MBONED WG                                                   Zheng. Zhang
Internet-Draft                                                 Cui. Wang
Intended status: Standards Track                         ZTE Corporation
Expires: August 25, 2018                                     Ying. Cheng
                                                           China Unicom
                                                      February 21, 2018

# Multicast YANG Data Model
## draft-zhang-mboned-multicast-yang-model-00

Abstract

This document intents to provide a general and all-round multicast YANG data model, which tries to stand at a high level to take full advantages of existed multicast protocol models to control the multicast network, and guides the deployment of multicast service. And also, there will define several possible RPCs about how to interact between multicast YANG data model and multicast protocol models.  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 used to deliver multicast service, as well as gathering some data from the network.

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 August 25, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors.  All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

## 1. Introduction

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 intents to provide a general and all-round multicast model, which tries to stand at a high level to take full advantages of these aforementioned models to control the multicast network, and guides the deployment of multicast service.

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 used to deliver multicast service, as well as gathering some data from the network.

```
                  +-------------------------+
                  |    Multicast Model   |
                  +-------------------------+
                    |        |          |
                    |        |          |
                    |  +----------+ +-----------+
                    |  | EMS/NMS |  |Controller|
                    |  +----------+ +-----------+
                    |        |          |
                    |        |          |
           +------------------------------------------------+
           |               Network Element1.....N           |
           +------------------------------------------------+
```

Figure 1: Example 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 who is responsible to implement some 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. For example, a multicast service need to be delopy in a network, supposed that the multicast flow is 239.0.0.0/8, the flow should be transport by BIER technology. Then we use this multicast YANG data model and set the correspond key (239.0.0.0) and associated transport technology with BIER, send the model from controller to every egde node in the network. Then there is an interaction among all the nodes to exchange the multicast flow information. The ingress node will encapsulate the multicast flow with BIER header and send it into the network. Intermediate nodes will 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

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.

## 3. UML Class like Diagram for Multicast YANG data Model

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

```
                                             +--------------------+
                                             |  Multicast  Model  |
                                             +--------------------+
                                               |   |   |   |Contain
     +------------------------------------------+   |   |   |
     |                         +------------------ -+   |
+----------------------------+
     |                         |
|                               |
+-----------+         +--------------------+           +----------------------
+         +--------------------+
|Multi-keys |         | Multicast Overlay |           | Multicast Transport
|         | Multicast Underlay |
+-----------+         +--------------------+           +----------------------
+         +--------------------+
|Group Addr |                |   |Contain                    | | | | |
invoke               | | | | | invoke
+-----------+       +---------+   +--------+             +----+ | | | +----
+            +----+ | | | +----+
|Source Addr|       |                    |              |      | | |
|            |      | | |      |
+-----------+ +------------+     +--------------+    +------+   | | |   +------
+     +------+  | | |   +------+
|VPN Info   | |Overlay Tech|     | Ing/Eg Nodes |    | PIM |   | | |   | MPLS
|     | OSPF |  | | |   | PIM  |
+-----------+ +------------+     +--------------+    +------+   | | |   +------
+     +------+  | | |   +------+
|VNI Info   | |     MLD    |     |Ingress Nodes |         +----+ | +-----
+               +----+ | +-----+
+-----------+ +------------+     +--------------+         |      |
|               |      |       |
              |     MVPN   |     |Egress Nodes  |   +-----------+ |  +--------
+        +------+   |    +------+
              +------------+     +--------------+   |Cisco Mode| |  |BIER-TE
|        |BABEL|   |    | BGP  |
              |     BGP    |             | relate   +-----------+ |  +--------
+        +------+   |    +------+
              +------------+           \|/                        +----
+                    +-----+
              |MLD-Snooping|    +-----------------+
|                    |
              +------------+    | BIER Nodes Info|                  +------
+            +-------+
                                +-----------------+                  | BIER
|            | ISIS |
                                | BFR-ID          |                  +------
+            +-------+
                                +-----------------+
```

Figure 2: UML like Class Diagram for Multicast YANG data Model

## 4.  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?     bier:sub-domain-id
          |  |  +--rw ingress-node?   bier:bfr-id
          |  |  +--rw egress-nodes* [egress-node]
          |  |     +--rw egress-node    bier:bfr-id
          |  +--rw overlay-tech-type?   enumeration
          +--rw multicast-transport
          |  +--rw bier
          |  |  +--rw sub-domain?        bier:sub-domain-id
          |  |  +--rw (encap-type)?
          |  |  |  +--:(mpls)
          |  |  |  +--:(eth)
          |  |  |  +--:(ipv6)
          |  |  +--rw bitstringlength?   bier:bsl
          |  |  +--rw set-identifier?    bier:si
          |  |  +--rw ecmp?              boolean
          |  |  +--rw frr?               boolean
          |  +--rw bier-te
          |  |  +--rw sub-domain?        bier:sub-domain-id
          |  |  +--rw (encap-type)?
          |  |  |  +--:(mpls)
          |  |  |  +--:(non-mpls)
          |  |  +--rw bitstringlength?   bier:bsl
          |  |  +--rw set-identifier?    bier:si
          |  |  +--rw ecmp?              boolean
          |  |  +--rw frr?               boolean
          |  +--rw cisco-mode
          |  |  +--rw p-group?            rt-types:ip-multicast-group-address
          |  |  +--rw graceful-restart?   boolean
          |  |  +--rw bfd?                boolean
          |  +--rw mpls
          |  |  +--rw (mpls-tunnel-type)?
          |  |     +--:(mldp)
          |  |     |  +--rw mldp-tunnel-id?       uint32
          |  |     |  +--rw mldp-frr?             boolean
          |  |     |  +--rw mldp-backup-tunnel?   boolean
          |  |     +--:(p2mp-te)
          |  |        +--rw te-tunnel-id?         uint32
          |  |        +--rw te-frr?               boolean
          |  |        +--rw te-backup-tunnel?     boolean
          |  +--rw pim
          |     +--rw graceful-restart?   boolean
          |     +--rw bfd?                boolean
          +--rw multicast-underlay
             +--rw underlay-requirement?   boolean
             +--rw bgp
             +--rw ospf
             |  +--rw topology-id?   uint8
             +--rw isis
             |  +--rw topology-id?   uint16
             +--rw 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?   enumeration
       +--ro transport-tech?      enumeration
       +--ro underlay-tech?       enumeration
```

## 5.  Multicast YANG data Model

```
<CODE BEGINS> file "ietf-multicast-model.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 "RFC6991";
    }

    import ietf-routing-types {
        prefix rt-types;
        reference "RFC8294";
    }

    import ietf-bier {
        prefix bier;
    }

    organization " IETF MBONED( MBONE Deployment ) Working Group";
    contact
        "WG List:  <mailto:bier@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>
        ";
```

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

          Copyright (c) 2018 IETF Trust and the persons
          identified as authors of the code.  All rights reserved.

          Redistribution and use in source and binary forms, with or
          without modification, is permitted pursuant to, and subject
          to the license terms contained in, the Simplified BSD License
          set forth in Section 4.c of the IETF Trust's Legal Provisions
          Relating to IETF Documents
          (http://trustee.ietf.org/license-info).
          This version of this YANG module has relationship with overall
multicast
          technologies, such as PIM(RFC7761), BIER(RFC8279), MVPN(RFC6513), and
so
          on; see the RFC itself for full legal notices.";

    revision 2018-02-22 {
        description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for multicast YANG.
          RFC 7761: Protocol Independent Multicast - Sparse Mode (PIM-SM):
                    Protocol Specification (Revised).
          RFC 8279: Multicast Using Bit Index Explicit Replication (BIER);
          RFC 6513: Multicast in MPLS/BGP IP VPNs";
    }

/*key*/

    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 type. See more detail in RFC7348.";
            }
            enum "virtual subnet" {
                description "The nvgre type. See more detail in RFC7637.";
            }
            enum "vni" {
                description "The geneve type. See more detail in [ietf-nvo3-
geneve].";
            }
        }
        description "The collection of virtual network type.";
    }

    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 (RFC 6513).";
            reference
                "RFC8294: Common YANG Data Types for the Routing Area.";
        }
        leaf source-address {
            type ip-multicast-source-address;
            description "The IPv4/IPv6 source address of 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.";
        }
    }
```

```
/*overlay*/

    grouping overlay-technology {
        leaf overlay-tech-type {
            type enumeration {
                enum mld {
                description "MLD technology is used for multicast overlay. See
more detail in [draft-ietf-bier-mld]";
                }
                enum mvpn {
                    description "MVPN technology is used for multicast overlay.
See more detail in RFC6513.";
```

```
                }
                enum bgp {
                    description "BGP technology is used for multicast overlay.
See more detail in RFC7716.";
                }
                enum mld-snooping {
                    description "MLD snooping technology is used for multicast
overlay. See more detail in RFC4541.";
                }
            }
            description "The possible overlay technologies for multicast
service.";
        }
        description "The possible overlay technologies for multicast service.";
    }

    grouping multicast-overlay {
        description "The multicast overlay information, includes ingress node
and egress nodes' information.";
        container ingress-egress {
            description "The ingress and egress nodes address collection.";
            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 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 of egress multicast nodes. See more
details in RFC6513.";
                }
            }
        }

        container bier-ids {
            description "The BFR-ids of ingress and egress BIER nodes for one
```

```
or more multicast flows.";
            leaf sub-domain {
                type bier:sub-domain-id;
                description "The sub-domain that this multicast flow belongs
to. See more details in RFC8279.";
            }
            leaf ingress-node {
                type bier:bfr-id;
                description "The ingress node of multicast flow. This is the
```

```
                      BFR-id of ingress nodes. See more details in RFC8279.";
            }
            list egress-nodes {
                key "egress-node";
                description "This ID information of one adjacency. See more
details in RFC8279.";

                leaf egress-node {
                    type bier:bfr-id;
                    description "The BFR-ids of egress multicast BIER nodes.
See more details in RFC8279.";
                }
            }
        }

        uses overlay-technology;
    }


/*transport*/

    grouping transport-pim {
        description "The requirement information of pim transportion. PIM
protocol is defined in RFC7761.";
        leaf graceful-restart {
            type boolean;
            description "If the graceful restart function should be
supported.";
        }
        leaf bfd {
            type boolean;
            description "If the bfd function should be supported.";
        }
    }

    grouping multicast-transport {
        description "The transport information of multicast service.";
        container bier {
            description "The transport technology is BIER. The BIER technology
is introduced in RFC8279. The parameter is consistent with the definition in
[ietf-bier-bier-yang].";
            leaf sub-domain {
                type bier:sub-domain-id;
                description "The subdomain id that the multicast flow belongs
to. See more details in RFC8279.";
            }
            choice encap-type {
                case mpls {
                    description "The BIER forwarding depends on mpls. See more
details in RFC8296.";
                }
                case eth {
                    description "The BIER forwarding depends on ethernet. See
more details in RFC8296.";
                }
                case ipv6 {
                    description "The BIER forwarding depends on IPv6.";
                }
                description "The encapsulation type in BIER.";
            }
            leaf bitstringlength {
                type bier:bsl;
                description "The bitstringlength used by BIER forwarding. See
more details in RFC8279.";
            }
            leaf set-identifier {
                type bier:si;
                description "The set identifier used by the multicast flow. See
more details in RFC8279.";
            }
            leaf ecmp {
                type boolean;
                description "The capability of ECMP. If this value is set to
true, ecmp mechanism should be enabled. See more details in RFC8279.";
            }
            leaf frr {
                type boolean;
                description "The capability of fast re-route. If this value is
set to true, fast re-route mechanism should be enabled. See more details in
RFC8279.";
            }
        }
        container bier-te {
            description "The transport technology is BIER-TE. BIER-TE
technology is introduced in [ietf-bier-te-arch].";
            leaf sub-domain {
                type bier:sub-domain-id;
                description "The subdomain id that the multicast flow belongs
to. See more details in [ietf-bier-te-arch].";
            }
            choice encap-type {
                case mpls {
                    description "The BIER-TE forwarding depends on mpls. See
more details in [ietf-bier-te-arch].";
                }
                case non-mpls {
                    description "The BIER-TE forwarding depends on non-mpls.
See more details in [ietf-bier-te-arch].";
                }
                description "The encapsulation type in BIER-TE.";
            }
            leaf bitstringlength {
                type bier:bsl;
                description "The bitstringlength used by BIER-TE forwarding.
See more details in [ietf-bier-te-arch].";
```

```
            }
            leaf set-identifier {
                type bier:si;
                description "The set identifier used by the multicast flow,
especially in BIER TE. See more details in [ietf-bier-te-arch].";
            }
            leaf ecmp {
                type boolean;
                description "The capability of ECMP. If this value is set to
true, ecmp mechanism should be enabled. See more details in [ietf-bier-te-
arch].";
            }
            leaf frr {
                type boolean;
                description "The capability of fast re-route. If this value is
set to true, fast re-route mechanism should be enabled. See more details in
[ietf-eckert-bier-te-frr].";
            }
        }
        container cisco-mode {
            description "The transport technology is cisco-mode. The Cisco MDT
multicast mechanism is defined in RFC6037.";
            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. The usage is introduced in
RFC6513.";
            choice mpls-tunnel-type {
                case mldp {
                    description "The mldp tunnel. The protocol detail is
defined in RFC6388.";
                    leaf mldp-tunnel-id {
                        type uint32;
                        description "The tunnel id that correspond this flow.
The detail is defined in RFC6388.";
                    }
                    leaf mldp-frr {
                        type boolean;
                        description "If the fast re-route function should be
supported. The detail is defined in RFC6388.";
                    }
                    leaf mldp-backup-tunnel {
                        type boolean;
                        description "If the backup tunnel function should be
supported. The detail is defined in RFC6388.";
                    }
                }
                case p2mp-te {
                    description "The p2mp te tunnel. The protocol detail is
defined in RFC4875.";
                    leaf te-tunnel-id {
                        type uint32;
                        description "The tunnel id that correspond this flow.
The detail is defined in RFC4875.";
```

```
                    }
                    leaf te-frr {
                        type boolean;
                        description "If the fast re-route function should be
supported. The detail is defined in RFC4875.";
                    }
                    leaf te-backup-tunnel {
                        type boolean;
                        description "If the backup tunnel function should be
supported. The detail is defined in RFC4875.";
                    }
                }
                description "The collection types of mpls tunnels";
            }
        }
        container pim {
            uses transport-pim;
            description "The transport technology is PIM. PIM [RFC7761] is used
commonly in traditional network.";
        }
    }

/*underlay*/

    grouping multicast-underlay {
        description "The underlay information relevant multicast service.
Underlay protocols are used to build transport layer. It is unnecessary in
traditional network that use PIM [RFC7761] to build multicast tree. Diversity
underlay protocols can be choosed to build BIER transport layer.";
        leaf underlay-requirement {
            type boolean;
            description "If the underlay technology is required.";
        }
        container bgp {
            description "The underlay technology is BGP. BGP protocol RFC4271
should be triggered to run if BGP is used as underlay protocol.";
        }
        container ospf {
            description "The underlay technology is OSPF. OSPF protocol RFC2328
should be triggered to run if OSPF is used as underlay protocol.";
            leaf topology-id {
                type uint8;
                description "The topology id of ospf instance. The topology id
can be assigned In some situations. More details is defined in RFC2328.";
            }
        }
        container isis {
            description "The underlay technology is ISIS. ISIS protocol should
be triggered to run if ISIS is used as underlay protocol. Details is defined in
RFC1195.";
            leaf topology-id {
                type uint16;
                description "The topology id of isis instance. The topology id
can be assigned In some situations.";
            }
        }
        container babel {
            description "The underlay technology is Babel. Babel protocol
should be triggered to run if Babel is used as underlay protocol.";
        }
    }

    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 technology can be choosed according to different deploy
consideration.";
                uses multicast-overlay;
            }
            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 multicast-transport;
            }
            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 controlling. Once a underlay
technology is choosed, associated protocol should be triggered to run.";
                uses multicast-underlay;
            }
            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 "Some new modules that can be used by multicast
flows finish loading.";
                }
                enum module-unloaded {
                    description "Some 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-technology;
        leaf transport-tech {
            type enumeration {
                enum bier {
                    description "BIER(RFC8279) technology can be used to
forward multicast flows.";
                }
                enum bier-te {
                    description "BIER-TE(draft-ietf-bier-te-arch) technology
can be used to forward multicast flows.";
                }
                enum cisco-mode {
                    description "Cisco mode(RFC6037) technology can be used to
forward multicast flows.";
                }
                enum mldp {
                    description "MLDP(RFC6388) technology can be used to
forward multicast flows.";
                }
                enum p2mp-te {
                    description "P2MP TE(RFC4875) technology can be used to
forward multicast flows.";
                }
                enum pim {
                    description "PIM(RFC7761) technology can be used to forward
multicast flows.";
                }
            }
            description "The modules can be used to forward multicast flows.";
        }
        leaf underlay-tech {
            type enumeration {
                enum bgp {
                    description "BGP protocol can be used to build multicast
transport layer.";
                }
                enum ospf {
                    description "OSPF protocol can be used to build multicast
transport layer.";
                }
                enum isis {
                    description "ISIS protocol can be used to build multicast
transport layer.";
                }
                enum babel {
                    description "Babel protocol can be used to build multicast
transport layer.";
                }
            }
            description "The modules can be used to build multicast transport
layer.";
        }
        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>
```

## 6.  Notifications

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.

## 7. Acknowledgements

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

## 8. Normative References

[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-03 (work in progress), February 2018.

[I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication (BIER-TE)", draft-ietf-bier-te-arch-00 (work in progress), January 2018.

[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-13 (work in progress), January 2018.

[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>.

[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>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <https://www.rfc-editor.org/info/rfc6087>.

[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>.

[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>.

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

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <https://www.rfc-editor.org/info/rfc7277>.

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

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[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>.

## Authors' Addresses

Zheng Zhang
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: zhang.zheng@zte.com.cn

Cui(Linda) Wang
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: lindawangjoy@gmail.com

Ying Cheng
China Unicom
Beijing
China

Email: chengying10@chinaunicom.cn