MBONE WG
Internet-Draft
Intended status: Standards Track
Expires: August 31, 2016

Zheng. Zhang
Cui. Wang
ZTE Corporation

February 28, 2016

Multicast Service YANG
draft-zhang-mbone-multicast-service-yang-01

Abstract

The purpose of this document defines a general multicast YANG service model. This document intent to provide a general and all-round multicast service model, and guides the deployment of multicast technology. This model includes many of the existed multicast technology. But this document does not define any specifically and detailed YANG model of multicast protocol. The multicast model can be used with YANG models of specifically multicast technologies, such as PIM, MLD, and BIER and so on.

This document proposes a general and all-round multicast service YANG model, which provides explanations and guidelines for the deployment of multicast service in all kinds of multicast scenarios. The multicast technologies include BIER multicast, PIM multicast, MPLS multicast and so on. And also, there defines several possible RPCs about how to interact between multicast service model and multicast device model.

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 August 31, 2016.

Table of Contents

1. Introduction

The purpose of this document defines a general multicast YANG service model. This document intent to provide a general and all-round multicast service model, and guides the deployment of multicast technology. This model includes many of the existed multicast technology. But this document does not define any specifically and detailed YANG model of multicast protocol. The multicast model can be used with YANG models of specifically multicast technologies, such as PIM, MLD, and BIER and so on.

This document proposes a general and all-round multicast service YANG model, which provides explanations and guidelines for the deployment of multicast service in all kinds of multicast scenarios. The multicast technologies include BIER multicast, PIM multicast, MPLS multicast and so on. And also, there defines several possible RPCs about how to interact between multicast service model and multicast device model.

2. Design of the multicast service model

This model includes multicast overlay, the transport layer and the possible underlay information.

Multicast overlay defines the feature of multicast flow, such as VPNID, multicast source and group information. Or the new feature such as virtual network identifier used in NVO3. Multicast overlay also define the node overlay information.

Multicast overlay defines the feature of multicast flow, such as (vpnid, multicast source and multicast group) information, (ingress-node,egress-nodes) nodes information. Additionally, BIER information including (Subdomain,ingress-node BFR-id,egress-nodes BFR-id) is also included to provide BIER multicast service. 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 transport defines the transport technologies that may be used to forward multicast flow, including BIER forwarding, mpls forwarding or pim forwarding and so on.

Multicast underlay defines the possible technologies that may be used to interaction the necessary information, such as OSPF, ISIS, and BGP and so on.

```
module: ietf-multicast-service
   +--rw multicast-service
      +--rw multicast-overlay
      |  +--rw (feature-type)
      |  |  +--:(pure-multicast)
      |  |  |  +--rw vpn-id                uint32
      |  |  |  +--rw source-address        inet:ip-address
      |  |  |  +--rw source-wildcard?      uint8
      |  |  |  +--rw group-address         inet:ip-address
      |  |  |  +--rw group-wildcard?       uint8
      |  |  +--:(nvo3)
      |  |     +--rw vni-type              virtual-type
      |  |     +--rw vni-value             uint32
      |  +--rw nodes-information
      |  |  +--rw ingress-node    inet:ip-address
      |  |  +--rw egress-nodes* [number]
      |  |     +--rw number         uint32
      |  |     +--rw egress-node    inet:ip-address
      |  +--rw bier-information
      |  |  +--rw sub-domain      sub-domain-id
      |  |  +--rw ingress-node    bfr-id
      |  |  +--rw egress-nodes* [number]
```

```
|  |     +--rw number         uint32
|  |     +--rw egress-node    bfr-id
|  +--rw overlay-technology
|     +--rw (overlay-tech-type)
|        +--:(mld)
|        +--:(mvpn)
+--rw multicast-transport
|  +--rw (transport-type)
|     +--:(bier)
|     |  +--rw sub-domain          sub-domain-id
|     |  +--rw (encap-type)
|     |  |  +--:(mpls)
|     |  +--rw bitstringlength?    uint16
|     |  +--rw set-identifier?     si
|     |  +--rw ecmp?               boolean
|     |  +--rw frr?                boolean
|     +--:(cisco-mode)
|     |  +--rw p-group             inet:ip-address
|     |  +--rw graceful-restart?   boolean
|     |  +--rw bfd?                boolean
|     +--:(mpls)
|     |  +--rw (mpls-tunnel-type)?
|     |     +--:(mldp)
|     |     |  +--rw tunnel-id?          uint32
|     |     |  +--rw frr?                boolean
|     |     |  +--rw backup-tunnel?      boolean
|     |     +--:(p2mp-te)
|     |        +--rw tunnel-id?          uint32
|     |        +--rw frr?                boolean
|     |        +--rw backup-tunnel?      boolean
|     +--:(pim)
|        +--rw graceful-restart?   boolean
|        +--rw bfd?                boolean
+--rw multicast-underlay
   +--rw underlay-requirement?   boolean
   +--rw (underlay-type)
      +--:(bgp)
      +--:(ospf)
      |  +--rw topology-id?            uint16
      +--:(isis)
      |  +--rw topology-id?            uint16
      +--:(pim)
```

2.1. Multicast overlay layer

This layer defines the feature of multicast service, and the possible overlay protocol may be used. The feature of multicast includes the

source and group information. And specific for nvo3, the feature of multicast service may be virtual network identifier.

The ingress and egress nodes information include the IP address of nodes and PEs. In BIER scenario, the nodes information may be BFR-ids.

The overlay technology, until now, MVPN and MLD are necessary.

2.2. Multicast transport layer

BIER is the transport layer technology. MPLS and PIM also is transport layer technology. The choice of transport layer protocol can be flexible.

2.3. Multicast underlay layer

This layer has a tight connection with the underlay protocol.

3. Notifications

TBD.

4. Multicast service YANG model

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

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

    prefix multicast-service;

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

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

    revision 2016-02-29 {
        description
        "Initial version.";
        reference "https://tools.ietf.org/html/draft-zhang-mboned-mservice-
yang";
    }
/*feature*/
    grouping general-multicast {
        description "The general multicast address information.";
        leaf source-address {
            type inet:ip-address;
            mandatory true;
            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;
            mandatory true;
            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;
            mandatory true;
```

```
            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;
            mandatory true;
            description "The type of virtual network identifier. Include the
Vxlan
              NVGRE and Geneve.";
        }
        leaf vni-value {
            type uint32;
            mandatory true;
            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.";
        choice feature-type {
            mandatory true;
            case pure-multicast {
                uses m-addr;
            }
            case nvo3 {
```

```
            uses multicast-nvo3;
        }
```

```
            description "The collection of all possible multicast feature.";
        }
    }

    grouping ip-node {
        description "The IP information of multicast nodes.";
        leaf ingress-node {
            type inet:ip-address;
            mandatory true;
            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 "number";
            description "This ID information of one adjacency.";
            leaf number {
                type uint32;
                mandatory true;
                description "The number of egress nodes.";
            }
            leaf egress-node {
                type inet:ip-address;
                mandatory true;
                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;
            mandatory true;
            description "The sub-domain that this multicast flow belongs to.";
        }
        leaf ingress-node {
            type bfr-id;
            mandatory true;
            description "The ingress node of multicast flow. This is the
              BFR-id of ingress nodes.";
        }
        list egress-nodes {
            key "number";
            description "This ID information of one adjacency.";
            leaf number {
                type uint32;
                mandatory true;
                description "The number of egress nodes.";
            }
            leaf egress-node {
                type bfr-id;
                mandatory true;
                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 {
            mandatory true;
            case mld {
                description "MLD technology is used for multicast overlay";
            }
            case mvpn {
```

```
                description "MVPN 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*/

    typedef bier-encap-type {
        type enumeration {
            enum "mpls" {
                description "The mpls forwarding function is used in BIER.";
            }
        }
        description "The encapsulation type of BIER transportion.";
    }
    grouping transport-bier {
        description "The BIER transport information.";
        leaf sub-domain {
            type sub-domain-id;
            mandatory true;
            description "The subdomain id that this multicast flow belongs
to.";
        }
        choice encap-type {
            mandatory true;
            case mpls {
```

```
            description "The BIER forwarding depend on mpls.";
        }
```

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

```
    }
```

```
    grouping transport-mpls {
        description "The mpls transportion information.";
        choice mpls-tunnel-type {
            case mldp {
                uses tunnel-feature;
                description "The mldp tunnel.";
            }
            case p2mp-te {
                uses 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;
            mandatory true;
            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 transport-mvpn {
        description "The transport information of MVPN";
        choice mvpn-type {
            case cisco-mode {
                uses transport-pim;
            }
            case mpls-bgp-mode {
                uses trans-mpls;
            }
            description "The transport solution of MVPN.";
        }
    }*/

    grouping multicast-transport {
        description "The transport information of multicast service.";
        choice transport-type {
            mandatory true;
            case bier {
```

```
                uses transport-bier;
            }
            case cisco-mode {
                uses transport-cisco-mode;
            }
            case mpls {
                uses transport-mpls;
            }
            case pim {
                uses transport-pim;
            }
            description "The collection of all possible multicast transport
technology.";
        }
    }

/*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-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.";
        }
```

```
        choice underlay-type {
```

```
            mandatory true;
            case bgp {
                uses underlay-bgp;
            }
            case ospf {
                uses underlay-ospf;
            }
            case isis {
                uses underlay-isis;
            }
            case pim {
                uses underlay-pim;
            }
            description "The collection of all possible multicast underlay
technology.";
        }
    }

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

5. Normative References

[I-D.chh-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-chh-bier-bier-yang-02 (work in progress), November 2015.

[I-D.ietf-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., P, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-03 (work in progress), January 2016.

[I-D.ietf-pim-yang]
Liu, X., McAllister, P., and A. Peter, "A YANG data model for Protocol-Independent Multicast (PIM)", draft-ietf-pim-yang-00 (work in progress), February 2016.

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

[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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6513>.

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

Authors' Addresses

Zheng(Sandy) 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: wang.cui1@zte.com.cn