Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 23, 2016

Z. Li
S. Zhuang
Huawei Technologies
X. Liu
Ericsson
J. Haas
S. Esale
Juniper Networks
B. Wen
Comcast

December 21, 2015

Yang Data Model for BGP/MPLS IP VPN
draft-li-bess-l3vpn-yang-01

Abstract

This document defines a YANG data model that can be used to configure and manage L3VPN (BGP/MPLS IP VPN).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 June 23, 2016.

Copyright Notice

Copyright (c) 2015 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 (http://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. Definitions and Acronyms 3
3. Design of the L3VPN Model 3
  3.1. Overview 3
  3.2. VPN Instance Configuration 4
    3.2.1. Per-Instance Configuration 4
    3.2.2. Address Family Configuration of L3VPN Instance . . . 4
  3.3. Yang Tree of L3VPN Yang Model 5
4. L3VPN YANG Model 7
5. IANA Considerations 15
6. Security Considerations 15
7. Normative References 15
Authors' Addresses 16

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encodings other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage L3VPN (BGP/MPLS IP VPN) [RFC4364].

2. Definitions and Acronyms

AF: Address Family

BGP: Border Gateway Protocol

JSON: JavaScript Object Notation

L3VPN: Layer 3 VPN

NETCONF: Network Configuration Protocol

ReST: Representational State Transfer, a style of stateless interface and protocol that is generally carried over HTTP

YANG: A data definition language for NETCONF

3. Design of the L3VPN Model

3.1. Overview

The L3VPN Yang module is to augment the routing instance Yang models proposed by the draft [I-D.ietf-netmod-routing-cfg]. It introduced the "l3vpn"container to define augmented parameters which can be applied for VRF Routing Instance and support both the IPv4 and IPv6 address families. The overview of the "l3vpn" container is shown in the following figure:

```
module: ietf-l3vpn
augment /rt:routing/rt:routing-instance:
   +--rw l3vpn
      +--rw ipv4-family
      |  +--rw bgp-parameters
      |  |  +--rw common
      |  |     +--rw route-distinguisher?   string
      |  |     +--rw vpn-targets* [rt-value]
      |  |        +--rw rt-value    string
      |  |        +--rw rt-type     bgp-rt-type
      |  ......
      +--rw ipv6-family
         +--rw bgp-parameters
         |  +--rw common
         |     +--rw route-distinguisher?   string
         |     +--rw vpn-targets* [rt-value]
         |        +--rw rt-value    string
         |        +--rw rt-type     bgp-rt-type
         ......
```

L3VPN inteface parameters can reuse those parameters defined by [I-D.ietf-netmod-routing-cfg].

BGP Protocols parameters for L3VPN is defined by the draft [I-D.ietf-idr-bgp-model]. The augment may be defined in the future version if necessary.

3.2. VPN Instance Configuration

An instance is created to comprise the VPN forwarding information for each VPN in a BGP/MPLS IP VPN. This instance is called a VPN instance or a VPN routing and forwarding (VRF) table. It is also called a per-site forwarding table in [RFC4364]. VPN instances must be created in all BGP/MPLS IP VPN solutions. VPN instances support both the IPv4 and IPv6 address families.

VPN instance configuration consists of the following components :

o Per-Instance Configuration : that contains the common writable configuration objects for VPN instance IPv4 and IPv6 address family.

o Address Family Configuration of L3VPN Instance: that contains the address family specific writable configuration objects.

3.2.1. Per-Instance Configuration

Per-instance parameters is defined by [I-D.ietf-netmod-routing-cfg] including instance name, description, etc.

3.2.2. Address Family Configuration of L3VPN Instance

l3vpn container contains the address family specific writable configuration objects, such as route-distinguisher, vpn-targets, apply-label-mode, etc. The parameters should be consistent between IPv4 family and IPv6 family.

```
      +--rw l3vpn
         +--rw ipv4-family
         |  +--rw bgp-parameters
         |  |  +--rw common
         |  |     +--rw route-distinguisher?   string
         |  |     +--rw vpn-targets* [rt-value]
         |  |        +--rw rt-value    string
         |  |        +--rw rt-type     bgp-rt-type
         |  +--rw apply-label-mode?      apply-label-mode-def
         |  +--rw import-route-policy?   string
         |  +--rw export-route-policy?   string
         |  +--rw tunnel-policy?         string
         |  +--rw prefix-limit
         |  |  +--rw prefix-limit-number?   uint32
         |  |  +--rw (prefix-limit-action)?
         |  |     +--:(enable-alert-percent)
         |  |     |  +--rw alert-percent-value?   uint8
         |  |     |  +--rw route-unchanged?       boolean
         |  |     +--:(enable-simple-alert)
         |  |        +--rw simple-alert?          boolean
         |  +--rw routing-table-limit
         |  |  +--rw routing-table-limit-number?   uint32
         |  |  +--rw (routing-table-limit-action)?
         |  |     +--:(enable-alert-percent)
         |  |     |  +--rw alert-percent-value?          uint8
         |  |     +--:(enable-simple-alert)
         |  |        +--rw simple-alert?                 boolean
         |  +--rw import-global-rib
         |     +--rw protocol?            enumeration
         |     +--rw processId?           uint32
         |     +--rw bgp-valid-route?     boolean
         |     +--rw route-policy-name?   string
         +--rw ipv6-family
            ......
```

3.3. Yang Tree of L3VPN Yang Model

The Yang tree of L3VPn Yang model is shown in the following figure:

```
module: ietf-l3vpn
augment /rt:routing/rt:routing-instance:
   +--rw l3vpn
      +--rw ipv4-family
      |  +--rw bgp-parameters
      |  |  +--rw common
      |  |     +--rw route-distinguisher?   string
      |  |     +--rw vpn-targets* [rt-value]
      |  |        +--rw rt-value    string
```

```
         |  |        +--rw rt-type     bgp-rt-type
         |  +--rw apply-label-mode?      apply-label-mode-def
         |  +--rw import-route-policy?   string
         |  +--rw export-route-policy?   string
         |  +--rw tunnel-policy?         string
         |  +--rw prefix-limit
         |  |  +--rw prefix-limit-number?   uint32
         |  |  +--rw (prefix-limit-action)?
         |  |     +--:(enable-alert-percent)
         |  |     |  +--rw alert-percent-value?   uint8
         |  |     |  +--rw route-unchanged?       boolean
         |  |     +--:(enable-simple-alert)
         |  |        +--rw simple-alert?          boolean
         |  +--rw routing-table-limit
         |  |  +--rw routing-table-limit-number?   uint32
         |  |  +--rw (routing-table-limit-action)?
         |  |     +--:(enable-alert-percent)
         |  |     |  +--rw alert-percent-value?          uint8
         |  |     +--:(enable-simple-alert)
         |  |        +--rw simple-alert?                boolean
         |  +--rw import-global-rib
         |     +--rw protocol?            enumeration
         |     +--rw processId?           uint32
         |     +--rw bgp-valid-route?     boolean
         |     +--rw route-policy-name?   string
         +--rw ipv6-family
            +--rw bgp-parameters
            |  +--rw common
            |     +--rw route-distinguisher?   string
            |     +--rw vpn-targets* [rt-value]
            |        +--rw rt-value    string
            |        +--rw rt-type     bgp-rt-type
            +--rw apply-label-mode?      apply-label-mode-def
            +--rw import-route-policy?   string
            +--rw export-route-policy?   string
            +--rw tunnel-policy?         string
            +--rw prefix-limit
            |  +--rw prefix-limit-number?   uint32
            |  +--rw (prefix-limit-action)?
            |     +--:(enable-alert-percent)
            |     |  +--rw alert-percent-value?   uint8
            |     |  +--rw route-unchanged?       boolean
            |     +--:(enable-simple-alert)
            |        +--rw simple-alert?          boolean
            +--rw routing-table-limit
            |  +--rw routing-table-limit-number?   uint32
            |  +--rw (routing-table-limit-action)?
            |     +--:(enable-alert-percent)
```

```
            |     |  +--rw alert-percent-value?          uint8
            |     +--:(enable-simple-alert)
            |        +--rw simple-alert?                 boolean
            +--rw import-global-rib
               +--rw protocol?            enumeration
               +--rw processId?           uint32
               +--rw bgp-valid-route?     boolean
               +--rw route-policy-name?   string
```

4. L3VPN YANG Model

```
//L3VPN YANG MODEL
<CODE BEGINS> file "ietf-l3vpn.yang"
module ietf-l3vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3vpn";
  // replace with IANA namespace when assigned
  prefix "l3vpn";

  import ietf-routing {
    prefix "rt";
    //draft-ietf-netmod-routing-cfg-19
  }

  organization  "IETF BGP Enabled Services WG";
  contact       "draft-li-bess-l3vpn-yang@tools.ietf.org";
  description
    "This YANG module defines the generic configuration data
     for L3VPN service.

     Terms and Acronyms

     BGP (bgp): Border Gateway Protocol
     IPv4 (ipv4):Internet Protocol Version 4
     IPv6 (ipv6): Internet Protocol Version 6

    ";

  revision 2015-10-09 {
    description
      "Initial revision.";
      reference "RFC4271, RFC4364, RFC4365, RFC4760.";
  }

  /* typedefs */

  typedef bgp-rt-type {
    type enumeration {
      enum import {
```

```
          description "For import";
        }
        enum export {
          description "For export";
        }
        enum both {
          description "For both import and export";
        }
      }
      description "BGP route-target type. Import from BGP YANG.";
    }

    typedef apply-label-mode-def {
      type enumeration {
        enum "per-route" {
          value 0;
          description
            "By default, the VPN instance IPv4 address family
             assigns a unique label to each route to be sent
             to the peer PE.";
        }
        enum "per-instance" {
          value 1;
          description
            "The apply-label per-instance command enables the
             one-label-per-VPN-instance mode.";
        }
      }
      description "...";
    }

    typedef routing-instance-type-ref {
      type leafref {
        path "/rt:routing/rt:routing-instance/rt:type";
      }
      description
        "This type is used for leafs that reference a routing
         instance configuration.";
    }

    grouping bgp-parameters-grp {
      description
        "BGP parameters grouping.";
      container bgp-parameters {
        description
          "Parameters for BGP.";
        container common {
          description
```

```
            "Common BGP parameters.";
          leaf route-distinguisher {
            type string;
            description "BGP RD.";
          }
          list vpn-targets {
            key rt-value;
            description
              "Route Targets.";
            leaf rt-value {
              type string;
              description
                "Route-Target value.";
            }
            leaf rt-type {
              type bgp-rt-type;
              mandatory true;
              description
                "Type of RT.";
            }
          }
        }
      }
    }


    grouping vpn-af-config {
      description
        "A set of configuration parameters that is applicable to both
         IPv4 and IPv6 address family for a VPN instance .";

      leaf apply-label-mode {
        type apply-label-mode-def;
        default "per-route";
        description ".";
      }

      leaf import-route-policy {
        type string {
          length "1..40";
        }
        description
          "The import route-policy command associates a VPN instance
           enabled with the IPv4 or IPv6 address family with an
           import routing policy.
           Only one import routing policy can be associated with a
           VPN instance enabled with the IPv4 or IPv6 address family.
           If the import route-policy command is run more than once,
```

```
             the latest configuration overrides the previous ones.";
       }

       leaf export-route-policy {
         type string {
           length "1..40";
         }
         description
           "The export route-policy command associates a VPN instance
            enabled with the IPv4 or IPv6 address family with an
            export routing policy.
            Only one export routing policy can be associated with a
            VPN instance enabled with the IPv4 or IPv6 address family.
            If the export route-policy command is run more than once,
            the latest configuration overrides the previous ones.";
       }

       leaf tunnel-policy {
          type string;
          description
            "Tunnel policy name.";
       }

       container prefix-limit {
         description
           "The prefix limit command sets a limit on the maximum
            number of prefixes supported in the existing VPN
            instance, preventing the PE from importing excessive
            VPN route prefixes.";

         leaf prefix-limit-number {
           type uint32 {
             range "1..4294967295";
           }
           description
             "Specifies the maximum number of prefixes supported in the
              VPN instance IPv4 or IPv6 address family.";
         }

         choice prefix-limit-action {
           description ".";
           case enable-alert-percent {
             leaf alert-percent-value {
               type uint8 {
                 range "1..100";
               }
               description
                 "Specifies the proportion of the alarm threshold to the
```

```
                 maximum number of prefixes.";
             }
             leaf route-unchanged {
               type boolean;
               default "false";
               description
                 "Indicates that the routing table remains unchanged.
                  By default, route-unchanged is not configured. When
                  the number of prefixes in the routing table is
                  greater than the value of the parameter number,
                  routes are processed as follows:
                  (1)If route-unchanged is configured, routes in the
                     routing table remain unchanged.
                  (2)If route-unchanged is not configured, all routes
                     in the routing table are deleted and then
                     re-added.";
             }
           }
           case enable-simple-alert {
             leaf simple-alert {
               type boolean;
               default "false";
               description
                 "Indicates that when the number of VPN route prefixes
                  exceeds number, prefixes can still join the VPN
                  routing table and alarms are displayed.";
             }
           }
         }
       }


       container routing-table-limit {
         description
           "The routing-table limit command sets a limit on the maximum
            number of routes that the IPv4 or IPv6 address family of a
            VPN instance can support.
            By default, there is no limit on the maximum number of
            routes that the IPv4 or IPv6 address family of a VPN
            instance can support, but the total number of private
            network and public network routes on a device cannot
            exceed the allowed maximum number of unicast routes.";

         leaf routing-table-limit-number {
           type uint32 {
             range "1..4294967295";
           }
           description
```

```
             "Specifies the maximum number of routes supported by a
              VPN instance. ";
         }

         choice routing-table-limit-action {
           description ".";
           case enable-alert-percent {
             leaf alert-percent-value {
               type uint8 {
                 range "1..100";
               }
               description
                 "Specifies the percentage of the maximum number of
                  routes. When the maximum number of routes that join
                  the VPN instance is up to the value
                  (number*alert-percent)/100, the system prompts
                  alarms. The VPN routes can be still added to the
                  routing table, but after the number of routes
                  reaches number, the subsequent routes are
                  dropped.";
             }
           }
           case enable-simple-alert {
             leaf simple-alert {
               type boolean;
               description
                 "Indicates that when VPN routes exceed number, routes
                  can still be added into the routing table, but the
                  system prompts alarms.
                  However, after the total number of VPN routes and
                  network public routes reaches the unicast route limit
                  specified in the License, the subsequent VPN routes
                  are dropped.";
             }
           }
         }
       }

       container import-global-rib {
         description
           "Route Leaking from a Global Routing Table into a VRF.";

         leaf protocol {
           type enumeration {
             enum ALL {
               value "0";
               description "ALL:";
             }
```

```
            enum Direct {
              value "1";
              description "Direct:";
            }
            enum OSPF {
              value "2";
              description "OSPF:";
            }
            enum ISIS {
              value "3";
              description "ISIS:";
            }
            enum Static {
              value "4";
              description "Static:";
            }
            enum RIP {
              value "5";
              description "RIP:";
            }
            enum BGP {
              value "6";
              description "BGP:";
            }
            enum OSPFV3 {
              value "7";
              description "OSPFV3:";
            }
            enum RIPNG {
              value "8";
              description "RIPNG:";
            }
            enum INVALID {
              value "9";
              description "INVALID:";
            }
          }
          description
            "Specifies the protocol from which routes are imported.
             At present, In the IPv4 unicast address family view,
             the protocol can be IS-IS,static, direct and BGP.";
        }

        leaf processId {
          type uint32 {
            range "0..4294967295";
          }
          default "0";
```

```
          description
            "Specifies the process ID if the protocol from routes
             are imported is IS-IS.";
        }

        leaf bgp-valid-route {
          type boolean;
          description ".";
        }

        leaf route-policy-name {
          type string;
          description
            "Policy Id for import routes";
        }

      }
    }

    augment "/rt:routing/rt:routing-instance" {
      description ".";
      container l3vpn {
        when "routing-instance-type-ref = 'vrf-routing-instance'" {
          description ".";
        }
        description ".";
        container ipv4-family {
          description
            "The IPv4 address family is enabled for the VPN
             instance.";

          uses bgp-parameters-grp;
          uses vpn-af-config;
        }

        container ipv6-family {
          description
            "The IPv6 address family is enabled for the VPN
             instance.";

          uses bgp-parameters-grp;
          uses vpn-af-config;
        }
      } //End of case type

    } //End of augment "/rt:routing/rt:routing-instance"

  }
```

</CODE ENDS>

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

This document does not introduce any new security risk.

7. Normative References

[I-D.ietf-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Alex, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-00 (work in progress), July 2015.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-20 (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <http://www.rfc-editor.org/info/rfc4760>.

[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>.

[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, <http://www.rfc-editor.org/info/rfc6241>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: zhuangshunwan@huawei.com

Xufeng Liu
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

Email: xufeng.liu@ericsson.com

Jeffrey Haas
Juniper Networks

Email: jhaas@juniper.net

Santosh Esale
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: sesale@juniper.net

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com