/ unloading. And the potential failure 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], [RFC8926], [RFC8966], [I-D.ietf-pim-yang], [I-D.ietf-bier-bier-yang], [I-D.ietf-bier-te-arch], [I-D.ietf-bier-mld], [I-D.ietf-bess-evpn-bum-procedure-updates], [I-D.ietf-bier-evpn], [I-D.ietf-bier-bierin6], [I-D.ietf-bier-pim-signaling], [I-D.ietf-rift-rift], [I-D.ietf-isis-yang-isis-cfg].

```
<CODE BEGINS> file "ietf-multicast-model@2021-08-26.yang"

module ietf-multicast-model {

  yang-version 1.1;

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

  import ietf-yang-types {
    prefix "yang";
    reference
      "RFC 6991: Common YANG Data Types";
  }

  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)";
  }

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

     Editor:   Zheng Zhang
               <mailto:zhang.zheng@zte.com.cn>
     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) 2021 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 2021-08-26 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for multicast YANG.";
}

/*
 *feature
 */
feature bier {
  description
    "Cooperation with BIER technology.";
  reference
    "RFC 8279:
       Multicast Using Bit Index Explicit Replication (BIER).";
}

/*
 *typedef
 */
typedef ip-multicast-source-address {
  type union {
    type enumeration {
```

```
      enum * {
        description
          "Any source address.";
      }
    }
    type inet:ipv4-address;
    type inet:ipv6-address;
  }
  description
    "Multicast source IP address type.";
}
typedef tree-sid {
  type union {
    type rt-types:mpls-label;
    type inet:ip-prefix;
  }
  description
    "The type of the Segment Identifier of a Replication segment
     is a SR-MPLS label or a SRv6 SID.";
}
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
        "RFC 8926: Geneve: Generic Network
         Virtualization Encapsulation.";
    }
  }
  description
    "The encapsulation type used for the flow.
     When this type is set, the associated vni-value
     MUST be set.";
```

```
} // virtual-type

/*
 * Identities
 */

identity multicast-model {
  base "rt:control-plane-protocol";
  description "Identity for the multicast model.";
}
identity overlay-type {
  description
    "Base identity for the type of multicast overlay technology.";
}
identity transport-type {
  description "Identity for the multicast transport technology.";
}
identity underlay-type {
  description "Identity for the multicast underlay technology.";
}
identity overlay-pim {
  base overlay-type;
  description
    "Using PIM as multicast overlay technology.
     For example, as BIER overlay.";
  reference
    "I-D.ietf-bier-pim-signaling:
       PIM Signaling Through BIER Core.";
}
identity mld {
  base overlay-type;
  description
    "Using MLD as multicast overlay technology.
     For example, as BIER overlay.";
  reference
    "I-D.ietf-bier-mld:
       BIER Ingress Multicast Flow Overlay
       using Multicast Listener Discovery Protocols.";
}
identity mld-snooping {
  base overlay-type;
  description
    "Using MLD as multicast overlay technology.
     For example, as BIER overlay.";
  reference
    "RFC 4541:
       Considerations for Internet Group Management
       Protocol (IGMP) and Multicast Listener
       Discovery (MLD) Snooping Switches.";
```

```
}
identity evpn {
  base overlay-type;
  description
    "Using EVPN as multicast overlay technology.";
  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.";
}
identity mvpn {
  base overlay-type;
  description
    "Using MVPN as multicast overlay technology.";
  reference
    "RFC 6513: Multicast in MPLS/BGP IP VPNs.
     RFC 7716:
       Global Table Multicast with BGP Multicast VPN
       (BGP-MVPN) Procedures.";
}
identity bier {
  base transport-type;
  description
    "Using BIER as multicast transport technology.";
  reference
    "RFC 8279:
       Multicast Using Bit Index Explicit Replication (BIER).";
}
identity bier-te {
  base transport-type;
  description
    "Using BIER-TE as multicast transport technology.";
  reference
    "I-D.ietf-bier-te-arch:
       Traffic Engineering for Bit Index Explicit Replication
       (BIER-TE)";
}
identity mldp {
  base transport-type;
  description
    "Using mLDP as multicast transport technology.";
  reference
    "RFC 6388:
       Label Distribution Protocol Extensions
       for Point-to-Multipoint and Multipoint-to-Multipoint
       Label Switched Paths.
     I-D.ietf-mpls-mldp-yang: YANG Data Model for MPLS mLDP.";
}
```

```
identity rsvp-te-p2mp {
  base transport-type;
  description
    "Using P2MP TE as multicast transport technology.";
  reference
    "RFC 4875:
       Extensions to Resource Reservation Protocol
       - Traffic Engineering (RSVP-TE) for Point-to-Multipoint
       TE Label Switched Paths (LSPs).";
}
identity sr-p2mp {
  base transport-type;
  description
    "Using Segment Routing  as multicast transport technology.";
  reference
    "I-D.ietf-pim-sr-p2mp-policy:
       Segment Routing Point-to-Multipoint Policy.";
}
identity cisco-mdt {
  base transport-type;
  description
    "Using cisco MDT for multicast transport technology.";
  reference
    "RFC 6037:
       Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs";
}
identity pim {
  base transport-type;
  base underlay-type;
  description
    "Using PIM as multicast transport technology.";
  reference
    "RFC 7761:
       Protocol Independent Multicast - Sparse Mode
       (PIM-SM): Protocol Specification (Revised).";
}
identity bgp {
  base underlay-type;
  description
    "Using BGP as underlay technology to build the multicast
     transport layer. For example, using BGP as BIER underlay.";
  reference
    "I-D.ietf-bier-idr-extensions: BGP Extensions for BIER.";
}
identity ospf  {
  base underlay-type;
  description
    "Using OSPF as multicast underlay technology.
     For example, using OSPF as BIER underlay.";
```

```
  reference
    "RFC 8444:
       OSPFv2 Extensions for Bit Index Explicit Replication (BIER),
     I-D.ietf-bier-ospfv3-extensions:
       OSPFv3 Extensions for BIER.";
}
identity isis {
  base underlay-type;
  description
    "Using ISIS as multicast underlay technology.
     For example, using ISIS as BIER underlay.";
  reference
    "RFC 8401:
       Bit Index Explicit Replication (BIER) Support via IS-IS";
}
identity babel {
  base underlay-type;
  description
    "Using BABEL as multicast underlay technology.
     For example, using BABEL as BIER underlay.";
  reference
    "RFC 8966: The Babel Routing Protocol
     I-D.zhang-bier-babel-extensions: BIER in BABEL";
}
identity rift {
  base underlay-type;
  description
    "Using RIFT as multicast underlay technology.
     For example, using RIFT as BIER underlay.";
  reference
    "I-D.ietf-rift-rift: RIFT: Routing in Fat Trees.
     I-D.zzhang-bier-rift: Supporting BIER with RIFT";
}

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
      "RFC 8294: Common YANG Data Types for the Routing Area.";
  }
  leaf mac-address {
    type yang:mac-address;
    description
      "The mac address of flow. In the EVPN situation, the L2
       flow that is called
       BUM (Broadcast, Unknown Unicast, Multicast)
       can be sent to the other PEs that
       are in a same broadcast domain.";
    reference
      "RFC 6991: Common YANG Data Types.
       RFC 7432: BGP MPLS-Based Ethernet VPN.";
  }
  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.
     This encapsulation acts as the inner encapsulation,
     as compare to the outer multicast-transport encapsulation.";
  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.ietf-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 {
  description
    "The transport technology selected for the multicast service.
     For one specific multicast flow, it's better to use only one
     transport technology for forwarding.";

  leaf type {
    type identityref {
      base transport-type;
    }
```

```
  description "The type of transport technology";
}
container bier {
  when "../type = 'ietf-multicast-model:bier'" {
     description
       "Only when BIER is used as transport technology.";
  }
  description
    "The transport technology is BIER. The BIER technology
     is introduced in RFC8279. The parameters are consistent
     with the definition in BIER YANG data model.";
  reference
    "I-D.ietf-bier-bier-yang:
       YANG Data Model for BIER Protocol.";
  uses bier-key;
}
container bier-te {
  when "../type = 'ietf-multicast-model:bier-te'" {
     description
       "Only when BIER-TE is used as transport technology.";
  }
  description
    "The BIER-TE parameter that may need to be set.
     The parameters are consistent with the definition in
     BIER and BIER TE YANG data model.";
  reference
    "I-D.ietf-bier-bier-yang:
       YANG Data Model for BIER Protocol
     I-D.ietf-bier-te-yang:
       A YANG data model for Traffic Engineering for Bit Index
       Explicit Replication (BIER-TE)";

  uses bier-key;

  list bitstring {
    key "name";
    leaf name {
      type string;
      description "The name of the bitstring";
    }
    list bier-te-adj {
      key "adj-id";
      leaf adj-id {
        type uint16;
        description
          "The link adjacency ID used for BIER TE forwarding.";
      }
      description
        "The adjacencies ID used for BIER TE bitstring
```

```
          encapsulation.";
      }
      description
        "The bitstring name and detail used for BIER TE
         forwarding encapsulation. One or more bitstring can be
         used for backup path.";
    }
  }
  container cisco-mdt {
    when "../type = 'ietf-multicast-model:cisco-mdt'" {
       description
         "Only when cisco MDT is used as transport technology.";
    }
    description "The MDT parameter that may need to be set.";
    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.";
    }
  }
  container rsvp-te-p2mp {
    when "../type = 'ietf-multicast-model:rsvp-te-p2mp'" {
     description
       "Only when RSVP TE P2MP is used as transport technology.";
    }
    description
      "The parameter that may be set. They are consistent with
      the definition in TE data model.";
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";

    leaf template-name {
      type string {
        pattern '/?([a-zA-Z0-9\-_.]+)(/[a-zA-Z0-9\-_.]+)*';
      }
      description
        "A type for the name of a TE node template or TE link
         template.";
    }
  }
  container pim {
    when "../type = 'ietf-multicast-model:pim'" {
       description
         "Only when PIM is used as transport technology.";
    }
    description "The PIM parameter that may need to be set.";
    uses pim;
```

```
  }
  container sr-p2mp {
    when "../type = 'ietf-multicast-model:sr-p2mp'" {
       description
         "Only when segment routing P2MP is used as transport
          technology.";
    }
    description "The SR-P2MP parameter that may need to be set.";
    list ir-segment-lists {
      key "name";
      leaf name {
        type string;
        description "Segment-list name";
      }
      description
        "The segment lists used for ingress replication.
         The name refers a segment list.";
    }

    list replication-segment {
      key "replication-id node-id";
      leaf replication-id {
        type tree-sid;
        description
          "The identifier for a Replication segment that is
           unique in context of the Replication Node.
           This is a SR-MPLS label or a SRv6 SID";
      }
      leaf node-id {
        type inet:ip-address;
        description
          "The address of the Replication Node that the
           Replication segment is for.";
      }
      description
        "A Multi-point service delivery could be realized via
         P2MP trees in a Segment Routing domain.
         It may consist of one or more Replication segment";
      reference
        "I-D.ietf-spring-sr-replication-segment:
           SR Replication Segment for Multi-point Service
           Delivery.";
    }
  } // sr-p2mp
} // transport-tech

grouping underlay-tech {
  description
    "The underlay technology selected for the transport layer.
```

```
   The underlay technology has no straight relationship with
   the multicast overlay, it is used for transport path
   building, for example BIER forwarding path building.";

leaf type {
  type identityref {
    base underlay-type;
  }
  description "The type of underlay technology";
}
container ospf {
  when "../type = 'ietf-multicast-model:ospf'" {
     description
       "Only when OSPF is used as underay technology.";
  }
  description
    "If OSPF protocol supports multiple topology feature,
     the associated topology name may be assigned.
     In case the topology name is assigned, the specific
     OSPF topology is used for underly to building the
     transport layer.";
  reference
    "RFC 4915: Multi-Topology Routing";
  leaf topology {
    type string;
    description
      "The designed topology name of ospf protocol.";
  }
}
container isis {
  when "../type = 'ietf-multicast-model:isis'" {
     description
       "Only when ISIS is used as underay technology.";
  }
  description
    "If ISIS protocol supports multiple topology feature,
     the associated topology name may be assigned.
     In case the topology name is assigned, the specific
     ISIS topology is used for underly to building the
     transport layer.";
  reference
    "RFC 5120: M-IS-IS: Multi Topology Routing in IS-IS";
  leaf topology {
    type string;
    description
      "The designed topology name of isis protocol.";
  }
}
container pim {
```

```
    when "../type = 'ietf-multicast-model:pim'" {
       description
         "Only when PIM is used as underay technology.";
    }
    description "The PIM parameter that may need to be set.";
    uses pim;
  }
} // underlay-tech

/*overlay*/

grouping overlay-tech {
  container dynamic-overlay {
    leaf type {
      type identityref {
        base overlay-type;
      }
      description "The type of overlay technology";
    }
    container mld {
      when "../type = 'ietf-multicast-model:mld'" {
         description
           "Only when MLD is used as overlay technology.";
      }
      description "The MLD parameter that may need to be set.";
      leaf mld-instance-group {
        type rt-types:ip-multicast-group-address;
        description
          "The multicast address used for multiple MLD instance
           support.";
      }
    }
    description
      "The dynamic overlay technologies and associated parameter
       that may be set.";
  }
  description "The overlay technology used for multicast service.";
} // overlay-tech

/*transport*/

grouping pim {
  description
    "The required information of pim transportion.";
  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;
    mandatory true;
    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
    "RFC 7761: Protocol Independent Multicast - Sparse Mode
               (PIM-SM): Protocol Specification (Revised).";
} //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 mac-address 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.";

      leaf vni-type {
        type virtual-type;
        description
          "The encapsulated type for the multicast flow,
           it is used to carry the virtual network identifier
           for the multicast service.";
      }

      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.";

    list ingress-nodes {
      key "ingress-node";
      description
        "The egress nodes of multicast flow.";

      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.
           Two or more ingress nodes may existed for the
           redundant ingress node protection.";
      }
    }

    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 {
    if-feature bier;
    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.";
    }
    list ingress-nodes {
      key "ingress-node";
      description
        "The ingress nodes of multicast flow.";
      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 ingress-egress-event {
    leaf event-type {
      type enumeration {
        enum down {
          description
            "There is something wrong with ingress or egress node,
             and node can't work properlay.";
        }
        enum protocol-enabled {
          description
            "The protocol that is used for multicast
             flows have been enabled.";
        }
        enum protocol-disabled {
          description
            "The protocol that is used by multicast
             flows have been disabled.";
        }
      }
      description "Event type.";
    }
    container multicast-key {
      uses general-multicast-key;
      description
        "The associated multicast keys that are influenced by
         ingress or egress 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 ingress or egress nodes. Like
       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, Gyan Mishra 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-bess-evpn-bum-procedure-updates] Zhang, Z., Lin, W., Rabadan, J., Patel, K., and A. Sajassi, "Updates on EVPN BUM Procedures", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-bum-procedure-updates-09, 9 June 2021, <https://www.ietf.org/

archive/id/draft-ietf-bess-evpn-bum-procedure-updates-09.txt>.

[I-D.ietf-bier-bier-yang] Chen, R., Hu, F., Zhang, Z., Dai, X., and M. Sivakumar, "YANG Data Model for BIER Protocol", Work in Progress, Internet-Draft, draft-ietf-bier-bier-yang-07, 8 September 2020, <https://www.ietf.org/archive/id/draft-ietf-bier-bier-yang-07.txt>.

[I-D.ietf-bier-bierin6] Zhang, Z., Zhang, Z., Wijnands, I., Mishra, M., Bidgoli, H., and G. Mishra, "Supporting BIER in IPv6 Networks (BIERin6)", Work in Progress, Internet-Draft, draft-ietf-bier-bierin6-00, 14 June 2021, <https://www.ietf.org/archive/id/draft-ietf-bier-bierin6-00.txt>.

[I-D.ietf-bier-evpn] Zhang, Z., Przygienda, A., Sajassi, A., and J. Rabadan, "EVPN BUM Using BIER", Work in Progress, Internet-Draft, draft-ietf-bier-evpn-04, 2 December 2020, <https://www.ietf.org/archive/id/draft-ietf-bier-evpn-04.txt>.

[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", Work in Progress, Internet-Draft, draft-ietf-bier-mld-05, 22 February 2021, <https://www.ietf.org/archive/id/draft-ietf-bier-mld-05.txt>.

[I-D.ietf-bier-pim-signaling]
Bidgoli, H., Xu, F., Kotalwar, J., Wijnands, I., Mishra, M., and Z. Zhang, "PIM Signaling Through BIER Core", Work in Progress, Internet-Draft, draft-ietf-bier-pim-signaling-12, 25 July 2021, <https://www.ietf.org/archive/id/draft-ietf-bier-pim-signaling-12.txt>.

[I-D.ietf-bier-te-arch] Eckert, T., Cauchie, G., and M. Menth, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-arch-09, 30 October 2020, <https://www.ietf.org/internet-drafts/draft-ietf-bier-te-arch-09.txt>.

[I-D.ietf-isis-yang-isis-cfg] Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for IS-IS Protocol", Work in Progress, Internet-Draft, draft-ietf-isis-yang-isis-cfg-42, 15 October 2019, <https://www.ietf.org/archive/id/draft-ietf-isis-yang-isis-cfg-42.txt>.

[I-D.ietf-ospf-yang] Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", Work in

Progress, Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019, <https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>.

[I-D.ietf-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", Work in Progress, Internet-Draft, draft-ietf-pim-sr-p2mp-policy-02, 19 February 2021, <https://www.ietf.org/archive/id/draft-ietf-pim-sr-p2mp-policy-02.txt>.

[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)", Work in Progress, Internet-Draft, draft-ietf-pim-yang-17, 19 May 2018, <https://www.ietf.org/archive/id/draft-ietf-pim-yang-17.txt>.

[I-D.ietf-rift-rift] Sharma, A., Thubert, P., Rijsman, B., and D. Afanasiev, "RIFT: Routing in Fat Trees", Work in Progress, Internet-Draft, draft-ietf-rift-rift-13, 12 July 2021, <https://www.ietf.org/archive/id/draft-ietf-rift-rift-13.txt>.

[I-D.szcl-mboned-redundant-ingress-failover] Shepherd, G., Zhang, Z., Liu, Y., and Y. Cheng, "Multicast Redundant Ingress Router Failover", Work in Progress, Internet-Draft, draft-szcl-mboned-redundant-ingress-failover-01, 8 July 2021, <https://www.ietf.org/archive/id/draft-szcl-mboned-redundant-ingress-failover-01.txt>.

[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>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/info/rfc8926>.

[RFC8966] Chroboczek, J. and D. Schinazi, "The Babel Routing Protocol", RFC 8966, DOI 10.17487/RFC8966, January 2021, <https://www.rfc-editor.org/info/rfc8966>.

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",
        "mac-address": "00:00:5e:00:53:01",
        "vni-value":0,
        "multicast-overlay":{
          "vni-type":"nvgre",
          "ingress-egress":{
            "ingress-nodes":[
              {
                "ingress-node":"146.150.100.0"
              }
            ],
            "egress-nodes":[
              {
                "egress-node":"110.141.168.0"
              }
            ]
          }
        },
        "multicast-transport":{
          "type": "ietf-multicast-model:bier",
          "bier":{
            "sub-domain":0,
            "bitstringlength":256,
            "set-identifier":0
          }
        },
        "multicast-underlay":{
          "type": "ietf-multicast-model:ospf",
          "ospf":{
            "topology":"2"
          }
        }
      }
    ]
  }
}
```

Authors' Addresses

Zheng Zhang
ZTE Corporation
China

Email: zhang.zheng@zte.com.cn

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,
United States of America

Email: sivakumar.mahesh@gmail.com