

IETF RTGWG  
Internet-Draft  
Intended status: Standards Track  
Expires: September 20, 2018

Fangwei Hu  
RongRong Hua  
ZTE Corporation  
Shujun Hu  
Rong Gu  
China Mobile  
Mar 19, 2018

YANG Data Model for Configuration Interface of Control-Plane and User-  
Plane separation BNG  
draft-hu-rtgwg-cu-separation-yang-model-03.txt

## Abstract

This document defines the YANG data model for operation management of Control-Plane and User-Plane separation BNG (Broadband Network Gateway).

## Status of This Memo

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

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

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

This Internet-Draft will expire on September 20, 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 . . . . .	2
2. Conventions used in this document . . . . .	4
2.1. Terminology . . . . .	4
2.2. Requirements Language . . . . .	4
3. Design Tree . . . . .	4
3.1. Global Configuration . . . . .	4
3.2. BNG-UP Interface Configuration . . . . .	5
3.3. Control Channel Configuration . . . . .	5
3.4. Service Channel Configuration . . . . .	6
3.5. Multicast Service . . . . .	6
3.6. PPPOX Configuration . . . . .	7
3.7. Acl Configuration . . . . .	8
3.8. QoS Configuration . . . . .	9
4. vBNG YANG Data Model . . . . .	9
5. Security Considerations . . . . .	29
6. Acknowledgements . . . . .	30
7. IANA Considerations . . . . .	30
8. References . . . . .	30
8.1. Normative References . . . . .	30
8.2. Informative References . . . . .	31
Authors' Addresses . . . . .	32

## 1. Introduction

The main idea of BNG Control-Plane and User-Plane separation is to extract and centralize the user management functions of multiple BNG devices, forming an unified and centralized control plane (CP), while the traditional router's Control Plane and forwarding plane are both preserved on BNG devices in the form of a user plane (UP). We name the control-Plane and User-plane separation BNG as vBNG.

The architecture of Control-plane and user-plane separated BNG is shown as the following figure.

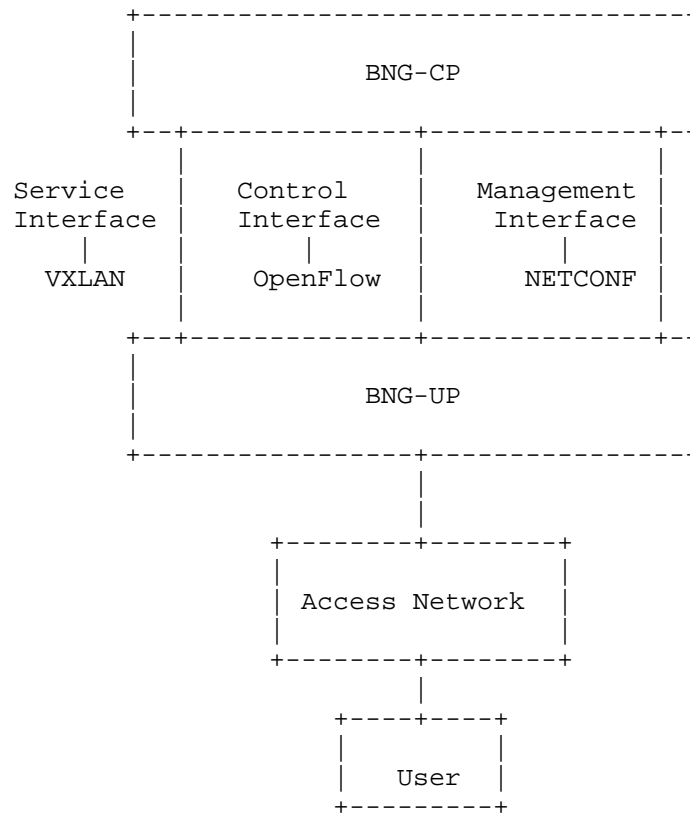


Figure 1: Architecture of C/U separated BNG

There are three interfaces between BNG-CP and BNG-UP: Service interface, control interface and management interface. The service interface is used to carry PPPoE/IPoE dialup packets between user plane and control plane. The requirement and possible solution is defined in the [I-D.huang-nvo3-vxlan-extension-for-vbras]. Control interface is used for setting forwarding entries of user plane through OpenFlow or other protocols. Management interface is used by BNG-CP to carry out related configurations of BNG-UP through NETCONF protocol [RFC6241].

This document defines the YANG data model for vBNG(BNG-CP and BNG-UP). There are three types of YANG data model for vBNG in this document: The YANG data models for BNG-CP, the YANG data models for BNG-UP by network management directly, and the YANG data models for BNG-UP through the management interfaces among the BNG-UP and BNG-CP.

The YANG data models through the management interfaces include: The BNG-UP interfaces configuration(Section 3.2), the controller channel configuration(Section 3.3), the ACL configuration for BNG-UP(Section 3.6) and QoS configuration for BNG-UP (section 3.7), etc.

## 2. Conventions used in this document

### 2.1. Terminology

**BNG:** Broadband Network Gateway. A broadband remote access server routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet service provider's (ISP) network.

**BNG-CP:** BNG Control Plane. The BNG-CP is a user control management component which support to manage UP's resources such as the user entry and forwarding policy.

**BNG-UP:** BNG User Plane. BNG-UP is a network edge and user policy implementation component.

**vBNG:** Virtualization Broadband Network Gateway. An vBNG is to extract and centralize the user management functions of multiple BNG devices, and to form an unified and centralized control plane (CP). The vBNG devices include BNG-UP and BNG-CP.

### 2.2. Requirements Language

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.

## 3. Design Tree

### 3.1. Global Configuration

The BNG-UP or BNG-CP part can be a physical or logical network element. The LNE model [I-D.ietf-rtgwg-lne-model] is augmented to define the YANG data models for BNG-UP and BNG-CP in this document.

```

module: ietf-vbng
augment /lne:logical-network-elements/lne:logical-network-element:
  +--rw ietf-vbng
    +--rw bng-cp
      | +--rw bng-cp-name?   string
      | +--rw enable?       boolean
    +--rw bng-up!
      | +--rw bng-up* [shelf-no]
      | | +--rw shelf-no      uint8
      | | +--rw bng-up-name?  string
      | | +--rw netconf-server!
      | | | +--rw ip          inet:ipv4-address
      | | | +--rw user-name?  string
      | | | +--rw password?   string
      | | | +--rw port?       uint32
      | | +--rw keepalive-sink? enumeration

```

### 3.2. BNG-UP Interface Configuration

The BNG-UP interface configuration is to configure the basic interface informations of BNG-UP element, such as interface name, the VLAN parameters for the sub-interface. The BNG-UP interface data models are configured through the management interfaces between BGN-UP and BNG-CP by netconf protocol.

The tree structure for BNG-UP interface configuration is as following:

```

+--rw interfaces
  | +--rw interface* [name]
  | | +--rw name      if:interface-ref
  | | +--rw ethernet
  | | | +--rw lacp?   boolean
  | | +--rw mac-offset? uint32
  | | +--rw vlans
  | | | +--rw tag* [index]
  | | | | +--rw index    uint8
  | | | | +--rw tag
  | | | | | +--rw tag-type? string
  | | | | +--rw vlan-id?  vlan-id

```

### 3.3. Control Channel Configuration

The control channel is to configure the control channel parameters. The control channel data models are configured through the management interfaces between BGN-UP and BNG-CP by netconf protocol

The control channel parameters include: name, id, port, disconnect. The tree structure for control channel configuration parameters are as following:

```

+--rw control-channel
|   +--rw address-family* [af]
|   |   +--rw af          address-family-type
|   |   +--rw control-ip?  inet:ip-address
|   +--rw name?           string
|   +--rw id?             uint32
|   +--rw port?           uint32
|   +--rw disconnect
|   |   +--rw (response-delay)?
|   |   |   +--:(nolimitflag)
|   |   |   |   +--rw forever?      enumeration
|   |   |   +--:(range)
|   |   |   +--rw delay-time?      uint32

```

### 3.4. Service Channel Configuration

The VXLAN tunnel is the suggestion service interface protocol between BNG-CP and BNG-UP. The VXLAN tunnel parameters include: tunnel-source-ip, tunnel-destination-ip, vxlan-id, vxlan-tunnel-id, vxlan-tunnel-name, etc.

```

+--rw vxlan-channel* [vxlan-tunnel-id]
|   +--rw vxlan-tunnel-id      uint32
|   +--rw vxlan-tunnel-name?   string
|   +--rw address-family* [af]
|   |   +--rw af              address-family-type
|   |   +--rw tunnel-source-ip?  inet:ip-address
|   |   +--rw tunnel-destination-ip?  inet:ip-address
|   |   +--rw bind-vxlan-id* [vxlan-id]
|   |   |   +--rw vxlan-id      vxlan-id

```

### 3.5. Multicast Service

The multicast service parameters are configured through management interfaces. Both IGMP and MLD multicast services are supported by bng. The multicast service YANG data model are only configured to BNG-CP.

```

+--rw multicast-service
|   +--rw multicast-global
|   |   +--rw keepalive-timer?    enumeration
|   |   +--rw query-interval?    uint16
|   +--rw igmp-service-profile
|   |   +--rw igmp-service-profile* [service-profile-num]
|   |   |   +--rw service-profile-num    uint8
|   |   |   +--rw access-group?          string
|   |   |   +--rw description?           string
|   |   |   +--rw max-groups?            uint8
|   |   |   +--rw max-prw-groups?        uint8
|   |   |   +--rw prw-group* [ipv4-address ipv4-address-mask]
|   |   |   |   +--rw ipv4-address        inet:ipv4-address
|   |   |   |   +--rw ipv4-address-mask    inet:ipv4-address
|   |   |   |   +--rw max-prw-count?      uint32
|   |   |   |   +--rw prw-resume-interval? uint32
|   |   |   +--rw static-group* [ipv4-address]
|   |   |   |   +--rw ipv4-address        inet:ipv4-address
|   +--rw mld-service-profile
|   |   +--rw mld-service-profile* [service-profile-num]
|   |   |   +--rw service-profile-num    uint8
|   |   |   +--rw access-group?          string
|   |   |   +--rw description?           string
|   |   |   +--rw max-groups?            uint8
|   |   |   +--rw max-prw-groups?        uint8
|   |   |   +--rw prw-group* [ipv6-address ipv6-prefixlen]
|   |   |   |   +--rw ipv6-address        inet:ipv6-address
|   |   |   |   +--rw ipv6-prefixlen      uint8
|   |   |   |   +--rw max-prw-count?      uint32
|   |   |   |   +--rw prw-resume-interval? uint32
|   |   |   +--rw static-group* [ipv6-address]
|   |   |   |   +--rw ipv6-address        inet:ipv6-address

```

### 3.6. PPPOX Configuration

The pppox parameters are only configured to BNG-CP. The parameters are as following:

```

+--rw bras-pppox
|   +--rw pppox-ipv6cp-cfg
|   |   +--rw ipv6cp-extension?    enumeration
|   |   +--rw ipv6cp-aftr?         uint8
|   |   +--rw ipv6cp-ipv6-address? uint8
|   |   +--rw ipv6cp-ipv6-prefix?  uint8
|   |   +--rw ipv6-dns!
|   |   |   +--rw ipv6cp-ipv6-dns-secondary    uint8
|   |   |   +--rw ipv6cp-ipv6-dns-primary      uint8

```

```

|   +--rw ipv4-dns!
|   |   +--rw ipv6cp-ipv4-dns-primary      uint8
|   |   +--rw ipv6cp-ipv4-dns-secondary   uint8
+--rw pppox-ipcp-cfg
|   +--rw ipcp-flag?                       enumeration
|   +--rw option-type?                     uint8
|   +--rw br-address?                      inet:ipv4-address
|   +--rw ipv6-rapid-deployment!
|   |   +--rw v4-mask-len                  uint8
|   |   +--rw v6-pref                     inet:ipv6-address
|   |   +--rw v6-mask-len                  uint8
+--rw pppoe-switch
|   +--rw delay-time?                      uint16
|   +--rw keepalive-timer?                 enumeration
|   +--rw ppp-max-payload?                 enumeration
|   +--rw service?                        enumeration
|   +--rw ppp-mru-verify?                  enumeration
|   +--rw keepalive-fast-reply?            enumeration
+--rw pppoe-cfg* [template]
|   +--rw template                        uint32
|   +--rw ppp-authentication?              enumeration
|   +--rw ppp-check-magic-num?             enumeration
|   +--rw ppp-mru?                         uint32
|   +--rw pppoe-ac-name?                   string
|   +--rw pppoe-service-name-omit?         enumeration
|   +--rw pppoe-ac-cookie-check?           enumeration
|   +--rw pppoe-password-string?           string
|   +--rw pppoe-username-string?           string
|   +--rw (ppp-quick-redial)?
|   |   +--:(quick-redial-disable)
|   |   |   +--rw ppp-quick-redial-disable? enumeration
|   |   +--:(fast-response)
|   |   |   +--rw ppp-fast-response?         enumeration
|   |   |   +--rw ppp-quick-redial-enable?   enumeration
+--rw ppp-keepalive
|   +--rw ppp-keepalive-timer?             uint32
|   +--rw ppp-keepalive-count?             uint16
+--rw ppp-timeout
|   +--rw ppp-timeout-negtimeoutsec?       uint8
|   +--rw ppp-timeout-authentication?      uint8

```

### 3.7. Acl Configuration

The acl information for BNG-UP is configured through netconf protocol from BNG-CP. The ACL information includes ipv4-acl, ipv6-acl, link-acl, etc. The YANG data model for ACL refers to [I-D.ietf-netmod-acl-model]



### 3.8. QoS Configuration

The QoS information for BNG-UP is also configured through netconf protocol from BNG-CP. The support QoS information includes IP-DSCP, MPLS, VPLS, VPWS etc. The YANG data model for QoS refers to [I-D.asechoud-rtgwg-qos-model]

## 4. vBNG YANG Data Model

```
<CODE BEGINS> file "ietf-vbng@2018-03-18.yang"
module ietf-vbng{
  namespace "urn:ietf:params:xml:ns:yang:ietf-vbng";
  prefix "vbng";

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

  import ietf-interfaces {
    prefix if;
  }

  import ietf-logical-network-element {
    prefix lne;
  }

  organization
    "IETF NETCONF Working Group";

  contact
    "
      WG List:  <mailto:netconf@ietf.org>

      Editor:   Fangwei Hu
                <mailto:hu.fangwei@zte.com.cn>
    ";

  description
    "The YANG module defines a generic configuration
    model for vbng";

  revision 2018-03-20{
    description "Change the control channel protocol name";
    reference
      "draft-hu-rtgwg-cu-separation-yang-model-03";
  }
}
```

```
    revision 2018-02-27{
    description "Correct some type of nodes.";
    reference
        "draft-hu-rtgwg-cu-separation-yang-model-02";
    }

    revision 2018-01-18{
    description "add multicast service configuration and pppox configuration,
        and update the OpenFlow channel parameters.";
    reference
        "draft-hu-rtgwg-cu-separation-yang-model-01";
    }

revision 2017-07-16{
    description "Initial revision";
    reference
        "draft-hu-rtgwg-cu-separation-yang-model-00";
    }

/* Typedefs */

    typedef vlan-id {
    type uint16 {
        range "0..4094";
    }
    description
        "Typedef for VLAN ID.";
    }

    typedef vxlan-id {
    type uint32;
    description
        "Typedef for VxLAN ID.";
    }

    typedef address-family-type {
    type enumeration {
        enum ipv4 {
            description
                "IPv4";
        }
        enum ipv6 {
            description
                "IPv6";
        }
    }
    description
        "Typedef for address family type.";
```

```
    }

    /* Configuration Data */

    augment /lne:logical-network-elements/lne:logical-network-element {
    container ietf-vbng{

        container bng-cp {
        leaf bng-cp-name {
            type string;
            description "configure bng-cp name";
        }

        leaf enable {
            type boolean;
            description "'true' to support bng control plane and user plane
separation";
        }
        description "configure bng-cp";
    }

    container bng-up {
        presence bng-up;
        list bng-up {
            key shelf-no ;
            leaf shelf-no {
                type uint8 {
                    range 1..127 ;
                }
                description 'Configure shelf-no of forwarder,1-127.';
            }
        }

        leaf bng-up-name {
            type string {
                length 1..31 ;
            }
            description 'Configure bng up name.' ;
        }
    }

    container netconf-server {
        presence netconf-server ;
        leaf ip {
            type inet:ipv4-address;
            mandatory true ;
            description 'Configure ip address of netconf server.';
        }

        leaf user-name {
```

```
    type string {
      length 1..65 ;
    }
    description 'configure user name, default: "who".';
  }

  leaf password {
    type string {
      length 3..32 ;
    }
    description 'configure password, default: "who".';
  }

  leaf port {
    type uint32;
    description 'Configure port.';
  }
  description 'Configure netconf server.';
}

leaf keepalive-sink {
  type enumeration {
    enum enable {
      value 1 ;
      description 'enable the keepalive-sink function';
    }
    enum disable {
      value 0 ;
      description 'disable keepalive-sink function';
    }
  }
  description "configure keepalive-sink";
}
description "configure bng up list";
}
description "configure bng up container";
}

container interfaces {
  list interface {
    key name;
    leaf name {
      type if:interface-ref;
      description "interface name";
    }
    container ethernet {
      leaf lacp {
        type boolean;
      }
    }
  }
}
```

```

        description "enable lacp function";
    }
        description "configure ethernet interface";
    }
    leaf mac-offset {
        type uint32;
        description "configure mac offset";
    }

    container vlans {
        list tag {
            key index;
            max-elements 2;

            leaf index {
                type uint8 {
                    range "0..1";
                }
                must ". = 0 or
count(.../.../tag[index = 0]/index) > 0" {
                    error-message "An inner tag can only be specified if an
                        outer tag has also been specified";
                    description "Ensure that an inner tag cannot be
                        specified without an outer tag'";
                }
            }

            description "The index into the tag stack, outermost tag
                assigned index 0";
        }

        container tag{
            leaf tag-type {
                type string;
                description "tag type";
            }
            leaf vlan-id {
                type vlan-id;
                description "vlan id value";
            }
        }

        description "tag";
    }
    description "tag list";
}
description "vlans";
}
description "interfaces list";
}
description "interface container";

```

```
    }

    container control-channel {
list address-family {
    key "af";
    leaf af {
        type address-family-type;
        description "Address family type value.";
    }
    leaf control-ip {
        type inet:ip-address;
        description
            "Set the IP address of for control channel protocol session";
    }
        description "Per-af params.";
    }

    leaf name {
        type string;
        description "control channel protocol logical name";
    }

    leaf id {
        type uint32;
        description "id value";
    }

    leaf of-port {
        type uint32;
        description "control channel udp port number";
    }

container disconnect {
    choice response-delay {
        default range ;
        case nolimitflag {
            leaf forever {
                type enumeration {
                    enum forever {
                        value 1 ;
                                description "Delay forever";
                    }
                }
            }
        }
        description 'Delay forever';
    }
    description 'The flag of no time limite';
}
    case range {
```

```
        leaf delay-time {
            type uint32 {
                range 0..2880 ;
            }
            description 'Delay time ,default 240 minutes';
        }
        description 'Set delay time range';
    }
    description 'Set delay time';
}
description 'Set delay time after control channel protocol discon
nect';
}
    description "configure control channel value";
}

list vxlan-channel{
    key vxlan-tunnel-id;
    leaf vxlan-tunnel-id {
        type uint32;
        description
            "Static VxLAN tunnel ID.";
    }

    leaf vxlan-tunnel-name {
        type string;
        description
            "Name of the static VxLAN tunnel.";
    }

    list address-family {
        key "af";
        leaf af {
            type address-family-type;
            description
                "Address family type value.";
        }

        leaf tunnel-source-ip {
            type inet:ip-address;
            description
                "Source IP address for the static VxLAN tunnel";
        }

        leaf tunnel-destination-ip {
            type inet:ip-address;
            description
                "Destination IP address for the static VxLAN tunnel";
        }
    }
}
```

```
        list bind-vxlan-id {
            key vxlan-id;
            leaf vxlan-id {
                type vxlan-id;
                description
                    "VxLAN ID.";
            }
            description
                "VxLAN ID list for the VTEP.";
        }

        description
            "Per-af params.";
    }
    description
        "Configure VxLAN channel";
}

container multicast-service{
    container multicast-global {
        leaf keepalive-timer {
            type enumeration {
                enum start {
                    value 1 ;
                    description 'open switch';
                }
                enum stop {
                    value 2 ;
                    description 'close switch';
                }
                enum always {
                    value 3 ;
                    description 'always keepalive';
                }
            }
            default start ;
            description 'the switch of sending keepalive packet';
        }
        leaf query-interval {
            type uint16 {
                range 1..65535 ;
            }
            default 125 ;
            description 'multicast query interval';
        }
        description 'multicast global configuration';
    }
}
```



```
container igmp-service-profile {
  list igmp-service-profile {
    key service-profile-num ;
    leaf service-profile-num {
      type uint8 {
        range 1..100 ;
      }
      description 'service profile number';
    }
    leaf access-group {
      type string {
        length 1..31 ;
      }
      description 'acl name';
    }

    leaf description {
      type string {
        length 1..31 ;
      }
      description 'description of service profile';
    }

    leaf max-groups {
      type uint8 {
        range 1..128 ;
      }
      default 10 ;
      description 'max groups of the service profile';
    }

    leaf max-prw-groups {
      type uint8 {
        range 1..128 ;
      }
      default 10 ;
      description 'max preview groups of the service profile';
    }

    list prw-group {
      key 'ipv4-address ipv4-address-mask';
      leaf ipv4-address {
        type inet:ipv4-address ;
        description 'ipv4 address of the preview group';
      }

      leaf ipv4-address-mask {
        type inet:ipv4-address;
      }
    }
  }
}
```

```
        description 'ipv4 mask of the preview group';
    }

    leaf max-prw-count {
        type uint32 {
            range 1..1800 ;
        }
        default 10 ;
        description 'max preview times';
    }

    leaf prw-resume-interval {
        type uint32 {
            range 1..86400 ;
        }
        default 60 ;
        description 'preview interval';
    }
    description 'configure preview group';
}

list static-group {
    key ipv4-address ;
    leaf ipv4-address {
        type inet:ipv4-address ;
        description 'ipv4 address of the static group';
    }
    description 'configure static group';
}
description 'configuration of igmp service profile';
}
description 'configuration of igmp service profile';
}

container mld-service-profile {
    list mld-service-profile {
        key service-profile-num ;
        leaf service-profile-num {
            type uint8 {
                range 1..100 ;
            }
            description 'service profile number';
        }
        leaf access-group {
            type string {
                length 1..31 ;
            }
            description 'acl name';
        }
    }
}
```

```
}
leaf description {
    type string {
        length 1..31 ;
    }
    description 'description of service profile';
}
leaf max-groups {
    type uint8 {
        range 1..128 ;
    }
    default 10 ;
    description 'max groups of the service profile';
}
leaf max-prw-groups {
    type uint8 {
        range 1..128 ;
    }
    default 10 ;
    description 'max preview groups of the service profile';
}
list prw-group {
    key 'ipv6-address ipv6-prefixlen'
    ;
    leaf ipv6-address {
        type inet:ipv6-address ;
        description 'ipv6 address of the preview group';
    }
    leaf ipv6-prefixlen {
        type uint8 {
            range 1..128 ;
        }
        description 'ipv6 prefix length';
    }
    leaf max-prw-count {
        type uint32 {
            range 1..1800 ;
        }
        default 10 ;
        description 'max preview times';
    }
    leaf prw-resume-interval {
        type uint32 {
            range 1..86400 ;
        }
        default 60 ;
    }
}
```

```
        description 'preview interval';
    }
    description 'configure preview group';
}
list static-group {
    key ipv6-address ;
    leaf ipv6-address {
        type inet:ipv6-address;
        description 'ipv6 address of the static group';
    }
    description 'configure static group';
}
description 'configuration of mld service profile';
}
description 'configuration of mld service profile';
}
    description 'multicast service configuration';
}

container bras-pppox {
    container pppox-ipv6cp-cfg {
        leaf ipv6cp-extension {
            type enumeration {
                enum enable {
                    value 1 ;
                    description "enable the IPv6cp extension!";
                }
                enum disable {
                    value 0 ;
                    description "disable the IPv6cp extension!";
                }
            }
            default disable ;
            description 'Ipv6cp extension flag';
        }
        leaf ipv6cp-aftr {
            type uint8 {
                range 1..255 ;
            }
            description 'AFTR option type value';
        }
        leaf ipv6cp-ipv6-address {
            type uint8 {
                range 1..255 ;
            }
        }
    }
}
```

```
        description 'Ipv6 address option type value';
    }
    leaf ipv6cp-ipv6-prefix {
        type uint8 {
            range 1..255 ;
        }
        description 'Ipv6 prefix option type value';
    }
    container ipv6-dns {
        presence ipv6-dns ;
        leaf ipv6cp-ipv6-dns-secondary {
            type uint8 {
                range 1..255 ;
            }
            mandatory true ;
            description 'IPV6 primary DNS option type value';
        }
        leaf ipv6cp-ipv6-dns-primary {
            type uint8 {
                range 1..255 ;
            }
            mandatory true ;
            description 'IPV6 secondary DNS option type value';
        }
        description 'Ipv6 DNS option type value';
    }
    container ipv4-dns {
        presence ipv4-dns ;
        leaf ipv6cp-ipv4-dns-primary {

            type uint8 {
                range 1..255 ;
            }
            mandatory true ;
            description 'IPV4 primary DNS option type value';
        }
        leaf ipv6cp-ipv4-dns-secondary {

            type uint8 {
                range 1..255 ;
            }
            mandatory true ;
            description 'IPV4 secondary DNS option type value';
        }
        description 'Ipv4 DNS option type value';
    }
    description 'Configuration about IPV6CP extension.';
}
```

```

container pppox-ipcp-cfg {
  leaf ipcp-flag {
    type enumeration {
      enum enable {
        value 1 ;
        description "enable ipcp extension flag";
      }
      enum disable {
        value 0 ;
        description "disable ipcp extension flag";
      }
    }
    default disable ;
    description 'Ipcp extension flag';
  }

  leaf option-type {
    type uint8 {
      range 1..255 ;
    }
    description 'Ipcp 6rd option type';
  }

  leaf br-address {
    type inet:ipv4-address {
      pattern '^(1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]{0,1}|
        + '0,1})\.){3}(0|1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|
        + '3[0-9]{0,1})$|^0\.(1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|
        + ')|25[0-5]|3[0-9]{0,1})\.){2}(1[0-9]{0,2}|2[0-9]{0,1}|2
        + '[0-4][0-9]|25[0-5]|3[0-9]{0,1})$|^((1[0-9]{0,2}|2[0-9]
        + '[0,1]|2[0-4][0-9]|25[0-5]|3[0-9]{0,1})\.)0\.(1[0-9]{0
        + ',2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-5]|3[0-9]{0,1})\.)
        + '(1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-5]|3[0-9]{0,1})
        + '$|^((1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-5]|3[0-9]
        + '){0,1})\.){2}0\.(1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-
        + '5]|3[0-9]{0,1})$|^((1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]
        + ')|25[0-5]|3[0-9]{0,1})\.){3}0$|255\.(0|1[0-9]{0,2}|2[
        + '0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]{0,1})\.){2}(0|1[0
        + '-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]{0,1})$
        + '|^((0|1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]
        + '[0,1])\.)255\.(0|1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25
        + '[0-4]|3[0-9]{0,1})\.)
        + '(0|1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]
        + '[0-9]|25[0-4]|3[0-9]{0,1})$|^((0|1[0-9]{0,2}|2[0-9]{0,
        + '1}|2[0-4][0-9]|25[0-4]|3[0-9]{0,1})\.){2}255\.(0|1[0-9
        + '){0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]{0,1})$|^
        + '((0|1[0-9]{0,2}|2[0-9]{0,1}|2[0-4][0-9]|25[0-4]|3[0-9]
        + '[0,1])\.){3}255$'
      ;
    }
  }
}

```

```
    }
    description '6RD BR IPv4 address';
  }
  container ipv6-rapid-deployment {
    presence ipv6-rapid-deployment ;
    leaf v4-mask-len {
      type uint8 {
        range 0..32 ;
      }
      mandatory true ;
      description 'IPv4 address mask length';
    }
    leaf v6-pref {
      type inet:ipv6-address ;
      mandatory true ;
      description 'IPv6 prefix';
    }
    leaf v6-mask-len {
      type uint8 {
        range 1..128 ;
      }
      mandatory true ;
      description 'IPv6 prefix length';
    }
    description 'Ipv6 rapid deployment';
  }
  description 'Configuration about IPCP extension.';
}
container pppoe-switch {
  leaf delay-time {
    type uint16 {
      range 1..300 ;
    }
    description 'Trigger user offline when VCC phys-interface down';
  }
  leaf keepalive-timer {
    type enumeration {
      enum start {
        value 1 ;
        description "start keepalive timer";
      }
      enum stop {
        value 0 ;
        description "stop keepalive timer";
      }
    }
    default start ;
    description 'Start or stop send keepalive packet';
  }
}
```

```
}
leaf ppp-max-payload {
  type enumeration {
    enum disable {
      value 0 ;
      description "disable ppp max payload";
    }
    enum enable {
      value 1 ;
      description "enable ppp max payload";
    }
  }
  default disable ;
  description 'Enable or disable pppoe ppp-max-payload';
}
leaf service {
  type enumeration {
    enum advertise {
      value 1 ;
      description "enable ppp service!";
    }
    enum disable {
      value 0 ;
      description "disable ppp service!";
    }
  }
  default advertise ;
  description 'Open or close pppoe service';
}
leaf ppp-mru-verify {
  type enumeration {
    enum open {
      value 1 ;
      description "enable ppp mru verify!";
    }
    enum close {
      value 0 ;
      description "disable ppp mru!";
    }
  }
  default close ;
  description 'set ppp lcp mru verify when mru over 1492';
}

leaf keepalive-fast-reply {
  type enumeration {
    enum enable {
      value 1 ;
```



```
        description 'Enable keepalive fast reply!';
    }
    enum disable {
        value 0 ;
        description 'Disable keepalive fast reply!';
    }
}
description 'Set keepalive fast reply flag.';
}
description 'Configuration about pppoe switch.';
}
list pppoe-cfg {
    key template ;
    leaf template {
        type uint32 {
            range 1..1000 ;
        }
        description 'PPPoX template number';
    }
}
leaf ppp-authentication {
    type enumeration {
        enum pap {
            value 1 ;
            description "configure pap authentication!";
        }
        enum chap {
            value 2 ;
            description "configure chap authentication!";
        }
        enum mschapv1 {
            value 6 ;
            description "configure mschapv1 authentication!";
        }
        enum mschapv2 {
            value 7 ;
            description "configure mschapv2 authentication!";
        }
        enum pap-chap {
            value 21 ;
            description "configure pap-chap authentication!";
        }
    }
}
default pap-chap ;
description 'Set ppp authentication';
}
leaf ppp-check-magic-num {
    type enumeration {
        enum disable {
```

```
        value 0 ;
        description 'disable ppp magic check';
    }
    enum enable {
        value 1 ;
        description 'enable ppp magic check';
    }
}
default enable ;
description 'Check magic number or not';
}
leaf ppp-mru {
    type uint32 {
        range 320..9000 ;
    }
    default 1492 ;
    description 'Set mru value';
}
leaf pppoe-ac-name {
    type string ;
    description 'Set ac-name';
}
leaf pppoe-service-name-omit {
    type enumeration {
        enum disable {
            value 0 ;
            description "disable pppoe service name omit";
        }
        enum enable {
            value 1 ;
            description "enable pppoe service name omit";
        }
    }
    default disable ;
    description 'Check service-name value';
}
leaf pppoe-ac-cookie-check {
    type enumeration {
        enum disable {
            value 0 ;
            description "disable pppoe ac cookie check";
        }
        enum enable {
            value 1 ;
            description "enable pppoe ac cookie check";
        }
    }
}
```

```
    default enable ;
        description 'Check options';
    }
    leaf pppoe-password-string {
        type string ;
        description 'Set authen fail password error string';
    }
    leaf pppoe-username-string {
        type string ;
        description 'Set authen fail username error string';
    }
}

choice ppp-quick-redial {
    case quick-redial-disable {
        leaf ppp-quick-redial-disable {
            type enumeration {
                enum disable {
                    value 0 ;
                    description "disable ppp quick redial";
                }
            }
        }
        default disable ;
        description 'disable quick-redial';
    }
    description 'disable quick-redial';
}

case fast-response {
    leaf ppp-fast-response {
        type enumeration {
            enum diable {
                value 0 ;
                description "disable ppp fast response";
            }
            enum enable {
                value 1 ;
                description "enable ppp fast response";
            }
        }
        description 'set Response the access request immediately';
    }
    leaf ppp-quick-redial-enable {
        type enumeration {
            enum enable {
                value 1 ;
                description "enable ppp quick redial";
            }
        }
        default enable ;
    }
}
```

```

        description 'Enable quick-redial';
    }
    description 'set quick-redial or Response the access request immediately';
}
    default quick-redial-disable ;
    description 'Enable or disable quick-redial';
}
container ppp-keepalive {
    leaf ppp-keepalive-timer {
        type uint32 {
            range 10..14400 ;
        }
        default 60 ;
        description 'Set keepalive time(unit:seconds)';
    }
    leaf ppp-keepalive-count {
        type uint16 {
            range 1..10 ;
        }
        default 3 ;
        description 'Set keepalive counter';
    }
    description 'Set keepalive time and counter';
}
container ppp-timeout {
    leaf ppp-timeout-negtimeoutsec {
        type uint8 {
            range 1..10 ;
        }
        default 3 ;
        description 'Set ppp negtimeoutsec timeout(unit:seconds)';
    }
    leaf ppp-timeout-authentication {
        type uint8 {
            range 1..10 ;
        }
        default 3 ;
        description 'Set ppp authentication timeout(unit:seconds)';
    }
    description 'Set ppp negtimeoutsec and authentication timeout';
}
    description 'Configuration pppoe template';
}
    description 'Configuration vBRAS PPPoX.';
}
    description "ietf-bng configuration!";
}
description "augment lne model";

```

```
}  
}  
<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 [RFC5246].

The NETCONF access control model [RFC6536] 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.

vBNG represents device and network configuration information based on the LNE. As such, the security of this information is important, but it is fundamentally no different than any other interface or device configuration information that has already been covered in other documents such as [I-D.ietf-rtgwg-lne-model].

The vulnerable "config true" parameters and subtree are the following:

/lne:logical-network-elements/lne:logical-network-element/ietf-vbng/bng-cp: this subtree specifies the global configuration of bng control plane. Modify the configuration can cause the bng control plane instance disabled.

/lne:logical-network-elements/lne:logical-network-element/ietf-vbng/bng-up: this subtree specifies the global configuration of BNG user plane. Modify the configuration can cause the BNG user plane instance disabled.

/lne:logical-network-elements/lne:logical-network-element/ietf-vbng/control-channel: this subtree specifies the configuration control channel parameters among bng user planes and control plane. Modify the configuration can cause the control channel and control channel protocol interrupted.

/lne:logical-network-elements/lne:logical-network-element/ietf-vbng/vxlan-channel: this subtree specifies the configuration VXLAN channel parameters among BNG user planes and control plane. Modify the configuration can cause the VXLAN channel interrupted.

Unauthorized access to any of these lists can adversely affect the security of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

## 6. Acknowledgements

## 7. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-vbng.

Registrant Contact: The IESG.

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name:	ietf-vbng
namespace:	urn:ietf:params:xml:ns:yang:ietf-vbng
prefix:	vbng
reference:	RFC XXXX

## 8. References

### 8.1. Normative References

[I-D.asechoud-rtgwg-qos-model]

Choudhary, A., Jethanandani, M., Strahle, N., Aries, E., and I. Chen, "YANG Model for QoS", draft-asechoud-rtgwg-qos-model-05 (work in progress), March 2018.

[I-D.ietf-netmod-acl-model]

Jethanandani, M., Huang, L., Agarwal, S., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-18 (work in progress), March 2018.

[I-D.ietf-rtgwg-lne-model]

Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", draft-ietf-rtgwg-lne-model-09 (work in progress), March 2018.

- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [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>>.

## 8.2. Informative References

- [I-D.huang-nvo3-vxlan-extension-for-vbras]  
Huang, L. and S. Hu, "VxLAN Extension Requirement for Signaling Exchange Between Control and User Plane of vBras", draft-huang-nvo3-vxlan-extension-for-vbras-00 (work in progress), March 2017.

Authors' Addresses

Fangwei Hu  
ZTE Corporation  
No.889 Bibo Rd  
Shanghai 201203  
China

Phone: +86 21 68896273  
Email: hu.fangwei@zte.com.cn

RongRong Hua  
ZTE Corporation  
No.50 Software Avenue,Yuhuatai District  
Nanjing, Jiangsu Province 210012  
China

Email: hua.rongrong@zte.com.cn

Shujun Hu  
China Mobile  
32 Xuanwumen West Ave, Xicheng District  
Beijing 100053  
China

Email: shujun\_hu@outlook.com

Rong Gu  
China Mobile  
32 Xuanwumen West Ave, Xicheng District  
Beijing, Beijing 100053  
China

Email: gurong\_cmcc@outlook.com