

BESS Working Group
Internet Draft
Intended status: Standards Track
Expires: June 2, 2020

Y. Liu
Individual
F. Guo
Huawei
S. Litkowski
Cisco
X. Liu
Volta Networks
R. Kebler
M. Sivakumar
Juniper
Dec 2, 2019

**Yang Data Model for Multicast in MPLS/BGP IP VPNs
draft-ietf-bess-mvpn-yang-02**

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on June 2, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Abstract

This document defines a YANG data model that can be used to configure and manage multicast in MPLS/BGP IP VPNs.

Table of Contents

1.	Introduction	2
1.1.	Terminology	3
1.2.	Tree Diagrams	3
1.3.	Prefixes in Data Node Names	4
2.	Design of Data Model	4
2.1.	Scope of Model	4
2.2.	Optional Capabilities	4
2.3.	Position of Address Family in Hierarchy	5
3.	Module Structure	5
4.	MVPN YANG Modules	13
5.	Security Considerations	36
6.	IANA Considerations	36
7.	References	37
7.1.	Normative References	37
7.2.	Informative References	38
8.	Acknowledgments	38
	Authors' Addresses	39

[1. Introduction](#)

YANG [[RFC6020](#)] [[RFC7950](#)] 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 encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interface, such as CLI and Programmatic APIs.

This document defines a YANG data model that can be used to configure and manage Multicast in MPLS/BGP IP VPN (MVPN). It includes Cisco systems' solution [[RFC6037](#)], BGP MVPN [[RFC6513](#)] [[RFC6514](#)] etc. This model will support the core MVPN protocols, as well as many other features mentioned in separate MVPN RFCs. In addition, Non-core features described in MVPN standards other than mentioned above RFC in separate documents.

1.1. Terminology

The terminology for describing YANG data models is found in [[RFC6020](#)] & [[RFC7950](#)].

The following abbreviations are used in this document and the defined model:

MVPN: Multicast Virtual Private Network [[RFC6513](#)].

PMSI: P-Multicast Service Interface [[RFC6513](#)].

PIM: Protocol Independent Multicast [[RFC7761](#)].

SM: Sparse Mode [[RFC7761](#)].

SSM: Source Specific Multicast [[RFC4607](#)].

BIDIR-PIM: Bidirectional Protocol Independent Multicast [[RFC5015](#)].

MLDP P2MP: Multipoint Label Distribution Protocol for Point to Multipoint [[RFC6388](#)].

MLDP MP2MP: Multipoint Label Distribution Protocol for Multipoint to Multipoint [[RFC6388](#)].

RSVP TE P2MP: Resource Reservation Protocol - Traffic Engineering for Point to Multipoint [[RFC4875](#)].

BIER: Bit Index Explicit Replication [[RFC8279](#)].

1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [[RFC8340](#)].

1.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1

Prefix	YANG module	Reference
ni	ietf-network-instance	[RFC8529]
l3vpn	ietf-bgp-l3vpn	[I-D.ietf-l3vpn-yang]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
acl	ietf-access-control-list	[RFC8519]

Table 1: Prefixes and Corresponding YANG Modules

2. Design of Data Model

2.1. Scope of Model

The model covers Rosen MVPN [RFC6037], BGP MVPN [RFC6513] [RFC6514]. The configuration of MVPN features, and the operational state fields and RPC definitions are not all included in this document of the data model. This model can be extended, though the structure of what has been written may be taken as representative of the structure of the whole model.

This model does not cover other MVPN related protocols such as MVPN Extranet [RFC7900] or MVPN MLDP In-band signaling [RFC7246] etc., these will be specified in separate documents.

2.2. Optional Capabilities

This model is designed to represent the capabilities of MVPN devices with various specifications, including some with basic subsets of the MVPN protocols. The main design goals of this document are that any major now-existing implementation may be said to support the basic model, and that the configuration of all implementations meeting the specification is easy to express through some

combination of the features in the basic model and simple vendor augmentations.

On the other hand, operational state parameters are not so widely designated as features, as there are many cases where the defaulting of an operational state parameter would not cause any harm to the system, and it is much more likely that an implementation without native support for a piece of operational state would be able to derive a suitable value for a state variable that is not natively supported.

For the same reason, wide constant ranges (for example, timer maximum and minimum) will be used in the model. It is expected that vendors will augment the model with any specific restrictions that might be required. Vendors may also extend the features list with proprietary extensions.

2.3. Position of Address Family in Hierarchy

The current draft contains MVPN IPv4 and IPv6 as separate schema branches in the structure. The reason for this is to inherit l3vpn yang model structure and make it easier for implementations which may optionally choose to support specific address families. And the names of some objects may be different between the IPv4 and IPv6 address families.

3. Module Structure

The MVPN YANG model follows the Guidelines for YANG Module Authors (NMDA) [[RFC8342](#)]. The operational state data is combined with the associated configuration data in the same hierarchy [[RFC8407](#)]. The MVPN modules define for both IPv4 and IPv6 in a two-level hierarchy as listed below:

Instance level: Only including configuration data nodes now. MVPN configuration attributes for the entire routing instance, including route-target, I-PMSI tunnel and S-PMSI number, common timer etc.

PMSI tunnel level: MVPN configuration attributes applicable to the I-PMSI and per S-PMSI tunnel configuration attributes, including tunnel mode, tunnel specific parameters and threshold etc. MVPN PMSI tunnel operational state attributes applicable to the I-PMSI and per S-PMSI tunnel operational state attributes, including tunnel mode, tunnel role, tunnel specific parameters and referenced private source and group address etc.

Where fields are not genuinely essential to protocol operation, they are marked as optional. Some fields will be essential but have a default specified, so that they need not be configured explicitly.

This MVPN model augments "/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4:" for IPv4 MVPN service and "/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6" for IPv6 MVPN service specified in [[I-D.ietf-l3vpn-yang](#)].

```
augment /ni:network-instances/ni:network-instance/ni:ni-type
  /l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4:
  +--rw multicast
    +--rw signaling-mode?          enumeration
    +--rw auto-discovery-mode?     enumeration
    +--rw mvpn-type?               enumeration
    +--rw is-sender-site?          boolean {mvpn-sender}?
    +--rw rpt-spt-mode?            enumeration
    +--rw ecmp-load-balance-mode?
      | enumeration {mvpn-ecmp-load-balance}?
    +--rw mvpn-route-targets {mvpn-separate-rt}?
      | +--rw mvpn-route-target* [mvpn-rt-type mvpn-rt-value]
      |   +--rw mvpn-rt-type       enumeration
      |   +--rw mvpn-rt-value      string
    +--rw mvpn-ipmsi-tunnel-ipv4
      | +--rw tunnel-type?          p-tunnel
      | +--rw (ipmsi-tunnel-attribute)?
      | | +--:(rsvp-te-p2mp)
      | | | +--rw rsvp-te-p2mp-template? string
      | | +--:(mldp-p2mp)
      | | +--:(pim-ssm)
      | | | +--rw ssm-default-group-addr?
      | | |   rt-types:ip-multicast-group-address
      | | +--:(pim-sm)
      | | | +--rw sm-default-group-addr?
      | | |   rt-types:ip-multicast-group-address
      | | +--:(bidir-pim)
      | | | +--rw bidir-default-group-addr?
      | | |   rt-types:ip-multicast-group-address
      | | +--:(ingress-replication)
      | | +--:(mldp-mp2mp)
      | | +--:(bier)
      | | +--rw inclusive-sub-domain-id? uint8
```

```

| |      +--rw inclusive-bitstring-length?  uint16
| +--ro (pmsi-tunnel-state-attribute)?
| |      +--:(rsvp-te-p2mp)
| | |      +--ro p2mp-id?                    uint16
| | |      +--ro tunnel-id?                  uint16
| | |      +--ro extend-tunnel-id?           uint16
| |      +--:(mldp-p2mp)
| | |      +--ro mldp-root-addr?              inet:ip-address
| | |      +--ro mldp-lsp-id?                 string
| |      +--:(pim-ssm)
| | |      +--ro ssm-group-addr?
| | | |      rt-types:ip-multicast-group-address
| |      +--:(pim-sm)
| | |      +--ro sm-group-addr?
| | | |      rt-types:ip-multicast-group-address
| |      +--:(bidir-pim)
| | |      +--ro bidir-group-addr?
| | | |      rt-types:ip-multicast-group-address
| |      +--:(ingress-replication)
| |      +--:(mldp-mp2mp)
| |      +--:(bier)
| | |      +--ro sub-domain-id?               uint8
| | |      +--ro bitstring-length?           uint16
| | |      +--ro bfir-id?                     uint16
| +--ro tunnel-role?                         enumeration
| +--ro upstream-vpn-label?
| |      rt-types:mpls-label {mvpn-aggregation-tunnel}?
| +--ro mvpn-pmsi-ipv4-ref-sg-entries
| |      +--ro mvpn-pmsi-ipv4-ref-sg-entries*
| | |      [ipv4-source-address ipv4-group-address]
| | |      +--ro ipv4-source-address          inet:ipv4-address
| | |      +--ro ipv4-group-address
| | | |      rt-types:ipv4-multicast-group-address
+--rw mvpn-spmsi-tunnels-ipv4
  +--rw switch-delay-time?                   uint8
  +--rw switch-back-holddown-time?          uint16
  +--rw tunnel-limit?                        uint16
  +--rw mvpn-spmsi-tunnel-ipv4* [tunnel-type]
    +--rw tunnel-type                        p-tunnel
    +--rw (spmsi-tunnel-attribute)?
    | +--:(rsvp-te-p2mp)
    | | +--rw rsvp-te-p2mp-template?         string

```

```

| +--:(p2mp-mldp)
| +--:(pim-ssm)
| | +--rw ssm-group-pool-addr?
| | | rt-types:ip-multicast-group-address
| | +--rw ssm-group-pool-masklength? uint8
| +--:(pim-sm)
| | +--rw sm-group-pool-addr?
| | | rt-types:ip-multicast-group-address
| | +--rw sm-group-pool-masklength? uint8
| +--:(bidir-pim)
| | +--rw bidir-group-pool-addr?
| | | rt-types:ip-multicast-group-address
| | +--rw bidir-group-pool-masklength? uint8
| +--:(ingress-replication)
| +--:(mldp-mp2mp)
| +--:(bier)
| | +--rw selective-sub-domain-id? uint8
| | +--rw selective-bitstring-length? uint16
+--rw switch-threshold? uint32
+--rw per-item-tunnel-limit? uint16
+--rw switch-wildcard-mode?
| enumeration {mvpn-switch-wildcard}?
+--rw explicit-tracking-mode?
| enumeration {mvpn-explicit-tracking}?
+--rw (address-mask-or-acl)?
| +--:(address-mask)
| | +--rw ipv4-group-addr?
| | | rt-types:ipv4-multicast-group-address
| | +--rw ipv4-group-masklength? uint8
| | +--rw ipv4-source-addr?
| | | inet:ipv4-address
| | +--rw ipv4-source-masklength? uint8
| +--:(acl-name)
| | +--rw group-acl-ipv4?
| | -> /acl:acls/acl/name
+--ro (pmsi-tunnel-state-attribute)?
| +--:(rsvp-te-p2mp)
| | +--ro p2mp-id? uint16
| | +--ro tunnel-id? uint16
| | +--ro extend-tunnel-id? uint16
| +--:(mldp-p2mp)

```

```

    | | +--ro mldp-root-addr?          inet:ip-address
    | | +--ro mldp-lsp-id?            string
    | +--:(pim-ssm)
    | | +--ro ssm-group-addr?
    | |         rt-types:ip-multicast-group-address
    | +--:(pim-sm)
    | | +--ro sm-group-addr?
    | |         rt-types:ip-multicast-group-address
    | +--:(bidir-pim)
    | | +--ro bidir-group-addr?
    | |         rt-types:ip-multicast-group-address
    | +--:(ingress-replication)
    | +--:(mldp-mp2mp)
    | +--:(bier)
    |   +--ro sub-domain-id?          uint8
    |   +--ro bitstring-length?       uint16
    |   +--ro bfir-id?                uint16
+--ro tunnel-role?                    enumeration
+--ro upstream-vpn-label?
|   rt-types:mpls-label {mvpn-aggregation-tunnel}?
+--ro mvpn-pmsi-ipv4-ref-sg-entries
    +--ro mvpn-pmsi-ipv4-ref-sg-entries*
        [ipv4-source-address ipv4-group-address]
        +--ro ipv4-source-address    inet:ipv4-address
        +--ro ipv4-group-address
            rt-types:ipv4-multicast-group-address
augment /ni:network-instances/ni:network-instance/ni:ni-type
    /l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6:
+--rw multicast
    +--rw signaling-mode?             enumeration
    +--rw auto-discovery-mode?        enumeration
    +--rw mvpn-type?                  enumeration
    +--rw is-sender-site?              boolean {mvpn-sender}?
    +--rw rpt-spt-mode?               enumeration
    +--rw ecmp-load-balance-mode?
    |   enumeration {mvpn-ecmp-load-balance}?
+--rw mvpn-route-targets {mvpn-separate-rt}?
|   +--rw mvpn-route-target* [mvpn-rt-type mvpn-rt-value]
|   |   +--rw mvpn-rt-type            enumeration
|   |   +--rw mvpn-rt-value           string
+--rw mvpn-ipmsi-tunnel-ipv6

```

```

| +--rw tunnel-type?                               p-tunnel
| +--rw (ipmsi-tunnel-attribute)?
| | +--:(rsvp-te-p2mp)
| | | +--rw rsvp-te-p2mp-template?                string
| | +--:(mldp-p2mp)
| | +--:(pim-ssm)
| | | +--rw ssm-default-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(pim-sm)
| | | +--rw sm-default-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(bidir-pim)
| | | +--rw bidir-default-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(ingress-replication)
| | +--:(mldp-mp2mp)
| | +--:(bier)
| | | +--rw inclusive-sub-domain-id?                uint8
| | | +--rw inclusive-bitstring-length?            uint16
| +--ro (pmsi-tunnel-state-attribute)?
| | +--:(rsvp-te-p2mp)
| | | +--ro p2mp-id?                                uint16
| | | +--ro tunnel-id?                              uint16
| | | +--ro extend-tunnel-id?                       uint16
| | +--:(mldp-p2mp)
| | | +--ro mldp-root-addr?                          inet:ip-address
| | | +--ro mldp-lsp-id?                            string
| | +--:(pim-ssm)
| | | +--ro ssm-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(pim-sm)
| | | +--ro sm-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(bidir-pim)
| | | +--ro bidir-group-addr?
| | | | rt-types:ip-multicast-group-address
| | +--:(ingress-replication)
| | +--:(mldp-mp2mp)
| | +--:(bier)
| | | +--ro sub-domain-id?                          uint8
| | | +--ro bitstring-length?                      uint16
| | | +--ro bfir-id?                                uint16

```

```

|   +--ro tunnel-role?                               enumeration
|   +--ro upstream-vpn-label?
|   |       rt-types:mpls-label {mvpn-aggregation-tunnel}?
|   +--ro mvpn-pmsi-ipv6-ref-sg-entries
|   |   +--ro mvpn-pmsi-ipv6-ref-sg-entries*
|   |   |       [ipv6-source-address ipv6-group-address]
|   |   +--ro ipv6-source-address    inet:ipv6-address
|   |   +--ro ipv6-group-address
|   |   |       rt-types:ipv6-multicast-group-address
+--rw mvpn-spmsi-tunnels-ipv6
  +--rw switch-delay-time?                uint8
  +--rw switch-back-holddown-time?        uint16
  +--rw tunnel-limit?                     uint16
  +--rw mvpn-spmsi-tunnel-ipv6* [tunnel-type]
    +--rw tunnel-type                                p-tunnel
    +--rw (spmsi-tunnel-attribute)?
      | +--:(rsvp-te-p2mp)
      | | +--rw rsvp-te-p2mp-template?              string
      | +--:(p2mp-mldp)
      | +--:(pim-ssm)
      | | +--rw ssm-group-pool-addr?
      | | |       rt-types:ip-multicast-group-address
      | | +--rw ssm-group-pool-masklength?          uint8
      | +--:(pim-sm)
      | | +--rw sm-group-pool-addr?
      | | |       rt-types:ip-multicast-group-address
      | | +--rw sm-group-pool-masklength?          uint8
      | +--:(bidir-pim)
      | | +--rw bidir-group-pool-addr?
      | | |       rt-types:ip-multicast-group-address
      | | +--rw bidir-group-pool-masklength?        uint8
      | +--:(ingress-replication)
      | +--:(mldp-mp2mp)
      | +--:(bier)
      | +--rw selective-sub-domain-id?              uint8
      | +--rw selective-bitstring-length?           uint16
  +--rw switch-threshold?                   uint32
  +--rw per-item-tunnel-limit?              uint16
  +--rw switch-wildcard-mode?
  |   enumeration {mvpn-switch-wildcard}?
  +--rw explicit-tracking-mode?
  |   enumeration {mvpn-explicit-tracking}?

```

```

+--rw (address-mask-or-acl)?
| +--:(address-mask)
| | +--rw ipv6-group-addr?
| | | rt-types:ipv6-multicast-group-address
| | +--rw ipv6-groupmasklength?          uint8
| | +--rw ipv6-source-addr?
| | | inet:ipv6-address
| | +--rw ipv6-source-masklength?        uint8
| +--:(acl-name)
|   +--rw group-acl-ipv6?
|   -> /acl:acls/acl/name
+--ro (pmsi-tunnel-state-attribute)?
| +--:(rsvp-te-p2mp)
| | +--ro p2mp-id?                      uint16
| | +--ro tunnel-id?                    uint16
| | +--ro extend-tunnel-id?             uint16
| +--:(mldp-p2mp)
| | +--ro mldp-root-addr?               inet:ip-address
| | +--ro mldp-lsp-id?                  string
| +--:(pim-ssm)
| | +--ro ssm-group-addr?
| | | rt-types:ip-multicast-group-address
| +--:(pim-sm)
| | +--ro sm-group-addr?
| | | rt-types:ip-multicast-group-address
| +--:(bidir-pim)
| | +--ro bidir-group-addr?
| | | rt-types:ip-multicast-group-address
| +--:(ingress-replication)
| +--:(mldp-mp2mp)
| +--:(bier)
|   +--ro sub-domain-id?                uint8
|   +--ro bitstring-length?             uint16
|   +--ro bfir-id?                      uint16
+--ro tunnel-role?                      enumeration
+--ro upstream-vpn-label?
|   rt-types:mpls-label {mvpn-aggregation-tunnel}?
+--ro mvpn-pmsi-ipv6-ref-sg-entries
  +--ro mvpn-pmsi-ipv6-ref-sg-entries*
    [ipv6-source-address ipv6-group-address]
    +--ro ipv6-source-address    inet:ipv6-address

```

```
+--ro ipv6-group-address
   rt-types:ipv6-multicast-group-address
```

4. MVPN YANG Modules

```
<CODE BEGINS> file ietf-mvpn@2019-12-02.yang
module ietf-mvpn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mvpn";
  prefix mvpn;

  import ietf-network-instance {
    prefix ni;
  }

  import ietf-bgp-l3vpn {
    prefix l3vpn;
  }

  import ietf-inet-types {
    prefix inet;
  }

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

  import ietf-access-control-list {
    prefix acl;
  }

  organization
    "IETF BESS(BGP Enabled Services) Working Group";
  contact
    "
    Yisong Liu
    <mailto:liuyisong.ietf@gmail.com>
    Stephane Litkowski
    <mailto:slitkows@cisco.com>
    Feng Guo
    <mailto:guofeng@huawei.com>
    Xufeng Liu
```

```
<mailto:xufeng.liu.ietf@gmail.com>
Robert Kebler
<mailto:rkebler@juniper.net>
Mahesh Sivakumar
<mailto:sivakumar.mahesh@gmail.com>";
description
  "This YANG module defines the generic configuration
  and operational state data for mvpn, which is common across
  all of the vendor implementations of the protocol. It is
  intended that the module will be extended by vendors to
  define vendor-specific mvpn parameters.";

revision 2019-12-02 {
  description
    "Update the contact information of co-authors.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2019-03-05 {
  description
    "Add bier as a type of P-Tunnel and Errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2018-11-08 {
  description
    "Update for leaf type and reference.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2018-05-10 {
  description
    "Update for Model structure and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2017-09-15 {
  description
    "Update for NMDA version and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
```

```
revision 2017-07-03 {
  description
    "Update S-PMSI configuration and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2016-10-28 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
/* Features */
feature mvpn-sender {
  description
    "Support configuration to specify the current PE as the
    sender PE";
}
feature mvpn-separate-rt {
  description
    "Support route-targets configuration of MVPN when they are
    different from the route-targets of unicast L3VPN.";
}
feature mvpn-switch-wildcard {
  description
    "Support configuration to use wildcard mode when multicast
    packets switch from I-PMSI to S-PMSI.";
}
feature mvpn-explicit-tracking {
  description
    "Support configuration to use explicit tracking for leaf PEs
    when multicast packets forward by I-PMSI or S-PMSI.";
}
feature mvpn-aggregation-tunnel {
  description
    "Support more than one VPN multicast service to use the same
    p-tunnel.";
}
feature mvpn-ecmp-load-balance {
  description
    "Support multicast entries in the private network to be
```

```
distributed on the ECMP path of bier in the public
network.";
}
```

```
typedef p-tunnel {
  type enumeration {
    enum no-tunnel-present {
      value 0;
      description "No tunnel information present";
    }
    enum rsvp-te-p2mp {
      value 1;
      description "RSVP TE P2MP tunnel";
    }
    enum mldp-p2mp {
      value 2;
      description "MLDP P2MP tunnel";
    }
    enum pim-ssm {
      value 3;
      description "PIM SSM tree in public net";
    }
    enum pim-sm {
      value 4;
      description "PIM SM tree in public net";
    }
    enum bidir-pim {
      value 5;
      description "BIDIR-PIM tree in public net";
    }
    enum ingress-replication {
      value 6;
      description "Ingress Replication p2p tunnel.";
    }
    enum mldp-mp2mp {
      value 7;
      description "MLDP MP2MP tunnel";
    }
    enum bier {
      value 11;
      description "bier underlay";
    }
  }
}
```

```
    }
    description "Provider tunnel type definition.";
  }

  grouping mvpn-instance-config {
    description "Mvpn basic configuration per instance.";

    leaf signaling-mode {
      type enumeration {
        enum none {
          value 0;
          description "No signaling";
        }
        enum bgp {
          value 1;
          description "bgp signaling";
        }
        enum pim {
          value 2;
          description "pim signaling";
        }
        enum mldp-in-band {
          value 3;
          description "mldp in-band signaling";
        }
      }
      default "none";
      description "Signaling mode for C-multicast route.";
    }

    leaf auto-discovery-mode {
      type enumeration {
        enum none {
          value 0;
          description "no auto-discovery signaling";
        }
        enum pim {
          value 1;
          description "auto-discovery by PIM signaling";
        }
        enum bgp {
          value 2;
          description "auto-discovery by BGP signaling";
        }
      }
    }
  }
}
```

```
    }
  }
  default "none";
  description "Auto discovery mode of MVPN PE members.";
}
leaf mvpn-type {
  type enumeration {
    enum rosen-mvpn {
      value 0;
      description "Rosen mvpn mode referenced RFC6037";
    }
    enum ng-mvpn {
      value 1;
      description
        "BGP/MPLS mvpn mode referenced RFC6513&RFC6514";
    }
  }
  default "ng-mvpn";
  description
    "Mvpn type, which can be rosen mvpn mode or ng mvpn mode.";
}
leaf is-sender-site {
  if-feature mvpn-sender;
  type boolean;
  default false;
  description "Configure the current PE as a sender PE.";
}
leaf rpt-spt-mode {
  type enumeration {
    enum spt-only {
      value 0;
      description
        "Only spt entries can cross the public net.";
    }
    enum rpt-spt {
      value 1;
      description
        "Both rpt and spt entries can corss the public net.";
    }
  }
  description
    "ASM mode in multicast private network for crossing
```

```
        public net.";
    }
    leaf ecmp-load-balance-mode {
        if-feature mvpn-ecmp-load-balance;
        type enumeration {
            enum none {
                value 0;
                description
                    "No load balancing for multicast entries.";
            }
            enum source {
                value 1;
                description
                    "Load balancing based on multicast source address.";
            }
            enum group {
                value 2;
                description
                    "Load balancing based on multicast group address.";
            }
            enum source-group {
                value 3;
                description
                    "Load balancing based on multicast source and group
                    address.";
            }
        }
        description
            "Distribution mode of multicast entries in the private
            network on the ECMP path of bier in the public network.";
    }
}/* mvpn-instance-config */

grouping mvpn-rt {
    description
        "May be different from l3vpn unicast route-targets.";
    container mvpn-route-targets {
        if-feature mvpn-separate-rt;
        description "Multicast vpn route-targets";
        list mvpn-route-target {
            key "mvpn-rt-type mvpn-rt-value" ;
            description

```

```
    "List of multicast route-targets" ;
  leaf mvpn-rt-type {
    type enumeration {
      enum export-extcommunity {
        value 0;
        description "export-extcommunity";
      }
      enum import-extcommunity {
        value 1;
        description "import-extcommunity";
      }
    }
    description
      "rt types are as follows:
      export-extcommunity: specifies the value of
      the extended community attribute of the
      route from an outbound interface to the
      destination vpn.
      import-extcommunity: receives routes that
      carry the specified extended community
      attribute";
  }
  leaf mvpn-rt-value {
    type string {
      length "3..21";
    }
    description
      "the available mvpn target formats are as
      follows:
      - 16-bit as number:32-bit user-defined
      number, for example, 1:3. an as number
      ranges from 0 to 65535, and a user-defined
      number ranges from 0 to 4294967295. The as
      number and user-defined number cannot be
      both 0s. That is, a vpn target cannot be 0:0.
      - 32-bit ip address:16-bit user-defined
      number, for example, 192.168.122.15:1.
      The ip address ranges from 0.0.0.0 to
      255.255.255.255, and the user-defined
      number ranges from 0 to 65535.";
  }
}
```

```
    }  
  }  
  
  grouping mvpn-ipmsi-tunnel-config {  
    description  
      "Configuration of default mdt for rosen mvpn  
      and I-PMSI for ng mvpn";  
    leaf tunnel-type {  
      type p-tunnel;  
      description "I-PMSI tunnel type.";  
    }  
    choice ipmsi-tunnel-attribute {  
      description "I-PMSI tunnel attributes configuration";  
      case rsvp-te-p2mp {  
        description "RSVP TE P2MP tunnel";  
        leaf rsvp-te-p2mp-template {  
          type string {  
            length "1..31";  
          }  
          description "RSVP TE P2MP tunnel template";  
        }  
      }  
      case mldp-p2mp {  
        description "MLDP P2MP tunnel";  
      }  
      case pim-ssm {  
        description "PIM SSM tree in the public net";  
        leaf ssm-default-group-addr {  
          type rt-types:ip-multicast-group-address;  
          description  
            "Default mdt or I-PMSI group address for SSM mode.";  
        }  
      }  
      case pim-sm {  
        description "PIM SM tree in the public net";  
        leaf sm-default-group-addr {  
          type rt-types:ip-multicast-group-address;  
          description  
            "Default mdt or I-PMSI group address for SM mode.";  
        }  
      }  
      case bidir-pim {
```

```
    description "BIDIR PIM tree in the public net";
    leaf bidir-default-group-addr {
      type rt-types:ip-multicast-group-address;
      description
        "Default mdt or I-PMSI group address for BIDIR mode.";
    }
  }
  case ingress-replication {
    description "Ingress replication p2p tunnel";
  }
  case mldp-mp2mp {
    description "MLDP MP2MP tunnel";
  }
  case bier {
    description "bier underlay";
    leaf inclusive-sub-domain-id {
      type uint8;
      description "Subdomain ID of bier.";
    }
    leaf inclusive-bitstring-length {
      type uint16 {
        range "64|128|256|512|1024|2048|4096";
      }
      description "BitString length of bier underlay.";
    }
  }
}
}/* mvpn-ipmsi-tunnel-config */

grouping mvpn-spmsi-tunnel-per-item-config {
  description "S-PMSI tunnel basic configuration";
  leaf tunnel-type {
    type p-tunnel;
    description "S-PMSI tunnel type.";
  }
  choice spmsi-tunnel-attribute {
    description "S-PMSI tunnel attributes configuration";
    case rsvp-te-p2mp {
      description "RSVP TE P2MP tunnel";
      leaf rsvp-te-p2mp-template {
        type string {
          length "1..31";
        }
      }
    }
  }
}
```

```
    }
    description "RSVP TE P2MP tunnel template";
  }
}
case p2mp-mldp {
  description "MLDP P2MP tunnel";
}
case pim-ssm {
  description "PIM SSM tree in the public net";
  leaf ssm-group-pool-addr {
    type rt-types:ip-multicast-group-address;
    description
      "Group pool address for data mdt or s-pmsi in SSM
mode";
  }
  leaf ssm-group-pool-masklength {
    type uint8 {
      range "8..128";
    }
    description
      "Group pool mask length for data mdt or s-pmsi in
      SSM mode";
  }
}
case pim-sm {
  description "PIM SM tree in the public net";
  leaf sm-group-pool-addr {
    type rt-types:ip-multicast-group-address;
    description
      "Group pool address for data mdt or s-pmsi in SM mode";
  }
  leaf sm-group-pool-masklength {
    type uint8 {
      range "8..128";
    }
    description
      "Group pool mask length for data mdt or s-pmsi in
      SM mode";
  }
}
case bidir-pim {
  description "BIDIR PIM tree in the public net";
```

```
    leaf bidir-group-pool-addr {
      type rt-types:ip-multicast-group-address;
      description
        "Group pool address for data mdt or s-pmsi in
        BIDIR mode";
    }
    leaf bidir-group-pool-masklength {
      type uint8 {
        range "8..128";
      }
      description
        "Group pool mask length for data mdt or s-pmsi in
        BIDIR mode";
    }
  }
}
case ingress-replication {
  description "Ingress replication p2p tunnel";
}
case mldp-mp2mp {
  description "MLDP MP2MP tunnel";
}
case bier {
  description "bier underlay";
  leaf selective-sub-domain-id {
    type uint8;
    description "Subdomain ID of bier.";
  }
  leaf selective-bitstring-length {
    type uint16 {
      range "64|128|256|512|1024|2048|4096";
    }
    description "BitString length of bier underlay.";
  }
}
}
leaf switch-threshold {
  type uint32 {
    range "0..4194304";
  }
  units kbps;
  default 0;
  description
```

```
    "Multicast packet rate threshold for
    triggering the switching from the
    I-PMSI to the S-PMSI. The value is
    an integer ranging from 0 to 4194304, in
    kbps. The default value is 0.";
  }
  leaf per-item-tunnel-limit {
    type uint16 {
      range "1..1024";
    }
    description
      "Maximum number of S-PMSI tunnels allowed
      per S-PMSI configuration item per mvpn instance.";
  }
  leaf switch-wildcard-mode {
    if-feature mvpn-switch-wildcard;
    type enumeration {
      enum source-group {
        value 0;
        description
          "Wildcard neither for source or group address.";
      }
      enum star-star {
        value 1;
        description
          "Wildcard for both source and group address.";
      }
      enum star-group {
        value 2;
        description
          "Wildcard only for source address.";
      }
      enum source-star {
        value 3;
        description
          "Wildcard only for group address.";
      }
    }
    description
      "I-PMSI switching to S-PMSI mode for private net
      wildcard mode, which including (*,*), (*,G), (S,*),
      (S,G) four modes.";
```

```
    }
    leaf explicit-tracking-mode {
      if-feature mvpn-explicit-tracking;
      type enumeration {
        enum no-leaf-info-required {
          value 0;
          description "No need to track leaf information.";
        }
        enum leaf-info-required {
          value 1;
          description "Need to track leaf information.";
        }
        enum leaf-info-required-per-flow {
          value 2;
          description
            "Need to track leaf information based on
             per multicast flow.";
        }
      }
      description "Tracking mode for leaf information.";
    }
  }/* mvpn-spmsi-tunnel-per-item-config */

grouping mvpn-spmsi-tunnel-common-config {
  description
    "Data mdt for rosen mvpn or S-PMSI for ng mvpn configuration
     attributes for both IPv4 and IPv6 private network";
  leaf switch-delay-time {
    type uint8 {
      range "3..60";
    }
    units seconds;
    default 5;
    description
      "Delay for switching from the I-PMSI to
       the S-PMSI. The value is an integer
       ranging from 3 to 60, in seconds. ";
  }
  leaf switch-back-holddown-time {
    type uint16 {
      range "0..512";
    }
  }
}
```

```
        units seconds;
        default 60;
        description
            "Delay for switching back from the S-PMSI
            to the I-PMSI. The value is an integer
            ranging from 0 to 512, in seconds. ";
    }
    leaf tunnel-limit {
        type uint16 {
            range "1..8192";
        }
        description
            "Maximum number of s-pmsi tunnels allowed
            per mvpn instance.";
    }
}/* mvpn-spmsi-tunnel-common-config */

grouping mvpn-pmsi-state {
    description "PMSI tunnel operational state information";

    choice pmsi-tunnel-state-attribute {
        config false;
        description
            "PMSI tunnel operational state information for each type";
        case rsvp-te-p2mp {
            description "RSVP TE P2MP tunnel";
            leaf p2mp-id {
                type uint16 {
                    range "0..65535";
                }
                description "P2MP ID of the RSVP TE P2MP tunnel";
            }
            leaf tunnel-id {
                type uint16 {
                    range "1..65535";
                }
                description "Tunnel ID of the RSVP TE P2MP tunnel";
            }
            leaf extend-tunnel-id {
                type uint16 {
                    range "1..65535";
                }
            }
        }
    }
}
```

```
        description
          "Extended tunnel ID of the RSVP TE P2MP Tunnel";
      }
  }
  case mldp-p2mp {
    description "MLDP P2MP tunnel";
    leaf mldp-root-addr {
      type inet:ip-address;
      description "IP address of the root of a MLDP P2MP lsp.";
    }
    leaf mldp-lsp-id {
      type string {
        length "1..256";
      }
      description "MLDP P2MP lsp ID.";
    }
  }
}
case pim-ssm {
  description "PIM SSM tree in the public net";
  leaf ssm-group-addr {
    type rt-types:ip-multicast-group-address;
    description "Group address for pim ssm";
  }
}
case pim-sm {
  description "PIM SM tree in the public net";
  leaf sm-group-addr {
    type rt-types:ip-multicast-group-address;
    description "Group address for pim sm";
  }
}
case bidir-pim {
  description "BIDIR PIM tree in the public net";
  leaf bidir-group-addr {
    type rt-types:ip-multicast-group-address;
    description "Group address for bidir-pim";
  }
}
case ingress-replication {
  description "Ingress replication p2p tunnel";
}
case mldp-mp2mp {
```

```
    description "MLDP MP2MP tunnel";
  }
  case bier {
    description "bier underlay";
    leaf sub-domain-id {
      type uint8;
      description "Subdomain ID of bier.";
    }
    leaf bitstring-length {
      type uint16 {
        range "64|128|256|512|1024|2048|4096";
      }
      description "BitString length of bier underlay.";
    }
    leaf bfir-id {
      type uint16;
      description "ID of BIER sender PE of MVPN.";
    }
  }
}
leaf tunnel-role {
  type enumeration {
    enum none {
      value 0;
      description "none";
    }
    enum root {
      value 1;
      description "root";
    }
    enum leaf {
      value 2;
      description "leaf";
    }
    enum root-and-leaf {
      value 3;
      description "root-and-leaf";
    }
  }
}
config false;
description "Role of a node for a p-tunnel.";
}
```

```
    leaf upstream-vpn-label {
      if-feature mvpn-aggregation-tunnel;
      type rt-types:mpls-label;
      config false;
      description
        "VPN context label for the multicast data of the VPN instance
         in an aggregation P-tunnel.";
    }
  }/* mvpn-pmsi-state */

  grouping mvpn-pmsi-ipv4-entry {
    description
      "Multicast entries in ipv4 mvpn referenced the pmsi tunnel";
    container mvpn-pmsi-ipv4-ref-sg-entries {
      config false;
      description
        "Multicast entries in ipv4 mvpn referenced the pmsi
tunnel";
      list mvpn-pmsi-ipv4-ref-sg-entries {
        key "ipv4-source-address ipv4-group-address";
        description
          "IPv4 source and group address of private network entry";
        leaf ipv4-source-address {
          type inet:ipv4-address;
          description
            "IPv4 source address of private network entry
             in I-PMSI or S-PMSI.";
        }
        leaf ipv4-group-address {
          type rt-types:ipv4-multicast-group-address;
          description
            "IPv4 group address of private network entry
             in I-PMSI or S-PMSI.";
        }
      }
    }
  }
}/* mvpn-pmsi-ipv4-entry */

  grouping mvpn-pmsi-ipv6-entry {
    description
      "Multicast entries in ipv6 mvpn referenced the pmsi tunnel";
```

```

    container mvpn-pmsi-ipv6-ref-sg-entries {
        config false;
        description
            "Multicast entries in ipv6 mvpn referenced the pmsi
tunnel";
        list mvpn-pmsi-ipv6-ref-sg-entries {
            key "ipv6-source-address ipv6-group-address";
            description
                "IPv6 source and group address of private network entry";
            leaf ipv6-source-address {
                type inet:ipv6-address;
                description
                    "IPv6 source address of private network entry
                    in I-PMSI or S-PMSI.";
            }
            leaf ipv6-group-address {
                type rt-types:ipv6-multicast-group-address;
                description
                    "IPv6 group address of private network entry
                    in I-PMSI or S-PMSI.";
            }
        }
    }
}/* mvpn-pmsi-ipv6-entry */

grouping mvpn-ipmsi-tunnel-info-ipv4 {
    description
        "Default mdt or I-PMSI configuration and
        operational state information";
    container mvpn-ipmsi-tunnel-ipv4 {
        description
            "Default mdt or I-PMSI configuration and
            operational state information";
        uses mvpn-ipmsi-tunnel-config;
        uses mvpn-pmsi-state;
        uses mvpn-pmsi-ipv4-entry;
    }
}

grouping mvpn-ipmsi-tunnel-info-ipv6 {
    description
        "Default mdt or I-PMSI configuration and

```

```
        operational state information";
    container mvpn-ipmsi-tunnel-ipv6 {
        description
            "Default mdt or I-PMSI configuration and
            operational state information";
        uses mvpn-ipmsi-tunnel-config;
        uses mvpn-pmsi-state;
        uses mvpn-pmsi-ipv6-entry;
    }
}

grouping mvpn-spmsi-tunnel-info-ipv4 {
    description
        "Data mdt for rosen mvpn or S-PMSI for ng mvpn in
        IPv4 private network";

    container mvpn-spmsi-tunnels-ipv4 {
        description
            "S-PMSI tunnel configuration and
            operational state information.";
        uses mvpn-spmsi-tunnel-common-config;

        list mvpn-spmsi-tunnel-ipv4 {
            key "tunnel-type";
            description
                "S-PMSI tunnel attributes configuration and
                operational state information.";

            uses mvpn-spmsi-tunnel-per-item-config;
            choice address-mask-or-acl {
                description
                    "Type of definition of private network
                    multicast address range";
                case address-mask {
                    description "Use the type of address and mask";
                    leaf ipv4-group-addr {
                        type rt-types:ipv4-multicast-group-address;
                        description
                            "Start address of the IPv4 group
                            address range in private net. ";
                    }
                    leaf ipv4-group-masklength {
```

```
        type uint8 {
            range "4..32";
        }
        description
            "Group mask length for the IPv4
            group address range in private net.";
    }
    leaf ipv4-source-addr {
        type inet:ipv4-address;
        description
            "Start address of the IPv4 source
            address range in private net.";
    }
    leaf ipv4-source-masklength {
        type uint8 {
            range "0..32";
        }
        description
            "Source mask length for the IPv4
            source address range in private net.";
    }
}
case acl-name {
    description "Use the type of acl";
    leaf group-acl-ipv4 {
        type leafref {
            path "/acl:acls/acl:acl/acl:name";
        }
        description
            "Specify the (s, g) entry on which the
            S-PMSI tunnel takes effect.
            The value is an integer ranging from 3000
            to 3999 or a string of 32 case-sensitive
            characters. If no value is specified, the
            switch-group address pool takes effect on
            all (s, g).";
    }
}
}
uses mvpn-pmsi-state;
uses mvpn-pmsi-ipv4-entry;
}/* list mvpn-spmsi-tunnel-ipv4 */
```

```
    }/* container mvpn-spmsi-tunnels-ipv4 */
  }/* grouping mvpn-spmsi-tunnel-info-ipv4 */
  grouping mvpn-spmsi-tunnel-info-ipv6 {
    description
      "Data mdt for rosen mvpn or S-PMSI for ng mvpn in
      IPv6 private network";

    container mvpn-spmsi-tunnels-ipv6 {
      description
        "S-PMSI tunnel configuration and
        operational state information.";
      uses mvpn-spmsi-tunnel-common-config;

      list mvpn-spmsi-tunnel-ipv6 {
        key "tunnel-type";
        description
          "S-PMSI tunnel attributes configuration and
          operational state information.";
        uses mvpn-spmsi-tunnel-per-item-config;

        choice address-mask-or-acl {
          description
            "Type of definition of private network
            multicast address range";
          case address-mask {
            description "Use the type of address and mask";

            leaf ipv6-group-addr {
              type rt-types:ipv6-multicast-group-address;
              description
                "Start address of the IPv6 group
                address range in private net. ";
            }
            leaf ipv6-groupmasklength {
              type uint8 {
                range "8..128";
              }
              description
                "Group mask length for the IPv6
                group address range in private net.";
            }
          }
          leaf ipv6-source-addr {
```

```
        type inet:ipv6-address;
        description
            "Start address of the IPv6 source
            address range in private net.";
    }
    leaf ipv6-source-masklength {
        type uint8 {
            range "0..128";
        }
        description
            "Source mask length for the IPv6
            source address range in private net.";
    }
}
case acl-name {
    description "Use the type of acl";
    leaf group-acl-ipv6 {
        type leafref {
            path "/acl:acls/acl:acl/acl:name";
        }
        description
            "Specify the (s, g) entry on which the
            S-PMSI tunnel takes effect.
            The value is an integer ranging from 3000
            to 3999 or a string of 32 case-sensitive
            characters. If no value is specified, the
            switch-group address pool takes effect on
            all (s, g).";
    }
}
}
uses mvpn-pmsi-state;
uses mvpn-pmsi-ipv6-entry;
}/* list mvpn-spmsi-tunnel-ipv6 */
}/* container mvpn-spmsi-tunnels-ipv6 */
}/* grouping mvpn-spmsi-tunnel-info-ipv6 */

augment "/ni:network-instances/ni:network-instance/ni:ni-type/"
    +"l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4" {
    description
        "Augment l3vpn ipv4 container for per multicast VRF
        configuration and operational state.";
```

```
    container multicast {
      description
        "Configuration and operational state of multicast IPv4 vpn
        specific parameters";
      uses mvpn-instance-config;
      uses mvpn-rtts;
      uses mvpn-ipmsi-tunnel-info-ipv4;
      uses mvpn-spmsi-tunnel-info-ipv4;
    }
  }

  augment "/ni:network-instances/ni:network-instance/ni:ni-type/"
    + "l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6" {
    description
      "Augment l3vpn ipv6 container for per multicast VRF
      configuration and operational state.";
    container multicast {
      description
        "Configuration and operational state of multicast IPv6 vpn
        specific parameters";
      uses mvpn-instance-config;
      uses mvpn-rtts;
      uses mvpn-ipmsi-tunnel-info-ipv6;
      uses mvpn-spmsi-tunnel-info-ipv6;
    }
  }
}
<CODE ENDS>
```

5. Security Considerations

TBD

6. IANA Considerations

TBD

[7. References](#)

[7.1. Normative References](#)

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010
- [RFC6037] Rosen, E., Cai, Y., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", [RFC 6037](#), October 2010.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", [RFC 6513](#), February 2012.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", [RFC 6514](#), February 2012.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), July 2013
- [RFC7246] IJ. Wijnands, P. Hitchen, N. Leymann, W. Henderickx, A. Gulko and J. Tantsura, " Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context ", [RFC 7246](#), June 2014.
- [RFC7900] Y. Rekhter, E. Rosen, R. Aggarwal, Arkatan, Y. Cai and T. Morin, " Extranet Multicast in BGP/IP MPLS VPNs ", [RFC 7900](#), June 2016.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), August 2016
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", [RFC 8279](#), November 2017
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", [RFC 8294](#), December 2017

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), March 2018
- [RFC8519] M. Jethanandani, S. Agarwal, L. Huang and D. Blair, "Yang Data Model for Network Access Control Lists (ACL) ", [RFC8519](#), March 2019
- [RFC8529] L. Berger, C. Hopps, A. Lindem, D. Bogdanovic and X. Liu, "YANG Data Model for Network Instances", [RFC8529](#), March 2019.
- [I-D.ietf-l3vpn-yang] D. Jain, K. Patel, P. Brissette, Z. Li, S. Zhuang, X. Liu, J. Haas, S. Esale and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", [draft-ietf-bess-l3vpn-yang-04](#)(expired), October 2018.

[7.2. Informative References](#)

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", [BCP 215](#), [RFC 8340](#), March 2018
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", [RFC8407](#), October 2018

[8. Acknowledgments](#)

The authors would like to thank the following for their valuable contributions of this document:

TBD

Authors' Addresses

Yisong Liu
Individual
China
Email: liuyisong.ietf@gmail.com

Feng Guo
Huawei Technologies
China
Email: guofeng@huawei.com

Stephane Litkowski
Cisco

Email: slitkows@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Robert Kebler
Juniper Networks
USA
Email: kebler@juniper.net

Mahesh Sivakumar
Juniper Networks
USA
Email: sivakumar.mahesh@gmail.com