

BESS Working Group
Internet Draft
Intended status: Standards Track
Expires: May 08, 2019

Y. Liu
F. Guo
Huawei Technologies
S. Litkowski
Orange
X. Liu
Volta Networks
R. Kebler
M. Sivakumar
Juniper Networks
November 08, 2018

Yang Data Model for Multicast in MPLS/BGP IP VPNs
draft-liu-bess-mvpn-yang-07

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/lid-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 May 8, 2019.

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	3
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	10
5. Security Considerations	28
6. IANA Considerations	28
7. References	28
7.1. Normative References	28
7.2. Informative References	29
8. Acknowledgments	29

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:

Multipoint Label Distribution Protocol [RFC6388].

P2MP TE:

Point to Multipoint Traffic Engineering [RFC4875].

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

Prefix	YANG module	Reference
ni	ietf-network-instance	[I-D.ietf-ni-model]
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	[I-D.ietf-acl-yang]

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 [I-D.ietf-netmod-rfc6087bis]. 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 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?           enumeration
      +--rw (ipmsi-tunnel-attribute)?
        +--:(p2mp-te)
          +--rw te-p2mp-template?  string
        +--:(p2mp-mldp)
        +--:(pim-ssm)
          +--rw ssm-default-group-addr?  rt-types:ip-multicast-gro
up-address
        +--:(pim-sm)
          +--rw sm-default-group-addr?  rt-types:ip-multicast-gro
up-address
        +--:(bidir-pim)
          +--rw bidir-default-group-addr?  rt-types:ip-multicast-gro
up-address
        +--:(ingress-replication)
        +--:(mp2mp-mldp)
      +--ro (pmsi-tunnel-state-attribute)?
        +--:(p2mp-te)
          +--ro te-p2mp-id?          uint16
          +--ro te-tunnel-id?        uint16
          +--ro te-extend-tunnel-id? uint16
        +--:(p2mp-mldp)
          +--ro mldp-root-addr?      inet:ip-address
          +--ro mldp-lsp-id?         string
        +--:(pim-ssm)
          +--ro ssm-group-addr?      rt-types:ip-multicast-gro
up-address
        +--:(pim-sm)
          +--ro sm-group-addr?       rt-types:ip-multicast-gro
up-address
        +--:(bidir-pim)
          +--ro bidir-group-addr?    rt-types:ip-multicast-gro
up-address
        +--:(ingress-replication)
        +--:(mp2mp-mldp)
      +--ro tunnel-role?            enumeration
      +--ro mvpn-pmsi-ipv4-ref-sg-entries

```

```

p-address] |      +---ro mvpn-pmsi-ipv4-ref-sg-entries* [ipv4-source-address ipv4-grou
|           +---ro ipv4-source-address      inet:ipv4-address
|           +---ro ipv4-group-address        rt-types:ipv4-multicast-group-addre
ss
+---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                        enumeration
+---rw (spmsi-tunnel-attribute)?
+---: (p2mp-te)
|   +---rw 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)
+---: (mp2mp-mldp)
+---rw switch-threshold?                   uint32
+---rw per-item-tunnel-limit?              uint16
+---rw switch-wildcard-mode?               enumeration {mvpn-switch-wil
dcard-mode}?
+---rw (address-mask-or-acl)?
+---: (address-mask)
|   +---rw ipv4-group-addr?                  rt-types:ipv4-multicas
t-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)?
+---: (p2mp-te)
|   +---ro te-p2mp-id?                      uint16
|   +---ro te-tunnel-id?                   uint16
|   +---ro te-extend-tunnel-id?            uint16
+---: (p2mp-mldp)
|   +---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)
    |   +---:(mp2mp-mldp)
+---ro tunnel-role?                                     enumeration
+---ro mvpn-pmsi-ipv4-ref-sg-entries
    +---ro mvpn-pmsi-ipv4-ref-sg-entries* [ipv4-source-address ipv4-g
roup-address]
    +---ro ipv4-source-address       inet:ipv4-address
    +---ro ipv4-group-address        rt-types:ipv4-multicast-group-ad
dress
augment /ni:network-instances/ni:network-instance/ni:ni-type/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 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
+---ro mvpn-ipmsi-tunnel-ipv6
    +---ro tunnel-type?              enumeration
    +---ro (ipmsi-tunnel-attribute)?
        +---:(p2mp-te)
        |   +---ro te-p2mp-template?    string
        +---:(p2mp-mldp)
        +---:(pim-ssm)
        |   +---ro ssm-default-group-addr?    rt-types:ip-multicast-gro
up-address
        +---:(pim-sm)
        |   +---ro sm-default-group-addr?    rt-types:ip-multicast-gro
up-address
        +---:(bidir-pim)
        |   +---ro bidir-default-group-addr?    rt-types:ip-multicast-gro
up-address
        +---:(ingress-replication)
        +---:(mp2mp-mldp)
    +---ro (pmsi-tunnel-state-attribute)?
        +---:(p2mp-te)
        |   +---ro te-p2mp-id?            uint16
        |   +---ro te-tunnel-id?         uint16
        |   +---ro te-extend-tunnel-id?   uint16
        +---:(p2mp-mldp)
        |   +---ro mldp-root-addr?        inet:ip-address
        |   +---ro mldp-lsp-id?          string
        +---:(pim-ssm)
        |   +---ro ssm-group-addr?        rt-types:ip-multicast-gro
up-address
        +---:(pim-sm)
        |   +---ro sm-group-addr?        rt-types:ip-multicast-gro
up-address
        +---:(bidir-pim)
        |   +---ro bidir-group-addr?      rt-types:ip-multicast-gro
up-address
        +---:(ingress-replication)
        +---:(mp2mp-mldp)
    +---ro tunnel-role?              enumeration

```



```

    |   +---ro mvpn-pmsi-ipv6-ref-sg-entries
    |   +---ro mvpn-pmsi-ipv6-ref-sg-entries* [ipv6-source-address ipv6-grou
p-address]
    |   +---ro ipv6-source-address      inet:ipv6-address
    |   +---ro ipv6-group-address       rt-types:ipv6-multicast-group-addre
ss
    +---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               enumeration
    |   +---rw (spmsi-tunnel-attribute)?
    |   |   +---: (p2mp-te)
    |   |   |   +---rw 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)
    |   |   +---: (mp2mp-mldp)
    |   +---rw switch-threshold?          uint32
    |   +---rw per-item-tunnel-limit?      uint16
    |   +---rw switch-wildcard-mode?       enumeration {mvpn-switch-wil
dcard-mode}?
    |   +---rw (address-mask-or-acl)?
    |   |   +---: (address-mask)
    |   |   |   +---rw ipv6-group-addr?      rt-types:ipv6-multicas
t-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)?
    |   |   +---: (p2mp-te)
    |   |   |   +---ro te-p2mp-id?           uint16
    |   |   |   +---ro te-tunnel-id?        uint16
    |   |   |   +---ro te-extend-tunnel-id? uint16
    |   |   +---: (p2mp-mldp)
    |   |   |   +---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)

```

```

group-address | | +--ro bidir-group-addr? rt-types:ip-multicast-
| | |
| | | +--:(ingress-replication)
| | | +--:(mp2mp-mldp)
+--ro tunnel-role? enumeration
+--ro mvpn-pmsi-ipv6-ref-sg-entries
+--ro mvpn-pmsi-ipv6-ref-sg-entries* [ipv6-source-address ipv6-g
roup-address]
+--ro ipv6-source-address inet:ipv6-address
+--ro ipv6-group-address rt-types:ipv6-multicast-group-ad
dress

```

4. MVPN YANG Modules

```

<CODE BEGINS> file ietf-mvpn@2018-11-08.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@huawei.com>
    Stephane Litkowski
    <mailto:stephane.litkowski@orange.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 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-mode {
    description
      "Support configuration to use wildcard mode when multicast
       packets switch from I-PMSI to S-PMSI.";
  }

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

    leaf signaling-mode {
      type enumeration {
        enum invalid {
          value "0";
          description "invalid";
        }
        enum bgp {
          value "1";
          description "bgp";
        }
        enum pim {
          value "2";
          description "pim";
        }
        enum mldp {
          value "3";
          description "mldp";
        }
      }
      default "invalid";
      description "Signaling mode for C-multicast route.";
    }
    leaf auto-discovery-mode {
      type enumeration {
        enum invalid {
          value "0";
          description "no auto-discovery";
        }
        enum pim {
          value "1";
          description "auto-discovery by PIM signaling";
        }
        enum bgp {
          value "2";
          description "auto-discovery by BGP signaling";
        }
      }
    }
  }
}
```

```
    }
    default "invalid";
    description "Auto discovery mode.";
  }
  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 mode for crossing public net.";
      }
      enum rpt-spt {
        value "1";
        description
          "Both rpt and spt mode for corssing public net.";
      }
    }
    description
      "ASM mode in multicast private net for crossing public net.";
  }
}/* 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 enumeration {
    enum no-tunnel {
      value "0";
      description "no tunnel information present";
    }
    enum p2mp-te {
      value "1";
      description "p2mp-te";
    }
    enum p2mp-mldp {
      value "2";
      description "p2mp-mldp";
    }
    enum pim-ssm {
      value "3";
      description "pim-ssm";
    }
    enum pim-sm {
      value "4";
      description "pim-sm";
    }
    enum bidir-pim {
      value "5";
      description "bidir-pim";
    }
    enum ingress-replication {
      value "6";
      description "ingress-replication";
    }
    enum mp2mp-mldp {
      value "7";
      description "mp2mp-mldp";
    }
  }
  description "I-PMSI tunnel type.";
}
choice ipmsi-tunnel-attribute {
  description "I-PMSI tunnel attributes configuration";
  case p2mp-te {
    description "P2mp TE tunnel";
    leaf te-p2mp-template {
      type string {
        length "1..31";
      }
      description "P2mp te tunnel template";
    }
  }
}
```

```
    }
  }
  case p2mp-mldp {
    description "Mldp tunnel";
  }
  case pim-ssm {
    description "Pim ssm tunnel";
    leaf ssm-default-group-addr {
      type rt-types:ip-multicast-group-address;
      description "Default mdt or I-PMSI group address.";
    }
  }
  case pim-sm {
    description "Pim sm tunnel";
    leaf sm-default-group-addr {
      type rt-types:ip-multicast-group-address;
      description "Default mdt or I-PMSI group address.";
    }
  }
  case bidir-pim {
    description "Bidir pim tunnel";
    leaf bidir-default-group-addr {
      type rt-types:ip-multicast-group-address;
      description "Default mdt or I-PMSI group address.";
    }
  }
  case ingress-replication {
    description "Ingress replication p2p tunnel";
  }
  case mp2mp-mldp {
    description "Mp2mp mldp tunnel";
  }
}
}/* mvpn-ipmsi-tunnel-config */

grouping mvpn-spmsi-tunnel-per-item-config {
  description "S-PMSI tunnel basic configuration";
  leaf tunnel-type {
    type enumeration {
      enum no-tunnel {
        value "0";
        description "no tunnel information present";
      }
      enum p2mp-te {
        value "1";
        description "p2mp-te";
      }
      enum p2mp-mldp {
        value "2";
      }
    }
  }
}
```



```
        description "p2mp-mldp";
    }
    enum pim-ssm {
        value "3";
        description "pim-ssm";
    }
    enum pim-sm {
        value "4";
        description "pim-sm";
    }
    enum bidir-pim {
        value "5";
        description "bidir-pim";
    }
    enum ingress-replication {
        value "6";
        description "ingress-replication";
    }
    enum mp2mp-mldp {
        value "7";
        description "mp2mp-mldp";
    }
}
description "S-PMSI tunnel type.";
}
choice spmsi-tunnel-attribute {
    description "S-PMSI tunnel attributes configuration";
    case p2mp-te {
        description "P2mp te tunnel";
        leaf te-p2mp-template {
            type string {
                length "1..31";
            }
            description "P2mp te tunnel template";
        }
    }
    case p2mp-mldp {
        description "Mldp tunnel";
    }
    case pim-ssm {
        description "Pim ssm tunnel";
        leaf ssm-group-pool-addr {
            type rt-types:ip-multicast-group-address;
            description "Group pool address for data mdt or pim s-pmsi.";
        }
        leaf ssm-group-pool-masklength {
            type uint8 {
                range "8..128";
            }
        }
    }
}
```

```
        description "Group pool mask for data mdt or pim s-pmsi";
    }
}
case pim-sm {
    description "Pim sm tunnel";
    leaf sm-group-pool-addr {
        type rt-types:ip-multicast-group-address;
        description "Group pool address for data mdt or pim s-pmsi.";
    }
    leaf sm-group-pool-masklength {
        type uint8 {
            range "8..128";
        }
        description "Group pool mask for data mdt or pim s-pmsi";
    }
}
case bidir-pim {
    description "Bidir pim tunnel";
    leaf bidir-group-pool-addr {
        type rt-types:ip-multicast-group-address;
        description "Group pool address for data mdt or pim s-pmsi.";
    }
    leaf bidir-group-pool-masklength {
        type uint8 {
            range "8..128";
        }
        description "Group pool mask for data mdt or pim s-pmsi";
    }
}
case ingress-replication {
    description "Ingress replication p2p tunnel";
}
case mp2mp-mldp {
    description "Mp2mp mldp tunnel";
}
}
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-mode;
  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.";
    }
  }
  default "source-group";
  description
    "I-PMSI switching to S-PMSI mode for private net
    wildcard mode, which including (*,*), (*,G), (S,*),
    (S,G) four modes.";
}
}/* 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 p2mp-te {
            description "P2mp te tunnel";
            leaf te-p2mp-id {
                type uint16 {
                    range "0..65535";
                }
                default "0";
                description "P2mp id of the p2mp tunnel.";
            }
            leaf te-tunnel-id {
                type uint16 {
                    range "1..65535";
                }
                description "Id of the p2mp tunnel.";
            }
        }
    }
}
```

```
    leaf te-extend-tunnel-id {
      type uint16 {
        range "1..65535";
      }
      description "P2mp extended tunnel interface id.";
    }
  }
  case p2mp-mldp {
    description "P2mp mldp tunnel";
    leaf mldp-root-addr {
      type inet:ip-address;
      description "Ip address of the root of a p2mp ldp lsp.";
    }
    leaf mldp-lsp-id {
      type string {
        length "1..256";
      }
      description "P2mp ldp lsp id.";
    }
  }
  case pim-ssm {
    description "Pim ssm tunnel";
    leaf ssm-group-addr {
      type rt-types:ip-multicast-group-address;
      description "Group address for pim ssm";
    }
  }
  case pim-sm {
    description "Pim sm tunnel";
    leaf sm-group-addr {
      type rt-types:ip-multicast-group-address;
      description "Group address for pim sm";
    }
  }
  case bidir-pim {
    description "Bidir pim tunnel";
    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 mp2mp-mldp {
    description "mp2mp mldp tunnel";
  }
}
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.";
}
}/* 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 {
    config false;
  }
}
```

```
        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 net 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 net 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 of multicast IPv4 vpn specific parameters and
            operational state of multicast IPv4 vpn specific parameters";
        uses mvpn-instance-config;
        uses mvpn-rt;
        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 of multicast IPv6 vpn specific parameters and
            operational state of multicast IPv6 vpn specific parameters";
        uses mvpn-instance-config;
        uses mvpn-rt;
        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
- [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
- [I-D.ietf-acl-yang] M. Jethanandani, L. Huang, S. Agarwal and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-19(work in progress), April 2018
- [I-D.ietf-ni-model] Berger, L., Hopps, C., Lindem, A., and D. Bogdanovic, X. Liu, "Network Instance Model", draft-ietf-rtgwg-ni-model-12(work in progress), March 2018.
- [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(work in progress), October 2018.

7.2. Informative References

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, March 2018
- [I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-20(work in progress), March 2018

8. Acknowledgments

The authors would like to thank Anish Peter, Stig Venaas for their valuable contributions.

Authors' Addresses

Yisong Liu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: liuyisong@huawei.com

Feng Guo
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: guofeng@huawei.com

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Robert Kebler
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

Email: rkebler@juniper.net

Maresh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA

Email: sivakumar.maresh@gmail.com

