

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 23, 2015

Y. Liu  
F. Guo  
Huawei  
M. Sivakumar  
Cisco  
March 22, 2015

YANG Data Model for PIM  
draft-liu-pim-yang-01

## Abstract

This document defines a YANG data model that can be used to configure and manage PIM.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on September 23, 2015.

## Copyright Notice

Copyright (c) 2015 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

Internet-Draft

PIM YANG Data Model

March 2015

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.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Design of Data Model . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Overview . . . . .	<a href="#">3</a>
<a href="#">3.2.</a>	PIM BSR Configuration . . . . .	<a href="#">5</a>
3.2.1.	Per-instance C-BSR Configuration of PIM BSR Instance	5
3.2.2.	Per-instance C-RP Configuration of PIM BSR Instance .	6
<a href="#">3.3.</a>	PIM BSR Operation State . . . . .	<a href="#">6</a>
<a href="#">3.4.</a>	PIM SM Configuration . . . . .	<a href="#">8</a>
<a href="#">3.4.1.</a>	Per-instance Parameters . . . . .	<a href="#">8</a>
3.4.2.	Per-static-RP configuration of the PIM SM instance .	9
3.4.3.	Per-SPT-switch configuration of the PIM SM instance .	10
3.4.4.	Per-Anycast-RP configuration of the PIM SM instance .	10
3.4.5.	Per-interface configuration of the PIM SM instance .	11
<a href="#">3.5.</a>	PIM SM Operation State . . . . .	<a href="#">12</a>
<a href="#">4.</a>	Notification . . . . .	<a href="#">18</a>
<a href="#">5.</a>	PIM Yang Module . . . . .	<a href="#">21</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">101</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">101</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">101</a>
<a href="#">9.</a>	References . . . . .	<a href="#">101</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">101</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">102</a>
	Authors' Addresses . . . . .	<a href="#">102</a>

## [1.](#) Introduction

YANG[RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF[RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces(e.g. REST) and 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 PIM. It includes PIM SM[RFC4601], PIM SSM[RFC4607][[RFC4608](#)], PIM BSR[RFC5059]. In addition, It can extend

PIM DM[RFC3973], BIDIR PIM[RFC5015] etc. features described in PIM standards other than mentioned above RFC in future version.

## [2.](#) Terminology

- o PIM: Protocol Independent Multicast
- o SM: Sparse Mode
- o SSM: Source-specific Multicast
- o DM: Dense Mode
- o BSR: Bootstrap Router
- o RP: Rendezvous Point
- o SPT: shortest-path tree
- o RPT: Rendezvous Point Tree
- o RPF: Reverse Path Forwarding

## [3.](#) Design of Data Model

### [3.1.](#) Overview

Because PIM SSM is a subset of PIM SM, it is not necessary to assign a separate container for PIM SSM. The PIM Yang module is divided into two main containers :

- o pim-bsr : that contains all pim bsr&rp writable configuration objects and operation states.
- o pim-sm : that contains all pim sm writable configuration objects and operation states.

The figure below describes the overall structure of the PIM Yang module :

```
module: pim
  +--rw pim
    +--rw pim-bsr
      |   +--rw pim-bsr-instances
      |   |   +--rw c-bsr-instance* [vrf-name address-family]
      |   |   |   +--...
      |   |   +--rw c-rp-instance* [vrf-name address-family]
      |   |   +--...
```

```
      |   +--rw pim-bsr-ebsr-infos
      |   |   +--ro pim-bsr-ebsr-info* [vrf-name address-family]
      |   |   +--...
      |   +--rw pim-bsr-cbsr-infos
      |   |   +--ro pim-bsr-cbsr-info* [vrf-name address-family]
      |   |   +--...
      |   +--rw pim-bsr-ebsr-rps
      |   |   +--ro pim-bsr-ebsr-rp* [vrf-name address-family]
      |   |   +--...
  +--rw pim-sm
    +--rw pim-sm-instances
      |   +--rw pim-sm-instance* [vrf-name address-family]
      |   +--...
      |   +--rw pim-static-rps
      |   |   +--rw pim-static-rp
      |   |   +--...
      |   +--rw pim-spt-switchs
      |   |   +--rw pim-spt-switch* [spt-threshhold]
      |   |   +--...
      |   +--rw pim-sm-anycast-rps
      |   |   +--rw pim-sm-anycast-rp
      |   |   +--...
      |   |   +--rw pim-sm-rp-peers
      |   |   |   +--rw pim-sm-rp-peer
      |   |   +--...
      |   +--rw pim-sm-interfaces
      |   |   +--rw pim-sm-interface* [vrf-name address-family if-name]
      |   |   +--...
    +--rw pim-sm-if-infos
```

```

|   +---ro pim-sm-if-info* [vrf-name address-family if-name]
|   +---...
+--rw pim-nbr-infos
|   +---ro pim-nbr-info* [vrf-name address-family if-name]
|   +---...
|   +---ro pim-second-addrs
|       +---ro pim-second-addr* [second-addr]
|       +---...
+--rw pim-routes
|   +---ro pim-route* [vrf-name address-family]
|   +---...
|   +---ro pim-route-downstreams
|       +---ro pim-route-downstream* [sg-ds-oif-name]
|       +---...
|       +---ro pim-route-downstream-vectors
|           +---ro pim-route-downstream-vector*
|           +---...
+--rw pim-rpf-routes
|   +---ro pim-rpf-route* [vrf-name address-family]

```

```

|   +---...
|   +---ro pim-claim-route-sgs
|       +---ro pim-claim-route-sg* [ipv4-source-addr ipv6-source-addr
pv4-group-addr ipv6-group-addr]
|       +---...
+--rw pim-embedded-rp-infos
|   +---ro pim-embedded-rp-info* [vrf-name]
|   +---...
+--rw pim-rp-group-infos
|   +---ro pim-rp-group-info* [vrf-name address-family]
|   +---...
+--rw pim-bfd-infos
|   +---ro pim-bfd-info* [vrf-name address-family if-name]
|   +---...
+--rw pim-vrf-packet-statistics
|   +---ro pim-vrf-packet-statistic* [vrf-name address-family]
|   +---...
|   +---ro pim-if-packet-statistics
|       +---ro pim-if-packet-statistic* [if-name]
|       +---...

```

Figure 1 The overview of PIM YANG data model

### [3.2.](#) PIM BSR Configuration

PIM BSR configuration container has only one main container:

- o `pim-bsr-instances` : that contains all `pim c-bsr` and `c-rp` writable configuration objects.

PIM BSR per-instance configuration container includes `c-bsr` and `c-rp` parameters of the public PIM instance or the PIM instance binding a specific VRF. PIM BSR per-instance configuration container is divided into:

- o Per-instance `c-bsr` configuration of the `pim-bsr` instance.
- o Per-instance `c-rp` configuration of the `pim-bsr` instance.

#### [3.2.1.](#) Per-instance C-BSR Configuration of PIM BSR Instance

The per-instance C-BSR configuration includes the name of the VRF bound by the `pim bsr` instance, and timer parameters , policies etc. it also includes administratively scoped BSR configuration.

```
|  +---rw pim-bsr-instances
|  |  +---rw c-bsr-instance* [vrf-name address-family]
|  |  |  +---rw vrf-name                string
|  |  |  +---rw address-family          enumeration
|  |  |  +---rw c-bsr-if-name            if-name
|  |  |  +---rw ipv4-c-bsr-if-addr?      inet:ipv4-address
|  |  |  +---rw ipv6-c-bsr-if-addr      inet:ipv6-address
|  |  |  +---rw c-bsr-holdtime?          uint32
|  |  |  +---rw c-bsr-interval?          uint32
|  |  |  +---rw c-bsr-hashlen?           uint32
|  |  |  +---rw c-bsr-priority?          uint8
|  |  |  +---rw c-bsr-ply-name?          string
|  |  |  +---rw c-bsr-ply-ipv6?          string
|  |  |  +---rw c-bsr-adminscope?        boolean
|  |  |  +---rw c-bsr-global-enable?     boolean
```

```

| | | +--rw c-bsr-global-hashlength?  uint32
| | | +--rw c-bsr-global-priority?    uint32
| | | +--rw is-fragable                boolean

```

Figure 2 The YANG data model of PIM C-BSR configuration

### 3.2.2. Per-instance C-RP Configuration of PIM BSR Instance

The per-instance C-RP configuration includes the name of the VRF bound by the pim bsr instance, specific interface or address for c-rp and timer parameters , policies etc.

```

| | +--rw c-rp-instance* [vrf-name address-family]
| |   +--rw vrf-name      string
| |   +--rw address-family enumeration
| |   +--rw c-rp-if-name  if-name
| |   +--rw c-rp-grpplly-name? string
| |   +--rw c-rp-priority? uint8
| |   +--rw c-rp-holdtime? uint16
| |   +--rw c-rp-adv-interval? uint16
| |   +--rw ipv4-c-rp-addr? inet:ipv4-address
| |   +--rw ipv6-c-rp-addr  inet:ipv6-address

```

Figure 3 The YANG data model of PIM C-RP configuration

### 3.3. PIM BSR Operation State

PIM BSR container provides operational states for BSR and RP information. This container is divided in three components:

- o pim-bsr-ebsr-infos : provide elected bsr state information.
- o pim-bsr-cbsr-infos : provide candidate bsr state information.

- o pim-bsr-ebsr-rps : provide rp and group mapping information from the elected bsr.

```

| +--rw pim-bsr-ebsr-infos
| |   +--ro pim-bsr-ebsr-info* [vrf-name address-family]
| |     +--ro vrf-name      string
| |     +--ro address-family enumeration
| |     +--ro ipv4-bsr-addr? inet:ipv4-address

```

```

| | +--ro ipv6-bsr-addr?      inet:ipv6-address
| | +--ro bsr-scope?         enumeration
| | +--ro ipv4-group-addr?   inet:ipv4-address
| | +--ro ipv6-group-addr?   inet:ipv6-address
| | +--ro ipv4-group-masklen? uint8
| | +--ro ipv6-group-masklen? uint8
| | +--ro priority?          uint8
| | +--ro hash-length?       uint8
| | +--ro fsm-state?         enumeration
| | +--ro up-time?           uint32
| | +--ro expire-time?       uint32
| | +--ro c-rp-count?        uint32

```

Figure 4 The YANG data model of PIM EBSR information

```

| +--rw pim-bsr-cbsr-infos
| | +--ro pim-bsr-cbsr-info* [vrf-name address-family]
| | | +--ro vrf-name          string
| | | +--ro address-family    enumeration
| | | +--ro ipv4-cbsr-addr?   inet:ipv4-address
| | | +--ro ipv6-cbsr-addr?   inet:ipv6-address
| | | +--ro bsr-scope?        enumeration
| | | +--ro ipv4-group-addr?   inet:ipv4-address
| | | +--ro ipv6-group-addr?   inet:ipv6-address
| | | +--ro ipv4-group-masklen? uint8
| | | +--ro ipv6-group-masklen? uint8
| | | +--ro priority?          uint8
| | | +--ro hash-length?       uint8
| | | +--ro fsm-state?         enumeration
| | | +--ro is-wait-to-bsr?    enumeration

```

Figure 5 The YANG data model of PIM CBSR information

```

| +--rw pim-bsr-ebsr-rps

```



```

|      +---ro pim-bsr-ebsr-rp* [vrf-name address-family]
|      |
|      |      +---ro vrf-name                string
|      |      +---ro address-family          enumeration
|      |      +---ro ipv4-rp-addr?          inet:ipv4-address
|      |      +---ro ipv6-rp-addr?          inet:ipv6-address
|      |      +---ro rp-addr-is-local?      boolean
|      |      +---ro ipv4-group-addr?       inet:ipv4-address
|      |      +---ro ipv6-group-addr?       inet:ipv6-address
|      |      +---ro ipv4-group-masklen?    uint8
|      |      +---ro ipv6-group-masklen?    uint8
|      |      +---ro priority?              uint8
|      |      +---ro up-time?               uint32
|      |      +---ro expiry-time?           uint32

```

Figure 6 The YANG data model of PIM RP information

### [3.4.](#) PIM SM Configuration

PIM SM configuration container has only one main container:

- o pim-sm-instances : that contains all pim sm writable configuration objects.

PIM SM per-instance configuration container includes pim sm protocol parameters of the public PIM instance or the PIM instance binding a specific VRF. PIM SM per-instance configuration container is divided into:

- o Per-instance parameters
- o Per-static-rp configuration of the pim-sm instance
- o Per-spt-switch configuration of the pim-sm instance
- o Per-anycast-rp configuration of the pim-sm instance
- o Per-interface configuration of the pim-sm instance

#### [3.4.1.](#) Per-instance Parameters

The per-instance parameter includes the name of the VRF bound by the PIM SM instance, and timer parameters such as hello interval etc., and varied policies etc

```

+--rw pim-sm-instances
|   +--rw pim-sm-instance* [vrf-name address-family]
|       +--rw vrf-name                string
|       +--rw address-family          enumeration
|       +--rw assert-holdtime?        uint16
|       +--rw jp-holdtime?            uint16
|       +--rw probe-interval?         uint16
|       +--rw jp-timer-interval?      uint16
|       +--rw dr-priority?            uint32
|       +--rw hello-holdtime?         uint16
|       +--rw hello-lan-delay?        uint16
|       +--rw hello-interval?         uint16
|       +--rw hello-override?         uint16
|       +--rw reg-checksum?           boolean
|       +--rw reg-suppress-interval?  uint16
|       +--rw embedded-rp?            boolean
|       +--rw nbr-check-recv?         boolean
|       +--rw nbr-check-send?         boolean
|       +--rw reg-ply-name?           string
|       +--rw reg-ply-ipv6?           string
|       +--rw ssm-ply-name?           string
|       +--rw ssm-ply-ipv6?           string
|       +--rw src-ply-name?           string
|       +--rw src-ply-ipv6?           string
|       +--rw bsr-ply-name?           string
|       +--rw bsr-ply-ipv6?           string
|       +--rw emb-rp-ply-name?        string
|       +--rw source-lifetime?        uint16
|       +--rw ipsec-name?             string
|       +--rw ipsec-type?             enumeration
|       +--rw uni-ipsec-name?         string

```

Figure 7 The YANG data model of PIM SM instance configuration

#### [3.4.2.](#) Per-static-RP configuration of the PIM SM instance

Per-static-RP configuration of the PIM SM instance includes static RP address, preference and policy for group range. For simply deployment, sometimes it is not necessary to deploy dynamic BSR/RP mechanism, and static RP mechanism is also satisfied.

```

|      +--rw pim-static-rps
|      |      +--rw pim-static-rp
|      |      |      +--rw ipv4-static-rp-addr      inet:ipv4-address
|      |      |      +--rw ipv6-static-rp-addr      inet:ipv6-address
|      |      |      +--rw static-rp-ply-name?      string
|      |      |      +--rw static-rp-ply-name-ipv6? string
|      |      |      +--rw preference?              enumeration

```

Figure 8 The YANG data mode of PIM static-RP configuration

#### 3.4.3. Per-SPT-switch configuration of the PIM SM instance

Per-SPT-switch configuration of the PIM SM instance includes spt switch threshold, group range by policy. PIM SM SPT switch process as described in [RFC4601] makes multicast flows from RPT to SPT in order to optimizing the forwarding paths.

```

|      +--rw pim-spt-switchs
|      |      +--rw pim-spt-switch* [spt-threshhold]
|      |      |      +--rw infinity?              boolean
|      |      |      +--rw spt-threshhold          uint32
|      |      |      +--rw is-acl-enable           boolean
|      |      |      +--rw spt-grp-ply-name?      string
|      |      |      +--rw spt-grp-ply-order?     uint32

```

Figure 9 The YANG data model of PIM SPT switch configuration

#### 3.4.4. Per-Anycast-RP configuration of the PIM SM instance

Anycast-RP as described in [RFC4610] is a mechanism that pim sm register packets have been used to exchange multicast source information and get fast convergence when a PIM Rendezvous Point (RP) router fails. Per-Anycast-RP configuration container is divided into :

- o Per-anycast-rp parameters
- o Per-rp-peer configuration of PIM SM Anycast RP

#### [3.4.4.1.](#) Per-Anycast-RP Parameters

Per-Anycast-RP parameter includes the RP for anycast address, and the local address to establish RP peer link.

```
|      +--rw pim-sm-anycast-rps
|      |      +--rw pim-sm-anycast-rp
|      |      |      +--rw ipv4-rp-address      inet:ipv4-address
|      |      |      +--rw ipv6-rp-address      inet:ipv6-address
|      |      |      +--rw local-ipv4-address   inet:ipv4-address
|      |      |      +--rw local-ipv6-address   inet:ipv6-address
```

Figure 10 The YANG data model of PIM SM anycast-RP configuration

#### [3.4.4.2.](#) Per-RP-peer Configuration of PIM SM Anycast RP

Per-RP-peer configuration of PIM SM anycast RP includes the peer address, source information forwarding policy etc.

```
|      |      +--rw pim-sm-rp-peers
|      |      |      +--rw pim-sm-rp-peer
|      |      |      |      +--rw ipv4-rp-peer-address   inet:ipv4-address
|      |      |      |      +--rw ipv6-rp-peer-address   inet:ipv6-address
|      |      |      |      +--rw fwd-sa-swt?            boolean
|      |      |      |      +--rw fwd-policy?            string
|      |      |      |      +--rw fwd-policy-ipv6?       string
```

Figure 11 The YANG data model of PIM SM anycast-RP peer configuration

#### [3.4.5.](#) Per-interface configuration of the PIM SM instance

Per-interface configuration of the PIM SM instance includes the interface name, and the VRF name bound by the interface, and time parameters, policies etc.

```
|      +---rw pim-sm-interfaces
|      |      +---rw pim-sm-interface* [vrf-name address-family if-name]
|      |      |      +---rw vrf-name                string
|      |      |      +---rw address-family          enumeration
|      |      |      +---rw if-name                if-name
|      |      |      +---rw pim-sm-enable          boolean
|      |      |      +---rw dr-priority?           uint32
|      |      |      +---rw hello-interval?        uint16
|      |      |      +---rw hello-holdtime?        uint16
|      |      |      +---rw hello-override?        uint16
|      |      |      +---rw hello-lan-delay?       uint16
|      |      |      +---rw jp-timer-interval?     uint16
|      |      |      +---rw jp-holdtime?           uint16
|      |      |      +---rw jp-ply-name?           string
|      |      |      +---rw jp-ply-ipv6?           string
|      |      |      +---rw jp-asm-ply-name?       string
|      |      |      +---rw jp-asm-ply-ipv6?       string
|      |      |      +---rw jp-ssm-ply-name?       string
|      |      |      +---rw jp-ssm-ply-ipv6?       string
|      |      |      +---rw nbr-ply-name?          string
|      |      |      +---rw nbr-ply-ipv6?          string
|      |      |      +---rw assert-holdtime?       uint16
|      |      |      +---rw require-gen-id?        boolean
|      |      |      +---rw pim-bsr-boundary?      enumeration
|      |      |      +---rw bfd-enable?            boolean
```

	+-rw bfd-min-tx?	uint16
	+-rw bfdminrx?	uint16
	+-rw bfd-multiplier?	uint8
	+-rw is-silent?	boolean
	+-rw is-dr-swt-delay?	boolean
	+-rw dr-swt-delay-interval?	uint16
	+-rw ipsec-name?	string
	+-rw ipsec-type?	enumeration

Figure 12 The YANG data model of PIM SM interface configuration

### 3.5. PIM SM Operation State

PIM SM container provides operational states for interface, neighbor, routing etc. information. This container is divided in multiple components:

- o pim-sm-if-infos : provide interface state and running parameters information for all of the PIM SM interfaces.
- o pim-nbr-infos : provide neighbor state information for all of the PIM SM neighbors.

- o pim-routes : provide routing state information for all of the PIM SM routing entries.
- o pim-rpf-routes : provide RPF routing state information for all of the PIM SM routing entries.
- o pim-embedded-rp-infos : provide embedded RP information for all of the PIM SM ipv6 groups.
- o pim-rp-group-infos : provide group RP mapping information for all of the PIM SM groups.
- o pim-bfd-infos : provide BFD session information for all of the BFD sessions created by PIM SM.
- o pim-vrf-packet-statistics : provide PIM SM packet statistical information for all of the PIM SM protocol packets.

```

+--rw pim-sm-if-infos
|   +--ro pim-sm-if-info* [vrf-name address-family if-name]
|       +--ro vrf-name          string
|       +--ro address-family    enumeration
|       +--ro if-name           if-name
|       +--ro if-up-state?      enumeration
|       +--ro neighbor-count?   uint32
|       +--ro dr-priority?      uint32
|       +--ro ipv4-dr-addr?     inet:ipv4-address
|       +--ro ipv6-dr-addr?     inet:ipv6-address
|       +--ro hello-interval?   uint16
|       +--ro hello-holdtime?   uint16
|       +--ro assert-holdtime?  uint16
|       +--ro jp-interval?      uint16
|       +--ro jp-holdtime?      uint16
|       +--ro lan-delay-neg?    uint16
|       +--ro hello-over-interval-neg?  uint16
|       +--ro gen-id?           uint32
|       +--ro non-dr-neighbor-num?  uint32
|       +--ro non-lan-delay-neighbor-num?  uint32
|       +--ro non-join-attribute-neighbor-num?  uint32
|       +--ro join-attribute?   boolean
|       +--ro ipsec-name?       string
|       +--ro ipsec-type?       enumeration

```

The Figure 13 The YANG data model of PIM SM interface information

```

+--rw pim-nbr-infos
|   +--ro pim-nbr-info* [vrf-name address-family if-name]
|       +--ro vrf-name          string
|       +--ro address-family    enumeration
|       +--ro if-name           if-name
|       +--ro ipv4-nbr-addr?    inet:ipv4-address
|       +--ro ipv6-nbr-addr?    inet:ipv6-address
|       +--ro up-time?          uint32
|       +--ro expire-time?      uint32
|       +--ro dr-priority-present?  boolean
|       +--ro dr-priority?      uint32

```

```

|      +---ro gen-id-present?          boolean
|      +---ro gen-id?                  uint32
|      +---ro holdtime?                uint16
|      +---ro lan-delay-present?       boolean
|      +---ro lan-delay?               uint16
|      +---ro override-interval?      uint16
|      +---ro join-attribute?          boolean
|      +---ro pim-second-addrs
|          +---ro pim-second-addr* [second-addr]
|              +---ro second-addr      inet:ipv6-address

```

The Figure 14 The YANG data model of PIM SM neighbor information

```

+---rw pim-routes
|   +---ro pim-route* [vrf-name address-family]
|       +---ro vrf-name                string
|       +---ro address-family          enumeration
|       +---ro ipv4-source-addr?       inet:ipv4-address
|       +---ro ipv6-source-addr?       inet:ipv6-address
|       +---ro ipv4-group-addr?        inet:ipv4-address
|       +---ro ipv6-group-addr?        inet:ipv6-address
|       +---ro sg-up-protocl-type?     enumeration
|       +---ro rp-mode-type?           enumeration
|       +---ro ipv4-rp-addr?           inet:ipv4-address
|       +---ro ipv6-rp-addr?           inet:ipv6-address
|       +---ro rp-addr-is-local?       boolean
|       +---ro sg-flag?                string
|       +---ro sg-up-time?             uint32
|       +---ro sg-iif-name?            string
|       +---ro sg-vector-rd?           string
|       +---ro ipv4-sg-vector-addr?    inet:ipv4-address
|       +---ro ipv6-sg-vector-addr?    inet:ipv6-address
|       +---ro sg-elect-vector-rd?     string
|       +---ro ipv4-sg-elect-vector-addr? inet:ipv4-address
|       +---ro ipv6-sg-elect-vector-addr? inet:ipv6-address
|       +---ro ipv4-sg-up-nbr-addr?    inet:ipv4-address
|       +---ro ipv6-sg-up-nbr-addr?    inet:ipv6-address

```

```

|      +---ro ipv4-sg-rpf-addr?        inet:ipv4-address
|      +---ro ipv6-sg-rpf-addr?        inet:ipv6-address
|      +---ro ipv4-sg-refer-addr?      inet:ipv4-address
|      +---ro ipv6-sg-refer-addr?      inet:ipv6-address

```



```

|      +---ro sg-rpf-masklen?                uint8
|      +---ro sg-rpf-metric?                 uint16
|      +---ro sg-rpf-metric-preference?      uint16
|      +---ro backup-sg-iif-name?            string
|      +---ro ipv4-backup-sg-up-nbr-addr?    inet:ipv4-address
|      +---ro ipv6-backup-sg-up-nbr-addr?    inet:ipv6-address
|      +---ro ipv4-backup-sg-rpf-addr?       inet:ipv4-address
|      +---ro ipv6-backup-sg-rpf-addr?       inet:ipv6-address
|      +---ro sg-up-state?                   enumeration
|      +---ro sg-up-exp-time?                uint32
|      +---ro sg-up-rp-tstate?               enumeration
|      +---ro sg-up-rpt-overtime?            uint32
|      +---ro sg-reg-state?                 enumeration
|      +---ro sg-reg-stop-time?             uint32
|      +---ro pim-route-downstreams
|          +---ro pim-route-downstream* [sg-ds-oif-name]
|              +---ro sg-ds-oif-name          string
|              +---ro sg-ds-protocol-type?    string
|              +---ro sg-ds-up-time?          uint32
|              +---ro sg-ds-expire-time?      uint32
|              +---ro sg-ds-dr-state?         boolean
|              +---ro sg-ds-state?           enumeration
|              +---ro sg-ds-ppt-time?        uint32
|              +---ro sg-ds-assert-state?    enumeration
|              +---ro sg-ds-assert-time?     uint32
|              +---ro ipv4-sg-ds-assert-win-addr? inet:ipv4-address
|              +---ro ipv6-sg-ds-assert-win-addr? inet:ipv6-address
|              +---ro sg-ds-assert-win-metric? uint16
|              +---ro sg-ds-ast-win-metric-pre? uint16
|              +---ro sg-ds-local-member?    boolean
|              +---ro sg-ds-rpt-state?       enumeration
|              +---ro sg-ds-rpt-ppt-time?    uint32
|              +---ro sg-ds-rpt-expire-time? uint32
|              +---ro sg-vector-num?         uint16
|              +---ro pim-route-downstream-vectors
|                  +---ro pim-route-downstream-vector*
|                      +---ro ipv4-neighbor-addr?    inet:ipv4-address
|                      +---ro ipv6-neighbor-addr?    inet:ipv6-address
|                      +---ro neighbor-vector-rd?    string
|                      +---ro ipv4-neighbor-vector-addr? inet:ipv4-address
|                      +---ro ipv6-neighbor-vector-addr? inet:ipv6-address
|                      +---ro neighbor-expire-time?  uint32

```

Figure 15 The YANG data model of PIM SM routing information

```

+--rw pim-rpf-routes
|   +--ro pim-rpf-route* [vrf-name address-family]
|       +--ro vrf-name          string
|       +--ro address-family    enumeration
|       +--ro ipv4-dest-addr?   inet:ipv4-address
|       +--ro ipv6-dest-addr?   inet:ipv6-address
|       +--ro if-name?          if-name
|       +--ro ipv4-rpf-addr?     inet:ipv4-address
|       +--ro ipv6-rpf-addr?     inet:ipv6-address
|       +--ro ipv4-refer-route-addr? inet:ipv4-address
|       +--ro ipv6-refer-route-addr? inet:ipv6-address
|       +--ro ipv4-refer-route-masklen? uint8
|       +--ro ipv6-refer-route-masklen? uint8
|       +--ro refer-route-type? enumeration
|       +--ro pim-claim-route-sgs
|           +--ro pim-claim-route-sg* [ipv4-source-addr ipv6-source-addr
pv4-group-addr ipv6-group-addr]
|               +--ro ipv4-source-addr    inet:ipv4-address
|               +--ro ipv6-source-addr    inet:ipv6-address
|               +--ro ipv4-group-addr     inet:ipv4-address
|               +--ro ipv6-group-addr     inet:ipv6-address

```

The Figure 16 The YANG data model of PIM SM RPF routing information

```

+--rw pim-embedded-rp-infos
|   +--ro pim-embedded-rp-info* [vrf-name]
|       +--ro vrf-name          string
|       +--ro ipv6-group-addr?   inet:ipv6-address
|       +--ro embedded-rp-addr?  inet:ipv6-address

```

The Figure 17 The YANG data model of PIM SM embedded RP information

```

+--rw pim-rp-group-infos
|   +--ro pim-rp-group-info* [vrf-name address-family]
|       +--ro vrf-name          string
|       +--ro address-family    enumeration
|       +--ro ipv4-group-addr?   inet:ipv4-address
|       +--ro ipv6-group-addr?   inet:ipv6-address
|       +--ro ipv4-bsr-rp-addr?   inet:ipv4-address
|       +--ro ipv6-bsr-rp-addr?   inet:ipv6-address
|       +--ro ipv4-static-rp-addr? inet:ipv4-address
|       +--ro ipv6-static-rp-addr? inet:ipv6-address
|       +--ro embedded-rp-addr?   inet:ipv6-address
|       +--ro ipv4-map-rp-addr?   inet:ipv4-address
|       +--ro ipv6-map-rp-addr?   inet:ipv6-address

```

Figure 18 The YANG data model of PIM SM group RP mapping information

```

+--rw pim-bfd-infos
|   +--ro pim-bfd-info* [vrf-name address-family if-name]
|       +--ro vrf-name          string
|       +--ro address-family    enumeration
|       +--ro if-name           if-name
|       +--ro ipv4-neighbor-addr? inet:ipv4-address
|       +--ro ipv6-neighbor-addr? inet:ipv6-address
|       +--ro act-tx?           uint32
|       +--ro act-rx?           uint32
|       +--ro act-multi?        uint32
|       +--ro local-port?       uint32
|       +--ro remote-port?      uint32
|       +--ro bfd-status?       enumeration

```

Figure 19 The YANG data model of PIM SM BFD session information

```
+---rw pim-vrf-packet-statistics
  +---ro pim-vrf-packet-statistic* [vrf-name address-family]
    +---ro vrf-name                string
    +---ro address-family          enumeration
    +---ro register-recv-num?      uint32
    +---ro register-send-num?     uint32
    +---ro register-invalid-num?   uint32
    +---ro register-filter-num?    uint32
    +---ro register-stop-recv-num? uint32
    +---ro register-stop-send-num? uint32
    +---ro register-stop-invalid-num? uint32
    +---ro register-stop-filter-num? uint32
    +---ro probe-recv-num?         uint32
    +---ro probe-send-num?         uint32
    +---ro probe-invalid-num?      uint32
    +---ro probe-filter-num?       uint32
    +---ro crp-recv-num?           uint32
    +---ro crp-send-num?           uint32
    +---ro crp-invalid-num?        uint32
    +---ro crp-filter-num?         uint32
    +---ro pim-if-packet-statistics
      +---ro pim-if-packet-statistic* [if-name]
        +---ro if-name             if-name
        +---ro assert-recv-num?    uint32
        +---ro assert-send-num?    uint32
        +---ro assert-invalid-num? uint32
        +---ro assert-filter-num?  uint32
        +---ro hello-recv-num?     uint32
        +---ro hello-send-num?     uint32
        +---ro hello-invalid-num?  uint32
        +---ro hello-filter-num?   uint32
        +---ro jp-recv-num?        uint32
        +---ro jp-send-num?        uint32
```

+-ro jp-invalid-num?	uint32
+-ro jp-filter-num?	uint32
+-ro bsr-recv-num?	uint32
+-ro bsr-send-num?	uint32
+-ro bsr-invalid-num?	uint32
+-ro bsr-filter-num?	uint32

Figure 20 The YANG data model of PIM SM packet statistics

#### 4. Notification

The pim yang data model introduces some notifications:

pim-neighbor-loss : This notification is sent when an adjacency with a neighbor was lost.

Liu, et al.

Expires September 23, 2015

[Page 18]

---

Internet-Draft

PIM YANG Data Model

March 2015

pim-invalid-register : This notification is sent when an invalid PIM Register message was received.

pim-invalid-join-prune : This notification is sent when an invalid PIM Join/Prune message was received.

pim-rp-mapping-change : This notification is sent when the active RP mapping has been changed.

pim-interface-election : This notification is sent when a new PIM DR has been elected on a network.

## notifications:

```
+---n pim-neighbor-loss
|  +--ro vrf-name?          string
|  +--ro address-family?    enumeration
|  +--ro if-name?           if-name
|  +--ro ipv4-neighbor-addr? inet:ipv4-address
|  +--ro ipv6-neighbor-addr? inet:ipv6-address
|  +--ro neighbor-up-time?   uint32
+---n pim-invalid-register
|  +--ro vrf-name?          string
|  +--ro address-family?    enumeration
|  +--ro ipv4-invalid-register-origin? inet:ipv4-address
|  +--ro ipv6-invalid-register-origin? inet:ipv6-address
|  +--ro ipv4-invalid-register-group?  inet:ipv4-address
|  +--ro ipv6-invalid-register-group?  inet:ipv6-address
|  +--ro ipv4-invalid-register-rp?     inet:ipv4-address
|  +--ro ipv6-invalid-register-rp?     inet:ipv6-address
+---n pim-invalid-join-prune
|  +--ro vrf-name?          string
```

```

| +--ro address-family?          enumeration
| +--ro ipv4-invalid-join-prune-origin?  inet:ipv4-address
| +--ro ipv6-invalid-join-prune-origin?  inet:ipv6-address
| +--ro ipv4-invalid-join-prune-group?   inet:ipv4-address
| +--ro ipv6-invalid-join-prune-group?   inet:ipv6-address
| +--ro ipv4-invalid-join-prune-rp?      inet:ipv4-address
| +--ro ipv6-invalid-join-prune-rp?      inet:ipv6-address
| +--ro neighbor-up-time?               uint32
+---n pim-rp-mapping-change
| +--ro vrf-name?                    string
| +--ro address-family?              enumeration
| +--ro group-mapping-origin?        enumeration
| +--ro ipv4-group-mapping-group-address?  inet:ipv4-address
| +--ro ipv6-group-mapping-group-address?  inet:ipv6-address
| +--ro group-mapping-group-prefix-length? uint8
| +--ro ipv4-group-mapping-rp-address?   inet:ipv4-address
| +--ro ipv6-group-mapping-rp-address?   inet:ipv6-address
| +--ro group-mapping-precedence?       uint32
+---n pim-interface-election
  +--ro vrf-name?                    string
  +--ro address-family?              enumeration
  +--ro if-name?                    if-name
  +--ro ipv4-if-addr?               inet:ipv4-address
  +--ro ipv6-if-addr?               inet:ipv6-address

```

Figure 21 The YANG data model of PIM SM notifications

## 5. PIM Yang Module

```

module pim {
  namespace "urn:huawei:params:xml:ns:yang:pim";
  // replace with IANA namespace when assigned -
  // urn:ietf:params:xml:ns:yang:1
  prefix "pim";
  import ietf-inet-types {
    prefix inet;
  }
  organization
    "IETF PIM(Protocol Independent Multicast) Working Group";

```

```

contact
  "liuyisong@huawei.com
  guofeng@huawei.com
  masivaku@cisco.com";
description
  "This YANG module defines the generic configuration
  data for PIM, 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
  PIM configuration parameters.";
revision 2015-03-09 {
  description
    "Add operation and notification content.";
}
revision 2014-10-21 {
  description
    "Initial revision.";
}

typedef if-name {
  description "if-name is like ethernet1/1/1/1";
  type string {
    length "1..63";
  }
}

container pim {

  container pim-bsr {

    container pim-bsr-instances {

      list c-bsr-instance {

        key "vrf-name address-family";
        max-elements "unbounded";

```

```

min-elements "0";
description "Specifies an pim bsr instance.";

leaf vrf-name {
  description "Name of a vpn instance.";

```



```

        config "true";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
    leaf address-family {
        config "true";
        mandatory "true";
        type enumeration {
            enum ipv4unicast {
                value "0";
                description
                    "Address family, which
                     determines whether an address belongs
                     to ipv4 or ipv6.";
            }
            enum ipv6unicast {
                value "1";
                description
                    "Address family, which determines
                     whether an address belongs to ipv4
                     or ipv6.";
            }
        }
    }
}
leaf c-bsr-if-name {
    description "interface name.";
    config "true";
    mandatory "true";
    type if-name;
}

leaf ipv4-c-bsr-if-addr {
    description
        "Global ipv4 unicast address of the c-bsr.";
    config "true";
    type inet:ipv4-address;
}
leaf ipv6-c-bsr-if-addr {
    description
        "Global ipv6 unicast address of the c-bsr.";
    config "true";
}

```

```

        mandatory "true";
        type inet:ipv6-address;
    }
    leaf c-bsr-holdtime {
        description
            "Timeout period (called holdtime) during which
            c-bsrs wait to receive bootstrap messages
            from the bsr. The value is an integer ranging
            from 1 to 214748364, in seconds. The default
            value is 130.To prevent frequent bsr elections,
            set the same holdtime for all c-bsrs in the
            same domain. If an interval (called bs_interval)
            at which bootstrap messages are sent has been
            set for c-bsrs, ensure that the specified
            holdtime is larger than the bs_interval.";
        config "true";
        default "130";
        type uint32 {
            range "1..214748364";
        }
    }
    leaf c-bsr-interval {
        description
            "Interval (called bs_interval) at which a bsr
            continuously sends bootstrap messages. The
            value is an integer ranging from 1 to
            107374177, in seconds. the default value is
            60.To prevent frequent bsr elections, set
            the same bs_interval for all c-bsrs in the
            same domain.If a timeout period (called
            holdtime) during which c-bsrs wait to
            receive bootstrap messages from the bsr has
            been set for c-bsrs, ensure that the
            specified bs_interval is smaller than the
            holdtime.";
        config "true";
        default "60";
        type uint32 {
            range "1..107374177";
        }
    }
    leaf c-bsr-hashlen {
        description
            "Global hash mask length for a c-bsr. the
            value is an integer ranging from 0 to 32.
            The default value is 30. in ipv4, the value
            is an integer ranging from 0 to 32, and the
            default value is 30. in ipv6, the value is

```

Internet-Draft

PIM YANG Data Model

March 2015

```
        an integer ranging from 0 to 128, and the
        default value is 126.";
    config "true";
    default "126";
    type uint32 {
        range "0..128";
    }
}
leaf c-bsr-priority {
    description
        "Globally specify a priority for all c-bsrs
        on the router. The greater the value, the
        higher the priority. The value is an integer
        ranging from 0 to 255. the default value is 0.";
    config "true";
    default "0";
    type uint8 {
        range "0..255";
    }
}
leaf c-bsr-ply-name {
    description
        "Policy for limiting the range of valid bsr
        addresses so that a router discards the
        messages received from the bsrs not in the
        specified address range. The value is an
        integer ranging from 2000 to 2999, or a
        string of 1 to 32 case-sensitive characters.
        By default, the range of valid bsr addresses
        is not limited.";
    config "true";
    type string {
        length "1..255";
        pattern "^[^ ]+$";
    }
}
leaf c-bsr-ply-ipv6 {
    description
        "Policy for limiting the range of valid bsr
        addresses so that a router discards the
        messages received from the bsrs not in the
        specified address range. The value is an
```

```
integer ranging from 2000 to 2999, or a
string of 1 to 32 case-sensitive characters.
By default, the range of valid bsr addresses
is not limited.";
config "true";
type string {
```

```
length "1..255";
pattern "^[^ ]+$";
}
}
leaf c-bsr-adminscope {
  description
    "One pim-sm domain is divided into multiple
    bsr administrative domains to implement
    rp-set advertisement. by default, there is
    only one bsr in the entire pim-sm domain.";
  config "true";
  type boolean
  {
  }
}
leaf c-bsr-global-enable {
  description
    "The router is a c-bsr in the global domain.
    By default, no c-bsr is configured in the
    global domain.";
  config "true";
  type boolean
  {
  }
}
leaf c-bsr-global-hashlength {
  description
    "Hash mask length for the c-bsr in the global
    domain. The value is an integer ranging
    from 0 to 32. The default value is 30.";
  config "true";
  default "30";
  type uint32 {
    range "0..32";
  }
}
```

```

}
leaf c-bsr-global-priority {
    description
        "Priority for the c-bsr in the global domain.
        The greater the value, the higher the
        priority. The value is an integer ranging
        from 0 to 255. the default value is 0.";
    config "true";
    default "0";
    type uint32 {
        range "0..255";
    }
}

```

```

leaf is-fragable {
    description
        "Enable the c-bsr message fragmentation
        function. By default, this function is
        disabled.";
    config "true";
    mandatory "true";
    type boolean
    {
    }
}

list c-rp-instance {

    key "vrf-name address-family";
    max-elements "unbounded";
    min-elements "0";
    description "specifies a c-rp instance.";

    leaf vrf-name {
        description "Name of a vpn name.";
        config "true";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
}

```

```

leaf address-family {
    config "true";
    mandatory "true";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description
                "Address family, which determines
                whether an address belongs to ipv4
                or ipv6.";
        }
        enum ipv6unicast {
            value "1";
            description
                "Address family, which determines
                whether an address belongs to ipv4
                or ipv6.";
        }
    }
}

```

```

leaf c-rp-if-name {
    description "interface name.";
    config "true";
    mandatory "true";
    type if-name;
}
leaf c-rp-grpply-name {
    description
        "Policy for limiting the range of valid group
        addresses. With this policy, a router
        discards messages received from the addresses
        not in the specified range. The value is an
        integer ranging from 2000 to 2999, or a
        string of 1 to 32 case-sensitive characters.";
    config "true";
    type string {
        length "1..255";
        pattern "^[^ ]+$";
    }
}
leaf c-rp-priority {

```

```

        description
            "Priority of a c-rp. The greater the value,
            the lower the priority. The value is an
            integer ranging from 0 to 255. The default
            value is 0.";
        config "true";
        default "0";
        type uint8 {
            range "0..255";
        }
    }
    leaf c-rp-holdtime {
        description
            "Timeout period during which a bsr waits to
            receive advertisement messages from a c-rp.
            The value is an integer ranging from 1 to
            65535, in seconds. The default value is 150.";
        config "true";
        default "150";
        type uint16 {
            range "1..65535";
        }
    }
    leaf c-rp-adv-interval {
        description
            "Interval at which a c-rp sends advertisement
            messages. The value is an integer ranging

```

```

            from 1 to 65535, in seconds. The default
            value is 60.";
        config "true";
        default "60";
        type uint16 {
            range "1..65535";
        }
    }
    leaf ipv4-c-rp-addr {
        description "specifies a c-rp address.";
        config "true";
        type inet:ipv4-address;
    }
    leaf ipv6-c-rp-addr {

```

```

        description "specifies a c-rp address.";
        config "true";
        mandatory "true";
        type inet:ipv6-address;
    }
}

container pim-bsr-ebsr-infos {

    list pim-bsr-ebsr-info {

        key "vrf-name address-family";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {
            config "false";
            type enumeration {
                enum ipv4unicast {
                    value "0";
                    description "ipv4unicast:ipv4unicast";
                }
                enum ipv6unicast {
                    value "1";
                    description "ipv6unicast:ipv6unicast";
                }
            }
        }
    }
}

```

```

    }
}
leaf ipv4-bsr-addr {
    description "address of an elected bsr.";
    config "false";
    type inet:ipv4-address;
}

```



```

}
leaf ipv6-bsr-addr {
    description "address of an elected bsr.";
    config "false";
    type inet:ipv6-address;
}
leaf bsr-scope {
    description
        "whether the bsr is a bsr in an
        administrative domain.";
    config "false";
    type enumeration {
        enum notscoped {
            value "0";
            description "notscoped:notscoped";
        }
        enum global {
            value "1";
            description "global:global";
        }
        enum adminscope {
            value "2";
            description "adminscope:adminscope";
        }
    }
}
leaf ipv4-group-addr {
    description
        "Multicast group address. In ipv4 address,
        the value ranges from 239.0.0.0 to
        239.255.255.255. ";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-group-addr {
    description
        "Multicast group address. In ipv6, the value
        is in hexadecimal notation and in the format
        of ffx:xxxx:xxxx::xxxx ";
    config "false";
    type inet:ipv6-address;
}

```

```

}
leaf ipv4-group-masklen {
    description
        "Mask length of a group address. The value is
        an integer ranging from 4 to 32.";
    config "false";
    type uint8 {
        range "4..32";
    }
}
leaf ipv6-group-masklen {
    description
        "Mask length of a group address. the value is
        an integer ranging from 8 to 128.";
    config "false";
    type uint8 {
        range "8..128";
    }
}
leaf priority {
    description
        "Priority for a c-bsr to participate in bsr
        election. The greater the value, the higher
        the priority. The value is an integer
        ranging from 0 to 255. The default value is
        0.";
    config "false";
    type uint8 {
        range "0..255";
    }
}
leaf hash-length {
    description
        "Hash length used by an elected bsr to perform
        rp calculation. The value is an integer
        ranging from 0 to 32.";
    config "false";
    type uint8 {
        range "0..32";
    }
}
leaf fsm-state {
    description "status of an elected bsr.";
    config "false";
    type enumeration {
        enum unknown {
            value "0";
            description "unknown:unknown";
        }
    }
}

```

Internet-Draft

PIM YANG Data Model

March 2015

```
    }
    enum acceptany {
        value "1";
        description "acceptany:acceptany";
    }
    enum acceptpreferred {
        value "2";
        description "acceptpreferred:acceptpreferred";
    }
    enum candidate {
        value "3";
        description "candidate:candidate";
    }
    enum pending {
        value "4";
        description "pending:pending";
    }
    enum elected {
        value "5";
        description "elected:elected";
    }
}
}
leaf up-time {
    description
        "time since the c-bsr was elected as the bsr.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf expire-time {
    description
        "period after which the elected bsr on the
        lan sends the next bootstrap message.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf c-rp-count {
    description "number of candidate rps.";
    config "false";
```

```

        type uint32 {
            range "0..4294967295";
        }
    }
}

```

```

    }

}

container pim-bsr-cbsr-infos {

    list pim-bsr-cbsr-info {

        key "vrf-name address-family";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {
            config "false";
            type enumeration {
                enum ipv4unicast {
                    value "0";
                    description "ipv4unicast:ipv4unicast";
                }
                enum ipv6unicast {
                    value "1";
                    description "ipv6unicast:ipv6unicast";
                }
            }
        }
        leaf ipv4-cbsr-addr {
            description "address of a c-bsr.";
            config "false";
            type inet:ipv4-address;
        }
    }
}

```

```

}
leaf ipv6-cbsr-addr {
    description "address of a c-bsr.";
    config "false";
    type inet:ipv6-address;
}
leaf bsr-scope {
    description
        "whether the bsr is a bsr in an administrative
        domain.";
    config "false";
    type enumeration {

```

```

        enum notscoped {
            value "0";
            description "notscoped:notscoped";
        }
        enum global {
            value "1";
            description "global:global";
        }
        enum adminscope {
            value "2";
            description "adminscope:adminscope";
        }
    }
}
leaf ipv4-group-addr {
    description
        "Multicast group address. In ipv4 address,
        the value ranges from 239.0.0.0 to
        239.255.255.255. ";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-group-addr {
    description
        "Multicast group address. In ipv6, the value
        is in hexadecimal notation and in the format
        of ffx:xxxx:xxxx::xxxx ";
    config "false";
    type inet:ipv6-address;
}

```

```

}
leaf ipv4-group-masklen {
    description
        "Mask length of a group address. The value is
        an integer ranging from 4 to 32.";
    config "false";
    type uint8 {
        range "4..32";
    }
}
leaf ipv6-group-masklen {
    description
        "Mask length of a group address. the value is
        an integer ranging from 8 to 128.";
    config "false";
    type uint8 {
        range "8..128";
    }
}
}

```

```

leaf priority {
    description
        "Priority for a c-bsr to participate in bsr
        election. The greater the value, the higher
        the priority. The value is an integer
        ranging from 0 to 255. the default value is
        0.";
    config "false";
    type uint8 {
        range "0..255";
    }
}
leaf hash-length {
    description
        "Hash length used by an elected bsr to
        perform rp calculation. The value is an
        integer ranging from 0 to 32.";
    config "false";
    type uint8 {
        range "0..32";
    }
}
}

```

```

leaf fsm-state {
    description "status of a c-bsr.";
    config "false";
    type enumeration {
        enum unknown {
            value "0";
            description "unknown:unknown";
        }
        enum acceptany {
            value "1";
            description "acceptany:acceptany";
        }
        enum acceptpreferred {
            value "2";
            description "acceptpreferred:acceptpreferred";
        }
        enum candidate {
            value "3";
            description "candidate:candidate";
        }
        enum pending {
            value "4";
            description "pending:pending";
        }
        enum elected {
            value "5";

```

```

        description "elected:elected";
    }
}
leaf is-wait-to-bsr {
    description
        "whether the current c-bsr is valid
        (0: valid; 1: invalid).";
    config "false";
    type enumeration {
        enum join {
            value "0";
            description "join:join";
        }
        enum notjoin {

```

```

        value "1";
        description "notjoin:notjoin";
    }
}
}
}

container pim-bsr-ebsr-rps {

    list pim-bsr-ebsr-rp {

        key "vrf-name address-family";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {
            config "false";
            type enumeration {
                enum ipv4unicast {
                    value "0";
                    description "ipv4unicast:ipv4unicast";
                }
                enum ipv6unicast {
                    value "1";

```

```

        description "ipv6unicast:ipv6unicast";
    }
}
}
leaf ipv4-rp-addr {
    description "rp address.";
    config "false";
    type inet:ipv4-address;

```



```

}
leaf ipv6-rp-addr {
    description "rp address.";
    config "false";
    type inet:ipv6-address;
}
leaf rp-addr-is-local {
    description
        "whether the rp address is a local address.";
    config "false";
    type boolean;
}
leaf ipv4-group-addr {
    description
        "Multicast group address. In ipv4 address,
        the value ranges from 239.0.0.0 to
        239.255.255.255. ";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-group-addr {
    description
        "Multicast group address. in ipv6, the value
        is in hexadecimal notation and in the format
        of ffx:xxxx:xxxx::xxxx ";
    config "false";
    type inet:ipv6-address;
}
leaf ipv4-group-masklen {
    description
        "Mask length of a group address. The value is
        an integer ranging from 4 to 32.";
    config "false";
    type uint8 {
        range "4..32";
    }
}
leaf ipv6-group-masklen {
    description
        "Mask length of a group address. The value is

```

an integer ranging from 8 to 128.";

```

        config "false";
        type uint8 {
            range "8..128";
        }
    }
    leaf priority {
        description
            "Priority of an rp. The greater the value,
            the higher the priority. The value is an
            integer ranging from 0 to 255. the default
            value is 0.";
        config "false";
        type uint8 {
            range "0..255";
        }
    }
    leaf up-time {
        description
            "time since the c-rp was elected as the rp.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf expiry-time {
        description
            "time after which an rp will expire, in
            seconds.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
}

}

}

container pim-sm {

    container pim-sm-instances {

        list pim-sm-instance {

            key "vrf-name address-family";

```

---

```
max-elements "unbounded";
min-elements "0";
description "Specifies an pim-sm instance.";

leaf vrf-name {
    description "Name of a vpn instance.";
    config "true";
    //default "_public_";
    type string {
        length "0..32";
    }
}

leaf address-family {
    config "true";
    mandatory "true";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description
                "Address family, which determines
                whether an address belongs to ipv4
                or ipv6.";
        }
        enum ipv6unicast {
            value "1";
            description
                "Address family, which determines
                whether an address belongs to ipv4
                or ipv6.";
        }
    }
}

leaf assert-holdtime {
    description
        "Timeout period during which pim interfaces
        wait to receive assert messages from the
        forwarder. The value is an integer ranging
        from 0 to 65535, in seconds. The default
        value is 180. ";
    config "true";
    default "180";
    type uint16 {
        range "0..65535";
    }
}
```

```
leaf jp-holdtime {  
    description
```

```
        "Holdtime for a join/prune message sent by a  
        pim interface. The value is an integer  
        ranging from 1 to 65535, in seconds. The  
        default value is 210. Commonly, the holdtime  
        is 3.5 times longer than the interval for  
        all the interfaces to send join/prune messages.";  
    config "true";  
    default "210";  
    type uint16 {  
        range "1..65535";  
    }  
}  
leaf probe-interval {  
    description  
        "Interval at which probe messages are sent  
        to an rp. The value is an integer ranging  
        from 1 to 3600, in seconds. The default  
        value is 5.";  
    config "true";  
    default "5";  
    type uint16 {  
        range "1..3600";  
    }  
}  
leaf jp-timer-interval {  
    description  
        "Interval at which join/prune messages are  
        sent. The value is an integer ranging from  
        1 to 2147483647, in seconds. It must be  
        shorter than the holdtime of join/prune  
        messages. The default value is 60. ";  
    config "true";  
    default "60";  
    type uint16 {  
        range "1..18000";  
    }  
}  
leaf dr-priority {  
    description
```

```

        "Dr election priority for a router. The
        greater the value, the higher the priority.
        The value is an integer ranging from 0 to
        4294967295. The default value is 1.";
    config "true";
    default "1";
    type uint32 {
        range "0..4294967295";
    }

```

```

    }
    leaf hello-holdtime {
        description
            "Timeout period during which a pim interface
            waits to receive hello messages from its
            neighbors. The value is an integer that
            ranging from 1 to 65535, in seconds. It must
            be longer than the interval for pim neighbors
            to send hello messages. The default value
            is 105. ";
        config "true";
        default "105";
        type uint16 {
            range "1..65535";
        }
    }
    leaf hello-lan-delay {
        description
            "Delay for transmitting prune messages on a
            shared network segment. The value is an
            integer ranging from 1 to 32767, in
            milliseconds. The default value is 500.";
        config "true";
        default "500";
        type uint16 {
            range "1..32767";
        }
    }
    leaf hello-interval {
        description
            "Specifies the interval at which hello
            messages are sent. The value is an integer

```

```

        ranging from 1 to 18000, in seconds.
        It must be shorter than the timeout period
        of pim neighbors. The default value is 30.";
    config "true";
    default "30";
    type uint16 {
        range "1..18000";
    }
}
leaf hello-override {
    description
        "Interval at which the prune action in a hello
        message is overridden. The value is an
        integer ranging from 1 to 65535, in
        milliseconds. The default value is 2500.";
    config "true";

```

```

        default "2500";
        type uint16 {
            range "1..65535";
        }
    }
    leaf reg-checksum {
        description
            "Configure a router to calculate the checksum
            based on all contents of a register message.
            By default, the checksum is calculated based
            on the header of a register message only.";
        config "true";
        default "false";
        type boolean {
        }
    }
    leaf reg-suppress-interval {
        description
            "Timeout period during which a router remains
            in the registration suppression state. The
            value is an integer ranging from 0 to 65535,
            in seconds. The default value is 60.";
        config "true";
        default "60";
        type uint16 {

```

```

        range "0..65535";
    }
}
leaf embedded-rp {
    description
        "Enable the embedded rp function. By default,
        the embedded rp function is enabled.";
    config "true";
    default "true";
    type boolean {
    }
}
leaf nbr-check-recv {
    description
        "Enable the pim neighbor check function to
        check whether received join/prune and assert
        messages are sent from a pim neighbor. If
        not, these messages are discarded. By
        default, the pim neighbor check function is
        disabled.";
    config "true";
    type boolean {
    }
}

```

```

}
leaf nbr-check-send {
    description
        "Enable the pim neighbor check function to
        check whether join/prune and assert messages
        are to be sent to an ipv4/v6 pim neighbor.
        If not, these messages are not sent. By
        default, the pim neighbor check function is
        disabled for join/prune and assert messages
        to be sent.";
    config "true";
    type boolean {
    }
}
leaf reg-ply-name {
    description
        "Policy for filtering register messages. The
        value is an integer ranging from 3000 to

```

```

        3999, or a string of 1 to 32 case-sensitive
        characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf reg-ply-ipv6 {
    description
        "Policy for filtering register messages. The
        value is an integer ranging from 3000 to
        3999, or a string of 1 to 32 case-sensitive
        characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf ssm-ply-name {
    description
        "Policy for limiting the range of valid ssm
        group addresses. The value is an integer
        ranging from 2000 to 2999, or a string of 1
        to 32 case-sensitive characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}

```

```

    }
}
leaf ssm-ply-ipv6 {
    description
        "Range of ssm group addresses. The value is
        an integer ranging from 2000 to 2999, or a
        string of 1 to 32 case-sensitive characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}

```



```

    }
}
leaf src-ply-name {
    description
        "Policy for filtering multicast entries based
        on source addresses or based on both source
        and group addresses. The value is an integer
        ranging from 2000 to 3999, or a string of 1
        to 32 case-sensitive characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf src-ply-ipv6 {
    description
        "Policy for filtering multicast entries based
        on source addresses or based on both source
        and group addresses. The value is an integer
        ranging from 2000 to 3999, or a string of 1
        to 32 case-sensitive characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf bsr-ply-name {
    description
        "Policy for limiting the range of valid bsr
        addresses so that a router discards the
        messages received from the bsrs not in the
        specified address range. The value is an
        integer ranging from 2000 to 2999, or a
        string of 1 to 32 case-sensitive characters.
        By default, the range of valid bsr addresses

```

```

        is not limited.";
    config "true";
    type string {
        length "1..32";

```

```

        pattern "^[^ ]+$";
    }
}
leaf bsr-ply-ipv6 {
    description
        "Policy for limiting the range of valid bsr
        addresses so that a router discards the
        messages received from the bsrs not in the
        specified address range. The value is an
        integer ranging from 2000 to 2999, or a
        string of 1 to 32 case-sensitive characters.
        By default, the range of valid bsr addresses
        is not limited.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf emb-rp-ply-name {
    description
        "Policy for limiting the range of multicast
        groups to which an embedded-rp applies. The
        value is an integer ranging from 2000 to
        2999, or a string of 1 to 32 case-sensitive
        characters.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf source-lifetime {
    description
        "Timeout period for (s, g) entries on a
        router. The value is an integer ranging from
        60 to 65535, in seconds. The default value
        is 210.";
    config "true";
    default "210";
    type uint16 {
        range "60..65535";
    }
}

```

```
leaf ipsec-name {
  description
    "Description of an sa. The value is a string
    of 1 to 15 characters.";
  config "true";
  type string {
    length "1..15";
    pattern "^[^ ]+$";
  }
}
leaf ipsec-type {
  description
    "Hello options based on the ipsec type.";
  config "true";
  type enumeration {
    enum forall {
      value "0";
      description
        "hello options based on the ipsec type.";
    }
    enum forhello {
      value "1";
      description
        "hello options based on the ipsec type.";
    }
  }
}
leaf uni-ipsec-name {
  description
    "Description of an sa. the value is a string
    of 1 to 15 characters.";
  config "true";
  type string {
    length "1..15";
    pattern "^[^ ]+$";
  }
}
container pim-static-rps {
  container pim-static-rp {
    leaf ipv4-static-rp-addr {
      description "Specifies a static rp address.";
      config "true";
      mandatory "true";
      type inet:ipv4-address;
    }
  }
}
```

```
        description "Specifies a static rp address.";
        config "true";
        mandatory "true";
        type inet:ipv6-address;
    }
    leaf static-rp-ply-name {
        description
            "Static rp policy. The value is an
            integer ranging from 2000 to 2999,
            or a string of 1 to 32
            case-sensitive characters.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf static-rp-ply-name-ipv6 {
        description
            "Static rp policy. the value is an
            integer ranging from 2000 to 2999,
            or a string of 1 to 32
            case-sensitive characters.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf preference {
        description
            "Whether the static rp is preferred.
            The value can be: 0: the static rp
            is not preferred. 1: the static rp
            is preferred. the default value is 0.";
        config "true";
        default "notprefer";
        type enumeration {
            enum notprefer {
                value "0";
```



```

        description
            "The receiver's dr never triggers the
            switchover from the rpt to the spt.";
        config "true";
        type boolean {
        }
    }
    leaf spt-threshhold {
        description
            "Multicast packet rate threshold that
            triggers the switchover from the rpt
            to the spt. The value is an integer
            ranging from 1 to 4194304, in kbit/s. ";
        config "true";
        type uint32 {

```

```

            range "1..4194304";
        }
    }
    leaf is-acl-enable {
        description
            "Whether to enable a group policy. ";
        config "true";
        mandatory "true";
        type boolean {
        }
    }
    leaf spt-grp-ply-name {
        description
            "Name of a named acl. The value is an
            integer ranging from 2000 to 2999,
            or a string of 1 to 32 case-sensitive
            characters. If the parameter is not
            set, the threshold is applied to all
            multicast groups. \"0\" means that
            no acl is specified.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
}

```

```

leaf spt-grp-ply-order {
  description
    "Adjust the order of the acls in the
    group-policy list. If a group matches
    multiple acls, the threshold is
    selected in the order specified by
    the order-value parameter.
    Order-value specifies the updated
    number. It is an integer. The value
    is any value other than original one
    in the current group-policy list. If
    the parameter is not set, the order
    of the acls in the group-policy list
    does not change.";
  config "true";
  default "4294967295";
  type uint32 {
    range "1..4294967295";
  }
}
}

```

```

}

container pim-sm-anycast-rps {

  container pim-sm-anycast-rp {

    leaf ipv4-rp-address {
      description
        "Address of an ipv4 anycast-rp peer";
      config "true";
      mandatory "true";
      type inet:ipv4-address;
    }

    leaf ipv6-rp-address {
      description
        "Address of an ipv6 anycast-rp peer";
      config "true";
      mandatory "true";
    }
  }
}

```

```

        type inet:ipv6-address;
    }

    leaf local-ipv4-address {
        description
            "Address of an ipv4 anycast-rp peer";
        config "true";
        mandatory "true";
        type inet:ipv4-address;
    }

    leaf local-ipv6-address {
        description
            "Address of an ipv4 anycast-rp peer";
        config "true";
        mandatory "true";
        type inet:ipv6-address;
    }

    container pim-sm-rp-peers {

        container pim-sm-rp-peer {

            leaf ipv4-rp-peer-address {
                description
                    "Address of an ipv4 anycast-rp peer";
                config "true";
                mandatory "true";
                type inet:ipv4-address;
            }

```

```

    }
    leaf ipv6-rp-peer-address {
        description
            "Address of an ipv6 anycast-rp peer";
        config "true";
        mandatory "true";
        type inet:ipv6-address;
    }
    leaf fwd-sa-swt {
        description
            "Whether to forward to
            anycast-rp peers source

```



```

        information learned through
        sa messages.";
        config "true";
        type boolean {
        }
    }
    leaf fwd-policy {
        description
            "Policy for filtering sa
            messages to be forwarded to
            anycast-rp peers. The value
            is an integer ranging from
            2000 to 2999, or a string of
            1 to 32 case-sensitive
            characters.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf fwd-policy-ipv6 {
        description
            "Policy for filtering sa
            messages to be forwarded to
            anycast-rp peers. The value
            is an integer ranging from
            2000 to 2999, or a string of
            1 to 32 case-sensitive
            characters.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
}

```

```

    }
}
}

```

```

}

container pim-sm-interfaces {

    list pim-sm-interface {

        key "vrf-name address-family if-name";
        max-elements "unbounded";
        min-elements "0";
        description "Specifies an pim sm interface.";

        leaf vrf-name {
            description
                "Name of an pim sm instance. If the
                 name string is empty the instance
                 means a public instance whose name
                 is _public_.";
            config "true";
            mandatory "true";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {
            config "true";
            mandatory "true";
            type enumeration {
                enum ipv4unicast {
                    value "0";
                    description
                        "Address family, which
                         determines whether an
                         address belongs to ipv4
                         or ipv6.";
                }
                enum ipv6unicast {
                    value "1";
                    description
                        "Address family, which
                         determines whether an
                         address belongs to ipv4

```

```

        or ipv6.";
    }
}
}
leaf if-name {
    description "Interface name.";
    config "true";
    mandatory "true";
    type if-name;
}
leaf pim-sm-enable {
    description
        "Enable pim-sm on an interface.";
    config "true";
    mandatory "true";
    type boolean {
    }
}
leaf dr-priority {
    description
        "Dr priority. The greater the value,
        the higher the priority. The value
        is an integer ranging from 0 to
        4294967295. The default value is 1.";
    config "true";
    default "1";
    type uint32 {
        range "0..4294967295";
    }
}
leaf hello-interval {
    description
        "Interval at which hello messages are
        sent. The value is an integer ranging
        from 1 to 2147483647, in seconds. It
        must be shorter than the timeout
        period of pim neighbors. The default
        value is 30.";
    config "true";
    default "30";
    type uint16 {
        range "1..18000";
    }
}
leaf hello-holdtime {
    description
        "Timeout period during which a router
        waits for hello messages sent from

```

Internet-Draft

PIM YANG Data Model

March 2015

```
        its pim neighbors. The value is an
        integer ranging from 1 to 65535, in
        seconds. It must be longer than the
        interval for pim neighbors to send
        hello messages. The default value
        is 105.";
    config "true";
    default "105";
    type uint16 {
        range "1..65535";
    }
}
leaf hello-override {
    description
        "Interval at which the prune action
        in a hello message is overridden.
        The value is an integer ranging from
        1 to 65535, in milliseconds. The
        default value is 2500.";
    config "true";
    default "2500";
    type uint16 {
        range "1..65535";
    }
}
leaf hello-lan-delay {
    description
        "Period from the time when a router
        receives a prune message from a
        downstream device to the time when
        the router performs the prune action.
        The value is an integer ranging from
        1 to 32767, in milliseconds. The
        default value is 500.";
    config "true";
    default "500";
    type uint16 {
        range "1..32767";
    }
}
leaf jp-timer-interval {
    description
```

"Interval at which join/prune messages are sent. The value is an integer ranging from 1 to 2147483647, in seconds. the default value is 60. The interval must be shorter than the holdtime of join/prune messages.";

```
        config "true";
        default "60";
        type uint16 {
            range "1..18000";
        }
    }
    leaf jp-holdtime {
        description
            "Holdtime for join/prune messages
            sent by a router. The value is an
            integer ranging from 1 to 65535, in
            seconds. The default value is 210.
            The holdtime must be longer than the
            interval at which join/prune
            messages are sent.";
        config "true";
        default "210";
        type uint16 {
            range "1..65535";
        }
    }
    leaf jp-ply-name {
        description
            "Policy for filtering join/prune
            messages. The value is an integer
            ranging from 2000 to 2999, or a
            string of 1 to 32 case-sensitive
            characters. By default, no policy
            is configured to filter join/prune
            messages.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
}
```

```

}
leaf jp-ply-ipv6 {
  description
    "Policy for filtering join/prune
    messages. The value is an integer
    ranging from 2000 to 2999, or a
    string of 1 to 32 case-sensitive
    characters. By default, no policy
    is configured to filter join/prune
    messages.";
  config "true";
  type string {
    length "1..32";
  }
}

```

```

    pattern "^[^ ]+$";
  }
}
leaf jp-asm-ply-name {
  description
    "Policy for filtering asm join/prune
    messages. The value is an integer
    ranging from 2000 to 2999, or a
    string of 1 to 32 case-sensitive
    characters. By default, no policy is
    configured to filter asm join/prune
    messages.";
  config "true";
  type string {
    length "1..32";
    pattern "^[^ ]+$";
  }
}
leaf jp-asm-ply-ipv6 {
  description
    "Policy for filtering asm join/prune
    messages. The value is an integer
    ranging from 2000 to 2999, or a
    string of 1 to 32 case-sensitive
    characters. By default, no policy is
    configured to filter asm join/prune
    messages.";
  config "true";
}

```

```

        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf jp-ssm-ply-name {
        description
            "Policy for filtering ssm join/prune
            messages. The value is an integer
            ranging from 3000 to 3999, or a
            string of 1 to 32 case-sensitive
            characters. By default, no policy is
            configured to filter ssm join/prune
            messages.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
}

```

```

    leaf jp-ssm-ply-ipv6 {
        description
            "Policy for filtering ssm join/prune
            messages. The value is an integer
            ranging from 3000 to 3999, or a
            string of 1 to 32 case-sensitive
            characters. By default, no policy is
            configured to filter ssm join/prune
            messages.";
        config "true";
        type string {
            length "1..32";
            pattern "^[^ ]+$";
        }
    }
    leaf nbr-ply-name {
        description
            "Policy for filtering pim neighbors.
            The value is an integer ranging from
            2000 to 2999, or a string of 1 to 32
            case-sensitive characters. By default,

```

```

        no policy is configured to filter
        pim neighbors.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf nbr-ply-ipv6 {
    description
        "Policy for filtering pim neighbors.
        The value is an integer ranging from
        2000 to 2999, or a string of 1 to 32
        case-sensitive characters. By default,
        no policy is configured to filter
        pim neighbors.";
    config "true";
    type string {
        length "1..32";
        pattern "^[^ ]+$";
    }
}
leaf assert-holddtime {
    description
        "Timeout period during which pim
        interfaces wait to receive assert
        messages from the forwarder. The

```

```

        value is an integer ranging from 0
        to 65535, in seconds.The default
        value is 180.";
    config "true";
    default "180";
    type uint16 {
        range "0..65535";
    }
}
leaf require-gen-id {
    description
        "Configure a pim interface to deny
        hello messages that do not carry
        generation ids. By default, a pim

```



```

        interface receives hello messages
        that do not carry generation ids.";
    config "true";
    type boolean {
    }
}
leaf pim-bsr-boundary {
    description
        "Boundary for a pim domain.
        The value can be:
        0: no boundary;
        1: bidirectional domain boundary;
        2: inbound unidirectional domain boundary.";
    config "true";
    default "none";
    type enumeration {
        enum none {
            value "0";
            description
                "Boundary for a pim domain.
                The value can be:
                0: no boundary;
                1: bidirectional domain boundary;
                2: inbound unidirectional domain boundary.";
        }
        enum both {
            value "1";
            description
                "Boundary for a pim domain.
                The value can be:
                0: no boundary;
                1: bidirectional domain boundary;
                2: inbound unidirectional domain boundary.";
        }
    }
}

```

```

enum incoming {
    value "2";
    description
        "Boundary for a pim domain.
        The value can be:
        0: no boundary;
        1: bidirectional domain boundary;

```

```

        2: inbound unidirectional domain bound
    }
}
leaf bfd-enable {
    description
        "Enable pim bfd on an interface.
        By default, pim bfd is not enabled
        on an interface.";
    config "true";
    type boolean {
    }
}
leaf bfd-min-tx {
    description
        "Minimum interval at which bfd
        messages are sent. The value is an
        integer ranging from 3 to 1000,
        in milliseconds. ";
    config "true";
    default "0";
    type uint16 {
        range "0..1000";
    }
}
leaf bfdminrx {
    description
        "Minimum interval at which bfd
        messages are received. The value is
        an integer ranging from 3 to 1000,
        in milliseconds. ";
    config "true";
    default "0";
    type uint16 {
        range "0..1000";
    }
}
leaf bfd-multiplier {
    description
        "Local detect multiplier for bfd
        messages. The value is an integer

```

```

        ranging from 3 to 50. The default
        value is 3.";
    config "true";
    default "3";
    type uint8 {
        range "3..50";
    }
}
leaf is-silent {
    description
        "Enable the pim silent function on an
        interface. By default, pim silent is
        disabled on an interface. ";
    config "true";
    default "false";
    type boolean {
    }
}
leaf is-dr-swt-delay {
    description
        "Enable dr switchover delay on an
        interface. By default, this function
        is disabled on an interface.";
    config "true";
    default "false";
    type boolean {
    }
}
leaf dr-swt-delay-interval {
    description
        "Delay for a dr switchover. The value
        is an integer ranging from 10 to 3600.
        The default value is 10. The setting
        is ineffective if the dr switchover
        delay function is not enabled.";
    config "true";
    default "10";
    type uint16 {
        range "10..3600";
    }
}
leaf ipsec-name {
    description
        "Description of an sa. The value is a
        string of 1 to 15 characters.";
    config "true";
    type string {
        length "1..15";
    }
}

```

Internet-Draft

PIM YANG Data Model

March 2015

```

        pattern "[^ ]+$";
    }
}
leaf ipsec-type {
    description
        "Hello options based on the ipsec type.";
    config "true";
    type enumeration {
        enum forall {
            value "0";
            description
                "Hello options based on the ipsec type.";
        }
        enum forhello {
            value "1";
            description
                "Hello options based on the ipsec type.";
        }
    }
}
}
}
}
}

container pim-sm-if-infos {

    list pim-sm-if-info {

        key "vrf-name address-family if-name";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {

```

```

config "false";
type enumeration {
    enum ipv4unicast {
        value "0";
        description "ipv4unicast:ipv4unicast";
    }
}

```

```

    }
    enum ipv6unicast {
        value "1";
        description "ipv6unicast:ipv6unicast";
    }
}
leaf if-name {
    description "Interface name.";
    config "false";
    type if-name;
}
leaf if-up-state {
    description
        "Status of an interface: up or down.";
    config "false";
    type enumeration {
        enum down {
            value "0";
            description "down:down";
        }
        enum up {
            value "1";
            description "up:up";
        }
    }
}
leaf neighbor-count {
    description
        "Number of pim neighbors on an interface.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf dr-priority {

```

```

        description "Dr priority.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf ipv4-dr-addr {
        description "Dr address.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-dr-addr {

```

```

        description "Dr address.";
        config "false";
        type inet:ipv6-address;
    }

    leaf hello-interval {
        description
            "Interval at which hello messages are sent.";
        config "false";
        type uint16 {
            range "1..18000";
        }
    }
    leaf hello-holdtime {
        description
            "Timeout period during which a router waits
            to receive hello messages from its neighbors.";
        config "false";
        type uint16 {
            range "1..65535";
        }
    }
    leaf assert-holdtime {
        description
            "Timeout period during which pim interfaces
            wait to receive assert messages from the
            forwarder. ";
        config "false";
        type uint16 {

```

```

        range "0..65535";
    }
}
leaf jp-interval {
    description
        "Interval at which join/prune messages are
        sent.";
    config "false";
    type uint16 {
        range "1..18000";
    }
}
leaf jp-holdtime {
    description
        "Holdtime for join/prune messages sent by a
        router.";
    config "false";
    type uint16 {
        range "1..65535";
    }
}

```

```

    }
}
leaf lan-delay-neg {
    description
        "Negotiated delay of the messages transmitted
        on an interface.";
    config "false";
    type uint16 {
        range "1..32767";
    }
}
leaf hello-over-interval-neg {
    description
        "Negotiated overriding interval on an interface.";
    config "false";
    type uint16 {
        range "1..65535";
    }
}
leaf gen-id {
    description "Generation id on an interface.";
    config "false";
}

```

```

        type uint32 {
            range "0..4294967295";
        }
    }
    leaf non-dr-neighbor-num {
        description "Number of non-dr neighbors.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf non-lan-delay-neighbor-num {
        description
            "Number of neighbors that do not use lan delay.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf non-join-attribute-neighbor-num {
        description
            "Number of neighbors that do not use the
            join attribute.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }

```

```

    }
}
leaf join-attribute {
    description
        "Negotiated join attribute of the messages
        transmitted on an interface.";
    config "false";
    type boolean;
}
leaf ipsec-name {
    description
        "Description of an sa. The value is a string
        of 1 to 15 characters.";
    config "false";
    type string {

```



```

        length "1..15";
    }
}
leaf ipsec-type {
    description
        "Hello options based on the ipsec type.";
    config "false";
    type enumeration {
        enum forall {
            value "0";
            description "Ipsec sa for all:";
        }
        enum forhello {
            value "1";
            description "Ipsec sa for hello:";
        }
    }
}
}
}
}

```

```

container pim-nbr-infos {

    list pim-nbr-info {

        key "vrf-name address-family if-name";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
        }
    }
}

```

```

        //default "_public_";
        type string {
            length "0..32";
        }
    }
    leaf address-family {
        config "false";
        type enumeration {

```

```

        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf if-name {
    description "Interface name.";
    config "false";
    type if-name;
}
leaf ipv4-nbr-addr {
    description "Address of a pim neighbor.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-nbr-addr {
    description "Address of a pim neighbor.";
    config "false";
    type inet:ipv6-address;
}
leaf up-time {
    description
        "Time since the pim neighbor relationship
        was established.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf expire-time {
    description
        "Time when a pim neighbor relationship expires.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}

```

}

```

}
leaf dr-priority-present {
    description "Whether a dr priority exists.";
    config "false";
    type boolean;
}
leaf dr-priority {
    description "Dr priority.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf gen-id-present {
    description "Whether a generation id exists.";
    config "false";
    type boolean;
}
leaf gen-id {
    description
        "Random number of the pim neighbor status.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf holdtime {
    description
        "Ttl of a pim neighbor relationship. The value
        65535 indicates that the pim neighbor
        relationship never times out.";
    config "false";
    type uint16 {
        range "1..65535";
    }
}
leaf lan-delay-present {
    description
        "Whether a lan delay for transmitting prune
        messages exists.";
    config "false";
    type boolean;
}
leaf lan-delay {
    description
        "Lan delay for transmitting prune messages.";
    config "false";

```

```
        type uint16 {
            range "1..32767";
        }
    }
    leaf override-interval {
        description
            "Interval for overriding the prune action.";
        config "false";
        type uint16 {
            range "1..65535";
        }
    }
    leaf join-attribute {
        description
            "Whether the join attribute function is
            enabled on a pim neighbor.";
        config "false";
        type boolean;
    }
    container pim-second-addrs {

        list pim-second-addr {

            description
                "List of secondary address information.";
            key "second-addr";
            config "false";

            leaf second-addr {
                description
                    "Secondary address of a pim neighbor.";
                config "false";
                type inet:ipv6-address;
            }

        }

    }

}

container pim-routes {

    list pim-route {
```

key "vrf-name address-family";

```
config "false";

leaf vrf-name {
    description "Name of a vpn instance.";
    config "false";
    //default "_public_";
    type string {
        length "0..32";
    }
}
leaf address-family {
    config "false";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf ipv4-source-addr {
    description "Multicast source address.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-source-addr {
    description "Multicast source address.";
    config "false";
    type inet:ipv6-address;
}
leaf ipv4-group-addr {
    description
        "Multicast group address. In ipv4 address,
        the value ranges from 224.0.0.0 to
        239.255.255.255. ";
    config "false";
```

```

        type inet:ipv4-address;
    }
    leaf ipv6-group-addr {
        description
            "Multicast group address. In ipv6, the value
             is in hexadecimal notation and in the format
             of ffx:xxxx:xxxx::xxxx ";
        config "false";
        type inet:ipv6-address;
    }

```

```

    }
    leaf sg-up-protocol-type {
        description "Pim protocol type.";
        config "false";
        type enumeration {
            enum none {
                value "0";
                description "none:none";
            }
            enum asm {
                value "1";
                description "asm:asm";
            }
            enum ssm {
                value "2";
                description "ssm:ssm";
            }
            enum bidir {
                value "3";
                description "bidir:bidir";
            }
            enum other {
                value "4";
                description "other:other";
            }
        }
    }
    leaf rp-mode-type {
        description "Rp type.";
        config "false";
        type enumeration {
            enum fixed {

```

```

        value "0";
        description "fixed:fixed";
    }
    enum static {
        value "1";
        description "static:static";
    }
    enum configssm {
        value "2";
        description "config-ssm:config-ssm";
    }
    enum bsr {
        value "3";
        description "bsr:bsr";
    }
    enum autorp {

```

```

        value "4";
        description "autorp:autorp";
    }
    enum embedded {
        value "5";
        description "embedded:embedded";
    }
    enum other {
        value "6";
        description "other:other";
    }
}
}
leaf ipv4-rp-addr {
    description "Rp address.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-rp-addr {
    description "Rp address.";
    config "false";
    type inet:ipv6-address;
}
leaf rp-addr-is-local {
    description

```

```

        "Whether the rp address is a local address.";
        config "false";
        type boolean;
    }
    leaf sg-flag {
        description "Flag of the pim (s, g) entry.";
        config "false";
        type string {
            length "1..50";
        }
    }
    leaf sg-up-time {
        description
            "Time since the pim (s, g) entry was generated.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf sg-iif-name {
        description
            "Inbound interface of the (s, g) entry.";
        config "false";
    }

```

```

        type string {
            length "0..48";
        }
    }
    leaf sg-vector-rd {
        description "Rd of the rpf vector.";
        config "false";
        type string {
            length "1..255";
        }
    }
    leaf ipv4-sg-vector-addr {
        description "Address of the rpf vector.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-sg-vector-addr {
        description "Address of the rpf vector.";
    }

```



```

        config "false";
        type inet:ipv6-address;
    }
    leaf sg-elect-vector-rd {
        description "Rd of the elected vector.";
        config "false";
        type string {
            length "1..255";
        }
    }
    leaf ipv4-sg-elect-vector-addr {
        description "Address of elected vector.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-sg-elect-vector-addr {
        description "Address of elected vector.";
        config "false";
        type inet:ipv6-address;
    }
    leaf ipv4-sg-up-nbr-addr {
        description
            "Upstream neighbor address of the (s, g) entry.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-sg-up-nbr-addr {
        description
            "Upstream neighbor address of the (s, g) entry.";
        config "false";
    }

```

```

        type inet:ipv6-address;
    }
    leaf ipv4-sg-rpf-addr {
        description "Rpf address of the (s, g) entry.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-sg-rpf-addr {
        description "Rpf address of the (s, g) entry.";
        config "false";
        type inet:ipv6-address;
    }

```

```

}
leaf ipv4-sg-refer-addr {
    description "Reference address of the (s,g) entry.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-sg-refer-addr {
    description "Reference address of the (s,g) entry.";
    config "false";
    type inet:ipv6-address;
}
leaf sg-rpf-masklen {
    description "Mask length of an rpf route.";
    config "false";
    type uint8 {
        range "0..32";
    }
}
leaf sg-rpf-metric {
    description "Metric of an rpf route.";
    config "false";
    type uint16 {
        range "0..65535";
    }
}
leaf sg-rpf-metric-preference {
    description "Metric priority of an rpf route.";
    config "false";
    type uint16 {
        range "0..65535";
    }
}
leaf backup-sg-iif-name {
    description "backup inbound interface of the (s, g) ent
    config "false";
    type string {
        length "1..48";

```

```

    }
}
leaf ipv4-backup-sg-up-nbr-addr {
    description

```

```

        "Backup upstream neighbor address of the
        (s, g) entry.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-backup-sg-up-nbr-addr {
    description
        "Backup upstream neighbor address of the
        (s, g) entry.";
    config "false";
    type inet:ipv6-address;
}
leaf ipv4-backup-sg-rpf-addr {
    description
        "Backup rpf address of the (s, g) entry.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-backup-sg-rpf-addr {
    description
        "Backup rpf address of the (s, g) entry.";
    config "false";
    type inet:ipv6-address;
}
leaf sg-up-state {
    description
        "Upstream up status of the (s, g) entry.";
    config "false";
    type enumeration {
        enum notjoined {
            value "0";
            description "notjoined:notjoined";
        }
        enum joined {
            value "1";
            description "joined:joined";
        }
    }
}
leaf sg-up-exp-time {
    description
        "Upstream up expiration time of the (s, g) entry.";
    config "false";
    type uint32 {

```

```
        range "0..4294967295";
    }
}
leaf sg-up-rp-tstate {
    description "Rpt status.";
    config "false";
    type enumeration {
        enum notjoined {
            value "0";
            description "notjoined:notjoined";
        }
        enum pruned {
            value "1";
            description "pruned:pruned";
        }
        enum notpruned {
            value "2";
            description "notpruned:notpruned";
        }
        enum max {
            value "3";
            description "max:max";
        }
    }
}
leaf sg-up-rpt-overtime {
    description "Rpt timeout period.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf sg-reg-state {
    description "Registration status.";
    config "false";
    type enumeration {
        enum noinfo {
            value "0";
            description "noinfo:noinfo";
        }
        enum join {
            value "1";
            description "join:join";
        }
        enum joinpending {
            value "2";
            description "joinpending:joinpending";
        }
    }
}
```

```
}
```

```
        enum prune {
            value "3";
            description "prune:prune";
        }
    }
}
leaf sg-reg-stop-time {
    description "Time when the registration stops.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}

container pim-route-downstreams {

    list pim-route-downstream {

        key "sg-ds-oif-name";
        config "false";

        description "Downstream information.";

        leaf sg-ds-oif-name {
            description
                "Name of the downstream interface.";
            config "false";
            type string {
                length "1..48";
            }
        }
        leaf sg-ds-protocol-type {
            description
                "Type of pim protocol configured on
                the downstream interface.";
            config "false";
            type string {
                length "1..50";
            }
        }
    }
}
```

```

leaf sg-ds-up-time {
  description
    "Time since the interface was added
    to the downstream interface list.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}

```

```

}
leaf sg-ds-expire-time {
  description
    "Timeout period of the downstream
    interface.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf sg-ds-dr-state {
  description "Dr status.";
  config "false";
  type boolean;
}
leaf sg-ds-state {
  description "Downstream status.";
  config "false";
  type enumeration {
    enum noinfo {
      value "0";
      description "noinfo:noinfo";
    }
    enum joined {
      value "1";
      description "joined:joined";
    }
    enum prunepending {
      value "2";
      description "prunepending:prunepending";
    }
  }
}
}

```

```

leaf sg-ds-ppt-time {
    description
        "Timeout period of the downstream ppt.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf sg-ds-assert-state {
    description "Assert state machine.";
    config "false";
    type enumeration {
        enum noinfo {
            value "0";
            description "noinfo:noinfo";
        }
    }
}

```

```

    }
    enum winner {
        value "1";
        description "winner:winner";
    }
    enum loser {
        value "2";
        description "loser:loser";
    }
}
leaf sg-ds-assert-time {
    description "Assert timeout period.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf ipv4-sg-ds-assert-win-addr {
    description "Address of an assert winner.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-sg-ds-assert-win-addr {
    description "Address of an assert winner.";
    config "false";
}

```

```

        type inet:ipv6-address;
    }
    leaf sg-ds-assert-win-metric {
        description "Metric of an assert winner.";
        config "false";
        type uint16 {
            range "0..65535";
        }
    }
    leaf sg-ds-ast-win-metric-pre {
        description
            "Metric priority of an assert winner.";
        config "false";
        type uint16 {
            range "0..65535";
        }
    }
    leaf sg-ds-local-member {
        description "Whether local receivers exist.";
        config "false";
        type boolean;
    }
}

```

```

    leaf sg-ds-rpt-state {
        description "Rpt status.";
        config "false";
        type enumeration {
            enum noinfo {
                value "0";
                description "noinfo:noinfo";
            }
            enum pruned {
                value "1";
                description "pruned:pruned";
            }
            enum prunepending {
                value "2";
                description "prunepending:prunepending";
            }
            enum max {
                value "3";
                description "max:max";
            }
        }
    }
}

```



```

    }
  }
}
leaf sg-ds-rpt-ppt-time {
  description
    "Ppt time of the rpt state machine.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf sg-ds-rpt-expire-time {
  description "Rpt timeout period.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf sg-vector-num {
  description "Number of vectors.";
  config "false";
  type uint16 {
    range "0..65535";
  }
}
container pim-route-downstream-vectors {
  list pim-route-downstream-vector {

```

```

    leaf ipv4-neighbor-addr {
      description "Neighbor address.";
      config "false";
      type inet:ipv4-address;
    }
    leaf ipv6-neighbor-addr {
      description "Neighbor address.";
      config "false";
      type inet:ipv6-address;
    }
    leaf neighbor-vector-rd {
      description "Vector rds of neighbors.";

```

```

        config "false";
        type string {
            length "1..255";
        }
    }
    leaf ipv4-neighbor-vector-addr {
        description
            "Address of a neighbor's elected vect
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-neighbor-vector-addr {
        description
            "Address of a neighbor's elected vect
        config "false";
        type inet:ipv6-address;
    }
    leaf neighbor-expire-time {
        description
            "Timeout period of the vector.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
}

}

}

}

}

```

```

    }

    container pim-rpf-routes {

        list pim-rpf-route {

```

```

key "vrf-name address-family";
config "false";

leaf vrf-name {
    description "Name of a vpn instance.";
    config "false";
    //default "_public_";
    type string {
        length "0..32";
    }
}
leaf address-family {
    config "false";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf ipv4-dest-addr {
    description "Destination address.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-dest-addr {
    description "Destination address.";
    config "false";
    type inet:ipv6-address;
}
leaf if-name {
    description "Name of an rpf interface.";
    config "false";
    type if-name;
}
leaf ipv4-rpf-addr {
    description "Rpf neighbor.";
    config "false";
}

```

```

        type inet:ipv4-address;
    }
    leaf ipv6-rpf-addr {
        description "Rpf neighbor.";
        config "false";
        type inet:ipv6-address;
    }
    leaf ipv4-refer-route-addr {
        description "Referenced route.";
        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-refer-route-addr {
        description "Referenced route.";
        config "false";
        type inet:ipv6-address;
    }
    leaf ipv4-refer-route-masklen {
        description "Mask of the referenced route.";
        config "false";
        type uint8 {
            range "0..32";
        }
    }
    leaf ipv6-refer-route-masklen {
        description "Mask of the referenced route.";
        config "false";
        type uint8 {
            range "0..128";
        }
    }
    leaf refer-route-type {
        description "Type of the referenced route.";
        config "false";
        type enumeration {
            enum unicast {
                value "0";
                description "unicast:unicast";
            }
            enum direct {
                value "1";
                description "direct:direct";
            }
            enum mbgp {
                value "2";
                description "mbgp:mbgp";
            }
            enum multicaststatic {

```

Internet-Draft

PIM YANG Data Model

March 2015

```
        value "3";
        description "multicaststatic:multicaststatic";
    }
}
}
container pim-claim-route-sgs {
    list pim-claim-route-sg {
        description "Claim routing entry.";
        key "ipv4-source-addr ipv6-source-addr ipv4-group-a
        config "false";

        leaf ipv4-source-addr {
            description "Multicast source address.";
            config "false";
            type inet:ipv4-address;
        }
        leaf ipv6-source-addr {
            description "Multicast source address.";
            config "false";
            type inet:ipv6-address;
        }
        leaf ipv4-group-addr {
            description
                "Multicast group address. In ipv4
                address, the value ranges from
                224.0.0.0 to 239.255.255.255. ";
            config "false";
            type inet:ipv4-address;
        }
        leaf ipv6-group-addr {
            description
                "Multicast group address. In ipv6,
                the value is in hexadecimal notation
                and in the format of
                ffx:xxxx:xxxx::xxxx ";
            config "false";
            type inet:ipv6-address;
        }
    }
}
```

```

    }
  }
}

```

```

container pim-embedded-rp-infos {
  list pim-embedded-rp-info {
    key "vrf-name";
    config "false";

    leaf vrf-name {
      description "Name of a vpn instance.";
      config "false";
      //default "_public_";
      type string {
        length "0..32";
      }
    }
    leaf ipv6-group-addr {
      description
        "Multicast group address. In ipv6, the value
         is in hexadecimal notation and in the format
         of ffx:xxxx:xxxx::xxxx ";
      config "false";
      type inet:ipv6-address;
    }
    leaf embedded-rp-addr {
      description "Address of an embedded rp.";
      config "false";
      type inet:ipv6-address;
    }
  }
}

container pim-rp-group-infos {

```

```

list pim-rp-group-info {

    key "vrf-name address-family";
    config "false";

    leaf vrf-name {
        description "Name of a vpn instance.";
        config "false";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
}

```

```

leaf address-family {
    config "false";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf ipv4-group-addr {
    description
        "Multicast group address. In ipv4 address,
        the value ranges from 224.0.0.0 to
        239.255.255.255. ";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-group-addr {
    description
        "Multicast group address. In ipv6, the value
        is in hexadecimal notation and in the format
        of ffx:xxxx:xxxx::xxxx ";
    config "false";
    type inet:ipv6-address;
}

```

```

}
leaf ipv4-bsr-rp-addr {
    description "Address of a candidate rp.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-bsr-rp-addr {
    description "Address of a candidate rp.";
    config "false";
    type inet:ipv6-address;
}
leaf ipv4-static-rp-addr {
    description "Address of a static rp.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-static-rp-addr {
    description "Address of a static rp.";
    config "false";
    type inet:ipv6-address;
}

```

```

}
leaf embedded-rp-addr {
    description "Address of an embedded rp.";
    config "false";
    type inet:ipv6-address;
}
leaf ipv4-map-rp-addr {
    description "Address of the designated rp.";
    config "false";
    type inet:ipv4-address;
}
leaf ipv6-map-rp-addr {
    description "Address of the designated rp.";
    config "false";
    type inet:ipv6-address;
}
}
}

```



```

container pim-bfd-infos {

    list pim-bfd-info {

        description "Information about a pim bfd session.";
        key "vrf-name address-family if-name";
        config "false";

        leaf vrf-name {
            description "Name of a vpn instance.";
            config "false";
            //default "_public_";
            type string {
                length "0..32";
            }
        }
        leaf address-family {
            description
                "Address family, which determines whether an
                address belongs to ipv4 or ipv6.";
            config "false";
            type enumeration {
                enum ipv4unicast {
                    value "0";
                    description "ipv4unicast:ipv4unicast";
                }
                enum ipv6unicast {

```

```

                    value "1";
                    description "ipv6unicast:ipv6unicast";
                }
            }
        }
        leaf if-name {
            description
                "Name of the interface to which a pim
                neighbor relationship belongs.";
            config "false";
            type if-name;
        }
        leaf ipv4-neighbor-addr {
            description "Address of a neighbor.";

```

```

        config "false";
        type inet:ipv4-address;
    }
    leaf ipv6-neighbor-addr {
        description "Address of a neighbor.";
        config "false";
        type inet:ipv6-address;
    }
    leaf act-tx {
        description
            "Actual minimum interval at which bfd packets
            are sent, in milliseconds.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf act-rx {
        description
            "Actual minimum interval at which bfd packets
            are received, in milliseconds.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf act-multi {
        description "Actual local detection multiplier.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf local-port {

```

```

        description
            "Local discriminator of a bfd session.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
}

```

```

    leaf remote-port {
      description
        "Remote discriminator of a bfd session.";
      config "false";
      type uint32 {
        range "0..4294967295";
      }
    }
  }
  leaf bfd-status {
    description "Status of a pim bfd session.";
    config "false";
    type enumeration {
      enum admin-down {
        value "0";
        description "admin down:admin down";
      }
      enum down {
        value "1";
        description "down:down";
      }
      enum init {
        value "2";
        description "init:init";
      }
      enum up {
        value "3";
        description "up:up";
      }
      enum bfd_session_number_exceed {
        value "4";
        description "bfd session number exceed:bfd sess
      }
    }
  }
}

container pim-vrf-packet-statistics {

  list pim-vrf-packet-statistic {

```

```

key "vrf-name address-family";
config "false";

leaf vrf-name {
    description "Name of a vpn instance.";
    config "false";
    //default "_public_";
    type string {
        length "0..32";
    }
}
leaf address-family {
    config "false";
    type enumeration {
        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf register-recv-num {
    description
        "Number of register messages received.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf register-send-num {
    description
        "Number of register messages sent.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf register-invalid-num {
    description
        "Number of invalid register messages received.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}

```

Internet-Draft

PIM YANG Data Model

March 2015

```
leaf register-filter-num {
  description
    "Number of register messages filtered.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf register-stop-recv-num {
  description
    "Number of register-stop messages received.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf register-stop-send-num {
  description
    "Number of register-stop messages sent.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf register-stop-invalid-num {
  description
    "Number of invalid register-stop messages
    received.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf register-stop-filter-num {
  description
    "Number of register-stop messages filtered.";
  config "false";
  type uint32 {
    range "0..4294967295";
  }
}
leaf probe-recv-num {
  description
```

```

        "Number of null register messages received.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }

```

```

    }
    leaf probe-send-num {
        description
            "Number of the null register messages sent.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf probe-invalid-num {
        description
            "Number of the invalid null register
            messages.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf probe-filter-num {
        description
            "Number of null register messages filtered.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf crp-recv-num {
        description
            "Number of c-rp messages received.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf crp-send-num {
        description

```

```

        "Number of c-rp messages sent.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf crp-invalid-num {
    description
        "Number of invalid c-rp messages.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}

```

```

    }
}
leaf crp-filter-num {
    description
        "Number of c-rp messages filtered.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
container pim-if-packet-statistics {
    list pim-if-packet-statistic {
        key "if-name";
        config "false";

        leaf if-name {
            description "Name of the interface.";
            config "false";
            type if-name;
        }
        leaf assert-recv-num {
            description
                "Number of assert messages received
                on the current interface.";
            config "false";
            type uint32 {
                range "0..4294967295";
            }
        }
    }
}

```

```

    }
}
leaf assert-send-num {
    description
        "Number of assert messages sent on
        the current interface.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf assert-invalid-num {
    description
        "Number of invalid assert messages.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
}

```

```

leaf assert-filter-num {
    description
        "Number of assert messages filtered
        on the current interface.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf hello-recv-num {
    description
        "Number of hello messages received
        on the current interface.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}
leaf hello-send-num {
    description
        "Number of hello messages sent on
        the current interface.";
}

```



```

        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf hello-invalid-num {
        description
            "Number of invalid hello messages.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf hello-filter-num {
        description
            "Number of hello messages filtered
            on the current interface.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf jp-recv-num {
        description
            "Number of join/prune messages
            received on the current interface.";
    }

```

```

        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf jp-send-num {
        description
            "Number of join/prune messages sent
            on the current interface.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf jp-invalid-num {

```

```

        description
            "Number of invalid join/prune
            messages.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf jp-filter-num {
        description
            "Number of join/prune messages
            filtered on the current interface.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf bsr-recv-num {
        description
            "Number of bsr messages received on
            the current interface.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }
    leaf bsr-send-num {
        description
            "Number of bsr messages sent on the
            current interface.";
        config "false";
        type uint32 {
            range "0..4294967295";
        }
    }

```

```

    }
}
leaf bsr-invalid-num {
    description
        "Number of invalid bsr messages.";
    config "false";
    type uint32 {
        range "0..4294967295";
    }
}

```



```

        value "0";
        description "ipv4unicast:ipv4unicast";
    }
    enum ipv6unicast {
        value "1";
        description "ipv6unicast:ipv6unicast";
    }
}
}
leaf if-name {
    description "Name of the interface.";
    type if-name;
}
leaf ipv4-neighbor-addr {
    description "pim neighbor address.";
    type inet:ipv4-address;
}
leaf ipv6-neighbor-addr {
    description "pim neighbor address.";
    type inet:ipv6-address;
}
leaf neighbor-up-time {
    description "pim neighbor up time.";
    type uint32 {
        range "0..4294967295";
    }
}
}
}

```

```

notification pim-invalid-register {

```

```

    description
        "A pim-invalid-register notification signifies that an invalid
        pim register message was received by this device. ";

    leaf vrf-name {
        description "Name of a vpn instance.";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
    leaf address-family {
        type enumeration {
            enum ipv4unicast {
                value "0";
                description "ipv4unicast:ipv4unicast";
            }

```

```
    }
    enum ipv6unicast {
        value "1";
        description "ipv6unicast:ipv6unicast";
    }
}
leaf ipv4-invalid-register-origin {
    description
        "The source address of the last invalid register message.";
    type inet:ipv4-address;
}
leaf ipv6-invalid-register-origin {
    description
        "The source address of the last invalid register message.";
    type inet:ipv6-address;
}
leaf ipv4-invalid-register-group {
    description
        "The multicast group address of the last invalid register
        message.";
    type inet:ipv4-address;
}
leaf ipv6-invalid-register-group {
    description
        "The multicast group address of the last invalid register
        message.";
    type inet:ipv6-address;
}
leaf ipv4-invalid-register-rp {
    description
        "The rp address of the last invalid register message.";
    type inet:ipv4-address;
}
leaf ipv6-invalid-register-rp {
    description
        "The rp address of the last invalid register message.";
    type inet:ipv6-address;
}
}

notification pim-invalid-join-prune {

    description
        "A pim-invalid-join-prune notification signifies that an
        invalid pim join/prune message was received by this device.";
```

```
leaf vrf-name {
```

```
    description "Name of a vpn instance.";
    //default "_public_";
    type string {
        length "0..32";
    }
}
leaf address-family {
    type enumeration {
        enum ipv4unicast {
            value "0";
            description "ipv4unicast:ipv4unicast";
        }
        enum ipv6unicast {
            value "1";
            description "ipv6unicast:ipv6unicast";
        }
    }
}
leaf ipv4-invalid-join-prune-origin {
    description
        "The source address of the last invalid join/prune
        message.";
    type inet:ipv4-address;
}
leaf ipv6-invalid-join-prune-origin {
    description
        "The source address of the last invalid join/prune
        message.";
    type inet:ipv6-address;
}
leaf ipv4-invalid-join-prune-group {
    description
        "The multicast group address of the last invalid
        join/prune message.";
    type inet:ipv4-address;
}
leaf ipv6-invalid-join-prune-group {
    description
        "The multicast group address of the last invalid
```

```

        join/prune message.";
        type inet:ipv6-address;
    }
    leaf ipv4-invalid-join-prune-rp {
        description
            "The rp address of the last invalid join/prune message.";
        type inet:ipv4-address;
    }
    leaf ipv6-invalid-join-prune-rp {

```

```

        description
            "The rp address of the last invalid join/prune message.";
        type inet:ipv6-address;
    }
    leaf neighbor-up-time {
        description "pim neighbor up time.";
        type uint32 {
            range "0..4294967295";
        }
    }
}

notification pim-rp-mapping-change {

    description
        "A pim-rp-mapping-change notification signifies a change to
        the active rp mapping on this device.";

    leaf vrf-name {
        description "Name of a vpn instance.";
        //default "_public_";
        type string {
            length "0..32";
        }
    }
    leaf address-family {
        type enumeration {
            enum ipv4unicast {
                value "0";
                description "ipv4unicast:ipv4unicast";
            }
            enum ipv6unicast {

```

```

        value "1";
        description "ipv6unicast:ipv6unicast";
    }
}
leaf group-mapping-origin {
    description
        "The mechanism by which this group mapping was learned.";
    type enumeration {
        enum fixed {
            value "1";
            description "fixed:fixed";
        }
        enum staticrp {
            value "2";
            description " staticrp:staticrp";
        }
    }
}

```

```

    }
    enum ssm {
        value "3";
        description " ssm:ssm";
    }
    enum bsr {
        value "4";
        description " bsr:bsr";
    }
    enum autorp {
        value "5";
        description " autorp:autorp";
    }
    enum embeddedrp {
        value "6";
        description " embeddedrp:embeddedrp";
    }
    enum other {
        value "7";
        description " other:other";
    }
}
leaf ipv4-group-mapping-group-address {
    description "Multicast group address for group mapping.";
}

```



```

        type inet:ipv4-address;
    }
    leaf ipv6-group-mapping-group-address {
        description "Multicast group address for group mapping.";
        type inet:ipv6-address;
    }
    leaf group-mapping-group-prefix-length {
        description
            "Multicast group prefix length. For ipv4, it is in the
            range 4-32. for ipv6, it is in the range 8-128.";
        type uint8 {
            range "4..128";
        }
    }
    leaf ipv4-group-mapping-rp-address {
        description "The rp address for group mapping.";
        type inet:ipv4-address;
    }
    leaf ipv6-group-mapping-rp-address {
        description "The rp address for group mapping.";
        type inet:ipv6-address;
    }
    leaf group-mapping-precedence {

```

```

        description
            "Numerically higher values indicate lower precedences,
            with the value zero denoting the highest precedence.";
        type uint32 {
            range "0..4294967295";
        }
    }
}

notification pim-interface-election {

    description
        "A pim-interface-election notification signifies that a new
        dr or df has been elected on a network. ";

    leaf vrf-name {
        description "Name of a vpn instance.";
        //default "_public_";
    }
}

```

```

        type string {
            length "0..32";
        }
    }
    leaf address-family {
        type enumeration {
            enum ipv4unicast {
                value "0";
                description "ipv4unicast:ipv4unicast";
            }
            enum ipv6unicast {
                value "1";
                description "ipv6unicast:ipv6unicast";
            }
        }
    }
    leaf if-name {
        description "Name of the interface.";
        type if-name;
    }
    leaf ipv4-if-addr {
        description "The address of the interface.";
        type inet:ipv4-address;
    }
    leaf ipv6-if-addr {
        description "The address of the interface.";
        type inet:ipv6-address;
    }
}

```

}

## [6.](#) IANA Considerations

This draft includes no request to IANA.

## [7.](#) Security Considerations

The data model defined does not create any security implications. This draft does not change any underlying security issues inherent in [\[I-D.ietf-netmod-routing-cfg\]](#).

## [8.](#) Acknowledgements

TBD

## [9.](#) References

### [9.1.](#) Normative References

- [I-D.ietf-netmod-routing-cfg]  
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", [draft-ietf-netmod-routing-cfg-17](#) (work in progress), March 2015.
- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast – Dense Mode (PIM-DM): Protocol Specification (Revised)", [RFC 3973](#), January 2005.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)", [RFC 4601](#), August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", [RFC 4607](#), August 2006.
- [RFC4608] Meyer, D., Rockell, R., and G. Shepherd, "Source-Specific Protocol Independent Multicast in 232/8", [BCP 120](#), [RFC 4608](#), August 2006.
- [RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", [RFC 4610](#), August 2006.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", [RFC 5015](#), October 2007.

- [RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", [RFC 5059](#), January 2008.

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.

## [9.2.](#) Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

### Authors' Addresses

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

Email: [liuyisong@huawei.com](mailto:liuyisong@huawei.com)

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

Email: [guofeng@huawei.com](mailto:guofeng@huawei.com)

Mahesh Sivakumar  
Cisco  
510 McCarthy Blvd  
Milpitas, California 95035  
USA

Email: [masivaku@cisco.com](mailto:masivaku@cisco.com)