Network Working Group   D. Jain
Internet-Draft   Cisco
Intended status: Standards Track   K. Patel
Expires: October 15, 2021   Arrcus, Inc
                                                       P. Brissette
                                                              Cisco
                                                               Z. Li
                                                          S. Zhuang
                                                Huawei Technologies
                                                             X. Liu
                                                              Jabil
                                                            J. Haas
                                                           S. Esale
                                                   Juniper Networks
                                                             B. Wen
                                                            Comcast
                                                     April 13, 2021

                  Yang Data Model for BGP/MPLS L3 VPNs
                       draft-ietf-bess-l3vpn-yang-05

Abstract

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

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 will expire on October 15, 2021.

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.

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 network-instance container defined in the network instance model
   [I-D.ietf-rtgwg-ni-model] 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.ietf-idr-bgp-model] .

3.2. VRF Specific Configuration

   IETF network instance model defines various ni-types, one of which is
   l3vpn. This provides an anchor point to add a new container l3vpn.
   Under this container per VPN parameters pertaining to L3VPN are
   added.

3.2.1. VRF interface

   To associate a VRF instance with an interface, bind-network-instance
   config should be used. This is covered in the base network instance
   model [I-D.ietf-rtgwg-ni-model].

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, RD is defined under l3vpn container
   under a network-instance. Yang datatype for RD is imported from
   [RFC8294].

3.2.3. Import and export route targets

   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,
   yang datatype for which is imported from [RFC8294]. An import rule
   is modeled as list of RTs or a leafref to the route policy
   [I-D.ietf-rtgwg-policy-model] specifying the list of RTs to be
   matched for importing the routes into the VRF. Similarly, an export
   rule is modeled as a list of RTs or a leafref the route policy [I-
   D.ietf-rtgwg-policy-model] specifying the list of RTs which should be

attached to routes exported from the 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 network instance as defined in IETF network instance model
[I-D.ietf-rtgwg-ni-model]. 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

```
module: ietf-bgp-l3vpn
module: ietf-bgp-l3vpn
   augment /ni:network-instances/ni:network-instance/ni:ni-type:
     +--:(l3vpn)
        +--rw l3vpn
           +--rw rd?         bgp-rd-type
           +--ro auto-rd?    rt-types:route-distinguisher
           +--rw ipv4
           |  +--rw unicast
           |     +--rw vpn-targets
           |     |  +--rw vpn-target* [route-target]
           |     |  |  +--rw route-target        rt-types:route-target
           |     |  |  +--rw route-target-type    rt-types:route-target-type
           |     |  +--rw route-policy?   -> /rt-pol:routing-policy/policy-definit
ions/policy-definition/name
           |        +--rw import-from-global
           |        |  +--rw enable?           boolean
           |        |  +--rw advertise-as-vpn?  boolean
           |        |  +--rw route-policy?        -> /rt-pol:routing-policy/policy-def
initions/policy-definition/name
           |        |  +--rw bgp-valid-route?   boolean
           |        |  +--rw protocol?          enumeration
           |        |  +--rw instance?          string
           |        +--rw export-to-global
           |        |  +--rw enable?   boolean
           |        +--rw routing-table-limit
           |        |  +--rw routing-table-limit-number?   uint32
           |        |  +--rw (routing-table-limit-action)?
           |        |     +--:(enable-alert-percent)
           |        |     |  +--rw alert-percent-value?          rt-types:percentage
           |        |     +--:(enable-simple-alert)
           |        |        +--rw simple-alert?                 boolean
           |        +--rw tunnel-params
           |           +--rw tunnel-policy?   string
           +--rw ipv6
           ...

   augment /if:interfaces/if:interface:
     +--rw forwarding-mode?        enumeration
     +--rw mpls-label-security
        +--rw rpf?   boolean
```

3.3. BGP Specific Configuration

   The BGP specific configuration for L3VPNs is defined by augmenting
   base BGP model [I-D.ietf-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 context of core instance. Base
BGP draft [I-D.ietf-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 vrf-root
as defined in [I-D.ietf-rtgwg-ni-model].

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 for inter-as
VPN peering across ASBRs with option-B method. Appropriate
containers under BGP AF are augmented.

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-route-targets
    |  +--rw all?            empty
    |  +--rw route-policy?   -> /rt-pol:routing-policy/policy-definitions/policy-
definition/name
    +--rw vpn-prefix-limit
       +--rw prefix-limit-number?   uint32
       +--rw (prefix-limit-action)?
          +--:(enable-alert-percent)
          |  +--rw alert-percent-value?   rt-types:percentage
          |  +--rw route-unchanged?       boolean
          +--:(enable-simple-alert)
             +--rw simple-alert?          boolean
...
augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
    +--rw label-mode?           bgp-label-mode
    +--rw routing-table-limit
       +--rw routing-table-limit-number?   uint32
       +--rw (routing-table-limit-action)?
          +--:(enable-alert-percent)
          |  +--rw alert-percent-value?           rt-types:percentage
          +--:(enable-simple-alert)
             +--rw simple-alert?                  boolean
...
```

4. BGP Yang Module

<CODE BEGINS> file "ietf-bgp-l3vpn@2018-04-17.yang"

```
module ietf-bgp-l3vpn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-l3vpn";
  // replace with IANA namespace when assigned
  prefix l3vpn ;

  import ietf-network-instance {
    prefix ni;
  }

  import ietf-routing-types {
    prefix rt-types;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-bgp {
    prefix bgp;
```

```
  }

  import ietf-routing-policy {
    prefix rt-pol;
  }

  organization
     "IETF BGP Enabled Services WG";

  contact
     "BESS working group - bess@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 network instance 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 2018-04-17 {
    description
      "Import latest revisions of ietf-network-instance" +
      "Added leafrefs to named policy defs from routing-policy model" +
      "Minor other text corrections";
    reference "";
  }

  revision 2017-10-15 {
    description
      "Removed state containers per NMDA aligntment" +
      "Changes for network instance ni-type alignment" +
      "Other cleanups";
    reference "";
  }
  revision 2017-04-25 {
    description
      "Reused ietf-roting-types.yang for vpn route-targets" +
      " and route distinguisher types";
    reference "";
  }

  revision 2016-09-09 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for BGP L3VPN config management";
  }

  // Local typedef for RD
  typedef bgp-rd-type {
    type union {
      // Either RD value as per IETF routing types or AUTO assigned value
      type rt-types:route-distinguisher;
      type enumeration {
        enum auto-assigned {
          description "Assigned by system";
        }
      }
    }
    description "BGP RD type augmentation for configured and Auto RD value";
  }

  //Label mode

  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";
  }

  //RD
  grouping route-distinguisher-params {
    description "Route distinguisher value as per RFC4364";
    leaf rd {
      type bgp-rd-type;
      description "Route distinguisher value as per RFC4364";
    }
    leaf auto-rd {
      type rt-types:route-distinguisher;
      config false;
      description
        "Automatically assigned RD value when rd AUTO is configured";
    }
  }

  //Fwding mode
  grouping forwarding-mode {
    description "Forwarding mode of interface for ASBR scenario";
    leaf forwarding-mode {
      type enumeration {
        enum mpls {
          description "Forwarding mode mpls";
        }
      }
      description "Forwarding mode of interface for ASBR scenario";
    }
  }

  grouping label-security {
    description "Mpls label security for ASBR option B scenario";
    container mpls-label-security {
      description "MPLS label secruity";
      leaf rpf {
        type boolean;
```

```
            description "Enable MPLS label security rpf on interface";
        }
    }
  }

  //per VPN instance table limit under BGP
  grouping  vpn-pfx-limit {
    description "Per VPN instance table limit under BGP";
    container vpn-prefix-limit {
      description
       "The prefix limit config 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 rt-types:percentage;
            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 global-imports {
      description "Grouping for imports from global routing table";
    container import-from-global {
      description "Import from global routing table";
      leaf enable {
        type boolean;
        description "Enable";
      }
      leaf advertise-as-vpn {
        type boolean;
        description
          "Advertise routes imported from global table as VPN routes";
      }
      leaf route-policy {
        type leafref {
          path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:name";
          require-instance true;
        }
        description "Route policy as a filter for importing 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:";
            }
          }
        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-exports {
    description "Grouping for exports routes to global table";
    container export-to-global {
      description "Export to global routing table";
      leaf enable {
        type boolean;
          description "Enable";
      }
    }
  }

  grouping route-target-params {
    description "Grouping to specify rules for route import and export";
    container vpn-targets {
      description
        "Set of route-targets to match for import and export routes
         to/from VRF";
      uses rt-types:vpn-route-targets;
      leaf route-policy {
        type leafref {
          path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:name";
          require-instance true;
        }
        description
          "Reference to the route policy containing set of route-targets.";
      }
    }
  }

  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 rt-types:percentage;
```

```
                    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.";

            uses route-tbl-limit-params;
        }
    }

  // Tunnel policy parameters
  grouping tunnel-params {
      description "Tunnel parameters";
      container tunnel-params {
```

```
      description "Tunnel config parameters";
         leaf tunnel-policy {
            type string;
            description
               "Tunnel policy to steer the VPN traffic into specific tunnel";
         }
      }
   }

// Grouping for the L3vpn specific parameters under VRF
// (network-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-target-params;
         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-target-params;
          uses global-imports;
          uses global-exports;
          uses routing-tbl-limit;
          uses tunnel-params;
       }
    }
}

grouping bgp-label-mode {
    description "MPLS/VPN 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";
       leaf all {
         type empty;
         description "Accept all route targets.";
       }
       leaf route-policy {
         type leafref {
           path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
           "rt-pol:policy-definition/rt-pol:name";
           require-instance true;
         }
         description "Reference to route policy containing set of route-targets to
 accept.";
       }
     }
   }

   //
   // VRF specific parameters.
   // RD and RTs and route import-export rules are added under
   // network instance container in network instance model, hence
   // per VRF scoped
   augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
      description
        "Augment network instance for per VRF L3vpn parameters";
     case l3vpn {
       container l3vpn {
         description "Configuration of L3VPN specific parameters";

         uses route-distinguisher-params;
         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;
  }

  // TBD Additional oper state leafs

  // TBD RPCs

}

<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-rtgwg-ni-model] and [I-D.ietf-idr-bgp-model].

7. Acknowledgements

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

8. References

   [I-D.ietf-idr-bgp-model]
              Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP
              YANG Model for Service Provider Networks", draft-ietf-idr-
              bgp-model-10 (work in progress), November 2020.

   [I-D.ietf-rtgwg-ni-model]
              Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X.
              Liu, "YANG Model for Network Instances", draft-ietf-rtgwg-
              ni-model-12 (work in progress), March 2018.

   [I-D.ietf-rtgwg-policy-model]
              Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data
              Model for Routing Policy", draft-ietf-rtgwg-policy-
              model-27 (work in progress), January 2021.

   [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>.

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

   [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>.

   [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger,
              "Common YANG Data Types for the Routing Area", RFC 8294,
              DOI 10.17487/RFC8294, December 2017,
              <https://www.rfc-editor.org/info/rfc8294>.

Authors' Addresses

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

   Email: dhanendra.ietf@gmail.com

   Keyur Patel
   Arrcus, Inc

   Email: keyur@arrcus.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
156 Beiqing Road
Beijing, 100095
China

Email: zhuangshunwan@huawei.com

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean, VA 22102
USA

Email: Xufeng_liu@jabil.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