Abstract

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

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 11, 2019.

Copyright Notice

Copyright (c) 2019 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
1. Introduction

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

The architecture of Control-plane and User-plane separated BNG is shown as the following figure.
There are three interfaces between vBNG-CP (vBNG Control Plane) and vBNG-UP (vBNG User Plane): 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 requirements and possible solution are defined in the [I-D.hu-nvo3-vxlan-gpe-extension-for-vbng]. The control interface is used for setting forwarding entries of the user plane using Simple CUSP (S-CUSP) [I-D.cuspdt-rtgw-cu-separation-bng-protocol] or other protocols. The management interface is used by vBNG-CP to carry out related configurations of vBNG-UP through NETCONF protocol [RFC6241].

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

CUSP: Control-plane and User-plane Separation Protocol.

S-CUSP: Simple CUSP.

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 vBNG-UP and vBNG-CP.

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

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

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. Yang Data model through Management Interfaces

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

The YANG data model for vBNG through the management interface includes vBNG-UP interface configuration, control channel and service channel configuration, ACL and QoS.

The vBNG-UP interface configuration is to configure the basic interface informations of a vBNG-UP element, such as interface name, the VLAN parameters for the sub-interface.
The control channel is to configure the S-CUSP parameters. The control channel parameters include: name, id, port, S-CUSP version, hello interval, dead time, and keepalive time.

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

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]

The QoS information includes IP-DSCP, MPLS, VPLS, VPWS etc. The YANG data model for QoS refers to [I-D.asechoud-rtgwg-qos-model]
module: ietf-vbng
    augment /lne:logical-network-elements/lne:logical-network-element:
    +--rw ietf-vbng
        +--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
                +--rw control-channel
                    +--rw name? string
                    +--rw id? uint32
                    +--rw port? uint32
                    +--rw version uint8
                    +--rw hellointerval uint32
                    +--rw deadtime uint32
                    +--rw keepalivetime uint32
                    +--rw service-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
            +--rw acl
            .......
            +--rw qos
            .......

3.2. YANG Data Model for vBNG-CP

The ietf-vbng-cp module is to configure vBNG-CP. The YANG data model includes: vbng-cp-name, netconf-server and PPPoE parameters, etc.
3.3. YANG Data Model for vBNG-UP

The ietf-vbng-up module is to configure the vBNG-UP. The YANG data model includes: shelf-number, vbng-up-name, netconf-client and keepalive-sink, etc.
4. vBNG YANG Data Model

4.1. ietf-vbng module

<CODE BEGINS> file "ietf-vbng@2019-03-08.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: Guangping Huang
            <mailto:huang.Guangping@zte.com.cn>
        ";

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

"
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 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";
}
container control-channel {
    leaf name {
        type string;
        description "control channel protocol logical name";
    }
    leaf id {
        type uint32;
        description "the s-cusp session id";
    }
    leaf port {
        type uint32;
        description "s-cusp tcp connection port number";
    }
    leaf version {
        type uint8;
        description "s-cusp version number";
    }
    leaf hellointerval {
        type uint32;
        description "s-cusp hello interval";
    }
    leaf deadtime {
        type uint32;
        description "s-cusp dead time";
    }
    leaf keepalivetime {
        type uint32;
        description "s-cusp keepalive time";
    }
}

list service-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";
}

description "ietf-bng configuration!";
}

description "augment lne model";
}

<CODE ENDS>
4.2. ietf-vbng-cp module

```yang
<CODE BEGINS> file "ietf-vbng-cp@2019-03-08.yang"
module ietf-vbng-cp{
    namespace "urn:ietf:params:xml:ns:yang:ietf-vbng-cp";
    prefix "vbng-cp";

    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: Guangping Huang
        <mailto:huang.guangping@zte.com.cn>
        
    ";

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

    revision 2019-03-08{
        description "Initial a new vbng control and user plane separation
        yang data model, it includes ietf-vbng, ietf-vbng-cp,and ietf-vbng-up, this
        is ietf-vbng-cp";
        reference
            "draft-cuspdt-rtgwg-cu-separation-yang-model-02";
    }

/* Typedefs */

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-cp{
        leaf bng-cp-name {
            type string;
            description "configure vbng-cp name";
        }
        leaf enable {
            type boolean;
            description "'true' to support vbng separation";
        }
        container netconf-server {
            presence netconf-server ;
            list address-family {
                key "af";
                leaf af {
                    type address-family-type;
                    description
                    "Address family type value.";
                }
                leaf ip {
                    type inet:ip-address;
                    mandatory true ;
                    description 'Configure ip address of netconf server.';
                }
                description "address family list";
            }
            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.';
}

container vbng-pppoe {
  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;
        }
    }
}
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 ;
}
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 authentication failure password string';
}

leaf pppoe-username-string {
  type string;
  description 'Set authentication failure 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 disable {
        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 PPPoE.';
} description "configure bng-cp";
} description "augment lne model";
}
</CODE ENDS>

4.3. ietf-vbng-up module

<CODE BEGINS> file "ietf-vbng-up@2019-03-08.yang"
  module ietf-vbng-up{
    namespace "urn:ietf:params:xml:ns:yang:ietf-vbng-up";
    prefix "vbng-up";

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

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

    ...
</CODE ENDS>
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-up{
        list vbng-up {
            key shelf-no ;
        }
    }
}
leaf shelf-no {
  type uint8 {
    range 1..127;
  }
  description 'Configure shelf-no of forwarder, 1-127.';
}

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

container netconf-client {
  presence netconf-client;
  list address-family {
    key "af";
    leaf af {
      type address-family-type;
      description "Address family type value.";
    }
    leaf ip {
      type inet:ip-address;
      mandatory true;
      description 'Configure ip address of netconf server.';
    }
    description "address family list";
  }
  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";
}

<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(vBNG-CP, vBNG-UP) 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:
interfaces: this subtree specifies vBNG-UP interface parameters configuration. Modify the configuration can cause the vBNG-UP interfaces disable.

control-channel: this subtree specifies control channel parameters configuration. Modify the configuration can cause the S-CUSP protocol sessions interrupted among the vBNG-CPs and vBNG-UPs.

service-channel: this subtree specifies the service channel parameters configuration among vBNG user plane and control plane. Modify the configuration can cause the VxLAN session interrupted among vBGN-UPs and vBNG-CPs.

cp/netconf-server: this subtree specifies netconf parameters of vBNG-CP. Modify the configuration can cause the netconf session among vBNG-CPs and vBNG-UPs interrupted.

cp/vbng-pppoe: this subtree specifies PPPoE parameters of vBNG-CP. Modify the configuration can cause the PPPoE session interrupted.

cp/netconf-client: this subtree specifies netconf parameters of vBNG-UP. Modify the configuration can cause the netconf session among vBNG-CP and vBNG-UP 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 three URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested to be made.


Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.
This document registers three YANG modules in the YANG Module Names registry [RFC6020].

name: ietf-vbng
prefix: vbng
reference: RFC XXXX

name: ietf-vbng
prefix: vbng-cp
reference: RFC XXXX

name: ietf-vbng
prefix: vbng-up
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-08 (work in progress), January 2019.

[I-D.ietf-netmod-acl-model]
Jethanandani, M., Agarwal, S., Huang, L., and D. Blair,
"Network Access Control List (ACL) YANG Data Model",
draft-ietf-netmod-acl-model-21 (work in progress),
November 2018.


8.2. Informative References

[I-D.cuspdt-rtgwg-cu-separation-bng-protocol]

[I-D.hu-nvo3-vxlan-gpe-extension-for-vbng]

Authors’ Addresses

Guangping Huang
ZTE Corporation
No.50, Software Avenue
Nanjing, Jiangsu 210012
China

Email: huang.guangping@zte.com.cn

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

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

Email: shujun_hu@outlook.com
Fengwei Qin  
China Mobile  
32 Xuanwumen West Ave, Xicheng District  
Beijing, Beijing 100053  
China  

Email: qinfengwei@chinamobile.com