PIM Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 11, 2018

X. Liu
Jabil
P. McAllister
Metaswitch Networks
A. Peter
Individual
M. Sivakumar
Cisco Systems
Y. Liu
Huawei Technologies
F. Hu
ZTE Corporation

December 8, 2017

A YANG data model for Protocol-Independent Multicast (PIM)
draft-ietf-pim-yang-12

Abstract

This document defines a YANG data model that can be used to configure and manage Protocol Independent Multicast (PIM) devices. The model covers the PIM protocol configuration, operational state, and event notifications data.

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 June 11, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction 2
  1.1. Terminology 3
  1.2. Prefixes in Data Node Names 4
2. Design of Data Model 5
  2.1. Scope of model 5
  2.2. Optional capabilities 5
  2.3. Datastore applicability 6
  2.4. Top-level structure 6
  2.5. Position of address family in hierarchy 7
3. Module Structure 7
  3.1. PIM base module 7
  3.2. PIM RP module 11
  3.3. PIM-SM module 13
  3.4. PIM-DM module 14
  3.5. PIM-BIDIR module 15
4. PIM YANG Modules 16
  4.1. PIM base module 16
  4.2. PIM RP module 34
  4.3. PIM-SM module 49
  4.4. PIM-DM module 54
  4.5. PIM-BIDIR module 56
5. Implementation Status 63
6. Security Considerations 64
7. IANA Considerations 64
8. Acknowledgements 66
9. References 66
  9.1. Normative References 66
  9.2. Informative References 68
Authors' Addresses 69

1. Introduction

YANG [RFC6020] [RFC7950] is a data definition language that was introduced to model the configuration and running state of a device managed using network management protocols such as NETCONF [RFC6241].

YANG is now also being used as a component of wider management interfaces, such as CLIs.

This document defines a YANG data model that can be used to configure and manage Protocol-Independent Multicast (PIM) devices. This model supports the core PIM protocol, as well as many other features described in Section 2.1. Non-core features are defined as optional in the provided data model.

1.1. Terminology

The terminology for describing YANG data models is found in [RFC6020] and [RFC7950].

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is defined in [I-D.ietf-netmod-yang-tree-diagrams].

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

ASM:
Any-Source Multicast service model [RFC3569] [RFC4607].

BFD:
Bidirectional Forwarding Detection [RFC5880].

BSR:
Bootstrap Router [RFC5059].

DF:
Designated Forwarder [RFC5015].

DR:
Designated Router [RFC7761].

IGMP:
Internet Group Management Protocol [RFC3376].

MLD:
Multicast Listener Discovery [RFC3810].

MSDP:
Multicast Source Discovery Protocol [RFC3618].

mLDP:
Multipoint extensions for LDP [RFC6388].

MRIB:
Multicast Routing Information Base [RFC3973] [RFC5015] [RFC7761].

mVPN:
Multicast VPN.

PIM:
Protocol Independent Multicast. [RFC3973] [RFC5015] [RFC7761].

PIM-BIDIR:
Protocol Independent Multicast - Bidirectional Mode [RFC5015].

PIM-DM:
Protocol Independent Multicast - Dense Mode [RFC3973].

PIM-SM:
Protocol Independent Multicast - Sparse Mode [RFC7761].

RP:
Rendezvous Point. [RFC7761].

RPA:
Rendezvous Point Address. [RFC5015].

RPF:
Reverse Path Forwarding. [RFC3973] [RFC5015] [RFC7761].

RPT:
Rendezvous-Point Tree. [RFC7761].

SPT:
Shortest Path Tree. [RFC7761].

SSM:
Source-Specific Multicast service model [RFC3569] [RFC4607].

VRF:
Virtual Routing and Forwarding.

1.2. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
if	ietf-interfaces	[I-D.ietf-netmod-rfc7223bis]
rt	ietf-routing	[I-D.ietf-netmod-rfc8022bis]
rt-types	ietf-routing-types	[I-D.ietf-rtgwg-routing-types]
bfd-types	ietf-bfd-types	[I-D.ietf-bfd-yang]

Table 1: Prefixes and Corresponding YANG Modules

2. Design of Data Model

2.1. Scope of model

The model covers PIM Sparse Mode [RFC7761], including the Source-Specific subset [RFC3569] [RFC4607], Dense Mode [RFC3973], and Bi-directional PIM [RFC5015].

The PIM extensions represented in the model include BSR [RFC5059] and Anycast-RP [RFC4610].

The data model can be used to configure and manage these protocol features. The operational state data and statistics can be retrieved by this model. The protocol specific notifications are also defined in the model.

This model does not cover other multicast protocols such as IGMP/MLD, MSDP, mVPN, or mLDP in-band signalling. It does not cover any configuration required to generate the MRIB. These will be specified in separate documents.

2.2. Optional capabilities

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

There is also value in widely-supported features being standardized, to save work for individual vendors, and so that mapping between different vendors' configuration is not needlessly complicated.

Therefore, these modules declare a number of features representing capabilities that not all deployed devices support.

The extensive use of feature declarations should also substantially simplify the capability negotiation process for a vendor's PIM implementation.

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

For the same reason, wide constant ranges (for example, timer maxima and minima) will be used in the model. It is expected that vendors will augment the model with any specific extensions and restrictions needed to adapt it to their vendor specific implementation.

2.3. Datastore applicability

This model conforms to the Network Management Datastore Architecture (NMDA) [I-D.ietf-netmod-revised-datastores]. The operational state data is combined with the associated configuration data in the same hierarchy [I-D.ietf-netmod-rfc6087bis].

2.4. Top-level structure

This model defines several separate modules for modelling PIM configuration, defined below. Again, this separation will make it easier to express the specific capabilities of a PIM device.

The hierarchy of PIM configuration is designed so that objects that are only relevant for one situation or feature are collected in a container for that feature. For example, the configuration for PIM-SM that is not relevant for an SSM-only implementation is collected in an ASM container.

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

This module structure also applies, where applicable, to the operational state and notifications of the model.

2.5. Position of address family in hierarchy

This document contains address-family as a node in the hierarchy multiple times: both under the interface list, and under the PIM instance.

The reasoning for this is to make it easier for implementations in which configuration options are not supported for specific address families.

For these implementations, the restriction that interface configuration must be address-family independent may either be expressed as a vendor augmentation of an address-family-independent parameter above the address-family level, or by a constraint on the base model objects of a form similar to:

```
must ". = ../../address-family[address-family='ipv4']/dr-priority" {
  error-app-tag "dr-priority-mismatch";
  error-message
    "Error: IPv6 DR priority must match IPv4 DR priority.";
}
```

3. Module Structure

3.1. PIM base module

The PIM base module defines the router-wide configuration options not specific to any PIM mode, and is included by the other modules. There are a couple of things worth mentioning here regarding where the PIM model fits in the overall routing hierarchy [I-D.ietf-netmod-rfc8022bis]:

1. This data model agrees to a routing-instance-centric (VRF) model view as opposed to protocol-centric mainly because it fits well into the routing-instance model, and it is easier to map from the VRF-centric to the protocol-centric than the other way around due to forward references.

2. The PIM base model augments "/rt:routing/rt:control-plane-protocols" as opposed to augmenting "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol", as the latter would allow multiple protocol instances per VRF, while the PIM protocol is designed to be enabled or disabled on the per VRF basis.

```
module: ietf-pim-base
  augment /rt:routing/rt:control-plane-protocols:
    +--rw pim!
       +--rw graceful-restart
```

```
          |  +--rw enabled?    boolean
          |  +--rw duration?   uint16
          +--rw address-family* [address-family]
          |  +--rw address-family        identityref
          |  +--rw graceful-restart
          |  |  +--rw enabled?    boolean
          |  |  +--rw duration?   uint16
          |  +--ro statistics
          |  |  +--ro discontinuity-time?   yang:date-and-time
          |  |  +--ro error
          |  |  |  +--ro assert?                      yang:counter64
          |  |  |  +--ro bsr?                         yang:counter64
          |  |  |  +--ro candidate-rp-advertisement?  yang:counter64
          |  |  |  +--ro df-election?                 yang:counter64
          |  |  |  +--ro hello?                       yang:counter64
          |  |  |  +--ro join-prune?                  yang:counter64
          |  |  |  +--ro register?                    yang:counter64
          |  |  |  +--ro register-stop?               yang:counter64
          |  |  |  +--ro state-refresh?               yang:counter64
          |  |  +--ro queue
          |  |  |  +--ro size?       uint32
          |  |  |  +--ro overflow?   yang:counter32
          |  |  +--ro received
          |  |  |  +--ro assert?                      yang:counter64
          |  |  |  +--ro bsr?                         yang:counter64
          |  |  |  +--ro candidate-rp-advertisement?  yang:counter64
          |  |  |  +--ro df-election?                 yang:counter64
          |  |  |  +--ro hello?                       yang:counter64
          |  |  |  +--ro join-prune?                  yang:counter64
          |  |  |  +--ro register?                    yang:counter64
          |  |  |  +--ro register-stop?               yang:counter64
          |  |  |  +--ro state-refresh?               yang:counter64
          |  |  +--ro sent
          |  |     +--ro assert?                      yang:counter64
          |  |     +--ro bsr?                         yang:counter64
          |  |     +--ro candidate-rp-advertisement?  yang:counter64
          |  |     +--ro df-election?                 yang:counter64
          |  |     +--ro hello?                       yang:counter64
          |  |     +--ro join-prune?                  yang:counter64
          |  |     +--ro register?                    yang:counter64
          |  |     +--ro register-stop?               yang:counter64
          |  |     +--ro state-refresh?               yang:counter64
          |  +--ro topology-tree-info
          |     +--ro ipv4-route* [group source-address is-rpt]
          |     |  +--ro group
    rt-types:ipv4-multicast-group-address
          |     |  +--ro source-address
    rt-types:ipv4-multicast-source-address
```

```
          |     |  +--ro is-rpt                boolean
          |     |  +--ro expiration?           timer-value
          |     |  +--ro incoming-interface?   if:interface-ref
          |     |  +--ro mode?                 pim-mode
          |     |  +--ro msdp-learned?         boolean
          |     |  +--ro rp-address?           inet:ip-address
          |     |  +--ro rpf-neighbor?         inet:ip-address
          |     |  +--ro spt-bit?              boolean
          |     |  +--ro up-time?              uint32
          |     |  +--ro outgoing-interface* [name]
          |     |     +--ro name          if:interface-ref
          |     |     +--ro expiration?   timer-value
          |     |     +--ro up-time?      uint32
          |     |     +--ro jp-state?     enumeration
          |     +--ro ipv6-route* [group source-address is-rpt]
          |        +--ro group
    rt-types:ipv6-multicast-group-address
          |        +--ro source-address
    rt-types:ipv6-multicast-source-address
          |        +--ro is-rpt                boolean
          |        +--ro expiration?           timer-value
          |        +--ro incoming-interface?   if:interface-ref
          |        +--ro mode?                 pim-mode
          |        +--ro msdp-learned?         boolean
          |        +--ro rp-address?           inet:ip-address
          |        +--ro rpf-neighbor?         inet:ip-address
          |        +--ro spt-bit?              boolean
          |        +--ro up-time?              uint32
          |        +--ro outgoing-interface* [name]
          |           +--ro name          if:interface-ref
          |           +--ro expiration?   timer-value
          |           +--ro up-time?      uint32
          |           +--ro jp-state?     enumeration
          +--rw interfaces
             +--rw interface* [interface]
                +--rw interface         if:interface-ref
                +--rw address-family* [address-family]
                   +--rw address-family       identityref
                   +--rw bfd {bfd}?
                   |  +--rw enable?                     boolean
                   |  +--rw local-multiplier?           multiplier
                   |  +--rw (interval-config-type)?
                   |     +--:(tx-rx-intervals)
                   |     |  +--rw desired-min-tx-interval     uint32
                   |     |  +--rw required-min-rx-interval    uint32
                   |     +--:(single-interval)
                   |        +--rw min-interval                uint32
                   +--rw dr-priority?         uint32 {intf-dr-priority}?
```

```
                  +--rw hello-interval?      timer-value
    {intf-hello-interval}?
                  +--rw (hello-holdtime-or-multiplier)?
                  |  +--:(holdtime) {intf-hello-holdtime}?
                  |  |  +--rw hello-holdtime?      timer-value
                  |  +--:(multiplier) {intf-hello-multiplier}?
                  |     +--rw hello-multiplier?
    rt-types:timer-multiplier
                  +--rw jp-interval?         timer-value
    {intf-jp-interval}?
                  +--rw (jp-holdtime-or-multiplier)?
                  |  +--:(holdtime) {intf-jp-holdtime}?
                  |  |  +--rw jp-holdtime?         timer-value
                  |  +--:(multiplier) {intf-jp-multiplier}?
                  |     +--rw jp-multiplier?
    rt-types:timer-multiplier
                  +--rw propagation-delay?   uint16
    {intf-propagation-delay}?
                  +--rw override-interval?   uint16
    {intf-override-interval}?
                  +--ro ipv4
                  |  +--ro address*      inet:ipv4-address
                  |  +--ro dr-address?   inet:ipv4-address
                  +--ro ipv6
                  |  +--ro address*      inet:ipv6-address
                  |  +--ro dr-address?   inet:ipv6-address
                  +--ro oper-status?         enumeration
                  +--ro hello-expiration?    timer-value
                  +--ro neighbor-ipv4* [address]
                  |  +--ro address        inet:ipv4-address
                  |  +--ro bfd-status?    enumeration
                  |  +--ro expiration?    timer-value
                  |  +--ro dr-priority?   uint32
                  |  +--ro gen-id?        uint32
                  |  +--ro up-time?       uint32
                  +--ro neighbor-ipv6* [address]
                     +--ro address        inet:ipv6-address
                     +--ro bfd-status?    enumeration
                     +--ro expiration?    timer-value
                     +--ro dr-priority?   uint32
                     +--ro gen-id?        uint32
                     +--ro up-time?       uint32

     notifications:
       +---n pim-neighbor-event
       |  +--ro event-type?          neighbor-event-type
       |  +--ro interface-ref?       -> /rt:routing
   /control-plane-protocols/pim-base:pim/interfaces/interface/interface
```

```
       |  +--ro interface-af-ref?    -> /rt:routing
   /control-plane-protocols/pim-base:pim/interfaces
   /interface[pim-base:interface = current()/../interface-ref]
   /address-family/address-family
       |  +--ro neighbor-ipv4-ref?   -> /rt:routing
   /control-plane-protocols/pim-base:pim/interfaces
   /interface[pim-base:interface = current()/../interface-ref]
   /address-family[pim-base:address-family = current()/..
   /interface-af-ref]/neighbor-ipv4/address
       |  +--ro neighbor-ipv6-ref?   -> /rt:routing
   /control-plane-protocols/pim-base:pim/interfaces
   /interface[pim-base:interface = current()/../interface-ref]
   /address-family[pim-base:address-family = current()/..
   /interface-af-ref]/neighbor-ipv6/address
       |  +--ro up-time?             uint32
       +---n pim-interface-event
          +--ro event-type?      interface-event-type
          +--ro interface-ref?   -> /rt:routing/control-plane-protocols
   /pim-base:pim/interfaces/interface/interface
          +--ro ipv4
          |  +--ro address*      inet:ipv4-address
          |  +--ro dr-address?   inet:ipv4-address
          +--ro ipv6
             +--ro address*      inet:ipv6-address
             +--ro dr-address?   inet:ipv6-address
```

3.2. PIM RP module

The PIM RP module contains configuration information scoped to RPs or ranges of group addresses. This does not belong in the hierarchy under any PIM mode, but is augmented by the individual mode-specific modules as appropriate.

```
   module: ietf-pim-rp
     augment /rt:routing/rt:control-plane-protocols/pim-base:pim
   /pim-base:address-family:
       +--rw rp
          +--rw static-rp
          |  +--rw ipv4-rp* [ipv4-address]
          |  |  +--rw ipv4-address    inet:ipv4-address
          |  +--rw ipv6-rp* [ipv6-address]
          |     +--rw ipv6-address    inet:ipv6-address
          +--rw bsr {bsr}?
          |  +--rw bsr-candidate!
          |  |  +--rw (interface-or-address)?
          |  |  |  +--:(interface) {candidate-interface}?
          |  |  |  |  +--rw interface           if:interface-ref
```

```
          |  |  |  +--:(ipv4-address) {candidate-ipv4}?
          |  |  |  |  +--rw ipv4-address        inet:ipv4-address
          |  |  |  +--:(ipv6-address) {candidate-ipv6}?
          |  |  |     +--rw ipv6-address        inet:ipv6-address
          |  |  +--rw hash-mask-length    uint8
          |  |  +--rw priority            uint8
          |  +--rw rp-candidate-interface* [interface]
    {candidate-interface}?
          |  |  +--rw interface    if:interface-ref
          |  |  +--rw policy?      string
          |  |  +--rw mode?        identityref
          |  +--rw rp-candidate-ipv4-address* [ipv4-address]
    {candidate-ipv4}?
          |  |  +--rw ipv4-address    inet:ipv4-address
          |  |  +--rw policy?         string
          |  |  +--rw mode?           identityref
          |  +--rw rp-candidate-ipv6-address* [ipv6-address]
    {candidate-ipv6}?
          |  |  +--rw ipv6-address    inet:ipv6-address
          |  |  +--rw policy?         string
          |  |  +--rw mode?           identityref
          |  +--ro bsr
          |  |  +--ro address?            inet:ip-address
          |  |  +--ro hash-mask-length?   uint8
          |  |  +--ro priority?           uint8
          |  |  +--ro up-time?            uint32
          |  +--ro (election-state)? {bsr-election-state}?
          |  |  +--:(candidate)
          |  |  |  +--ro candidate-bsr-state?               enumeration
          |  |  +--:(non-candidate)
          |  |     +--ro non-candidate-bsr-state?           enumeration
          |  +--ro bsr-next-bootstrap?                uint16
          |  +--ro rp
          |  |  +--ro rp-address?     inet:ip-address
          |  |  +--ro group-policy?   string
          |  |  +--ro up-time?        uint32
          |  +--ro rp-candidate-next-advertisement?   uint16
          +--ro rp-list
          |  +--ro ipv4-rp* [ipv4-address mode]
          |  |  +--ro ipv4-address           inet:ipv4-address
          |  |  +--ro mode                   identityref
          |  |  +--ro info-source-address?   inet:ipv4-address
          |  |  +--ro info-source-type?      identityref
          |  |  +--ro up-time?               uint32
          |  |  +--ro expiration?            pim-base:timer-value
          |  +--ro ipv6-rp* [ipv6-address mode]
          |     +--ro ipv6-address           inet:ipv6-address
          |     +--ro mode                   identityref
```

```
          |     +--ro info-source-address?   inet:ipv6-address
          |     +--ro info-source-type?      identityref
          |     +--ro up-time?               uint32
          |     +--ro expiration?            pim-base:timer-value
          +--ro rp-mappings
             +--ro ipv4-rp* [group rp-address]
             |  +--ro group         inet:ipv4-prefix
             |  +--ro rp-address    inet:ipv4-address
             |  +--ro up-time?      uint32
             |  +--ro expiration?   pim-base:timer-value
             +--ro ipv6-rp* [group rp-address]
                +--ro group         inet:ipv6-prefix
                +--ro rp-address    inet:ipv6-address
                +--ro up-time?      uint32
                +--ro expiration?   pim-base:timer-value

     notifications:
       +---n pim-rp-event
          +--ro event-type?        rp-event-type
          +--ro instance-af-ref?   -> /rt:routing
   /control-plane-protocols/pim-base:pim/address-family/address-family
          +--ro group?             rt-types:ip-multicast-group-address
          +--ro rp-address?        inet:ip-address
          +--ro is-rpt?            boolean
          +--ro mode?              pim-base:pim-mode
          +--ro message-origin?    inet:ip-address
```

3.3. PIM-SM module

This module covers Sparse Mode configuration, including PIM-ASM and PIM-SSM.

```
module: ietf-pim-sm
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family:
    +--rw sm
       +--rw asm
       |  +--rw anycast-rp!
       |  |  +--rw ipv4
       |  |  |  +--rw ipv4-anycast-rp* [anycast-address rp-address]
       |  |  |     +--rw anycast-address    inet:ipv4-address
       |  |  |     +--rw rp-address         inet:ipv4-address
       |  |  +--rw ipv6
       |  |     +--rw ipv6-anycast-rp* [anycast-address rp-address]
       |  |        +--rw anycast-address    inet:ipv6-address
       |  |        +--rw rp-address         inet:ipv6-address
       |  +--rw spt-switch
       |     +--rw infinity! {spt-switch-infinity}?
       |        +--rw policy-name?   string {spt-switch-policy}?
       +--rw ssm!
          +--rw range-policy?   string
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:interfaces/pim-base:interface/pim-base:address-family:
    +--rw sm!
       +--rw passive?   empty
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/pim-rp:ipv4-rp:
    +--rw sm!
       +--rw policy-name?   string
       +--rw override?      boolean {static-rp-override}?
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/pim-rp:ipv6-rp:
    +--rw sm!
       +--rw policy-name?   string
       +--rw override?      boolean {static-rp-override}?
```

3.4. PIM-DM module

This module will cover Dense Mode configuration.

```
module: ietf-pim-dm
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family:
    +--rw dm!
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:interfaces/pim-base:interface/pim-base:address-family:
    +--rw dm!
```

3.5. PIM-BIDIR module

This module will cover Bidirectional PIM configuration.

```
module: ietf-pim-bidir
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family:
    +--rw bidir!
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:interfaces/pim-base:interface/pim-base:address-family:
    +--rw bidir!
       +--rw df-election {intf-df-election}?
          +--rw offer-interval?     pim-base:timer-value
          +--rw backoff-interval?   pim-base:timer-value
          +--rw offer-multiplier?   uint8
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/pim-rp:ipv4-rp:
    +--rw bidir!
       +--rw policy-name?   string
       +--rw override?      boolean {static-rp-override}?
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/pim-rp:ipv6-rp:
    +--rw bidir!
       +--rw policy-name?   string
       +--rw override?      boolean {static-rp-override}?
  augment /rt:routing/rt:control-plane-protocols/pim-base:pim
/pim-base:address-family/pim-rp:rp:
    +--ro bidir
       +--ro df-election
       |  +--ro ipv4-rp* [ipv4-address]
       |  |  +--ro ipv4-address    inet:ipv4-address
       |  +--ro ipv6-rp* [ipv6-address]
       |     +--ro ipv6-address    inet:ipv6-address
       +--ro interface-df-election
          +--ro ipv4-rp* [ipv4-address interface-name]
          |  +--ro ipv4-address       inet:ipv4-address
          |  +--ro interface-name     if:interface-ref
          |  +--ro df-address?        inet:ipv4-address
          |  +--ro interface-state?   identityref
          +--ro ipv6-rp* [ipv6-address interface-name]
             +--ro ipv6-address       inet:ipv6-address
             +--ro interface-name     if:interface-ref
             +--ro df-address?        inet:ipv6-address
             +--ro interface-state?   identityref
```

4. PIM YANG Modules

4.1. PIM base module

```
<CODE BEGINS> file "ietf-pim-base@2017-12-08.yang"
module ietf-pim-base {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pim-base";
  prefix pim-base;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

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

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-bfd-types {
    prefix "bfd-types";
  }

  organization
    "IETF PIM Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/pim/>
     WG List:  <mailto:pim@ietf.org>

     WG Chair: Stig Venaas
               <mailto:stig@venaas.com>

     WG Chair: Mike McBride
               <mailto:mmcbride7@gmail.com>

     Editor:   Xufeng Liu
```

```
                  <mailto:Xufeng_Liu@jabil.com>

         Editor:   Pete McAllister
                   <mailto:pete.mcallister@metaswitch.com>

         Editor:   Anish Peter
                   <mailto:anish.ietf@gmail.com>

         Editor:   Mahesh Sivakumar
                   <mailto:masivaku@cisco.com>

         Editor:   Yisong Liu
                   <mailto:liuyisong@huawei.com>

         Editor:   Fangwei Hu
                   <mailto:hu.fangwei@zte.com.cn>";

     description
       "The module defines a collection of YANG definitions common for
        all PIM (Protocol Independent Multicast) modes.";

     revision 2017-12-08 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }

     /*
      * Features
      */
     feature bfd {
       description
         "Support BFD (Bidirectional Forwarding Detection).";
     }

     feature global-graceful-restart {
       description
         "Global configuration for graceful restart support as per
          RFC5306.";
     }

     feature intf-dr-priority {
       description
         "Support configuration of interface DR (Designated Router)
          priority.";
     }
```

```
     feature intf-hello-holdtime {
       description
         "Support configuration of interface hello holdtime.";
     }

     feature intf-hello-interval {
       description
         "Support configuration of interface hello interval.";
     }

     feature intf-hello-multiplier {
       description
         "Support configuration of interface hello multiplier.";
     }

     feature intf-jp-interval {
       description
         "Support configuration of interface join prune interval.";
     }

     feature intf-jp-holdtime {
       description
         "Support configuration of interface join prune holdtime.";
     }

     feature intf-jp-multiplier {
       description
         "Support configuration of interface join prune multiplier.";
     }

     feature intf-propagation-delay {
       description
         "Support configuration of interface propagation delay.";
     }

     feature intf-override-interval {
       description
         "Support configuration of interface override interval.";
     }

     feature per-af-graceful-restart {
       description
         "Per address family configuration for graceful restart support
          as per RFC5306.";
     }

     /*
      * Typedefs
```

```
 */
typedef interface-event-type {
  type enumeration {
    enum up {
      description
        "Neighbor status changed to up.";
    }
    enum down {
      description
        "Neighbor status changed to down.";
    }
    enum new-dr {
      description
        "A new DR (Designated Router) was elected on the connected
         network.";
    }
    enum new-df {
      description
        "A new DF (Designated Forwarder) was elected on the
         connected network.";
    }
  }
  description "Operational status event type for notifications.";
}

typedef neighbor-event-type {
  type enumeration {
    enum up {
      description
        "Neighbor status changed to up.";
    }
    enum down {
      description
        "Neighbor status changed to down.";
    }
  }
  description "Operational status event type for notifications.";
}

typedef pim-mode {
  type enumeration {
    enum none {
      description
        "PIM is not operating.";
    }
    enum ssm {
      description
        "PIM operates in the Sparse Mode with Source-Specific
```

```
              Multicast (SSM).";
         }
         enum asm {
           description
             "PIM operates in the Sparse Mode with Any Source Multicast
              (ASM).";
         }
         enum bidir {
           description
             "PIM operates in the Bidirectional Mode.";
         }
         enum dm {
           description
             "PIM operates in the Dense Mode (DM).";
         }
         enum other {
           description
             "PIM operates in any other mode.";
         }
       }
       description
         "The PIM mode in which a group is operating.";
     }

     typedef timer-value {
       type rt-types:timer-value-seconds16;
       description "Timer value type.";
     } // timer-value

     /*
      * Identities
      */

     /*
      * Groupings
      */
     grouping global-attributes {
       description
         "A Grouping defining global configuration attributes.";
       uses graceful-restart-container {
         if-feature global-graceful-restart;
       }
     } // global-attributes

     grouping graceful-restart-container {
       description
         "A grouping defining a container of graceful restart
         attributes.";
```

```
        container graceful-restart {
          leaf enabled {
            type boolean;
            description
              "Enable or disable graceful restart.";
          }
          leaf duration {
            type uint16;
            units seconds;
            description
              "Maximum time for graceful restart to finish.";
          }
          description
            "Container of graceful restart attributes.";
        }
      } // graceful-restart-container

      grouping interface-config-attributes {
        description
          "A grouping defining interface attributes.";
        container bfd {
          if-feature bfd;
          description
            "BFD (Bidirectional Forwarding Detection) operation.";
          uses bfd-types:client-cfg-parms;
        }
        leaf dr-priority {
          if-feature intf-dr-priority;
          type uint32;
          description "DR (Designated Router) priority";
        }
        leaf hello-interval {
          if-feature intf-hello-interval;
          type timer-value;
          description "Hello interval";
        }
        choice hello-holdtime-or-multiplier {
          description "Use holdtime or multiplier";
          case holdtime {
            if-feature intf-hello-holdtime;
            leaf hello-holdtime {
              type timer-value;
              description "Hello holdtime";
            }
          }
          case multiplier {
            if-feature intf-hello-multiplier;
            leaf hello-multiplier {
```

```
              type rt-types:timer-multiplier;
              description
                "Hello multiplier is the number by which the hello
                 interval is multplied to obtain the hold time";
            }
          }
        }
        leaf jp-interval {
          if-feature intf-jp-interval;
          type timer-value;
          description "Join prune interval";
        }
        choice jp-holdtime-or-multiplier {
          description "Use holdtime or multiplier";
          case holdtime {
            if-feature intf-jp-holdtime;
            leaf jp-holdtime {
              type timer-value;
              description "Join prune holdtime";
            }
          }
          case multiplier {
            if-feature intf-jp-multiplier;
            leaf jp-multiplier {
              type rt-types:timer-multiplier;
              description
                "Join prune multiplier is the number by which the join
                 prune interval is multplied to obtain the hold time";
            }
          }
        }
        leaf propagation-delay {
          if-feature intf-propagation-delay;
          type uint16;
          units milliseconds;
          description "Propagation description";
        }
        leaf override-interval {
          if-feature intf-override-interval;
          type uint16;
          units milliseconds;
          description "Override interval";
        }
      } // interface-config-attributes

      grouping interface-state-attributes {
        description
          "A grouping defining interface attributes.";
```

```
        container ipv4 {
          when "../address-family = 'rt:ipv4'" {
            description
              "Only applicable to IPv4 address family.";
          }
          config false;
          description "Interface state attributes for IPv4.";
          leaf-list address {
            type inet:ipv4-address;
            description "List of addresses.";
          }
          leaf dr-address {
            type inet:ipv4-address;
            description "DR (Designated Router) address.";
          }
        }
        container ipv6 {
          when "../address-family = 'rt:ipv6'" {
            description
              "Only applicable to IPv6 address family.";
          }
          config false;
          description "Interface state attributes for IPv6.";
          leaf-list address {
            type inet:ipv6-address;
            description "List of addresses.";
          }
          leaf dr-address {
            type inet:ipv6-address;
            description "DR (Designated Router) address.";
          }
        }
        uses interface-state-af-attributes;
      } // interface-state-attributes

      grouping interface-state-af-attributes {
        description
          "A grouping defining interface per address family attributes.";

        leaf oper-status {
          type enumeration {
            enum up {
              description
              "Ready to pass packets.";
            }
            enum down {
              description
              "The interface does not pass any packets.";
```

```
          }
        }
        config false;
        description "Operational status.";
      }

      leaf hello-expiration {
        type timer-value;
        config false;
        description "Hello interval expiration time.";
      }

      list neighbor-ipv4 {
        when "../address-family = 'rt:ipv4'" {
          description
            "Only applicable to IPv4 address family.";
        }
        key "address";
        config false;
        description "Neighbor state information.";
        leaf address {
          type inet:ipv4-address;
          description "Neighbor address.";
        }
        uses neighbor-state-af-attributes;
      } // list neighbor-ipv4

      list neighbor-ipv6 {
        when "../address-family = 'rt:ipv6'" {
          description
            "Only applicable to IPv6 address family.";
        }
        key "address";
        config false;
        description "Neighbor state information.";
        leaf address {
          type inet:ipv6-address;
          description "Neighbor address.";
        }
        uses neighbor-state-af-attributes;
      } // list neighbor-ipv6
    } // interface-state-af-attributes

    grouping multicast-route-attributes {
      description
        "A grouping defining multicast route attributes.";

      leaf expiration {
```

```
          type timer-value;
          description "When the route will expire.";
        }
        leaf incoming-interface {
          type if:interface-ref;
          description
            "Reference to an entry in the global interface
             list.";
        }
        leaf mode {
          type pim-mode;
          description "PIM mode.";
        }
        leaf msdp-learned {
          type boolean;
          description
            "'true' if route is learned from MSDP (Multicast Source
              Discovery Protocol).";
        }
        leaf rp-address {
          type inet:ip-address;
          description "RP (Rendezvous Point) address.";
        }
        leaf rpf-neighbor {
          type inet:ip-address;
          description "RPF (Reverse Path Forwarding) neighbor address.";
        }
        leaf spt-bit {
          type boolean;
          description "'true' if SPT (Shortest Path Tree) bit is set.";
        }
        leaf up-time {
          type uint32;
          units seconds;
          description "Up time duration.";
        }
        list outgoing-interface {
          key "name";
          description
            "A list of outgoing interfaces.";

          leaf name {
            type if:interface-ref;
            description
              "Interface name.";
          }

          leaf expiration {
```

```
          type timer-value;
          description "Expiring information.";
        }

        leaf up-time {
          type uint32;
          units seconds;
          description "Up time duration.";
        }

        leaf jp-state {
          type enumeration {
            enum "no-info" {
              description
                "The interface has Join state and no timers running";
            }
            enum "join" {
              description
                "The interface has Join state.";
            }
            enum "prune-pending" {
              description
                "The router has received a Prune on this interface from
                 a downstream neighbor and is waiting to see whether
                 the prune will be overridden by another downstream
                 router.  For forwarding purposes, the Prune-Pending
                 state functions exactly like the Join state.";
            }
          }
          description "Join-prune state.";
        }
      }
    } // multicast-route-attributes

    grouping neighbor-state-af-attributes {
      description
        "A grouping defining neighbor per address family attributes.";
      leaf bfd-status {
        type enumeration {
          enum up {
            description
              "BFD (Bidirectional Forwarding Detection) is up.";
          }
          enum down {
            description
              "BFD (Bidirectional Forwarding Detection) is down.";
          }
        }
```

```
        description "BFD (Bidirectional Forwarding Detection) status.";
      }
      leaf expiration {
        type timer-value;
        description "Neighbor expiring information.";
      }
      leaf dr-priority {
        type uint32;
        description "DR (Designated Router) priority";
      }
      leaf gen-id {
        type uint32;
        description "Generation ID.";
      }
      leaf up-time {
        type uint32;
        units seconds;
        description "Up time duration.";
      }
    } // neighbor-state-af-attributes

    grouping per-af-attributes {
      description
        "A grouping defining per address family attributes.";
      uses graceful-restart-container {
        if-feature per-af-graceful-restart;
      }
    } // per-af-attributes

    grouping pim-instance-af-state-ref {
      description
        "An absolute reference to a PIM instance address family.";
      leaf instance-af-ref {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
            + "pim-base:pim/pim-base:address-family/"
            + "pim-base:address-family";
        }
        description
          "Reference to a PIM instance address family.";
      }
    } // pim-instance-state-af-ref

    grouping pim-interface-state-ref {
      description
        "An absolute reference to a PIM interface state.";
      leaf interface-ref {
        type leafref {
```

```
            path "/rt:routing/rt:control-plane-protocols/"
              + "pim-base:pim/pim-base:interfaces/pim-base:interface/"
              + "pim-base:interface";
          }
          description
            "Reference to a PIM interface.";
        }
      } // pim-interface-state-ref

      grouping pim-neighbor-state-ref {
        description
          "An absolute reference to a PIM neighbor state.";
        uses pim-interface-state-ref;
        leaf interface-af-ref {
          type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
              + "pim-base:pim/pim-base:interfaces/pim-base:interface"
              + "[pim-base:interface = "
              + "current()/../interface-ref]/"
              + "pim-base:address-family/pim-base:address-family";
          }
          description
            "Reference to a PIM interface address family.";
        }
        leaf neighbor-ipv4-ref {
          when "../interface-af-ref = 'rt:ipv4'" {
            description "Only applicable to IPv4 address family.";
          }
          type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
              + "pim-base:pim/pim-base:interfaces/pim-base:interface"
              + "[pim-base:interface = "
              + "current()/../interface-ref]/"
              + "pim-base:address-family"
              + "[pim-base:address-family = "
              + "current()/../interface-af-ref]/"
              + "pim-base:neighbor-ipv4/pim-base:address";
          }
          description
            "Reference to a PIM IPv4 neighbor.";
        }
        leaf neighbor-ipv6-ref {
          when "../interface-af-ref = 'rt:ipv6'" {
            description "Only applicable to IPv6 address family.";
          }
          type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
              + "pim-base:pim/pim-base:interfaces/pim-base:interface"
```

```
             + "[pim-base:interface = "
             + "current()/../interface-ref]/"
             + "pim-base:address-family"
             + "[pim-base:address-family = "
             + "current()/../interface-af-ref]/"
             + "pim-base:neighbor-ipv6/pim-base:address";
         }
         description
           "Reference to a PIM IPv6 neighbor.";
       }
     } // pim-neighbor-state-ref

     grouping statistics-container {
       description
         "A container defining statistics attributes.";
       container statistics {
         config false;
         description "A container defining statistics attributes.";
         leaf discontinuity-time {
           type yang:date-and-time;
           description
             "The time on the most recent occasion at which any one
              or more of the statistic counters suffered a
              discontinuity. If no such discontinuities have occurred
              since the last re-initialization of the local
              management subsystem, then this node contains the time
              the local management subsystem re-initialized itself.";
         }
         container error {
           description "Containing error statistics.";
           uses statistics-error;
         }
         container queue {
           description "Containing queue statistics.";
           uses statistics-queue;
         }
         container received {
           description "Containing statistics of received messages.";
           uses statistics-sent-received;
         }
         container sent {
           description "Containing statistics of sent messages.";
           uses statistics-sent-received;
         }
       }
     } // statistics-container

     grouping statistics-error {
```

```
        description
          "A grouping defining error statistics
           attributes.";
        uses statistics-sent-received;
      } // statistics-error

      grouping statistics-queue {
        description
          "A grouping defining queue statistics
          attributes.";
        leaf size {
          type uint32;
          description
            "The size of the input queue.";
        }
        leaf overflow {
          type yang:counter32;
          description
            "The number of the input queue overflows.";
        }
      } // statistics-queue

      grouping statistics-sent-received {
        description
          "A grouping defining sent and received statistics
          attributes.";
        leaf assert {
          type yang:counter64;
          description
            "The number of assert messages.";
        }
        leaf bsr {
          type yang:counter64;
          description
            "The number of BSR (Bootstrap Router) messages.";
        }
        leaf candidate-rp-advertisement {
          type yang:counter64;
          description
            "The number of Candidate-RP-advertisement messages.";
        }
        leaf df-election {
          type yang:counter64;
          description
            "The number of DF (Designated Forwarder) election messages.";
        }
        leaf hello {
          type yang:counter64;
```

```
        description
          "The number of hello messages.";
      }
      leaf join-prune {
        type yang:counter64;
        description
          "The number of join/prune messages.";
      }
      leaf register {
        type yang:counter64;
        description
          "The number of register messages.";
      }
      leaf register-stop {
        type yang:counter64;
        description
          "The number of register stop messages.";
      }
      leaf state-refresh {
        type yang:counter64;
        description
          "The number of state refresh messages.";
      }
    } // statistics-sent-received

    /*
     * Configuration data nodes
     */

    augment "/rt:routing/rt:control-plane-protocols" {
      description
        "PIM augmentation to routing instance configuration.";

      container pim {
        presence "Container for PIM protocol.";
        description
          "PIM configuration data.";

        uses global-attributes;

        list address-family {
          key "address-family";
          description
            "Each list entry for one address family.";
          uses rt:address-family;
          uses per-af-attributes;

          uses statistics-container;
```

```
            container topology-tree-info {
              config false;
              description "Containing topology tree information.";
              list ipv4-route {
                when "../../address-family = 'rt:ipv4'" {
                  description
                    "Only applicable to IPv4 address family.";
                }
                key "group source-address is-rpt";
                description "A list of IPv4 routes.";
                leaf group {
                  type rt-types:ipv4-multicast-group-address;
                  description "Group address.";
                }
                leaf source-address {
                  type rt-types:ipv4-multicast-source-address;
                  description "Source address.";
                }
                leaf is-rpt {
                  type boolean;
                  description
                    "'true' if the tree is RPT (Rendezvous-Point Tree).";
                }

                uses multicast-route-attributes;
              } // ipv4-route

              list ipv6-route {
                when "../../address-family = 'rt:ipv6'" {
                  description
                    "Only applicable to IPv6 address family.";
                }
                key "group source-address is-rpt";
                description "A list of IPv6 routes.";
                leaf group {
                  type rt-types:ipv6-multicast-group-address;
                  description "Group address.";
                }
                leaf source-address {
                  type rt-types:ipv6-multicast-source-address;
                  description "Source address.";
                }
                leaf is-rpt {
                  type boolean;
                  description
                    "'true' if the tree is RPT (Rendezvous-Point Tree).";
                }
```

```
              uses multicast-route-attributes;
            } // ipv6-route
          } // topology-tree-info
        } // address-family

        container interfaces {
          description
            "Containing a list of interfaces.";
          list interface {
            key "interface";
            description
              "List of pim interfaces.";
            leaf interface {
              type if:interface-ref;
              description
                "Reference to an entry in the global interface
                list.";
            }
            list address-family {
              key "address-family";
              description
                "Each list entry for one address family.";
              uses rt:address-family;
              uses interface-config-attributes;
              uses interface-state-attributes;
            } // address-family
          } // interface
        } // interfaces
      } // pim
    } // augment

    /*
     * Notifications
     */
    notification pim-neighbor-event {
      description "Notification event for neighbor.";
      leaf event-type {
        type neighbor-event-type;
        description "Event type.";
      }
      uses pim-neighbor-state-ref;
      leaf up-time {
        type uint32;
        units seconds;
        description "Up time duration.";
      }
    }
    notification pim-interface-event {
```

```
      description "Notification event for interface.";
      leaf event-type {
        type interface-event-type;
        description "Event type.";
      }
      uses pim-interface-state-ref;
      container ipv4 {
        description "Containing IPv4 information.";
        leaf-list address {
          type inet:ipv4-address;
          description "List of addresses.";
        }
        leaf dr-address {
          type inet:ipv4-address;
          description "DR (Designated Router) address.";
        }
      }
      container ipv6 {
        description "Containing IPv6 information.";
        leaf-list address {
          type inet:ipv6-address;
          description "List of addresses.";
        }
        leaf dr-address {
          type inet:ipv6-address;
          description "DR (Designated Router) address.";
        }
      }
    }
  }
  <CODE ENDS>
```

4.2. PIM RP module

```
  <CODE BEGINS> file "ietf-pim-rp@2017-12-08.yang"
  module ietf-pim-rp {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pim-rp";
    prefix pim-rp;

    import ietf-inet-types {
      prefix "inet";
    }

    import ietf-routing-types {
      prefix "rt-types";
```

```
     }

     import ietf-interfaces {
       prefix "if";
     }

     import ietf-routing {
       prefix "rt";
     }

     import ietf-pim-base {
       prefix "pim-base";
     }

     organization
       "IETF PIM Working Group";

     contact
       "WG Web:   <http://tools.ietf.org/wg/pim/>
        WG List:  <mailto:pim@ietf.org>

        WG Chair: Stig Venaas
                  <mailto:stig@venaas.com>

        WG Chair: Mike McBride
                  <mailto:mmcbride7@gmail.com>

        Editor:   Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>

        Editor:   Pete McAllister
                  <mailto:pete.mcallister@metaswitch.com>

        Editor:   Anish Peter
                  <mailto:anish.ietf@gmail.com>

        Editor:   Mahesh Sivakumar
                  <mailto:masivaku@cisco.com>

        Editor:   Yisong Liu
                  <mailto:liuyisong@huawei.com>

        Editor:   Fangwei Hu
                  <mailto:hu.fangwei@zte.com.cn>";

     description
       "The YANG module defines a PIM (Protocol Independent Multicast)
        RP (Rendezvous Point) model.";
```

```
     revision 2017-12-08 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }

     /*
      * Features
      */
     feature bsr {
       description
         "This feature indicates that the system supports BSR
          (Bootstrap Router).";
     }

     feature bsr-election-state {
       description
         "This feature indicates that the system supports providing
          BSR election state.";
       reference
         "RFC5059: Bootstrap Router (BSR) Mechanism for Protocol
          Independent Multicast (PIM).";
     }

     feature static-rp-override {
       description
         "This feature indicates that the system supports configuration
          of static RP (Rendezvous Point) override.";
     }

     feature candidate-interface {
       description
         "This feature indicates that the system supports using
          an interface to configure a BSR or RP candidate.";
     }

     feature candidate-ipv4 {
       description
         "This feature indicates that the system supports using
          an IPv4 address to configure a BSR or RP candidate.";
     }

     feature candidate-ipv6 {
       description
         "This feature indicates that the system supports using
          an IPv6 address to configure a BSR or RP candidate.";
     }
```

```
/*
 * Typedefs
 */
typedef rp-event-type {
  type enumeration {
    enum invalid-jp {
      description
        "An invalid JP (Join/Prune) message has been received.";
    }
    enum invalid-register {
      description
        "An invalid register message has been received.";
    }
    enum mapping-created {
      description
        "A new mapping has been created.";
    }
    enum mapping-deleted {
      description
        "A mapping has been deleted.";
    }
  }
  description "Operational status event type for notifications.";
}

/*
 * Identities
 */
identity rp-mode {
  description
    "The mode of an RP, which can be SM (Sparse Mode) or
     BIDIR (bi-directional).";
}

identity rp-info-source-type {
  description
    "The information source of an RP.";
}
identity static {
  base rp-info-source-type;
  description
    "The RP is statically configured.";
}
identity bootstrap {
  base rp-info-source-type;
  description
    "The RP is learned from bootstrap.";
}
```

```
     /*
      * Groupings
      */
     grouping bsr-config-attributes {
       description
         "Grouping of BSR config attributes.";
       container bsr-candidate {
         presence
           "Present to serve as a BSR candidate";
         description
           "BSR candidate attributes.";

         choice interface-or-address {
           description
             "Use either interface or ip-address.";
           case interface {
             if-feature candidate-interface;
             leaf interface {
               type if:interface-ref;
               mandatory true;
               description
                 "Interface to be used by BSR.";
             }
           }
           case ipv4-address {
             when "../../../pim-base:address-family = 'rt:ipv4'" {
               description
                 "Only applicable to IPv4 address family.";
             }
             if-feature candidate-ipv4;
             leaf ipv4-address {
               type inet:ipv4-address;
               mandatory true;
               description
                 "IP address to be used by BSR.";
             }
           }
           case ipv6-address {
             when "../../../pim-base:address-family = 'rt:ipv6'" {
               description
                 "Only applicable to IPv6 address family.";
             }
             if-feature candidate-ipv6;
             leaf ipv6-address {
               type inet:ipv6-address;
               mandatory true;
               description
                 "IP address to be used by BSR.";
```

```
            }
          }
        }

        leaf hash-mask-length{
          type uint8 {
            range "0..128";
          }
          mandatory true;
          description
            "Value contained in BSR messages used by all routers to
             hash (map) to an RP.";
        }

        leaf priority {
          type uint8 {
            range "0..255";
          }
          mandatory true;
          description
            "BSR election priority among different candidate BSRs.
            A larger value has a higher priority over a smaller
            value.";
        }
      } // bsr-candidate

      list rp-candidate-interface {
        if-feature candidate-interface;
        key "interface";
        description
          "A list of RP candidates";
        leaf interface {
          type if:interface-ref;
          description
            "Interface that the RP candidate uses.";
        }
        uses rp-candidate-attributes;
      }

      list rp-candidate-ipv4-address {
        when "../../../pim-base:address-family = 'rt:ipv4'" {
          description
            "Only applicable to IPv4 address family.";
        }
        if-feature candidate-ipv4;
        key "ipv4-address";
        description
          "A list of RP candidates";
```

```
          leaf ipv4-address {
            type inet:ipv4-address;
            description
              "IPv4 address that the RP candidate uses.";
          }
          uses rp-candidate-attributes;
        }

        list rp-candidate-ipv6-address {
          when "../../../pim-base:address-family = 'rt:ipv6'" {
            description
              "Only applicable to IPv6 address family.";
          }
          if-feature candidate-ipv6;
          key "ipv6-address";
          description
            "A list of RP candidates";
          leaf ipv6-address {
            type inet:ipv6-address;
            description
              "IPv6 address that the RP candidate uses.";
          }
          uses rp-candidate-attributes;
        }
      } // bsr-config-attributes

      grouping bsr-state-attributes {
        description
          "Grouping of BSR state attributes.";
        container bsr {
          config false;
          description
            "BSR information.";
          leaf address {
            type inet:ip-address;
            description "BSR address";
          }
          leaf hash-mask-length {
            type uint8 {
              range "0..128";
            }
            description "Hash mask length.";
          }
          leaf priority {
            type uint8 {
              range "0..255";
            }
            description "Priority.";
```

```
          }
          leaf up-time {
            type uint32;
            units seconds;
            description "Up time duration.";
          }
        }
        choice election-state {
          if-feature bsr-election-state;
          config false;
          description "BSR election state.";
          case candidate {
            leaf candidate-bsr-state {
              type enumeration {
                enum "candidate" {
                  description
                    "The router is a candidate to be the BSR for the
                     scope zone, but currently another router is the
                     preferred BSR.";
                }
                enum "pending" {
                  description
                    "The router is a candidate to be the BSR for the
                     scope zone. Currently, no other router is the
                     preferred BSR, but this router is not yet the
                     elected BSR.  This is a temporary state that
                     prevents rapid thrashing of the choice of BSR
                     during BSR election.";
                }
                enum "elected" {
                  description
                    "The router is the elected BSR for the scope zone
                     and it must perform all the BSR functions.";
                }
              }
              description
                "Candidate-BSR state.";
              reference
                "RFC5059, Section 3.1.1.";
            }
          }
          case "non-candidate" {
            leaf non-candidate-bsr-state {
              type enumeration {
                enum "no-info" {
                  description
                    "The router has no information about this scope
                     zone.";
```

```
              }
              enum "accept-any" {
                description
                  "The router does not know of an active BSR, and will
                   accept the first Bootstrap message it sees as giving
                   the new BSR's identity and the RP-Set.";
              }
              enum "accept" {
                description
                  "The router knows the identity of the current BSR,
                   and is using the RP-Set provided by that BSR. Only
                   Bootstrap messages from that BSR or from a
                   Candidate-BSR (C-BSR) with higher weight than the
                   current BSR will be accepted.";
              }
            }
            description
              "Non-candidate-BSR state.";
            reference
              "RFC5059, Section 3.1.2.";
          }
        }
      } // election-state
      leaf bsr-next-bootstrap {
        type uint16;
        units seconds;
        config false;
        description "The time when next bootstrap will be sent.";
      }

      container rp {
        config false;
        description
          "State information of the RP.";
        leaf rp-address {
          type inet:ip-address;
          description "RP address.";
        }
        leaf group-policy {
          type string;
          description "Group policy.";
        }
        leaf up-time {
          type uint32;
          units seconds;
          description "Up time duration.";
        }
      }
```

```
        leaf rp-candidate-next-advertisement {
          type uint16;
          units seconds;
          config false;
          description
            "When the next advertisement will be sent as RP candidate";
        }
      } // bsr-state-attributes

      grouping rp-mapping-state-attributes {
        description
          "Grouping of RP mapping attributes.";
        leaf up-time {
          type uint32;
          units seconds;
          description "Up time duration.";
        }
        leaf expiration {
          type pim-base:timer-value;
          description "Expiration time.";
        }
      } // rp-mapping-state-attributes

      grouping rp-state-attributes {
        description
          "Grouping of RP state attributes.";
        leaf info-source-type {
          type identityref {
            base rp-info-source-type;
          }
          description "The information source of an RP.";
        } // info-source-type
        leaf up-time {
          type uint32;
          units seconds;
          description "Up time duration.";
        }
        leaf expiration {
          type pim-base:timer-value;
          description "Expiration time.";
        }
      } // rp-state-attributes

      grouping static-rp-attributes {
        description
          "Grouping of static RP attributes, used in augmenting
           modules.";
        leaf policy-name {
```

```
        type string;
        description
          "Static RP policy.";
      }
      leaf override {
        if-feature static-rp-override;
        type boolean;
        description
          "When there is a conflict between static RP and dynamic
           RP, setting this attribute to 'true' will ask the
           system to use static RP.";
      }
    } // static-rp-attributes

    grouping static-rp-container {
      description
        "Grouping of static RP container.";
      container static-rp {
        description
          "Containing static RP attributes.";
        list ipv4-rp {
          when "../../../pim-base:address-family = 'rt:ipv4'" {
            description
              "Only applicable to IPv4 address family.";
          }
          key "ipv4-address";
          description
            "A list of IPv4 RP addresses.";
          leaf ipv4-address {
            type inet:ipv4-address;
            description
              "Specifies a static RP address.";
          }
        }

        list ipv6-rp {
          when "../../../pim-base:address-family = 'rt:ipv6'" {
            description
              "Only applicable to IPv6 address family.";
          }
          key "ipv6-address";
          description
            "A list of IPv6 RP addresses.";
          leaf ipv6-address {
            type inet:ipv6-address;
            description
              "Specifies a static RP address.";
          }
```

```
        }
      } // static-rp
    } // static-rp-container

    grouping rp-candidate-attributes {
      description
        "Grouping of RP candidate attributes.";
      leaf policy {
        type string;
        description
          "ACL (Access Control List) policy used to filter group
           addresses.";
      }
      leaf mode {
        type identityref {
          base rp-mode;
        }
        description
          "RP mode.";
      }
    } // rp-candidate-attributes

    /*
     * Configuration data nodes
     */

    augment "/rt:routing/rt:control-plane-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description "PIM RP augmentation.";

      container rp {
        description
          "PIM RP configuration data.";

        uses static-rp-container;

        container bsr {
          if-feature bsr;
          description
            "Containing BSR (BootStrap Router) attributes.";
          uses bsr-config-attributes;
          uses bsr-state-attributes;
        } // bsr

        container rp-list {
          config false;
          description
            "Containing a list of RPs.";
```

```
            list ipv4-rp {
              when "../../../pim-base:address-family = 'rt:ipv4'" {
                description
                  "Only applicable to IPv4 address family.";
              }
              key "ipv4-address mode";
              description
                "A list of IPv4 RP addresses.";
              leaf ipv4-address {
                type inet:ipv4-address;
                description
                  "RP address.";
              }
              leaf mode {
                type identityref {
                  base rp-mode;
                }
                description
                  "RP mode.";
              }
              leaf info-source-address {
                type inet:ipv4-address;
                description
                  "The address where RP information is learned.";
              }
              uses rp-state-attributes;
            }

            list ipv6-rp {
              when "../../../pim-base:address-family = 'rt:ipv6'" {
                description
                  "Only applicable to IPv6 address family.";
              }
              key "ipv6-address mode";
              description
                "A list of IPv6 RP addresses.";
              leaf ipv6-address {
                type inet:ipv6-address;
                description
                  "RP address.";
              }
              leaf mode {
                type identityref {
                  base rp-mode;
                }
                description
                  "RP mode.";
              }
```

```
            leaf info-source-address {
              type inet:ipv6-address;
              description
                "The address where RP information is learned.";
            }
            uses rp-state-attributes;
          }
        } // rp-list

        container rp-mappings {
          config false;
          description
            "Containing a list of group-to-RP mappings.";
          list ipv4-rp {
            when "../../../pim-base:address-family = 'rt:ipv4'" {
              description
                "Only applicable to IPv4 address family.";
            }
            key "group rp-address";
            description
              "A list of group-to-RP mappings.";
            leaf group {
              type inet:ipv4-prefix;
              description
                "Group prefix.";
            }
            leaf rp-address {
              type inet:ipv4-address;
              description
                "RP address.";
            }
            uses rp-mapping-state-attributes;
          }

          list ipv6-rp {
            when "../../../pim-base:address-family = 'rt:ipv6'" {
              description
                "Only applicable to IPv6 address family.";
            }
            key "group rp-address";
            description
              "A list of IPv6 RP addresses.";
            leaf group {
              type inet:ipv6-prefix;
              description
                "Group prefix.";
            }
            leaf rp-address {
```

```
                type inet:ipv6-address;
                description
                  "RP address.";
              }
              uses rp-mapping-state-attributes;
            }
          } // rp-mappings
        } // rp
      } // augment

      /*
       * Notifications
       */
      notification pim-rp-event {
        description "Notification event for RP.";
        leaf event-type {
          type rp-event-type;
          description "Event type.";
        }
        uses pim-base:pim-instance-af-state-ref;
        leaf group {
          type rt-types:ip-multicast-group-address;
          description "Group address.";
        }
        leaf rp-address {
          type inet:ip-address;
          description "RP address.";
        }
        leaf is-rpt {
          type boolean;
          description "'true' if the tree is RPT (RP-Tree).";
        }
        leaf mode {
          type pim-base:pim-mode;
          description "PIM mode.";
        }
        leaf message-origin {
          type inet:ip-address;
          description "Where the message is originated.";
        }
      }
    }
    <CODE ENDS>
```

4.3. PIM-SM module

```
<CODE BEGINS> file "ietf-pim-sm@2017-12-08.yang"
module ietf-pim-sm {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pim-sm";
  prefix pim-sm;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-pim-base {
    prefix "pim-base";
  }

  import ietf-pim-rp {
    prefix "pim-rp";
  }

  organization
    "IETF PIM Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/pim/>
     WG List:  <mailto:pim@ietf.org>

     WG Chair: Stig Venaas
               <mailto:stig@venaas.com>

     WG Chair: Mike McBride
               <mailto:mmcbride7@gmail.com>

     Editor:   Xufeng Liu
               <mailto:Xufeng_Liu@jabil.com>

     Editor:   Pete McAllister
               <mailto:pete.mcallister@metaswitch.com>

     Editor:   Anish Peter
               <mailto:anish.ietf@gmail.com>

     Editor:   Mahesh Sivakumar
               <mailto:masivaku@cisco.com>
```

```
        Editor:   Yisong Liu
                  <mailto:liuyisong@huawei.com>

        Editor:   Fangwei Hu
                  <mailto:hu.fangwei@zte.com.cn>";

     description
       "The YANG module defines a PIM (Protocol Independent Multicast)
        SM (Sparse Mode) model.";

     revision 2017-12-08 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }

     /*
      * Features
      */
     feature spt-switch-infinity {
       description
         "This feature indicates that the system supports configuration
          choice whether to trigger the switchover from the RPT
          (Rendezvous Point Tree) to the SPT (Shortest Path Tree).";
     }

     feature spt-switch-policy {
       description
         "This feature indicates that the system supports configuring
          policy for the switchover from the RPT to the SPT.";
     }

     /*
      * Identities
      */
     identity sm {
       base pim-rp:rp-mode;
       description
         "SM (Sparse Mode).";
     }

     /*
      * Groupings
      */
     grouping af-sm-container {
       description
         "Grouping of address family SM container.";
```

```
       container sm {
         description
           "PIM SM configuration data.";

         container asm {
           description
             "ASM (Any Source Multicast) attributes.";

           container anycast-rp {
             presence
               "Present to enable anycast RP (Rendezvous Point).";
             description
               "Anycast RP attributes.";

             container ipv4 {
               when "../../../../pim-base:address-family = 'rt:ipv4'" {
                 description
                   "Only applicable to IPv4 address family.";
               }
               description
                 "IPv4 attributes. Only applicable when
                  pim-base:address-family is IPv4.";
               list ipv4-anycast-rp {
                 key "anycast-address rp-address";
                 description
                   "A list of anycast RP settings.";
                 leaf anycast-address {
                   type inet:ipv4-address;
                   description
                     "IP address of the anycast RP set. This IP address
                      is used by the multicast groups or sources to join
                      or register.";
                 }

                 leaf rp-address {
                   type inet:ipv4-address;
                   description
                     "IP address of the router configured with anycast
                      RP. This is the IP address where the Register
                      messages are forwarded.";
                 }
               }
             }
             container ipv6 {
               when "../../../../pim-base:address-family = 'rt:ipv6'" {
                 description
                   "Only applicable to IPv6 address family.";
               }
```

```
                description
                  "IPv6 attributes. Only applicable when
                   pim-base:address-family is IPv6.";
                list ipv6-anycast-rp {
                  key "anycast-address rp-address";
                  description
                    "A list of anycast RP settings.";
                  leaf anycast-address {
                    type inet:ipv6-address;
                    description
                      "IP address of the anycast RP set. This IP address
                       is used by the multicast groups or sources to join
                       or register.";
                  }

                  leaf rp-address {
                    type inet:ipv6-address;
                    description
                      "IP address of the router configured with anycast
                       RP. This is the IP address where the Register
                       messages are forwarded.";
                  }
                }
              }
            }

            container spt-switch {
              description
                "SPT (Shortest Path Tree) switching attributes.";
              container infinity {
                if-feature spt-switch-infinity;
                presence
                  "Present if SPT switchover threshold is set to
                   infinity, according to the policy specified below.";
                description
                  "The receiver's DR (Designated Router) never triggers
                   the switchover from the RPT to the SPT.";
                leaf policy-name {
                  if-feature spt-switch-policy;
                  type string;
                  description
                    "Switch policy.";
                }
              } // infinity
            }
          } // asm

          container ssm {
```

```
          presence
            "Present to enable SSM (Source-Specific Multicast).";
          description
            "SSM (Source-Specific Multicast) attributes.";

          leaf range-policy {
            type string;
            description
              "Policy used to define SSM address range.";
          }
        } // ssm
      } // sm
    } // af-sm-container

    grouping interface-sm-container {
      description
        "Grouping of interface SM container.";
      container sm {
        presence "Present to enable sparse-mode.";
        description
          "PIM SM configuration data.";

        leaf passive {
          type empty;
          description
            "Specifies that no PIM messages are sent or accepted on
             this PIM interface, but the interface can be included in a
             multicast forwarding entry.";
        }
      } // sm
    } // interface-sm-container

    grouping static-rp-sm-container {
      description
        "Grouping that contains SM attributes for static RP.";
      container sm {
        presence
          "Indicate the support of sparse mode.";
        description
          "PIM SM configuration data.";

        uses pim-rp:static-rp-attributes;
      } // sm
    } // static-rp-sm-container

    /*
     * Configuration data nodes
     */
```

```
  augment "/rt:routing/rt:control-plane-protocols/pim-base:pim/"
    + "pim-base:address-family" {
    description "PIM SM augmentation.";

    uses af-sm-container;
  } // augment

  augment "/rt:routing/rt:control-plane-protocols/pim-base:pim/"
    + "pim-base:interfaces/pim-base:interface/"
    + "pim-base:address-family" {
    description "PIM SM augmentation.";

    uses interface-sm-container;
  } // augment

  augment "/rt:routing/rt:control-plane-protocols/pim-base:pim/"
    + "pim-base:address-family/pim-rp:rp/"
    + "pim-rp:static-rp/pim-rp:ipv4-rp" {
    description "PIM SM augmentation.";

    uses static-rp-sm-container;
  } // augment

  augment "/rt:routing/rt:control-plane-protocols/pim-base:pim/"
    + "pim-base:address-family/pim-rp:rp/"
    + "pim-rp:static-rp/pim-rp:ipv6-rp" {
    description "PIM SM augmentation.";

    uses static-rp-sm-container;
  } // augment
}
<CODE ENDS>
```

4.4. PIM-DM module

```
<CODE BEGINS> file "ietf-pim-dm@2017-12-08.yang"
module ietf-pim-dm {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pim-dm";
  prefix pim-dm;

  import ietf-routing {
    prefix "rt";
  }

  import ietf-pim-base {
```

```
      prefix "pim-base";
    }

    organization
      "IETF PIM Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/pim/>
       WG List:  <mailto:pim@ietf.org>

       WG Chair: Stig Venaas
                 <mailto:stig@venaas.com>

       WG Chair: Mike McBride
                 <mailto:mmcbride7@gmail.com>

       Editor:   Xufeng Liu
                 <mailto:Xufeng_Liu@jabil.com>

       Editor:   Pete McAllister
                 <mailto:pete.mcallister@metaswitch.com>

       Editor:   Anish Peter
                 <mailto:anish.ietf@gmail.com>

       Editor:   Mahesh Sivakumar
                 <mailto:masivaku@cisco.com>

       Editor:   Yisong Liu
                 <mailto:liuyisong@huawei.com>

       Editor:   Fangwei Hu
                 <mailto:hu.fangwei@zte.com.cn>";

    description
      "The YANG module defines a PIM (Protocol Independent Multicast)
       DM (Dense Mode) model.";

    revision 2017-12-08 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for PIM";
    }

    /*
     * Configuration data nodes
     */
```

```
  augment "/rt:routing/rt:control-plane-protocols/"
    + "pim-base:pim/pim-base:address-family" {
    description "PIM DM (Dense Mode) augmentation.";

    container dm {
      presence "Present to enable dense-mode.";
      description
        "PIM DM configuration data.";
    } // Dm
  } // augment

  augment "/rt:routing/rt:control-plane-protocols/"
    + "pim-base:pim/pim-base:interfaces/pim-base:interface/"
    + "pim-base:address-family" {
    description "PIM DM augmentation to PIM base interface.";

    container dm {
      presence "Present to enable dense-mode.";
      description
        "PIM DM configuration data.";
    } // sm
  } // augment
}
<CODE ENDS>
```

4.5. PIM-BIDIR module

```
<CODE BEGINS> file "ietf-pim-bidir@2017-12-08.yang"
module ietf-pim-bidir {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pim-bidir";
  prefix pim-bidir;

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-pim-base {
```

```
      prefix "pim-base";
    }

    import ietf-pim-rp {
      prefix "pim-rp";
    }

    organization
      "IETF PIM Working Group";

    contact
      "WG Web:   <http://tools.ietf.org/wg/pim/>
       WG List:  <mailto:pim@ietf.org>

       WG Chair: Stig Venaas
                 <mailto:stig@venaas.com>

       WG Chair: Mike McBride
                 <mailto:mmcbride7@gmail.com>

       Editor:   Xufeng Liu
                 <mailto:Xufeng_Liu@jabil.com>

       Editor:   Pete McAllister
                 <mailto:pete.mcallister@metaswitch.com>

       Editor:   Anish Peter
                 <mailto:anish.ietf@gmail.com>

       Editor:   Mahesh Sivakumar
                 <mailto:masivaku@cisco.com>

       Editor:   Yisong Liu
                 <mailto:liuyisong@huawei.com>

       Editor:   Fangwei Hu
                 <mailto:hu.fangwei@zte.com.cn>";

    description
      "The YANG module defines a PIM (Protocol Independent Multicast)
       BIDIR (Bidirectional) mode model.";

    revision 2017-12-08 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: A YANG Data Model for PIM";
    }
```

```
/*
 * Features
 */
feature intf-df-election {
  description
    "Support configuration of interface DF election.";
}

/*
 * Identities
 */
identity bidir {
  base pim-rp:rp-mode;
  description
    "BIDIR (Bidirectional) mode.";
}

identity df-state {
  description
    "DF (Designated Forwarder) election state type.";
  reference
    "RFC5015: Bidirectional Protocol Independent Multicast
     (BIDIR-PIM).";
}

identity df-state-offer {
  base df-state;
  description
    "Initial election state. When in the Offer state, a router
     thinks it can eventually become the winner and periodically
     generates Offer messages.";
}

identity df-state-lose {
  base df-state;
  description
    "There either is a different election winner or that no
     router on the link has a path to the RPA (Rendezvous-Point
     Address).";
}

identity df-state-win {
  base df-state;
  description
    "The router is the acting DF without any contest.";
}

identity df-state-backoff {
```

```
      base df-state;
      description
        "The router is the acting DF but another router has made a
         bid to take over.";
    }

    /*
     * Typedefs
     */

    /*
     * Groupings
     */
    grouping df-election-container {
      description
        "Grouping that contains DF (Designated Forwarder) election
         attributes.";
      container df-election {
        if-feature intf-df-election;
        description
          "DF election attributes.";
        leaf offer-interval {
          type pim-base:timer-value;
          description
            "Offer interval specifies the interval between repeated
             DF election messages.";
        }
        leaf backoff-interval {
          type pim-base:timer-value;
          description
            "This is the interval that the acting DF waits between
             receiving a better DF Offer and sending the Pass message
             to transfer DF responsibility";
        }
        leaf offer-multiplier {
          type uint8;
          description
            "This is number of transmission attempts for DF election
             messages.
             When a DF election Offer or Winner message fails to be
             received, the message is retransmitted.
             The offer-multiplier sets the minimum number of DF
             election messages that must fail to be received for DF
             election to fail.
             If a router receives from a neighbor a better offer than
             its own, the router stops participating in the election
             for a period of offer-multiplier * offer-interval.
             Eventually, all routers except the best candidate stop
```

```
              sending Offer messages.";
          }
        } // df-election
      } // df-election-container

      grouping static-rp-bidir-container {
        description
          "Grouping that contains BIDIR (Bidirectional) attributes for
           static RP (Rendezvous-Point).";
        container bidir {
          presence
            "Indicate the support of BIDIR mode.";
          description
            "PIM BIDIR configuration data.";

          uses pim-rp:static-rp-attributes;
        } // bidir
      } // static-rp-bidir-container

      /*
       * Configuration data nodes
       */

      augment "/rt:routing/rt:control-plane-protocols/"
        + "pim-base:pim/pim-base:address-family" {
        description "PIM BIDIR (Bidirectional) augmentation.";

        container bidir {
          presence "Present to enable BIDIR mode.";
          description
            "PIM BIDIR configuration data.";
        } // bidir
      } // augment

      augment "/rt:routing/rt:control-plane-protocols/"
        + "pim-base:pim/pim-base:interfaces/pim-base:interface/"
        + "pim-base:address-family" {
        description "PIM BIDIR augmentation.";

        container bidir {
          presence "Present to enable BIDIR mode.";
          description
            "PIM BIDIR configuration data.";

          uses df-election-container;
        } // bidir
      } // augment
```

```
     augment "/rt:routing/rt:control-plane-protocols/"
       + "pim-base:pim/pim-base:address-family/pim-rp:rp/"
       + "pim-rp:static-rp/pim-rp:ipv4-rp" {
       description "PIM BIDIR augmentation.";

       uses static-rp-bidir-container;
     } // augment

     augment "/rt:routing/rt:control-plane-protocols/"
       + "pim-base:pim/pim-base:address-family/pim-rp:rp/"
       + "pim-rp:static-rp/pim-rp:ipv6-rp" {
       description "PIM BIDIR augmentation.";

       uses static-rp-bidir-container;
     } // augment

     /*
      * Operational state data nodes
      */

     augment "/rt:routing/rt:control-plane-protocols/"
       + "pim-base:pim/pim-base:address-family/pim-rp:rp" {
       description "PIM BIDIR augmentation.";

       container bidir {
         config false;
         description
           "PIM BIDIR state data.";
         container df-election {
           description
             "DF election data.";
           list ipv4-rp {
             when "../../../../pim-base:address-family = 'rt:ipv4'" {
               description
                 "Only applicable to IPv4 address family.";
             }
             key "ipv4-address";
             description
               "A list of IPv4 RP addresses.";
             leaf ipv4-address {
               type inet:ipv4-address;
               description
                 "The address of the RP.";
             }
           } // ipv4-rp
           list ipv6-rp {
             when "../../../../pim-base:address-family = 'rt:ipv6'" {
               description
```

```
                "Only applicable to IPv6 address family.";
            }
            key "ipv6-address";
            description
              "A list of IPv6 RP addresses.";
            leaf ipv6-address {
              type inet:ipv6-address;
              description
                "The address of the RP.";
            }
          } // ipv6-rp
        } // df-election

        container interface-df-election {
          description
            "Interface DF election data.";
          list ipv4-rp {
            when "../../../../pim-base:address-family = 'rt:ipv4'" {
              description
                "Only applicable to IPv4 address family.";
            }
            key "ipv4-address interface-name";
            description
              "A list of IPv4 RP addresses.";
            leaf ipv4-address {
              type inet:ipv4-address;
              description
                "The address of the RP.";
            }
            leaf interface-name {
              type if:interface-ref;
              description
                "The address of the RP.";
            }
            leaf df-address {
              type inet:ipv4-address;
              description
                "DF address.";
            }
            leaf interface-state {
              type identityref {
                base df-state;
              }
              description
                "Interface state.";
            }
          } // ipv4-rp
          list ipv6-rp {
```

```
            when "../../../../pim-base:address-family = 'rt:ipv6'" {
              description
                "Only applicable to IPv6 address family.";
            }
            key "ipv6-address interface-name";
            description
              "A list of IPv6 RP addresses.";
            leaf ipv6-address {
              type inet:ipv6-address;
              description
                "The address of the RP.";
            }
            leaf interface-name {
              type if:interface-ref;
              description
                "The address of the RP.";
            }
            leaf df-address {
              type inet:ipv6-address;
              description
                "DF address.";
            }
            leaf interface-state {
              type identityref {
                base df-state;
              }
              description
                "Interface state.";
            }
          } // ipv6-rp
        } // interface-df-election
      }
    } // augment
  }
  <CODE ENDS>
```

5. Implementation Status

This section to be removed by the RFC editor.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation

here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

This document is the work result of the PIM working group's YANG multicast design team. The following wiki page contains the information on the design team members, the meeting discussions, lists of modeled features, and which features are supported by which existing implementations:

https://trac.ietf.org/trac/pim/wiki/yang

6. Security Considerations

Configuration and state data defined in this document are designed to be accessed via a management protocol with secure transport layer, such as NETCONF [RFC6241]. The NETCONF access control model [RFC6536] provides the means to restrict access for specific users to a pre- configured subset of all available operations and contents.

The models defined in this document contain a number of configuration data nodes that are writable, creatable, and deletable. Unauthorised access to the configuration data can adversely affect the routing subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations and other problems.

7. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-pim-base
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-pim-bidir
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-pim-dm
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-pim-rp
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-pim-sm
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------
```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```
---------------------------------------------------------------------
name:         ietf-pim-base
namespace:    urn:ietf:params:xml:ns:yang:ietf-pim-base
prefix:       pim-base
reference:    RFC XXXX
---------------------------------------------------------------------

---------------------------------------------------------------------
name:         ietf-pim-bidir
namespace:    urn:ietf:params:xml:ns:yang:ietf-pim-bidir
prefix:       pim-bidir
reference:    RFC XXXX
---------------------------------------------------------------------
```

```
--------------------------------------------------------------------
name:         ietf-pim-dm
namespace:    urn:ietf:params:xml:ns:yang:ietf-pim-dm
prefix:       pim-dm
reference:    RFC XXXX
--------------------------------------------------------------------

--------------------------------------------------------------------
name:         ietf-pim-rp
namespace:    urn:ietf:params:xml:ns:yang:ietf-pim-rp
prefix:       pim-rp
reference:    RFC XXXX
--------------------------------------------------------------------

--------------------------------------------------------------------
name:         ietf-pim-sm
namespace:    urn:ietf:params:xml:ns:yang:ietf-pim-sm
prefix:       pim-sm
reference:    RFC XXXX
--------------------------------------------------------------------
```

8. Acknowledgements

The authors would like to thank Steve Baillargeon, Guo Feng, Robert Kebler, Tanmoy Kundu, and Stig Venaas for their valuable contributions.

9. References

9.1. Normative References

[RFC3569] Bhattacharyya, S., Ed., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, DOI 10.17487/RFC3569, July 2003, <https://www.rfc-editor.org/info/rfc3569>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, DOI 10.17487/RFC3973, January 2005, <https://www.rfc-editor.org/info/rfc3973>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <https://www.rfc-editor.org/info/rfc4610>.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <https://www.rfc-editor.org/info/rfc5015>.

[RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", RFC 5059, DOI 10.17487/RFC5059, January 2008, <https://www.rfc-editor.org/info/rfc5059>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[I-D.ietf-netmod-revised-datastores] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-07 (work in progress), November 2017.

[I-D.ietf-netmod-rfc7223bis] Bjorklund, M., "A YANG Data Model for Interface Management", draft-ietf-netmod-rfc7223bis-00 (work in progress), October 2017.

[I-D.ietf-netmod-rfc8022bis]
Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NDMA Version)", draft-ietf-netmod-rfc8022bis-02 (work in progress), November 2017.

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

[I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-07 (work in progress), October 2017.

9.2. Informative References

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <https://www.rfc-editor.org/info/rfc3376>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <https://www.rfc-editor.org/info/rfc3618>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-14 (work in progress), September 2017.

[I-D.ietf-netmod-yang-tree-diagrams] Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-02 (work in progress), October 2017.

Authors' Addresses

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

EMail: Xufeng_Liu@jabil.com

Pete McAllister
Metaswitch Networks
100 Church Street
Enfield EN2 6BQ
UK

EMail: pete.mcallister@metaswitch.com

Anish Peter
Individual

EMail: anish.ietf@gmail.com

Mahesh Sivakumar
Cisco Systems
510 McCarthy Boulevard
Milpitas, California
USA

EMail: masivaku@cisco.com

Yisong Liu
Huawei Technologies
Huawei Administration Building
Longgang, Guangdong 518129
China

EMail: liuyisong@huawei.com

Fangwei Hu
ZTE Corporation
889 Bibo Road
Shanghai, Shanghai 201203
China

EMail: hu.fangwei@zte.com.cn