BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2018

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

October 18, 2017

Yang Data Model for BGP/MPLS L3 VPNs
draft-ietf-bess-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 https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 21, 2018.

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 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, RDs are defined l3vpn container under network-instance.

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. 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 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
  augment /ni:network-instances/ni:network-instance/ni:ni-type:
    +--:(l3vpn)
       +--rw l3vpn
          +--rw rd?        union
          +--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?   string
          |     +--rw import-from-global
          |     |  +--rw enable?             boolean
          |     |  +--rw advertise-as-vpn?   boolean
          |     |  +--rw route-policy?       string
          |     |  +--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

  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 address-family containers under BGP base model are augmented for this.

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?   string
    +--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@2017-10-18.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;
    revision-date 2017-09-27;
  }

  import ietf-routing-types {
    prefix rt-types;
    revision-date 2017-10-13;
  }

  import ietf-interfaces {
    prefix if;
```

```
  }

  import ietf-bgp {
    prefix bgp;
    revision-date 2016-06-21;
  }

  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 2017-10-18 {
    description
      "Removed state containers per NMDA alignment" +
      "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";
  }

  //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 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 "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 string;
        description "Route policy as 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 string;
        description
          "Reference to the route policy containing set of route-targets.
           TBD: leafref to policy xpath in IETF route policy model";
       }
    }
  }

  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 "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";
      }
    }
  }

  //
  // 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;
  }
}
```

<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

8.1. Normative References

[I-D.ietf-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Zhdankin, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-02 (work in progress), July 2016.

[I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Network Instances", draft-ietf-rtgwg-ni-model-04 (work in progress), September 2017.

[I-D.ietf-rtgwg-routing-types]
Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Routing Area Common YANG Data Types", draft-ietf-rtgwg-routing-types-17 (work in progress), October 2017.

[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>.

8.2. Informative References

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

[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, <https://www.rfc-editor.org/info/rfc4271>.

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

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: 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
Huawei Bld., No.156 Beiqing Rd.
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