MBONED WG
Internet-Draft
Intended status: Standards Track
Expires: February 22, 2018

Zheng. Zhang
Cui. Wang
ZTE Corporation
Ying. Cheng
China Unicom

August 21, 2017

Multicast Model
draft-zhang-mboned-multicast-info-model-02

Abstract

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 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 info model and multicast protocol models. This multicast information 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 http://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 February 22, 2018.

Table of Contents

1. Introduction

Currently, there are many multicast 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 information 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.

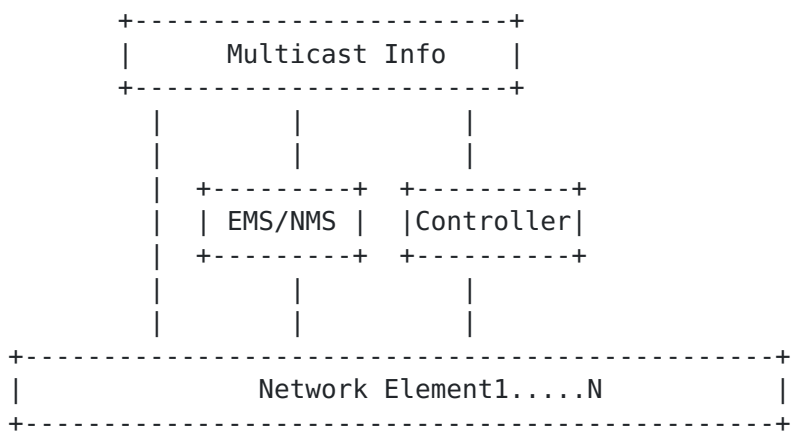


Figure 1: Example usage of Multicast Model

Detailly,in figure 1, there is an example of usage of this multicast model. Network operators can input this model to a controller who is responsible to translate the information and invoke the corresponding protocol models into configurations to configure the network elements through NETCONF/RESTCONF/CLI. Or network operators can input this model to the EMS/NMS to manage the network elements or configure the network elements directly. On the other hand, when the network elements detect failure or some other changes, the network operators can collect these kind of notifications through this model to assist locating the exact failure and responsing immediatly. for example, when the network element suffers a failure of one MVPN neighbor, it can notify to the EMS/NMS or Controller or to other Multicast Model management tool directly to let the network operator take actions immediately.

Specifically, in section 3, it provides a human readability of the whole multicast network through UML class diagram, which frames different multicast components and correlates them in a readable fashion. Then, based on this UML 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 three layers: the multicast overlay, the transport layer and the multicast underlay information.

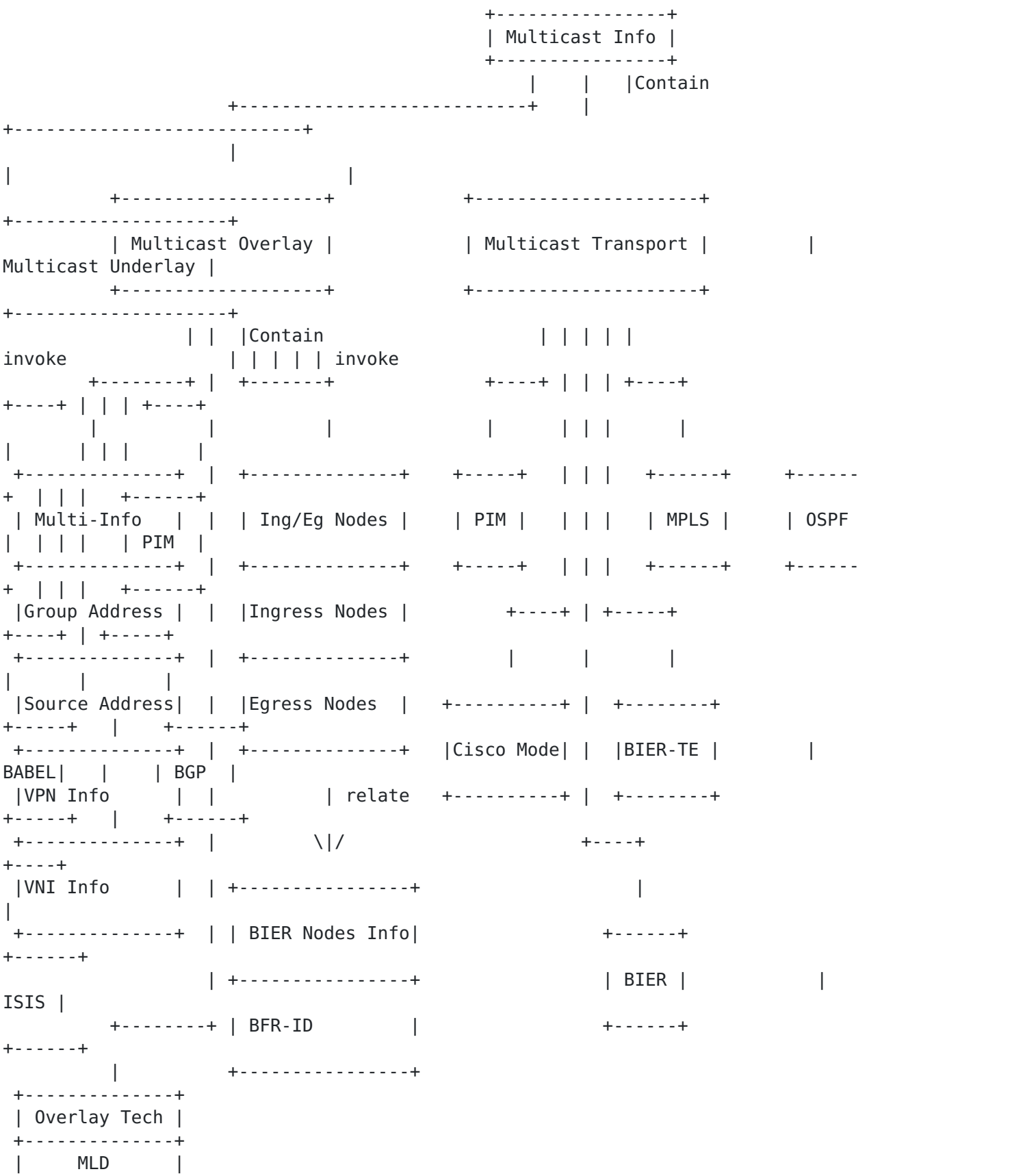
Multicast overlay defines the features of multicast flow, such as(vpnid, multicast source and multicast group) information, and (ingress-node,egress-nodes) nodes information. If the transport layer is BIER, there may define BIER information including (Subdomain, ingress-node BFR-id, egress-nodes BFR-id). 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. 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 information 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. As for the specific parameters for each underlay technology, this multicast information model can depend the corresponding protocol model to configure them as well.

3. UML Class Diagram for Multicast Info Model

The following is a UML diagram for Multicast Info Model.


Multicast Info
Contain
Multicast Overlay
Multicast Transport
Multicast Underlay
Contain
invoke
invoke
Multi-Info
Ing/Eg Nodes
PIM
MPLS
OSPF
PIM
Group Address
Ingress Nodes
Source Address
Egress Nodes
Cisco Mode
BIER-TE
VPN Info
relate
BABEL
BGP
VNI Info
BIER Nodes Info
BIER
ISIS
BFR-ID
Overlay Tech
MLD

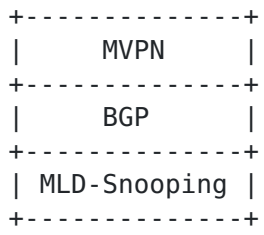


Figure 2: UML Class Diagram for Multicast Info Model

4. Model Structure

```
module: ietf-multicast-information
    +--rw multicast-information
       +--rw multicast-info* [vpn-id source-address source-wildcard group-
address group-wildcard vni-type vni-value]
          +--rw vpn-id                uint32
          +--rw source-address        inet:ip-address
          +--rw source-wildcard       uint8
```

```
          +--rw group-address          inet:ip-address
          +--rw group-wildcard         uint8
          +--rw vni-type               virtual-type
          +--rw vni-value              uint32
          +--rw multicast-overlay
          |  +--rw nodes-information
          |  |  +--rw ingress-node?   inet:ip-address
          |  |  +--rw egress-nodes* [egress-node]
          |  |     +--rw egress-node    inet:ip-address
          |  +--rw bier-information
          |  |  +--rw sub-domain?     sub-domain-id
          |  |  +--rw ingress-node?   bfr-id
          |  |  +--rw egress-nodes* [egress-node]
          |  |     +--rw egress-node    bfr-id
          |  +--rw overlay-technology
          |     +--rw (overlay-tech-type)?
          |        +--:(mld)
          |        +--:(mvpn)
          |        +--:(bgp)
          |        +--:(mld-snooping)
          +--rw multicast-transport
          |  +--rw bier
          |  |  +--rw sub-domain?        sub-domain-id
          |  |  +--rw (encap-type)?
          |  |  |  +--:(mpls)
          |  |  |  +--:(non-mpls)
          |  |  |  +--:(ipv6)
          |  |  +--rw bitstringlength?   uint16
          |  |  +--rw set-identifier?    si
          |  |  +--rw ecmp?              boolean
          |  |  +--rw frr?               boolean
          |  +--rw bier-te
          |  |  +--rw sub-domain?        sub-domain-id
          |  |  +--rw (encap-type)?
          |  |  |  +--:(mpls)
          |  |  |  +--:(non-mpls)
          |  |  +--rw bitstringlength?   uint16
          |  |  +--rw set-identifier?    si
          |  |  +--rw ecmp?              boolean
          |  |  +--rw frr?               boolean
          |  +--rw cisco-mode
          |  |  +--rw p-group?            inet:ip-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?   uint16
             +--rw isis
             |  +--rw topology-id?   uint16
             +--rw babel
             +--rw pim
```

5. Multicast Information Model

```
<CODE BEGINS> file "ietf-multicast-information.yang"
   module ietf-multicast-information {

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

    prefix multicast-info;

    import ietf-inet-types {
        prefix "inet";
    }

    organization " IETF MBONED( MBONE Deployment ) Working Group";
    contact
        "WG List:  <mailto:bier@ietf.org>
         WG Chair: Greg Shepherd
                   <mailto:gjshep@gmail.com>
         WG Chair: Leonard Giuliano
                   <mailto:lenny@juniper.net>

         Editor:   Zheng Zhang
                   <mailto:zhang.zheng@zte.com.cn>
         Editor:   Cui Wang
                   <mailto:wang.cui1@zte.com.cn>
         Editor:   Ying Cheng
                   <mailto:chengying10@chinaunicom.cn>
        ";
```

```
    description
        "This module contains a collection of YANG definitions for
         managing multicast information.";

    revision 2017-08-20 {
        description
        "Add BGP and MLD-snooping overlay and BIER-TE transport.";
        reference "https://tools.ietf.org/html/draft-zhang-mboned-multicast-
info-model";
    }

    revision 2016-12-08 {
        description
        "Initial version.";
        reference "https://tools.ietf.org/html/draft-zhang-mboned-multicast-
info-model";
    }
/*feature*/
    grouping general-multicast {
        description "The general multicast address information.";
        leaf source-address {
            type inet:ip-address;
            description "The address of multicast source. The value set to zero
              means that the receiver interests in all source that relevant to
              one group.";
        }
        leaf source-wildcard {
            type uint8;
            description "The wildcard information of source.";
        }
        leaf group-address {
            type inet:ip-address;
            description "The address of multicast group.";
        }
        leaf group-wildcard {
            type uint8;
            description "The wildcard information of group.";
        }
    }

        grouping m-addr {
        description "The vpn multicast information.";
        leaf vpn-id {
            type uint32;
            description "The vpn-id of the multicast flow.
              If there is global instance, the vpnid value should be zero.";
        }
        uses general-multicast;
    }
```

```
    typedef virtual-type {
        type enumeration {
            enum "vxlan" {
                description "The vxlan type.";
            }
            enum "virtual subnet" {
                description "The nvgre type";
            }
            enum "vni" {
                description "The geneve type";
            }
        }
        description "The collection of virtual network type.";
    }

    grouping multicast-nvo3 {
        description "The nvo3 multicast information.";
        leaf vni-type {
            type virtual-type;
             description "The type of virtual network identifier. Include the
Vxlan
              NVGRE and Geneve.";
        }
        leaf vni-value {
            type uint32;
            description "The value of Vxlan network identifier, virtual subnet
ID
            or virtual net identifier.";
        }
    }

    grouping multicast-feature {
        description
          "This group describe the different multicast information
           in various deployments.";
        uses m-addr;
        uses multicast-nvo3;
    }

    grouping ip-node {
        description "The IP information of multicast nodes.";
        leaf ingress-node {
            type inet:ip-address;
            description "The ingress node of 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 "This ID information of one adjacency.";

            leaf egress-node {
                type inet:ip-address;
                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.";
            }
        }
    }
    /* should import from BIER yang */
    typedef bfr-id {
        type uint16;
        description "The BFR id of nodes.";
    }

    typedef si {
        type uint16;
        description
          "The type for set identifier";
    }

    typedef sub-domain-id {
        type uint16;
        description
          "The type for sub-domain-id";
    }

    typedef bit-string {
        type uint16;
        description
          "The bit mask of one bitstring.";
    }

    grouping bier-node {
        description "The BIER information of multicast nodes.";
        leaf sub-domain {
            type sub-domain-id;
            description "The sub-domain that this multicast flow belongs to.";
        }
        leaf ingress-node {
            type bfr-id;
            description "The ingress node of multicast flow. This is the
              BFR-id of ingress nodes.";
        }
        list egress-nodes {
```

```
            key "egress-node";
            description "This ID information of one adjacency.";

            leaf egress-node {
                type bfr-id;
                description
                  "The egress multicast nodes of multicast flow.
                  This is the BFR-id of egress nodes.";
            }
        }
    }

    grouping overlay-tech {
        description "The possible overlay technologies for multicast service.";
        choice overlay-tech-type {
            case mld {
                description "MLD technology is used for multicast overlay";
            }
            case mvpn {
                description "MVPN technology is used for multicast overlay";
            }
            case bgp {
                description "BGP technology is used for multicast overlay";
            }
            case mld-snooping {
                description "MLD snooping technology is used for multicast
overlay";
            }
            description "The collection of multicast overlay technology";
        }
    }

    grouping multicast-overlay {
        description "The node information that connect the ingress multicast
          flow, and the nodes information that connect the egress multicast
          flow.";
        /*uses multicast-feature;*/
        container nodes-information {
            description "The ingress and egress nodes information.";
            uses ip-node;
        }
        container bier-information {
            description "The ingress and egress BIER nodes information.";
            uses bier-node;
        }
        container overlay-technology {
            description "The possible overlay technologies for multicast
service.";
            uses overlay-tech;
        }
```

```
    }


/*transport*/

    grouping transport-bier {
        description "The BIER transport information.";
        leaf sub-domain {
            type sub-domain-id;
            description "The subdomain id that this multicast flow belongs to.";
        }
        choice encap-type {
            case mpls {
                description "The BIER forwarding depend on mpls.";
            }
            case non-mpls {
                description "The BIER forwarding depend on non-mpls.";
            }
            case ipv6 {
                description "The BIER forwarding depend on IPv6.";
            }
            description "The encapsulation type in BIER.";
        }
         leaf bitstringlength {
            type uint16;
            description "The bitstringlength used by BIER forwarding.";
        }
        leaf set-identifier {
            type si;
            description "The set identifier used by this multicast flow.";
        }
        leaf ecmp {
            type boolean;
            description "The capability of ECMP.";
        }
        leaf frr {
            type boolean;
            description "The capability of fast re-route.";
        }
    }

        grouping transport-bier-te {
        description "The BIER-TE transport information.";
        leaf sub-domain {
            type sub-domain-id;
            description "The subdomain id that this multicast flow belongs to.";
        }
        choice encap-type {
```

```
            case mpls {
                description "The BIER-TE forwarding depend on mpls.";
            }
            case non-mpls {
                description "The BIER-TE forwarding depend on non-mpls.";
            }
            description "The encapsulation type in BIER-TE.";
        }
         leaf bitstringlength {
            type uint16;
            description "The bitstringlength used by BIER-TE forwarding.";
        }
        leaf set-identifier {
            type si;
            description "The set identifier used by this multicast flow,
especially in BIER TE.";
        }
        leaf ecmp {
            type boolean;
            description "The capability of ECMP.";
        }
        leaf frr {
            type boolean;
            description "The capability of fast re-route.";
        }
    }

    grouping transport-pim {
        description "The requirement information of pim transportion.";
        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 mldp-tunnel-feature {
        description "The tunnel feature.";
        leaf mldp-tunnel-id {
            type uint32;
            description "The tunnel id that correspond this flow.";
        }
        leaf mldp-frr {
            type boolean;
            description "If the fast re-route function should be supported.";
        }
```

```
        leaf mldp-backup-tunnel {
            type boolean;
            description "If the backup tunnel function should be supported.";
        }
    }

    grouping p2mp-te-tunnel-feature {
        description "The tunnel feature.";
        leaf te-tunnel-id {
            type uint32;
            description "The tunnel id that correspond this flow.";
        }
        leaf te-frr {
            type boolean;
            description "If the fast re-route function should be supported.";
        }
        leaf te-backup-tunnel {
            type boolean;
            description "If the backup tunnel function should be supported.";
        }
    }

    /*typedef sub-domain-id {
        type uint16;
        description
          "The type for sub-domain-id";
    }*/

    grouping transport-mpls {
        description "The mpls transportion information.";
        choice mpls-tunnel-type {
            case mldp {
                uses mldp-tunnel-feature;
                description "The mldp tunnel.";
            }
            case p2mp-te {
                uses p2mp-te-tunnel-feature;
                description "The p2mp te tunnel.";
            }
            description "The collection types of mpls tunnels";
        }
    }

    grouping cisco-multicast {
        description "The Cisco MDT multicast information in RFC6037.";
        leaf p-group {
            type inet:ip-address;
            description "The address of p-group.";
```

```
        }
    }

        grouping transport-cisco-mode {
        description "The transport information of Cisco mode, RFC6037.";
        uses cisco-multicast;
        uses transport-pim;
    }

    grouping multicast-transport {
        description "The transport information of multicast service.";
        container bier {
            uses transport-bier;
            description "The transport technology is BIER.";
        }
        container bier-te {
            uses transport-bier-te;
            description "The transport technology is BIER-TE.";
        }
        container cisco-mode {
            uses transport-cisco-mode;
            description "The transport technology is cisco-mode.";
        }
        container mpls {
            uses transport-mpls;
            description "The transport technology is mpls.";
        }
        container pim {
            uses transport-pim;
            description "The transport technology is PIM.";
        }
    }

/*underlay*/
    grouping underlay-bgp {
        description "Underlay information of BGP.";
    }

    grouping underlay-ospf {
        description "Underlay information of OSPF.";
        leaf topology-id {
            type uint16;
            description "The topology id of ospf instance.";
        }
    }

    grouping underlay-isis {
        description "Underlay information of ISIS.";
```

```
        leaf topology-id {
            type uint16;
            description "The topology id of isis instance.";
        }
    }

    grouping underlay-babel {
        description "Underlay information of Babel.";
        /* If there are some necessary information should be defined? */
    }

    grouping underlay-pim {
        description "Underlay information of PIM.";
        /* If there are some necessary information should be defined? */
    }

    grouping multicast-underlay {
        description "The underlay information relevant multicast service.";
        leaf underlay-requirement {
            type boolean;
            description "If the underlay technology should be required.";
        }
        container bgp {
            uses underlay-bgp;
            description "The underlay technology is BGP.";
        }
        container ospf {
            uses underlay-ospf;
            description "The underlay technology is OSPF.";
        }
        container isis {
            uses underlay-isis;
            description "The underlay technology is ISIS.";
        }
        container babel {
            uses underlay-babel;
            description "The underlay technology is Babel.";
        }
        container pim {
            uses underlay-pim;
            description "The underlay technology is PIM.";
        }
    }

    container multicast-information {
        description "The model of multicast service. Include overlay, transport
and underlay.";

        list multicast-info{
```

```
            key "vpn-id source-address source-wildcard group-address group-
wildcard vni-type vni-value";
            uses multicast-feature;
            description "The detail multicast information.";

            container multicast-overlay {
                description "The overlay information of multicast service.";
                uses multicast-overlay;
            }
            container multicast-transport {
                description "The transportion of multicast service.";
                uses multicast-transport;
            }
            container multicast-underlay {
                description "The underlay of multicast service.";
                uses multicast-underlay;
            }
        }
    }
}
<CODE ENDS>
```

6. Notifications

TBD.

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-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-07 (work in progress), June 2017.

[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-02 (work in progress), August 2017.

[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-08 (work in progress), April 2017.

[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>.

[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>.

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