BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 20, 2017

D. Jain
K. Patel
P. Brissette
Cisco
Z. Li
S. Zhuang
Huawei Technologies
X. Liu
Ericsson
J. Haas
S. Esale
Juniper Networks
B. Wen
Comcast

August 19, 2016

Yang Data Model for BGP/MPLS L3 VPNs
draft-dhjain-bess-bgp-l3vpn-yang-02.txt

Abstract

This document defines a YANG data model that can be used to configure and manage BGP Layer 3 VPNs.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 20, 2017.

Table of Contents

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 model that can be used to configure and manage BGP L3VPNs [RFC4364]. It contains VRF sepcific parameters as well as BGP specific parameters applicable for L3VPNs. The individual containers defined in this model contain control knobs for configuration for that purpose, as well as a few data nodes that can be used to monitor health and gather statistics.

1.1. 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].

2. Definitions and Acronyms

AF: Address Family

AS: Autonomous System

ASBR: Autonomous System Border Router

BGP: Border Gateway Protocol

CE: Customer Edge

PE: Provider Edge

L3VPN: Layer 3 VPN

NETCONF: Network Configuration Protocol

RD: Route Distinguisher

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

RTFilter: Route Filter

VPN: Virtual Private Network

VRF: Virtual Routing and Forwarding

YANG: Data definition language for NETCONF

3. Design of BGP L3VPN Data Model

3.1. Overview

There are two parts of the BGP L3VPN yang data model. The first part of the model defines VRF specific parameters for L3VPN by augmenting the routing-instance container defined in the routing model [I-D.ietf-netmod-routing-cfg] and the second part of the model defines BGP specific parameters for the L3VPN by augmenting the base BGP data model defined in [I-D.shaikh-idr-bgp-model].

3.2. VRF Specific Configuration

Routing-instance defined in the IETF routing model defines a default instance when routing-instance type is default-routing-instance and named vrf instance when type is vrf-routing-instance. For L3VPN, the VRF specific parameters are defined by augmenting the routing-instance container corresponding to named vrf instance. A new container l3vpn is added for VPN parameters.

3.2.1. VRF interface

To associate a VRF instance with an interface, the interface should be defined in the context of routing-instance representing a VRF. This is covered in base routing model [I-D.ietf-netmod-routing-cfg].

3.2.2. Route distinguisher

Route distinguisher (RD) is an unique identifier used in VPN routes to distinguish prefixes across different VPNs. RD is 8 byte field as defined in the [RFC4364]. Where the first two bytes refer to type followed by 6 bytes of value. The format of the value is dependent on type. In the yang model, RDs are defined l3vpn container under routing-instance.

3.2.3. Import and export route target

Route-target (RT) is an extended community used to specify the rules for importing and exporting the routes for each VRF as defined in [RFC4364]. This is applicable in the context of an address-family under the VRF. Under the l3vpn container, statements for import and export route-targets are added for ipv4 and ipv6 address family. Both import and export sets are modeled as a list of rout-targets. An import rule is modeled as list of RTs or a policy leafref specifying the list of RTs to be matched for importing routes into the VRF. Similarly an export rule is set or RTs or a policy leafref specifying the list of RTs which should be attached to routes exported from this VRF. In the case where policy is used to specify the RTs, a reference to the policy via leafref is used in this model, but actual definition of policy is outside the scope of this document. In addition, this section also defines parameters for the import from global routing table and export to global routing table, as well as route limit per VPN instance for ipv4 and ipv6 address family.

3.2.4. Forwarding mode

This configuration augments interface list under interface container under a routing-instance as defined in IETF routing model [I-D.ietf-netmod-routing-cfg]. Forwarding mode configuration is required under the ASBR facing interface to enable mpls forwarding for directly connected BGP peers for inter-as option B peering.

3.2.5. Label security

For inter-as option-B peering across ASs, under the ASBR facing interface, mpls label security enables the checks for RPF label on incoming packets. Ietf-interface container is augmented to add this config.

3.2.6. Yang tree

```
 augment /rt:routing/rt:routing-instance:
   +--rw l3vpn
      +--rw route-distinguisher
      |  +--rw config
      |  |  +--rw rd?   string
      |  +--ro state
      |     +--ro rd?   string
      +--rw ipv4
      |  +--rw unicast
      |     +--rw import-routes
```

```
          |     |  +--rw config
          |     |  |  +--rw route-targets
          |     |  |  |  +--rw rts* [rt]
          |     |  |  |     +--rw rt    string
          |     |  |  +--rw route-policy?    string
          |     |  +--ro state
          |     |     +--ro route-targets
          |     |     |  +--ro rts* [rt]
          |     |     |     +--ro rt    string
          |     |     +--ro route-policy?    string
          |     +--rw export-routes
          |     |  +--rw config
          |     |  |  +--rw route-targets
          |     |  |  |  +--rw rts* [rt]
          |     |  |  |     +--rw rt    string
          |     |  |  +--rw route-policy?    string
          |     |  +--ro state
          |     |     +--ro route-targets
          |     |     |  +--ro rts* [rt]
          |     |     |     +--ro rt    string
          |     |     +--ro route-policy?    string
          |     +--rw import-export-routes
          |     |  +--rw config
          |     |  |  +--rw route-targets
          |     |  |  |  +--rw rts* [rt]
          |     |  |  |     +--rw rt    string
          |     |  |  +--rw route-policy?    string
          |     |  +--ro state
          |     |     +--ro route-targets
          |     |     |  +--ro rts* [rt]
          |     |     |     +--ro rt    string
          |     |     +--ro route-policy?    string
          |     +--rw import-from-global
          |     |  +--rw config
          |     |  |  +--rw enable?             boolean
          |     |  |  +--rw advertise-as-vpn?   boolean
          |     |  |  +--rw route-policy?       string
          |     |  |  +--rw bgp-valid-route?    boolean
          |     |  |  +--rw protocol?           enumeration
          |     |  |  +--rw instance?           string
          |     |  +--ro state
          |     |     +--ro enable?             boolean
          |     |     +--ro advertise-as-vpn?   boolean
          |     |     +--ro route-policy?       string
          |     |     +--ro bgp-valid-route?    boolean
          |     |     +--ro protocol?           enumeration
          |     |     +--ro instance?           string
          |     +--rw export-to-global
```

```
         |     |  +--rw config
         |     |  |  +--rw enable?   boolean
         |     |  +--ro state
         |     |     +--ro enable?   boolean
         |     +--rw routing-table-limit
         |     |  +--rw config
         |     |  |  +--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
         |     |  +--ro state
         |     |     +--ro routing-table-limit-number?   uint32
         |     |     +--ro (routing-table-limit-action)?
         |     |        +--:(enable-alert-percent)
         |     |        |  +--ro alert-percent-value?          uint8
         |     |        +--:(enable-simple-alert)
         |     |           +--ro simple-alert?                 boolean
         |     +--rw tunnel-params
         |        +--rw config
         |        |  +--rw tunnel-policy?   string
         |        +--ro state
         |           +--ro tunnel-policy?   string
```

```
augment /if:interfaces/if:interface:
   +--rw forwarding-mode
   |  +--rw config
   |  |  +--rw forwarding-mode?   fwd-mode-type
   |  +--ro state
   |     +--ro forwarding-mode?   fwd-mode-type
   +--rw mpls-label-security
      +--rw config
      |  +--rw rpf?   boolean
      +--ro state
         +--ro rpf?   boolean
```

3.3. BGP Specific Configuration

The BGP specific configuration for L3VPNs is defined by augmenting base BGP model [I-D.shaikh-idr-bgp-model]. In particular, specific knobs are added under neighbor and address family containers to handle VPN routes and ASBR peering.

3.3.1. VPN peering

For Peering between PE routers, specific VPN address family needs to be enabled under BGP container in the default routing-instance. Base BGP draft [I-D.shaikh-idr-bgp-model] has l3vpn address family in the list of identity refs for AFs under global and neighbor modes. The same is augmented here for additional knobs. For peering with CE routers the VRF specific BGP configurations such as neighbors and address-family are covered in base BGP config, except that such configuration will be in the context of a VRF. The instance of BGP in this case would be a separate instance in the context of routing instance realizing a VRF.

3.3.2. VPN prefix limits

Limits for max number of VPN prefixes for a PE router is defined in the context of VPN address family under BGP. This would be the total number of prefixes in VPN table per AF in the context of BGP protocol. Route table limit for ipv4 and ipv6 address family for each VPN instance is also defined under BGP. The total prefix limit per VPN, including all the protocols is defined in the context of VRF address family under routing instance.

3.3.3. Label Mode

Label mode knobs control the label allocation behavior for VRF routes. Such as to specify Per-site, Per-vpn and Per-route label allocation. These knobs augment BGP global AF containers in the context of default routing instance.

3.3.4. ASBR options

This includes few specific knobs for ASBR peering methods illustrated in [RFC4364]. Such as route target retention on ASBRs and rewrite next hop to self, for inter-as VPN peering across ASBRs with option-B method. Similarly next hop unchanged on ASBRs for option-C peering. Appropriate containers under BGP AF and NBR modes are augmented for these parameters. As a note, when a knob is applicable for neighbor, it is also defined under corresponding peer-group container.

3.3.5. Yang tree

```
module: ietf-bgp-l3vpn
augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
   +--rw retain-rts
      +--rw config
```

```
      |  +--rw all?            empty
      |  +--rw route-policy?   string
      +--ro state
         +--ro all?            empty
         +--ro route-policy?   string
   +--rw prefix-limit
      +--rw config
      |  +--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
      +--ro state
         +--ro prefix-limit-number?   uint32
         +--ro (prefix-limit-action)?
            +--:(enable-alert-percent)
            |  +--ro alert-percent-value?   uint8
            |  +--ro route-unchanged?       boolean
            +--:(enable-simple-alert)
               +--ro simple-alert?          boolean       ...

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
   +--rw config
   |  +--rw label-mode?   bgp-label-mode
   +--ro state
      +--ro label-mode?   bgp-label-mode
   +--rw routing-table-limit
      +--rw config
      |  +--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
      +--ro state
         +--ro routing-table-limit-number?   uint32
         +--ro (routing-table-limit-action)?
            +--:(enable-alert-percent)
            |  +--ro alert-percent-value?          uint8
            +--:(enable-simple-alert)
               +--ro simple-alert?                 boolean
      ...

augment /bgp:bgp/bgp:neighbors/bgp:neighbor:
   +--rw nexthop-options
      +--rw config
```

```
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:peer-groups/bgp:peer-group:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:neighbors/bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:peer-groups/bgp:peer-group/bgp:afi-safis/bgp:afi-safi:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean
```

4. BGP Yang Module

<CODE BEGINS> file "ietf-bgp-l3vpn@2016-02-22.yang"

```
module ietf-bgp-l3vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-l3vpn";
  // replace with IANA namespace when assigned
  prefix l3vpn ;

  import ietf-routing {
    prefix rt;
```

```
    revision-date 2015-10-16;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-bgp {
    prefix bgp;
    revision-date 2016-01-06;
  }

  organization
     "IETF BGP Enabled Services WG";

  contact
     "draft-dhjain-bess-l3vpn-yang@tools.ietf.org";

  description
    "This YANG module defines a YANG data model to configure and manage BGP
Layer3 VPNs.
     It augments the IETF bgp yang model and IETF routing model to add L3VPN
specific
     configuration and operational knobs.


     Terms and Acronyms

     AF : Address Family

     AS : Autonomous System

     ASBR : Autonomous Systems Border Router

     BGP (bgp) : Border Gateway Protocol

     CE  : Customer Edge

     IP (ip) : Internet Protocol

     IPv4 (ipv4):Internet Protocol Version 4

     IPv6 (ipv6): Internet Protocol Version 6

     L3VPN: Layer 3 VPN

     PE : Provider Edge

     RT : Route Target
```

```
     RD : Route Distinguisher

     VPN : Virtual Private Network

     VRF : Virtual Routing and Forwarding

    ";

  revision 2016-02-22 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for BGP L3VPN config management";
  }


  grouping bgp-rd-spec {
    description "Route distinguisher specification as per RFC4364";
      leaf rd {
        type string;
        description "Route distinguisher value as per RFC4364";
    }
  }
  grouping bgp-rd {
    description "BGP route distinguisher";
    container route-distinguisher {
      description "Route distinguisher";
        container config {
          description "Configuration parameters for route distinguisher";
          uses bgp-rd-spec ;
        }
        container state {
          config "false" ;
          description "State information for route distinguisher";
          uses bgp-rd-spec ;
        }
     }
  }


  typedef bgp-label-mode {
    type enumeration {
      enum per-ce {
        description "Allocate labels per CE";
      }
      enum per-route {
        description "Allocate labels per prefix";
      }
```

```
      enum per-vpn {
        description "Allocate labels per VRF";
      }
    }
    description "BGP label allocation mode";
  }

  typedef fwd-mode-type {
    type enumeration {
      enum mpls {
        description "Forwarding mode mpls";
      }
    }
    description "Enable forwarding mode under ASBR facing interface";
  }

  grouping forwarding-mode {
    description "Forwarding mode of interface for ASBR scenario";
    container forwarding-mode {
      description "Forwarding mode of interface for ASBR scenario";
      container config {
        description "Configuration of Forwarding mode";
        leaf forwarding-mode {
          type  fwd-mode-type;
          description "Forwarding mode for this interface";
        }
      }
      container state {
        config "false";
        description "State information of Forwarding mode";
        leaf forwarding-mode {
        type  fwd-mode-type;
          description "Forwarding mode for this interface";
        }
      }
    }
 }

  grouping label-security {
    description "Mpls label security for ASBR option B scenario";
    container mpls-label-security {
      description "MPLS label secruity";
      container config {
        description "Configuration parameters";
        leaf rpf {
          type boolean;
          description "Enable MPLS label security rpf on interface";
        }
```

```
      }
      container state {
        config "false";
        description "State information";
        leaf rpf {
          type boolean;
          description "MPLS label security rpf on interface";
        }
      }
    }
  }


  //per VPN instance table limit under BGP
  grouping 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.";
        }
      }
    }
  }

  grouping  vpn-pfx-limit {
    description "Per VPN instance table limit under BGP";
    container vpn-prefix-limit {
      description "Prefix limit for this table";
      container config {
         description "Config parameters";
         uses prefix-limit;
      }
      container state {
         config "false";
         description "State parameters";
         uses prefix-limit;
     }
   }
 }

  grouping route-target-set {
    description
      "Extended community route-target set ";
    container route-targets {
      description
        "Route-target" ;
      list rts {
        key "rt" ;
        description
```

```
          "List of route-targets" ;
        leaf rt {
          type string {
            pattern '([0-9]+:[0-9]+)';
          }
          description "Route target extended community as per RFC4360";
        }
      }
    }
    leaf route-policy {
      type string;
      description
        "Reference to the policy containing set of routes.
               TBD: leafref to policy entry in IETF policy model";
    }
  }

  grouping import-from-gbl {
    description "Import from global routing table";
    leaf enable {
      type boolean;
        description "Enable";
    }
    leaf advertise-as-vpn {
      when "../from-default-vrf == TRUE" {
        description "This option is valid only when importing from global
routing table";
      }
      type boolean;
      description "Advertise routes imported from global table as VPN routes";
    }
    leaf route-policy {
      type string;
      description "Policy name or import routes";
    }

    leaf bgp-valid-route {
      type boolean;
      description "Enable all valid routes (including non-best paths) to be
candidate
                   for import";
    }

    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 instance {
        type string;
        description
             "Specifies the instance id of the protocol";
      }
  }
```

```
  grouping global-imports {
      description "Grouping for imports from global routing table";
    container import-from-global {
      description "Import from global global routing table";
      container config {
          description "Configuration";
          uses import-from-gbl;
      }
      container state {
          config "false";
          description "State";
          uses import-from-gbl;
      }
    }
  }


  grouping export-to-gbl {
    description "Export routes to default VRF";
    leaf enable {
      type boolean;
        description "Enable";
    }
  }

  grouping global-exports {
    description "Grouping for exports routes to global table";
    container export-to-global {
      description "Export to global routing table";
      container config {
         description "Configuration";
         uses export-to-gbl;
      }
      container state {
        config "false";
        description "State";
        uses export-to-gbl;
      }
    }
  }

  grouping route-import-set {
    description "Grouping to specify rules for route import";
    container import-routes {
      description "Set of route-targets to match to import routes into VRF";
      container config {
         description
               "Configuration parameters for import routes";
```

```
          uses route-target-set ;
      }
      container state {
        config "false" ;
        description
               "State information for the import routes";
          uses route-target-set ;
      }
    }
  }
  grouping route-export-set {
    description "Grouping to specify rules for route export";
    container export-routes {
      description "Set of route-targets to attach with exported routes from
VRF";
      container config {
        description
             "Configuration parameters for export routes";
          uses route-target-set ;
      }
      container state {
        config "false" ;
        description
              "State information for export routes";
          uses route-target-set ;
      }
    }
  }

  grouping route-import-export-set {
    description "Grouping to specify rules for route import/export both";
    container import-export-routes {
      description "Set of route-targets for import/export both";
      container config {
        description "Both import/export routes";
        uses route-target-set;
      }
      container state {
        config "false" ;
          description "Both import/export routes";
          uses route-target-set;
      }
    }
  }

  grouping route-tbl-limit-params {
     description "Grouping for VPN table prefix limit config";
     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.";
             }
           }
       }
   }

   grouping routing-tbl-limit {
      description ".";
      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.";
          container config {
             description "Config parameters";
             uses route-tbl-limit-params;
          }
          container state {
              config "false";
              description "State parameters";
              uses route-tbl-limit-params;
          }
       }
   }

  // Tunnel policy parameters
  grouping tunnel-params {
     description "Tunnel parameters";
     container tunnel-params {
        description "Tunnel config parameters";
        container config {
           description "configuration parameters";
           leaf tunnel-policy {
              type string;
              description
                  "Tunnel policy name.";
           }
        }
        container state {
           config "false";
           description "state parameters";
           leaf tunnel-policy {
              type string;
              description
                   "Tunnel policy name.";
           }
        }
     }
  }

  // Grouping for the L3vpn specific parameters under VRF (aka routing-
instance)
  grouping l3vpn-vrf-params {
      description "Specify route filtering rules for import/export";
      container ipv4 {
         description "Specify route filtering rules for import/export";
```

```
        container unicast {
```

```
            description "Specify route filtering rules for import/export";
            uses route-import-set;
            uses route-export-set;
            uses route-import-export-set;
            uses global-imports;
            uses global-exports;
            uses routing-tbl-limit;
            uses tunnel-params;
          }
      }
      container ipv6 {
         description "Ipv6 address family specific rules for import/export";
         container unicast {
             description "Ipv6 unicast address family";
             uses route-import-set;
             uses route-export-set;
             uses route-import-export-set;
             uses global-imports;
             uses global-exports;
             uses routing-tbl-limit;
             uses tunnel-params;
         }
     }
  }

  grouping bgp-label-mode {
       description "MPLS/VPN label allocation mode";
       container config {
         description "Configuration parameters for label allocation mode";
         leaf label-mode {
           type bgp-label-mode;
           description "Label allocation mode";
         }
       }
       container state {
         config "false" ;
         description "State information for label allocation mode";
         leaf label-mode {
           type bgp-label-mode;
           description "Label allocation mode";
         }
       }
  }

  grouping retain-route-targets {
       description "Grouping for route target accept";
    container retain-route-targets {
      description "Control route target acceptance behavior for ASBRs";
```

```
      container config {
         description "Configuration parameters for retaining route targets";
         leaf all {
            type empty;
            description "Disable filtering of all route-targets";
         }
         leaf route-policy {
            type string;
            description "Filter routes as per filter policy name
                         TBD: leafref to IETF routing policy model";
         }
      }
      container state {
         config "false" ;
         description "State information for retaining route targets";
         leaf all {
            type empty;
            description "Disable filtering of all route-targets";
         }
         leaf route-policy {
            type string;
            description "Filter routes as per filter policy name";
         }
      }
    }
  }

  grouping nexthop-opts {
     description "Next hop control options for inter-as route exchange";
     leaf next-hop-self {
        type boolean;
        description "Set nexthop of the route to self when advertising routes";
     }
     leaf next-hop-unchanged {
        type boolean;
        description "Enforce no nexthop change when advertising routes";
     }
  }

  grouping asbr-nexthop-options {
     description "Nexthop parameters for inter-as VPN options ";
     container nexthop-options {
         description "Nexthop related options for inter-as options";
         container config {
             description "Configuration parameters for nexthop options";
             uses nexthop-opts;
         }
         container state {
```

```
            config "false";
            description "State information for nexthop options" ;
            uses nexthop-opts;
         }
      }
   }

   //
   // VRF specific parameters.
   // RD and RTs are added in VRF routing-intance, therefore per per VRF scoped.
   //

   // route import-export rules in VRF context
   // (routing instance container in ietf-routing model).
   augment "/rt:routing/rt:routing-instance" {
      description "Augment routing instance container for per VRF import/export
config";
      container l3vpn {
         when "../type='rt:vrf-routing-instance'" {
            description "This container is only valid for vrf routing
instance.";
         }
         description "Configuration of L3VPN specific parameters";

         uses bgp-rd;
         uses l3vpn-vrf-params ;
      }
   }

   // bgp mpls forwarding enable required for inter-as option AB.
   augment "/if:interfaces/if:interface" {
     description "BGP mpls forwarding mode configuration on interface for ASBR
scenario";
     uses forwarding-mode ;
     uses label-security;
   }

   //
   // BGP Specific Paramters
   //

   //
   // Retain route-target for inter-as option ASBR knob.
   // vpn prefix limits
   // vpnv4/vpnv6 address-family only.
   augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-
unicast" {
     description "Retain route targets for ASBR scenario";
```

```
    uses retain-route-targets;
    uses vpn-pfx-limit;
  }
```

```
  augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-
unicast" {
    description "Retain route targets for ASBR scenario";
    uses retain-route-targets;
    uses vpn-pfx-limit;
  }

  // Label allocation mode configuration. Certain AFs only.
  augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast" {
     description "Augment BGP global AF mode for label allocation mode
configuration";
     uses bgp-label-mode ;
     uses routing-tbl-limit;
  }

  augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast" {
     description "Augment BGP global AF mode for label allocation mode
configuration";
     uses bgp-label-mode ;
     uses routing-tbl-limit;
  }


  // Nexthop options for the inter-as ASBR peering.
  augment "/bgp:bgp/bgp:neighbors/bgp:neighbor" {
     description "Augment BGP NBR mode with nexthop options for inter-as
ASBRs";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:peer-groups/bgp:peer-group" {
     description "Augment BGP peer-group mode with nexthop options for inter-as
ASBRs";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:neighbors/bgp:neighbor/bgp:afi-safis/bgp:afi-safi" {
     description "Augment BGP NBR AF mode with nexthop options for inter-as
ASBRs";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:peer-groups/bgp:peer-group/bgp:afi-safis/bgp:afi-
safi" {
     description "Augment BGP peer-group AF mode with nexthop options for
inter-as ASBRs";
     uses asbr-nexthop-options;
  }
```

```
}
```

<CODE ENDS>

5. IANA Considerations

6. Security Considerations

The transport protocol used for sending the BGP L3VPN data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg] and [I-D.shaikh-idr-bgp-model].

7. Acknowledgements

The authors would like to thank TBD for their detail reviews and comments.

8. References

8.1. Normative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-15 (work in progress), May 2014.

[I-D.shaikh-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-shaikh-idr-bgp-model-02 (work in progress), June 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>.

[RFC2547] Rosen, E. and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, DOI 10.17487/RFC2547, March 1999, <http://www.rfc-editor.org/info/rfc2547>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <http://www.rfc-editor.org/info/rfc3552>.

[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>.

8.2. Informative References

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <http://www.rfc-editor.org/info/rfc5492>.

Authors' Addresses

Dhanendra Jain
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dhjain@cisco.com

Keyur Patel
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: keyupate@cisco.com

Patrice Brissette
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: pbrisset@cisco.com

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: xliu@kuatrotech.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