PIM Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 10, 2016

X. Liu
Ericsson
P. McAllister
Metaswitch Networks
A. Peter
Juniper Networks

January 7, 2016

A YANG data model for Protocol-Independent Multicast (PIM)
draft-mcallister-pim-yang-02

Abstract

This document defines a YANG data model that can be used to configure Protocol Independent Multicast (PIM) devices.

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 July 10, 2016.

Copyright Notice

Copyright (c) 2016 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. Requirements Language 3
1.2. Terminology 3
2. Design of Data Model 3
2.1. Scope of model 3
2.2. Optional capabilities 3
2.3. Top-level structure 4
2.4. Position of address family in hierarchy 5
3. Unresolved Issues 5
3.1. Current status of work in progress 5
3.2. Group range mappings 5
3.3. Issues blocked on other model designers 6
4. Module Structure 6
4.1. PIM base module 6
4.2. PIM RP module 10
4.3. PIM-SM module 13
4.4. PIM-DM module 14
4.5. PIM-BIDIR module 15
5. PIM YANG Modules 16
5.1. PIM base module 16
5.2. PIM RP module 36
5.3. PIM-SM module 50
5.4. PIM-DM module 57
5.5. PIM-BIDIR module 59
6. TODO list 67
7. Security Considerations 68
8. IANA Considerations 68
9. Acknowledgements 68
10. References 68
10.1. Normative References 68
10.2. Informative References 68
Authors' Addresses 69

1. Introduction

YANG [RFC6020] [RFC6087] is a data definition language that was introduced to model the configuration and running state of a device managed using NETCONF [RFC6241]. YANG is now also being used as a component of wider management interfaces, such as CLIs.

This document defines a draft YANG data model that can be used to configure and manage Protocol-Independent Multicast (PIM) devices. Currently this model is incomplete, but it will support the core PIM protocol, as well as many other features mentioned in separate PIM RFCs. Non-core features are defined as optional in the provided data model.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Terminology

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

This draft employs YANG tree diagrams, which are explained in [I-D.ietf-netmod-rfc6087bis].

2. Design of Data Model

2.1. Scope of model

The model covers PIM Sparse Mode [RFC4601], including the Source-Specfic subset [RFC3569], Dense Mode [RFC3973], and Bi-directional PIM [RFC5015].

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

The representation of some of these features is not completely specified in this draft of the data model. This model is being circulated in its current form for early oversight and review of the basic hierarchy.

The operational state fields and notifications of this model are also incomplete, though the structure of what has been written may be taken as representative of the structure of the model when complete.

This model does not cover other multicast protocols such as IGMP/MLD, MSDP, mVPN, or m-LDP in-band signalling. It does not cover any configuration required to generate the MRIB. These will be covered by future Internet Drafts.

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 draft 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 restrictions that might be required. Vendors may also extend the features list with proprietary extensions.

2.3. 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.

2.4. Position of address family in hierarchy

The current draft contains address-family as a node in the hierarchy multiple times: both under the interface list, and under the PIM instance. This is similar to the IS-IS yang draft model.

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 must 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']/
interface[interface=current()/sibling:interface]/dr-priority" {
error-app-tag dr-priority-mismatch; error-message "Error: IPv6 DR
priority must match IPv4 DR priority "; }
```

3. Unresolved Issues

3.1. Current status of work in progress

The model so far details how the PIM modules interact and covers the higher levels of their hierarchy. Some details of interface configuration, RP configuration, and PIM-ASM-specific parameters are also complete.

For a list of the most substantial areas still to cover, please see the "TODO list" section below.

3.2. Group range mappings

There is currently no convenient way in the operational state model to map from a group address to the PIM mode and RP information that it will be forwarded according to, which complicates reasoning about the running state and the diagnosis of conflicting policy configuration.

A hypothetical group range state reporting object indexed by group range would be desirable for this purpose, but would be inconvenient for many implementations that index the relevant information on RP address (not applicable for DM or unroutable group-ranges), and difficult to express directly in YANG (as it is difficult to express a container indexed on an arbitrary, proprietary policy structure).

3.3. Issues blocked on other model designers

Some questions must be resolved with reference to other yang models, and so the resolution may be blocked on a decision from another body. Currently these issues are:

1. The position of BFD in the configuration; does the BFD model augment various different protocols, or do the protocols have to augment themselves individually to support BFD?

2. The abstract concepts of a "set of IP addresses" or "set of IP prefixes", as represented in implementations by prefixes, ACLs or policy statements, is a general concept out of the scope of this document. It will presumably have a yang data-type the instantiation of which is vendor-specific.

4. Module Structure

4.1. 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:

1. Our current direction is to agree to a routing-instance-centric (VRF) model 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 will augment "/rt:routing/rt:routing-instance/rt:routing-protocols:" as opposed to augmenting "/rt:routing/rt:routing-instance/rt:routing-protocols:/rt:routing-protocol" as the latter would allow multiple protocol instances per VRF, which does not make sense for PIM.

```
         module: ietf-pim-base
augment /rt:routing/rt:routing-instance/rt:routing-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
```

```
      +--rw interfaces
         +--rw interface* [interface]
            +--rw interface         if:interface-ref
            +--rw address-family* [address-family]
               +--rw address-family       identityref
               +--rw bfd
               |  +--rw enabled?                    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-multipler)?
               |  +--:(holdtime) {intf-hello-holdtime}?
               |  |  +--rw hello-holdtime?      timer-value
               |  +--:(multipler) {intf-hello-multipler}?
               |     +--rw hello-multipler?     uint8
               +--rw jp-interval?         timer-value {intf-jp-interval}?
               +--rw (jp-holdtime-or-multipler)?
               |  +--:(holdtime) {intf-jp-holdtime}?
               |  |  +--rw jp-holdtime?         timer-value
               |  +--:(multipler) {intf-jp-multipler}?
               |     +--rw jp-multipler?        uint8
               +--rw propagation-delay?   uint16 {intf-propagation-delay}?
               +--rw override-interval?   uint16 {intf-override-interval}?
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
   +--ro pim
      +--ro address-family* [address-family]
      |  +--ro address-family        identityref
      |  +--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 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 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 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-addr is-rpt]
      |     |  +--ro group                 inet:ipv4-address
      |     |  +--ro source-addr           union
      |     |  +--ro is-rpt                boolean
      |     |  +--ro expire?               uint32
      |     |  +--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 expire?     timer-value
      |     |     +--ro up-time?    uint32
      |     |     +--ro jp-state?   enumeration
      |     +--ro ipv6-route* [group source-addr is-rpt]
      |        +--ro group                 inet:ipv6-address
      |        +--ro source-addr           union
      |        +--ro is-rpt                boolean
      |        +--ro expire?               uint32
      |        +--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 expire?     timer-value
      |           +--ro up-time?    uint32
      |           +--ro jp-state?   enumeration
      +--ro interfaces
         +--ro interface* [interface]
            +--ro interface         if:interface-ref
            +--ro address-family* [address-family]
               +--ro address-family       identityref
               +--ro bfd
               |  +--ro enabled?                    boolean
               |  +--ro local-multiplier?           multiplier
               |  +--ro (interval-config-type)?
               |     +--:(tx-rx-intervals)
               |     |  +--ro desired-min-tx-interval     uint32
               |     |  +--ro required-min-rx-interval    uint32
               |     +--:(single-interval)
               |        +--ro min-interval                uint32
               +--ro dr-priority?         uint32 {intf-dr-priority}?
               +--ro hello-interval?      timer-value {intf-hello-interval}?
               +--ro (hello-holdtime-or-multipler)?
               |  +--:(holdtime) {intf-hello-holdtime}?
               |  |  +--ro hello-holdtime?      timer-value
               |  +--:(multipler) {intf-hello-multipler}?
               |     +--ro hello-multipler?     uint8
               +--ro jp-interval?         timer-value {intf-jp-interval}?
               +--ro (jp-holdtime-or-multipler)?
               |  +--:(holdtime) {intf-jp-holdtime}?
               |  |  +--ro jp-holdtime?         timer-value
               |  +--:(multipler) {intf-jp-multipler}?
               |     +--ro jp-multipler?        uint8
               +--ro propagation-delay?   uint16 {intf-propagation-delay}?
               +--ro override-interval?   uint16 {intf-override-interval}?
               +--ro ipv4
               |  +--ro address*   inet:ipv4-address
               |  +--ro dr-addr?   inet:ipv4-address
               +--ro ipv6
               |  +--ro address*   inet:ipv6-address
               |  +--ro dr-addr?   inet:ipv6-address
               +--ro oper-status?         enumeration
               +--ro hello-expire?        timer-value
               +--ro neighbor-ipv4* [address]
               |  +--ro address        inet:ipv4-address
               |  +--ro bfd-status?    enumeration
               |  +--ro expire?        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 expire?        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 routing-instance-state-ref?   rt:routing-instance-state-ref
   |  +--ro interface-state-ref?          leafref
   |  +--ro interface-af-state-ref?       leafref
   |  +--ro neighbor-ipv4-state-ref?      leafref
   |  +--ro neighbor-ipv6-state-ref?      leafref
   |  +--ro up-time?                      uint32
   +---n pim-interface-event
      +--ro event-type?                   interface-event-type
      +--ro routing-instance-state-ref?   rt:routing-instance-state-ref
      +--ro interface-state-ref?          leafref
      +--ro ipv4
      |  +--ro address*   inet:ipv4-address
      |  +--ro dr-addr?   inet:ipv4-address
      +--ro ipv6
         +--ro address*   inet:ipv6-address
         +--ro dr-addr?   inet:ipv6-address
```

4.2. 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:routing-instance/rt:routing-protocols/pim-base:pim/
pim-base:address-family:
   +--rw rp
      +--rw static-rp
      |  +--rw ipv4-rp* [ipv4-addr]
      |  |  +--rw ipv4-addr    inet:ipv4-address
      |  +--rw ipv6-rp* [ipv6-addr]
      |     +--rw ipv6-addr    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
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:address-family:
   +--ro rp
      +--ro static-rp
      |  +--ro ipv4-rp* [ipv4-addr]
      |  |  +--ro ipv4-addr    inet:ipv4-address
      |  +--ro ipv6-rp* [ipv6-addr]
      |     +--ro ipv6-addr    inet:ipv6-address
      +--ro bsr {bsr}?
      |  +--ro bsr-candidate!
      |  |  +--ro (interface-or-address)?
      |  |  |  +--:(interface) {candidate-interface}?
      |  |  |  |  +--ro interface           if:interface-ref
      |  |  |  +--:(ipv4-address) {candidate-ipv4}?
      |  |  |  |  +--ro ipv4-address        inet:ipv4-address
      |  |  |  +--:(ipv6-address) {candidate-ipv6}?
      |  |  |     +--ro ipv6-address        inet:ipv6-address
      |  |  +--ro hash-mask-length    uint8
      |  |  +--ro priority            uint8
      |  +--ro rp-candidate-interface* [interface] {candidate-interface}?
      |  |  +--ro interface    if:interface-ref
      |  |  +--ro policy?      string
      |  |  +--ro mode?        identityref
      |  +--ro rp-candidate-ipv4-address* [ipv4-address] {candidate-ipv4}?
      |  |  +--ro ipv4-address    inet:ipv4-address
      |  |  +--ro policy?         string
      |  |  +--ro mode?           identityref
      |  +--ro rp-candidate-ipv6-address* [ipv6-address] {candidate-ipv6}?
      |  |  +--ro ipv6-address    inet:ipv6-address
```

```
      |  |  +--ro policy?         string
      |  |  +--ro mode?           identityref
      |  +--ro bsr
      |  |  +--ro addr?               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-addr mode]
      |  |  +--ro ipv4-addr           inet:ipv4-address
      |  |  +--ro mode                identityref
      |  |  +--ro info-source-addr?   inet:ipv4-address
      |  |  +--ro info-source-type?   identityref
      |  |  +--ro up-time?            uint32
      |  |  +--ro expire?             pim-base:timer-value
      |  +--ro ipv6-rp* [ipv6-addr mode]
      |     +--ro ipv6-addr           inet:ipv6-address
      |     +--ro mode                identityref
      |     +--ro info-source-addr?   inet:ipv6-address
      |     +--ro info-source-type?   identityref
      |     +--ro up-time?            uint32
      |     +--ro expire?             pim-base:timer-value
      +--ro rp-mappings
         +--ro ipv4-rp* [group rp-addr]
         |  +--ro group      inet:ipv4-prefix
         |  +--ro rp-addr    inet:ipv4-address
         |  +--ro up-time?   uint32
         |  +--ro expire?    pim-base:timer-value
         +--ro ipv6-rp* [group rp-addr]
            +--ro group      inet:ipv6-prefix
            +--ro rp-addr    inet:ipv6-address
            +--ro up-time?   uint32
            +--ro expire?    pim-base:timer-value
notifications:
   +---n pim-rp-event
      +--ro event-type?                   rp-event-type
      +--ro routing-instance-state-ref?   rt:routing-instance-state-ref
```

```
      +--ro instance-af-state-ref?        leafref
      +--ro group?                        inet:ip-address
      +--ro rp-address?                   inet:ip-address
      +--ro is-rpt?                       boolean
      +--ro mode?                         pim-base:pim-mode
      +--ro message-origin?               inet:ip-address
```

4.3. PIM-SM module

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

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

```
   +--ro sm
      +--ro asm
      |  +--ro anycast-rp!
      |  |  +--ro ipv4
      |  |  |  +--ro ipv4-anycast-rp* [anycast-addr rp-addr]
      |  |  |     +--ro anycast-addr    inet:ipv4-address
      |  |  |     +--ro rp-addr         inet:ipv4-address
      |  |  +--ro ipv6
      |  |     +--ro ipv6-anycast-rip* [anycast-addr rp-addr]
      |  |        +--ro anycast-addr    inet:ipv6-address
      |  |        +--ro rp-addr         inet:ipv6-address
      |  +--ro spt-switch
      |     +--ro infinity! {spt-switch-infinity}?
      |        +--ro policy-name?   string {spt-switch-policy}?
      +--ro ssm!
         +--ro range-poligy?   string
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:interfaces/pim-base:interface/
pim-base:address-family:
   +--ro sm!
      +--ro passive?   empty
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/
pim-rp:ipv4-rp:
   +--ro sm!
      +--ro policy-name?   string
      +--ro override?      boolean {static-rp-override}?
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:address-family/pim-rp:rp/pim-rp:static-rp/
pim-rp:ipv6-rp:
   +--ro sm!
      +--ro policy-name?   string
      +--ro override?      boolean {static-rp-override}?
```

4.4. PIM-DM module

This module will cover Dense Mode configuration.

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

4.5. PIM-BIDIR module

This module will cover Bidirectional PIM configuration.

```
          module: ietf-pim-bidir
augment /rt:routing/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:address-family:
   +--rw bidir
augment /rt:routing/rt:routing-instance/rt:routing-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-multipler?    uint8
augment /rt:routing/rt:routing-instance/rt:routing-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:routing-instance/rt:routing-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-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:address-family:
   +--ro bidir
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols/
pim-base:pim/pim-base:interfaces/pim-base:interface/
pim-base:address-family:
```

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

5. PIM YANG Modules

5.1. PIM base module

```
   <CODE BEGINS> file "ietf-pim-base.yang"

   module ietf-pim-base {
     namespace "urn:ietf:params:xml:ns:yang:ietf-pim-base";
     // replace with IANA namespace when assigned
```

```
     prefix pim-base;

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

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

     import ietf-interfaces {
       prefix "if";
     }

     import ietf-routing {
       prefix "rt";
     }

     import ietf-bfd {
       prefix "bfd";
     }

     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>

        Editors:   ";

     description
       "The module defines a collection of YANG definitions common for
       all PIM modes.";

     revision 2015-12-07 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }
```

```
/*
 * Features
 */
feature bfd-protocol-parms {
  description
    "BFD protocol specific parameters support.";
}

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

feature intf-dr-priority {
  description
    "Support configuration of interface dr 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-multipler {
  description
    "Support configuration of interface hello multipler.";
}

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-multipler {
  description
    "Support configuration of interface join prune multipler.";
}
```

```
     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 AF configuraiont 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 was elected on the connected network.";
         }
         enum new-df {
           description
             "A new DF 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
            "Source-Specific Multicast (SSM) with PIM Sparse Mode.";
        }
        enum asm {
          description
            "Any Source Multicast (ASM) with PIM Sparse Mode.";
        }
        enum bidir {
          description
          "Bidirectional PIM.";
        }
        enum dm {
          description
            "PIM Dense Mode.";
        }
        enum other {
          description
            "Any other PIM mode.";
        }
      }
      description
        "The PIM mode in which a group is operating.";
    }

    typedef timer-value {
      type union {
        type uint16;
        type enumeration {
          enum "infinity" {
            description "The timer is set to infinity.";
          }
          enum "no-expiry" {
            description "The timer is not set.";
          }
```

```
        }
      }
      units seconds;
      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 {
        description "BFD operation.";
        leaf enabled {
```

```
            type boolean;
            description
              "True if BFD is enabled for the interface.";
          }
          uses bfd:bfd-grouping-base-cfg-parms {
            if-feature bfd-protocol-parms;
          }
        }
        leaf dr-priority {
          if-feature intf-dr-priority;
          type uint32;
          description "DR priority";
        }
        leaf hello-interval {
          if-feature intf-hello-interval;
          type timer-value;
          description "Hello interval";
        }
        choice hello-holdtime-or-multipler {
          description "Use holdtime or multipler";
          case holdtime {
            if-feature intf-hello-holdtime;
            leaf hello-holdtime {
              type timer-value;
              description "Hello holdtime";
            }
          }
          case multipler {
            if-feature intf-hello-multipler;
            leaf hello-multipler {
              type uint8;
              description "Hello multipler";
            }
          }
        }
        leaf jp-interval {
          if-feature intf-jp-interval;
          type timer-value;
          description "Join prune interval";
        }
        choice jp-holdtime-or-multipler {
          description "Use holdtime or multipler";
          case holdtime {
            if-feature intf-jp-holdtime;
            leaf jp-holdtime {
              type timer-value;
              description "Join prune holdtime";
            }
```

```
        }
        case multipler {
          if-feature intf-jp-multipler;
          leaf jp-multipler {
            type uint8;
            description "Join prune multipler";
          }
        }
      }
      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.";
        }
        description "";
        leaf-list address {
          type inet:ipv4-address;
          description "";
        }
        leaf dr-addr {
          type inet:ipv4-address;
          description "";
        }
      }
      container ipv6 {
        when "../../../address-family = 'rt:ipv6'" {
          description
            "Only applicable to ipv6 address family.";
        }
        description "";
        leaf-list address {
```

```
          type inet:ipv6-address;
          description "";
        }
        leaf dr-addr {
          type inet:ipv6-address;
          description "";
        }
      }
      uses interface-state-af-attributes;
    } // interface-state-attributes

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

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

      leaf hello-expire {
        type timer-value;
        description "Hello interval exiration time.";
      }

      list neighbor-ipv4 {
        when "../../../address-family = 'rt:ipv4'" {
          description
            "Only applicable to ipv4 address family.";
        }
        key "address";
        description "";
        leaf address {
          type inet:ipv4-address;
          description "";
        }
        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";
          description "";
          leaf address {
            type inet:ipv6-address;
            description "";
          }
          uses neighbor-state-af-attributes;
        } // list neighbor-ipv4
      } // interface-state-af-attributes

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

        leaf expire {
          type uint32;
          units seconds;
          description "";
        }
        leaf incoming-interface {
          type if:interface-ref;
          description
            "Reference to an entry in the global interface
             list.";
        }
        leaf mode {
          type pim-mode;
          description "";
        }
        leaf msdp-learned {
          type boolean;
          description "";
        }
        leaf rp-address {
          type inet:ip-address;
          description "";
        }
        leaf rpf-neighbor {
          type inet:ip-address;
          description "";
        }
        leaf spt-bit {
          type boolean;
```

```
        description "";
      }
      leaf up-time {
        type uint32;
        units seconds;
        description "";
      }
      list outgoing-interface {
        key "name";
        description
          "A list of outgoing interfaces.";

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

        leaf expire {
          type timer-value;
          description "Expiring information.";
        }

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

        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 "";
        }
      }
    } // multicast-route-attributes

    grouping neighbor-state-af-attributes {
      description
        "A grouping defining neighbor per af attributes.";
      leaf bfd-status {
        type enumeration {
          enum up {
            description
            "";
          }
          enum down {
            description
            "";
          }
        }
        description "";
      }
      leaf expire {
        type timer-value;
        description "Neighbor expiring information.";
      }
      leaf dr-priority {
        type uint32;
        description "DR priority";
      }
      leaf gen-id {
        type uint32;
        description "Generation ID.";
      }
      leaf up-time {
        type uint32;
        units seconds;
        description "";
      }
    } // 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-state-ref {
       description
         "An absolute reference to a PIM instance.";
       leaf routing-instance-state-ref {
         type rt:routing-instance-state-ref;
         description
           "Reference to the routing instance state.";
       }
     } // pim-instance-state-ref

     grouping pim-instance-af-state-ref {
       description
         "An absolute reference to a PIM instance address family.";
       uses pim-instance-state-ref;
       leaf instance-af-state-ref {
         type leafref {
           path "/rt:routing-state/rt:routing-instance"
             + "[rt:name = current()/../routing-instance-state-ref]/"
             + "rt:routing-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.";
       uses pim-instance-state-ref;
       leaf interface-state-ref {
         type leafref {
           path "/rt:routing-state/rt:routing-instance"
             + "[rt:name = current()/../routing-instance-state-ref]/"
             + "rt:routing-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-state-ref {
         type leafref {
```

```
            path "/rt:routing-state/rt:routing-instance"
              + "[rt:name = current()/../routing-instance-state-ref]/"
              + "rt:routing-protocols/pim-base:pim/pim-base:interfaces/"
              + "pim-base:interface"
              + "[pim-base:interface = "
              + "current()/../interface-state-ref]/"
              + "pim-base:address-family/pim-base:address-family";
          }
          description
            "Reference to a PIM interface address family.";
        }
        leaf neighbor-ipv4-state-ref {
          when "../interface-af-state-ref = 'rt:ipv4'" {
            description "";
          }
          type leafref {
            path "/rt:routing-state/rt:routing-instance"
              + "[rt:name = current()/../routing-instance-state-ref]/"
              + "rt:routing-protocols/pim-base:pim/pim-base:interfaces/"
              + "pim-base:interface"
              + "[pim-base:interface = "
              + "current()/../interface-state-ref]/"
              + "pim-base:address-family"
              + "[pim-base:address-family = "
              + "current()/../interface-af-state-ref]/"
              + "pim-base:neighbor-ipv4/pim-base:address";
          }
          description
            "Reference to a PIM IPv4 neighbor.";
        }
        leaf neighbor-ipv6-state-ref {
          when "../interface-af-state-ref = 'rt:ipv6'" {
            description "";
          }
          type leafref {
            path "/rt:routing-state/rt:routing-instance"
              + "[rt:name = current()/../routing-instance-state-ref]/"
              + "rt:routing-protocols/pim-base:pim/pim-base:interfaces/"
              + "pim-base:interface"
              + "[pim-base:interface = "
              + "current()/../interface-state-ref]/"
              + "pim-base:address-family"
              + "[pim-base:address-family = "
              + "current()/../interface-af-state-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 {
        description "";
        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 "";
          uses statistics-error;
        }
        container queue {
          description "";
          uses statistics-queue;
        }
        container received {
          description "";
          uses statistics-sent-received;
        }
        container sent {
          description "";
          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 messages.";
        }
        leaf candidate-rp-advertisement {
          type yang:counter64;
          description
            "The number of Candidate-RP-advertisement 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:routing-instance/"
      + "rt:routing-protocols" {
      description
        "PIM augmentation to routing instance configuration.";

      container pim {
        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;

        } // 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;
            } // address-family
          } // interface
        } // interfaces
      } // pim
    } // augment

    /*
     * Operational state data nodes
     */

    augment "/rt:routing-state/rt:routing-instance/"
      + "rt:routing-protocols" {
      description
        "PIM augmentation to routing instance state.";
      container pim {
        description
          "PIM state data.";

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

          uses statistics-container;

          container topology-tree-info {
            description "";
            list ipv4-route {
              when "../../../address-family = 'rt:ipv4'" {
                description
                  "Only applicable to ipv4 address family.";
              }
              key "group source-addr is-rpt";
              description "";
              leaf group {
                type inet:ipv4-address;
                description "";
              }
              leaf source-addr {
                type union {
                  type enumeration {
                    enum '*' {
```

```
                    description "";
                  }
                }
                type inet:ipv4-address;
              }
              description "";
            }
            leaf is-rpt {
              type boolean;
              description "";
            }

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

          list ipv6-route {
            when "../../../address-family = 'rt:ipv6'" {
              description
                "Only applicable to ipv4 address family.";
            }
            key "group source-addr is-rpt";
            description "";
            leaf group {
              type inet:ipv6-address;
              description "";
            }
            leaf source-addr {
              type union {
                type enumeration {
                  enum '*' {
                    description "";
                  }
                }
                type inet:ipv4-address;
              }
              description "";
            }
            leaf is-rpt {
              type boolean;
              description "";
            }

            uses multicast-route-attributes;
          } // ipv6-route
        } // routes
      } // 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

    /*
     * RPCs
     */

    /*
     * 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 "";
      }
    }
    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 "";
        leaf-list address {
          type inet:ipv4-address;
          description "";
        }
        leaf dr-addr {
          type inet:ipv4-address;
          description "";
        }
      }
      container ipv6 {
        description "";
        leaf-list address {
          type inet:ipv6-address;
          description "";
        }
        leaf dr-addr {
          type inet:ipv6-address;
          description "";
        }
      }
    }
  }

  <CODE ENDS>
```

5.2. PIM RP module

```
  <CODE BEGINS> file "ietf-pim-rp.yang"

  module ietf-pim-rp {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pim-rp";
    // replace with IANA namespace when assigned
    prefix pim-rp;

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

    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>

        Editors:   ";

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

     revision 2015-12-09 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }

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

     feature bsr-election-state {
       description
         "This feature indicates that the system supports providing
          BSR election state.";
     }
```

```
     feature static-rp-override {
       description
         "This feature indicates that the system supports configuration
         of static RP 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 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
         "Gouring 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 "../../../../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 "../../../../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..32";
          }
          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 "../../../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 "../../../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
        "Gouring of BSR state attributes.";
      container bsr {
        description
          "BSR information.";
        leaf addr {
          type inet:ip-address;
          description "BSR address";
        }
        leaf hash-mask-length {
          type uint8;
          description "";
        }
        leaf priority {
          type uint8 {
            range "0..255";
          }
          description "";
        }
        leaf up-time {
          type uint32;
          units seconds;
          description "";
        }
      }
      choice election-state {
        if-feature bsr-election-state;
        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 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;
```

```
        description "";
      }

      container rp {
        description
          "State information of the RP.";
        leaf rp-address {
          type inet:ip-address;
          description "";
        }
        leaf group-policy {
          type string;
          description "";
        }
        leaf up-time {
          type uint32;
          units seconds;
          description "";
        }
      }
      leaf rp-candidate-next-advertisement {
        type uint16;
        units seconds;
        description "";
      }
    } // bsr-state-attributes

    grouping rp-mapping-state-attributes {
      description
        "Gouring of RP mapping attributes.";
      leaf up-time {
        type uint32;
        units seconds;
        description "";
      }
      leaf expire {
        type pim-base:timer-value;
        description "";
      }
    } // rp-mapping-state-attributes

    grouping rp-state-attributes {
      description
        "Gouring of RP state attributes.";
      leaf info-source-type {
        type identityref {
          base rp-info-source-type;
        }
```

```
        description "";
      } // info-source-type
      leaf up-time {
        type uint32;
        units seconds;
        description "";
      }
      leaf expire {
        type pim-base:timer-value;
        description "";
      }
    } // rp-state-attributes

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

```
                "Specifies a static RP address.";
            }
          }

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

      grouping rp-candidate-attributes {
        description
          "Gouring of RP candidate attributes.";
        leaf policy {
          type string;
          description
            "ACL 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:routing-instance/"
        + "rt:routing-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;
        } // bsr
      } // rp
    } // augment

    /*
     * Operational state data nodes
     */

    augment "/rt:routing-state/rt:routing-instance/"
      + "rt:routing-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description
        "PIM SM state.";

      container rp {
        description
          "PIM RP state 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 {
          description
            "Containing a list RPs.";
          list ipv4-rp {
            when "../../../address-family = 'rt:ipv4'" {
              description
                "Only applicable to ipv4 address family.";
            }
            key "ipv4-addr mode";
            description
```

```
              "A list of IPv4 RP addresses.";
            leaf ipv4-addr {
              type inet:ipv4-address;
              description
                "RP address.";
            }
            leaf mode {
              type identityref {
                base rp-mode;
              }
              description
                "RP mode.";
            }
            leaf info-source-addr {
              type inet:ipv4-address;
              description
                "The address where RP information is learned.";
            }
            uses rp-state-attributes;
          }

          list ipv6-rp {
            when "../../../address-family = 'rt:ipv6'" {
              description
                "Only applicable to ipv6 address family.";
            }
            key "ipv6-addr mode";
            description
              "A list of IPv6 RP addresses.";
            leaf ipv6-addr {
              type inet:ipv6-address;
              description
                "RP address.";
            }
            leaf mode {
              type identityref {
                base rp-mode;
              }
              description
                "RP mode.";
            }
            leaf info-source-addr {
              type inet:ipv6-address;
              description
                "The address where RP information is learned.";
            }
            uses rp-state-attributes;
          }
```

```
        } // rp-list

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

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

```
     } // augment

     /*
      * RPCs
      */

     /*
      * 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 inet:ip-address;
         description "";
       }
       leaf rp-address {
         type inet:ip-address;
         description "";
       }
       leaf is-rpt {
         type boolean;
         description "";
       }
       leaf mode {
         type pim-base:pim-mode;
         description "";
       }
       leaf message-origin {
         type inet:ip-address;
         description "";
       }
     }
   }

   <CODE ENDS>
```

5.3. PIM-SM module

```
   <CODE BEGINS> file "ietf-pim-sm.yang"

   module ietf-pim-sm {
```

```
namespace "urn:ietf:params:xml:ns:yang:ietf-pim-sm";
// replace with IANA namespace when assigned
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>

   Editors:   ";

description
  "The YANG module defines a sparse mode PIM model.";

revision 2015-12-09 {
  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 to the
         spt.";
     }

     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 (Spars Mode).";
     }

     /*
      * Groupings
      */
     grouping af-sm-container {
       description
         "Gouping 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.";
             description
               "Anycast RP attributes.";

             container ipv4 {
               when "../../../../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-addr rp-addr";
                description
                  "A list of anycast RP setttings.";
                leaf anycast-addr {
                  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-addr {
                  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 "../../../../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-rip {
                key "anycast-addr rp-addr";
                description
                  "A list of anycast RP setttings.";
                leaf anycast-addr {
                  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-addr {
                  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-switch is set to infinity.";
              description
                "The receiver's dr 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-poligy {
            type string;
            description
              "Policy used to define SSM address range.";
          }
        } // ssm
      } // sm
    } // af-sm-container

    grouping interface-sm-container {
      description
        "Gouping 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 out of the PIM
            interface, but the interface can be included in a multicast
            forwarding entry.";
        }
      } // sm
    } // interface-sm-container

    grouping static-rp-sm-container {
      description
        "Gouping 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:routing-instance/"
      + "rt:routing-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description "PIM SM augmentation.";

      uses af-sm-container;
    } // augment

    augment "/rt:routing/rt:routing-instance/"
      + "rt:routing-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:routing-instance/"
      + "rt:routing-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:routing-instance/"
      + "rt:routing-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

    /*
     * Operational state data nodes
     */

    augment "/rt:routing-state/rt:routing-instance/"
      + "rt:routing-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description
        "PIM SM state.";

      uses af-sm-container;
    } // augment

    augment "/rt:routing-state/rt:routing-instance/"
      + "rt:routing-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-state/rt:routing-instance/"
      + "rt:routing-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-state/rt:routing-instance/"
      + "rt:routing-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

    /*
     * RPCs
     */

    /*
     * Notifications
     */
  }

  <CODE ENDS>
```

5.4. PIM-DM module

```
  <CODE BEGINS>

  module ietf-pim-dm {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pim-dm";
    // replace with IANA namespace when assigned
    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>
```

```
        Editors:   ";

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

     revision 2015-10-08 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: A YANG Data Model for PIM";
     }

     /*
      * Features
      */

     /*
      * Identities
      */

     /*
      * Groupings
      */

     /*
      * Configuration data nodes
      */

     augment "/rt:routing/rt:routing-instance/"
       + "rt:routing-protocols/pim-base:pim/"
       + "pim-base:address-family" {
       description "PIM DM augmentation.";

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

     augment "/rt:routing/rt:routing-instance/"
       + "rt:routing-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

    /*
     * Operational state data nodes
     */

    augment "/rt:routing-state/rt:routing-instance/"
      + "rt:routing-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description
        "PIM DM state.";
      container dm {
        description
          "PIM DM state data.";
      } // dm
    } // augment

    /*
     * RPCs
     */

    /*
     * Notifications
     */
  }
  <CODE ENDS>
```

5.5. PIM-BIDIR module

```
  <CODE BEGINS> file "ietf-pim-bidir.yang"

  module ietf-pim-bidir {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pim-bidir";
    // replace with IANA namespace when assigned
    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>

        Editors:   ";

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

     revision 2015-12-22 {
       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 election state type.";
  reference
    "RFC5015: Bidirectional Protocol Independent Multicast
    (BIDIR-PIM).";
}

identity df-state-offer {
  base df-state;
  description
    "Initial election state..";
}

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.";
}

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 static-rp-bidir-container {
       description
         "Gouping that contains BIDIR attributes for static RP.";
       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:routing-instance/"
       + "rt:routing-protocols/pim-base:pim/"
       + "pim-base:address-family" {
       description "PIM BIDIR augmentation.";

       container bidir {
         description
           "PIM BIDIR configuration data.";
       } // bidir
     } // augment

     augment "/rt:routing/rt:routing-instance/"
       + "rt:routing-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.";

         container df-election {
           if-feature intf-df-election;
           description
             "DF election attributes.";
           leaf offer-interval {
             type pim-base:timer-value;
             description "Offer interval";
```

```
          }
          leaf backoff-interval {
            type pim-base:timer-value;
            description "Backoff interval";
          }
          leaf offer-multipler {
            type uint8;
            description "Offer multipler";
          }
        } // df-election
      } // bidir
    } // augment

    augment "/rt:routing/rt:routing-instance/"
      + "rt:routing-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:routing-instance/"
      + "rt:routing-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-state/rt:routing-instance/"
      + "rt:routing-protocols/pim-base:pim/"
      + "pim-base:address-family" {
      description
        "PIM BIDIR state.";

      container bidir {
        description
          "PIM BIDIR state data.";
      } // bidir
    } // augment

    augment "/rt:routing-state/rt:routing-instance/"
```

```
       + "rt:routing-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.";

         container df-election {
           if-feature intf-df-election;
           description
             "DF election attributes.";
           leaf offer-interval {
             type pim-base:timer-value;
             description "Offer interval";
           }
           leaf backoff-interval {
             type pim-base:timer-value;
             description "Backoff interval";
           }
           leaf offer-multipler {
             type uint8;
             description "Offer multipler";
           }
         } // df-election
       } // bidir
     } // augment

     augment "/rt:routing-state/rt:routing-instance/"
       + "rt:routing-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-state/rt:routing-instance/"
       + "rt:routing-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
```

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

       container bidir {
         description
           "PIM BIDIR state data.";
         container df-election {
           description
             "DF election data.";
           list ipv4-rp {
             when "../../../../address-family = 'rt:ipv4'" {
               description
                 "Only applicable to ipv4 address family.";
             }
             key "ipv4-addr";
             description
               "A list of IPv4 RP addresses.";
             leaf ipv4-addr {
               type inet:ipv4-address;
               description
                 "The address of the RP.";
             }
           } // ipv4-rp
           list ipv6-rp {
             when "../../../../address-family = 'rt:ipv6'" {
               description
                 "Only applicable to ipv6 address family.";
             }
             key "ipv6-addr";
             description
               "A list of IPv6 RP addresses.";
             leaf ipv6-addr {
               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 "../../../../address-family = 'rt:ipv4'" {
               description
                 "Only applicable to ipv4 address family.";
```

```
              }
              key "ipv4-addr interface-name";
              description
                "A list of IPv4 RP addresses.";
              leaf ipv4-addr {
                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 "../../../../address-family = 'rt:ipv6'" {
                description
                  "Only applicable to ipv6 address family.";
              }
              key "ipv6-addr interface-name";
              description
                "A list of IPv6 RP addresses.";
              leaf ipv6-addr {
                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

    /*
     * RPCs
     */

    /*
     * Notifications
     */
  }

  <CODE ENDS>
```

6. TODO list

In this draft, several aspects of the model are incomplete. The PIM-DM and BI-DIR modules are just placeholders for now to demonstrate the hierarchy of the model. Other things the draft will include when complete but does not yet include:

- Interaction with BFD protocol

- Constraints (constant ranges, validation)

- State-limiting and policy

- Statistics.

For these subjects, we will employ the same design principles of expressing a highly general model; vendors may use features and add augmentations in order to express which subsets of this general model are valid in their implementations.

7. Security Considerations

The data model defined does not introduce any security implications.

This draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg].

8. IANA Considerations

TBD

9. Acknowledgements

The authors would like to thank Steve Baillargeon, Guo Feng, Hu Fangwei, Robert Kebler, Tanmoy Kundu, Liu Yisong, Mahesh Sivakumar, and Stig Venaas for their valuable contributions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

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

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

[I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-05 (work in progress), October 2015.

10.2. Informative References

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

[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, <http://www.rfc-editor.org/info/rfc3973>.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <http://www.rfc-editor.org/info/rfc4601>.

[RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc5059>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <http://www.rfc-editor.org/info/rfc6087>.

[I-D.ietf-netmod-routing-cfg] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-20 (work in progress), October 2015.

Authors' Addresses

Liu Xufeng
Ericsson
1595 Spring Hill Road, Suite 500
Vienna VA 22182
USA

EMail: xufeng.liu@ericsson.com

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

EMail: pete.mcallister@metaswitch.com

Anish Peter
Juniper Networks
Electra, Exora Business Park
Bangalore, KA 560103
India

EMail: anishp@juniper.net