PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: July 07, 2019

H. Zhao
Ericsson
X. Liu
Volta Networks
Y. Liu
Huawei
M. Sivakumar
Juniper
A. Peter
Individual

January 8, 2019

A Yang Data Model for IGMP and MLD Snooping
draft-ietf-pim-igmp-mld-snooping-yang-07.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

This Internet-Draft will expire on July 07, 2019.

Copyright Notice

Table of Contents

1. Introduction.....3
   1.1. Terminology.....3
   1.2. Tree Diagrams.....3
2. Design of Data Model.....3
   2.1. Overview.....4
   2.2. IGMP Snooping Instances.....4
   2.3. MLD Snooping Instances.....6
   2.4. IGMP and MLD Snooping Instances Reference.....8
   2.5. IGMP and MLD Snooping RPC.....8
3. IGMP and MLD Snooping YANG Module.....9
4. Security Considerations.....31
5. IANA Considerations.....32
6. Normative References.....33
Appendix A. Data Tree Example.....35
   A.1 Bridge scenario.....35
   A.2 L2VPN scenario.....38
Authors' Addresses.....42

1. Introduction

This document defines a YANG [RFC6020] data model for the management of Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

1.1. Terminology

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

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

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).

- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.

- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches [RFC4541].

The goal of this document is to define a data model that provides a common user interface to IGMP and MLD Snooping.

2.1. Overview

The IGMP and MLD Snooping YANG module defined in this document has all the common building blocks for the IGMP and MLD Snooping protocol.

The YANG module includes IGMP and MLD Snooping instance definition, instance reference in the scenario of BRIDGE and L2VPN. The module also includes the RPC methods for clearing IGMP and MLD Snooping group tables.

This YANG module conforms to Network Management Datastore Architecture (NMDA)[RFC8342]. This NMDA architecture provides an architectural framework for datastores as they are used by network management protocols such as NETCONF [RFC6241], RESTCONF [RFC8040] and the YANG [RFC7950] data modeling language.

2.2. IGMP Snooping Instances

The YANG module defines igmp-snooping-instance which augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol.

All the IGMP Snooping related attributes have been defined in the igmp-snooping-instance. The read-write attribute means configurable data. The read-only attribute means state data.

One igmp-snooping-instance could be referenced in one BRIDGE instance or L2VPN instance. One igmp-snooping-instance corresponds to one BRIDGE instance or L2VPN instance.

The value of scenario in igmp-snooping-instance is bridge or l2vpn. When it is bridge, the igmp-snooping-instance will be referenced in the BRIDGE scenario. When it is l2vpn, the igmp-snooping-instance will be referenced in the L2VPN scenario.

The value of bridge-mrouter-interface, l2vpn-mrouter-interface-ac, l2vpn-mrouter-interface-pw are filled by snooping device dynamically. They are different from static-bridge-mrouter-interface, static-l2vpn-mrouter-interface-ac, and static-l2vpn-mrouter-interface-pw which are configured statically.

The attributes under the interfaces show the statistics of IGMP Snooping related packets.

```
module: ietf-igmp-mld-snooping
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw igmp-snooping-instance {feature-igmp-snooping}?
    |  +--rw scenario?                            snooping-scenario-type
    |  +--rw enable?                              boolean
    |  +--rw forwarding-mode?                     enumeration
    |  +--rw explicit-tracking?                   boolean {explicit-tracking}?
    |  +--rw exclude-lite?                        boolean {exclude-lite}?
    |  +--rw send-query?                          boolean
    |  +--rw immediate-leave?                     empty {immediate-leave}?
    |  +--rw last-member-query-interval?          uint16
    |  +--rw query-interval?                      uint16
    |  +--rw query-max-response-time?             uint16
    |  +--rw require-router-alert?                boolean {require-router-alert}
?
    |  +--rw robustness-variable?                 uint8
    |  +--rw static-bridge-mrouter-interface*     if:interface-ref {static-mrout
er-
interface}?
    |  +--rw static-l2vpn-mrouter-interface-ac*   if:interface-ref {static-mrout
er-
interface}?
    |  +--rw static-l2vpn-mrouter-interface-pw*   pw:pseudowire-ref {static-mrou
ter-
interface}?
    |  +--rw version?                             uint8
    |  +--rw querier-source?                      inet:ipv4-address
    |  +--rw static-l2-multicast-group* [group source-addr] {static-l2-multicast
-
group}?
    |  |  +--rw group                       rt-types:ipv4-multicast-group-addre
ss
    |  |  +--rw source-addr                 rt-types:ipv4-multicast-source-addr
ess
    |  |  +--rw bridge-outgoing-interface*  if:interface-ref
    |  |  +--rw l2vpn-outgoing-ac*          if:interface-ref
    |  |  +--rw l2vpn-outgoing-pw*          pw:pseudowire-ref
    |  +--ro entries-count?                      uint32
    |  +--ro bridge-mrouter-interface*           if:interface-ref
    |  +--ro l2vpn-mrouter-interface-ac*         if:interface-ref
    |  +--ro l2vpn-mrouter-interface-pw*         pw:pseudowire-ref
    |  +--ro group* [address]
    |  |  +--ro address          rt-types:ipv4-multicast-group-address
    |  |  +--ro mac-address?     yang:phys-address
    |  |  +--ro expire?          rt-types:timer-value-seconds16
    |  |  +--ro up-time          uint32
    |  |  +--ro last-reporter?   inet:ipv4-address
    |  |  +--ro source* [address]
    |  |     +--ro address                      rt-types:ipv4-multicast-source-a
ddress
    |  |     +--ro bridge-outgoing-interface*   if:interface-ref
    |  |     +--ro l2vpn-outgoing-ac*           if:interface-ref
    |  |     +--ro l2vpn-outgoing-pw*           pw:pseudowire-ref
    |  |     +--ro up-time                      uint32
    |  |     +--ro expire?                      rt-types:timer-value-seconds16
    |  |     +--ro host-count?                  uint32 {explicit-tracking}?
    |  |     +--ro last-reporter?               inet:ipv4-address
    |  |     +--ro host* [host-address] {explicit-tracking}?
    |  |        +--ro host-address        inet:ipv4-address
    |  |        +--ro host-filter-mode    filter-mode-type
```

```
     |  +--ro interfaces
     |     +--ro interface* [name]
     |        +--ro name          if:interface-ref
     |        +--ro statistics
     |           +--ro received
     |           |  +--ro num-query?                  yang:counter64
     |           |  +--ro num-membership-report-v1?   yang:counter64
     |           |  +--ro num-membership-report-v2?   yang:counter64
     |           |  +--ro num-membership-report-v3?   yang:counter64
     |           |  +--ro num-leave?                  yang:counter64
     |           |  +--ro num-non-member-leave?       yang:counter64
     |           |  +--ro num-pim-hello?              yang:counter64
     |           +--ro sent
     |              +--ro num-query?                  yang:counter64
     |              +--ro num-membership-report-v1?   yang:counter64
     |              +--ro num-membership-report-v2?   yang:counter64
     |              +--ro num-membership-report-v3?   yang:counter64
     |              +--ro num-leave?                  yang:counter64
     |              +--ro num-non-member-leave?       yang:counter64
     |              +--ro num-pim-hello?              yang:counter64
```

2.3. MLD Snooping Instances

The YANG module defines mld-snooping-instance which could be referenced in the BRIDGE or L2VPN scenario to enable MLD Snooping.

The mld-snooping-instance is the same as IGMP snooping except changing IPv4 addresses to IPv6 addresses.

```
module: ietf-igmp-mld-snooping
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw igmp-snooping-instance {feature-igmp-snooping}?
    | ...
    +--rw mld-snooping-instance {feature-mld-snooping}?
       +--rw scenario?                           snooping-scenario-type
       +--rw enable?                             boolean
       +--rw forwarding-mode?                    enumeration
       +--rw explicit-tracking?                  boolean {explicit-tracking}?
       +--rw exclude-lite?                       boolean {exclude-lite}?
       +--rw send-query?                         boolean
       +--rw immediate-leave?                    empty {immediate-leave}?
       +--rw last-member-query-interval?         uint16
       +--rw query-interval?                     uint16
       +--rw query-max-response-time?            uint16
       +--rw require-router-alert?               boolean {require-router-alert}
?
       +--rw robustness-variable?                uint8
       +--rw static-bridge-mrouter-interface*    if:interface-ref {static-mrout
er-
interface}?
```

```
       +--rw static-l2vpn-mrouter-interface-ac*   if:interface-ref {static-mrout
er-
interface}?
       +--rw static-l2vpn-mrouter-interface-pw*   pw:pseudowire-ref {static-mrou
ter-
interface}?
       +--rw version?                             uint8
       +--rw querier-source?                      inet:ipv6-address
       +--rw static-l2-multicast-group* [group source-addr] {static-l2-multicast
-
group}?
       |  +--rw group                       rt-types:ipv6-multicast-group-addre
ss
       |  +--rw source-addr                 rt-types:ipv6-multicast-source-addr
ess
       |  +--rw bridge-outgoing-interface*   if:interface-ref
       |  +--rw l2vpn-outgoing-ac*           if:interface-ref
       |  +--rw l2vpn-outgoing-pw*           pw:pseudowire-ref
       +--ro entries-count?                      uint32
       +--ro bridge-mrouter-interface*           if:interface-ref
       +--ro l2vpn-mrouter-interface-ac*         if:interface-ref
       +--ro l2vpn-mrouter-interface-pw*         pw:pseudowire-ref
       +--ro group* [address]
       |  +--ro address          rt-types:ipv6-multicast-group-address
       |  +--ro mac-address?     yang:phys-address
       |  +--ro expire?          rt-types:timer-value-seconds16
       |  +--ro up-time          uint32
       |  +--ro last-reporter?   inet:ipv6-address
       |  +--ro source* [address]
       |     +--ro address                      rt-types:ipv6-multicast-source-a
ddress
       |     +--ro bridge-outgoing-interface*   if:interface-ref
       |     +--ro l2vpn-outgoing-ac*           if:interface-ref
       |     +--ro l2vpn-outgoing-pw*           pw:pseudowire-ref
       |     +--ro up-time                      uint32
       |     +--ro expire?                      rt-types:timer-value-seconds16
       |     +--ro host-count?                  uint32 {explicit-tracking}?
       |     +--ro last-reporter?               inet:ipv6-address
       |     +--ro host* [host-address] {explicit-tracking}?
       |        +--ro host-address        inet:ipv6-address
       |        +--ro host-filter-mode    filter-mode-type
       +--ro interfaces
          +--ro interface* [name]
             +--ro name          if:interface-ref
             +--ro statistics
                +--ro received
                |  +--ro num-query?       yang:counter64
                |  +--ro num-report-v1?   yang:counter64
                |  +--ro num-report-v2?   yang:counter64
                |  +--ro num-done?        yang:counter64
                |  +--ro num-pim-hello?   yang:counter64
                +--ro sent
                   +--ro num-query?       yang:counter64
                   +--ro num-report-v1?   yang:counter64
                   +--ro num-report-v2?   yang:counter64
                   +--ro num-done?        yang:counter64
                   +--ro num-pim-hello?   yang:counter64
```

2.4. IGMP and MLD Snooping Instances Reference

The igmp-snooping-instance could be referenced in the scenario of BRIDGE or L2VPN to configure the IGMP Snooping.

For the BRIDGE scenario this model augments /dot1q:bridges/dot1q:bridge to reference igmp-snooping-instance. It means IGMP Snooping is enabled in the whole bridge.

It also augments /dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan to reference igmp-snooping-instance. It means IGMP Snooping is enabled in the certain VLAN of the bridge.

```
   augment /dot1q:bridges/dot1q:bridge:
       +--rw igmp-snooping-instance?   igmp-snooping-instance-ref
      +--rw mld-snooping-instance?    mld-snooping-instance-ref

   augment /dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:v
lan:
       +--rw igmp-snooping-instance?   igmp-snooping-instance-ref
       +--rw mld-snooping-instance?    mld-snooping-instance-ref
```

For the L2VPN scenario this model augments /ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn to reference igmp-snooping-instance. It means IGMP Snooping is enabled in the specified l2vpn instance.

```
   augment /ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn:
       +--rw igmp-snooping-instance?   igmp-snooping-instance-ref
       +--rw mld-snooping-instance?    mld-snooping-instance-ref
```

The mld-snooping-instance could be referenced in concurrence with igmp-snooping-instance to configure the MLD Snooping.

2.5. IGMP and MLD Snooping RPC

IGMP and MLD Snooping RPC clears the specified IGMP and MLD Snooping group tables.

```
   rpcs:
       +---x clear-igmp-snooping-groups {rpc-clear-groups}?
       |  +---w input
       |     +---w name?     igmp-mld-snooping-instance-ref
       |     |       {feature-igmp-snooping}?
       |     +---w group?    rt-types:ipv4-multicast-group-address
       |     +---w source?   rt-types:ipv4-multicast-source-address
       +---x clear-mld-snooping-groups {rpc-clear-groups}?
          +---w input
             +---w name?     igmp-mld-snooping-instance-ref
             |       {feature-mld-snooping}?
             +---w group?    rt-types:ipv6-multicast-group-address
             +---w source?   rt-types:ipv6-multicast-source-address
```

3. IGMP and MLD Snooping YANG Module

```
<CODE BEGINS> file ietf-igmp-mld-snooping@2019-01-08.yang
module ietf-igmp-mld-snooping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping";

  prefix ims;

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

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

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-routing {
    prefix "rt";
  }

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

  import ietf-l2vpn {
    prefix "l2vpn";
  }

  import ietf-network-instance {
    prefix "ni";
  }
```

```
  import ietf-pseudowires {
    prefix "pw";
  }

  import ieee802-dot1q-bridge {
    prefix "dot1q";
  }

  organization
    "IETF PIM Working Group";

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

     Editors:  Hongji Zhao
               <mailto:hongji.zhao@ericsson.com>

               Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

               Yisong Liu
               <mailto:liuyisong@huawei.com>

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

               Mahesh Sivakumar
               <mailto:sivakumar.mahesh@gmail.com>

    ";

  description
    "The module defines a collection of YANG definitions common for
     all Internet Group Management Protocol (IGMP) and Multicast
     Listener Discovery (MLD) Snooping devices.

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

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

  revision 2019-01-08 {
    description
```

```
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";
  }


  /*
   * Features
   */

  feature feature-igmp-snooping {
    description
      "Support IGMP snooping protocol.";
    reference
      "RFC 4541, Section 1";
  }

  feature feature-mld-snooping {
    description
      "Support MLD snooping protocol.";
    reference
      "RFC 4541, Section 1";
  }

  feature immediate-leave {
    description
      "Support configuration of immediate-leave.";
    reference
      "RFC 2236, Section 10";
  }

  feature require-router-alert {
    description
      "Support configuration of require-router-alert.";
    reference
      "RFC 3376, Section 5.2";
  }

  feature static-l2-multicast-group {
    description
      "Support configuration of L2 multicast static-group.";
    reference
      "RFC 4541, Section 2.1";
  }

  feature static-mrouter-interface {
    description
      "Support configuration of mrouter interface.";
    reference
      "RFC 4541, Section 2.1";
  }
```

```
  feature rpc-clear-groups {
    description
      "Support clearing statistics by RPC for IGMP & MLD snooping.";
    reference
      "RFC 4541, Section 2.1";
  }

  feature explicit-tracking {
    description
      "Support configuration of per instance explicit-tracking.";
    reference
      "RFC 3376, Appendix B";
  }

  feature exclude-lite {
    description
      "Support configuration of per instance exclude-lite.";
    reference
      "RFC 5790, Section 3";
  }

  /* identities */

  identity scenario-type {
    description
      "Base identity for scenario type in IGMP & MLD snooping";
  }

  identity bridge {
    base scenario-type;
    description
      "This identity represents BRIDGE scenario.";
  }

  identity l2vpn {
    base scenario-type;
    description
      "This identity represents L2VPN scenario.";
  }

  identity filter-mode {
    description
      "Base identity for filter mode in IGMP & MLD snooping";
  }

  identity include {
    base filter-mode;
    description
      "This identity represents include mode.";
  }

  identity exclude {
```

```
    base filter-mode;
    description
      "This identity represents exclude mode.";
  }

  identity igmp-snooping {
    base rt:control-plane-protocol;
    description
      "IGMP snooping protocol";
  }

  identity mld-snooping {
    base rt:control-plane-protocol;
    description
      "MLD snooping protocol";
  }

  /*
   * Typedefs
   */

  typedef snooping-scenario-type {
    type identityref {
      base "scenario-type";
    }
    description "The IGMP & MLD snooping scenario type";
  }

  typedef filter-mode-type {
    type identityref {
      base "filter-mode";
    }
    description "The host filter mode";
  }

  typedef igmp-mld-snooping-instance-ref {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol/rt:name";
    }
    description
      "This type is used by data models which need to
       reference IGMP & MLD snooping instance.";
  }



  /*
   * Groupings
   */

  grouping instance-config-attributes-igmp-snooping {
```

```
    description
      "IGMP snooping configuration for each BRIDGE or L2VPN instance.";

    uses instance-config-attributes-igmp-mld-snooping;

    leaf version {
      type uint8 {
        range "1..3";
      }
      default 2;
      description "IGMP snooping version.";
    }

    leaf querier-source {
      type inet:ipv4-address;
      description
        "Use the IGMP snooping querier to support IGMP
         snooping in a VLAN where PIM and IGMP are not configured.
         The IPv4 address is used as source address in messages.";
    }

    list static-l2-multicast-group {
      if-feature static-l2-multicast-group;
      key "group source-addr";
      description
        "A static multicast route, (*,G) or (S,G).";

      leaf group {
        type rt-types:ipv4-multicast-group-address;
        description
          "Multicast group IPv4 address";
      }

      leaf source-addr {
        type rt-types:ipv4-multicast-source-address;
        description
          "Multicast source IPv4 address.";
      }

      leaf-list bridge-outgoing-interface {
        when 'derived-from-or-self(../../scenario,"ims:bridge")';
        type if:interface-ref;
        description "Outgoing interface in BRIDGE forwarding";
      }

      leaf-list l2vpn-outgoing-ac {
        when 'derived-from-or-self(../../scenario,"ims:l2vpn")';
        type if:interface-ref;
        description "Outgoing AC in L2VPN forwarding";
      }

      leaf-list l2vpn-outgoing-pw {
```

```
        when 'derived-from-or-self(../../scenario,"ims:l2vpn")';
        type pw:pseudowire-ref;
        description "Outgoing PW in L2VPN forwarding";
      }
    } // static-l2-multicast-group
  } // instance-config-attributes-igmp-snooping

  grouping instance-config-attributes-igmp-mld-snooping {
    description
      "IGMP and MLD snooping configuration of each VLAN.";

    leaf enable {
      type boolean;
      default false;
        description
          "Set the value to true to enable IGMP & MLD snooping.";
    }

    leaf forwarding-mode {
      type enumeration {
        enum "mac" {
          description
          "MAC-based lookup mode";
        }
        enum "ip"  {
          description
          "IP-based lookup mode";
        }
      }
      default "ip";
      description "The default forwarding mode is ip";
    }

    leaf explicit-tracking {
      if-feature explicit-tracking;
      type boolean;
      default false;
      description
        "Track the IGMP & MLD snooping v3 membership reports
         from individual hosts. It contributes to saving network
         resources and shortening leave latency.";
    }

    leaf exclude-lite {
      if-feature exclude-lite;
      type boolean;
      default false;
      description
        "Track the Lightweight IGMPv3 and MLDv2 protocol report";
      reference "RFC5790";
    }
```

```
    leaf send-query {
      type boolean;
      default false;
      description
        "Enable quick response for topology changes.
         To support IGMP snooping in a VLAN where PIM and IGMP are
         not configured. It cooperates with parameter querier-source.";
    }

    leaf immediate-leave {
      if-feature immediate-leave;
      type empty;
      description
        "When immediate leave is enabled, the IGMP software assumes
         that no more than one host is present on each VLAN port.";
    }

    leaf last-member-query-interval {
      type uint16 {
        range "1..1023";
      }
      units seconds;
      default 1;
      description
        "Last Member Query Interval, which may be tuned to modify
         the leave latency of the network.";
      reference "RFC3376. Sec. 8.8.";
    }

    leaf query-interval {
      type uint16;
      units seconds;
      default 125;
      description
        "The Query Interval is the interval between General Queries
         sent by the Querier.";
      reference "RFC3376. Sec. 4.1.7, 8.2, 8.14.2.";
    }

    leaf query-max-response-time {
      type uint16;
      units seconds;
      default 10;
      description
        "Query maximum response time specifies the maximum time
         allowed before sending a responding report.";
      reference "RFC3376. Sec. 4.1.1, 8.3, 8.14.3.";
    }

    leaf require-router-alert {
      if-feature require-router-alert;
      type boolean;
```

```
      default false;
      description
        "When the value is true, router alert should exist
         in the IP head of IGMP or MLD packet.";
    }

    leaf robustness-variable {
      type uint8 {
        range "1..7";
      }
      default 2;
      description
        "Querier's Robustness Variable allows tuning for the
         expected packet loss on a network.";
      reference "RFC3376. Sec. 4.1.6, 8.1, 8.14.1.";
    }

    leaf-list static-bridge-mrouter-interface {
      when 'derived-from-or-self(../scenario,"ims:bridge")';
      if-feature static-mrouter-interface;
      type if:interface-ref;
      description "static mrouter interface in BRIDGE forwarding";
    }

    leaf-list static-l2vpn-mrouter-interface-ac {
      when 'derived-from-or-self(../scenario,"ims:l2vpn")';
      if-feature static-mrouter-interface;
      type if:interface-ref;
      description
        "static mrouter interface whose type is interface
         in L2VPN forwarding";
    }

    leaf-list static-l2vpn-mrouter-interface-pw {
      when 'derived-from-or-self(../scenario,"ims:l2vpn")';
      if-feature static-mrouter-interface;
      type pw:pseudowire-ref;
      description
        "static mrouter interface whose type is PW
         in L2VPN forwarding";
    }
  } // instance-config-attributes-igmp-mld-snooping

  grouping instance-config-attributes-mld-snooping {
    description "MLD snooping configuration of each VLAN.";

    uses instance-config-attributes-igmp-mld-snooping;

    leaf version {
      type uint8 {
        range "1..2";
      }
```

```
      default 2;
      description "MLD snooping version.";
    }

    leaf querier-source {
      type inet:ipv6-address;
      description
        "Use the MLD snooping querier to support MLD snooping where
         PIM and MLD are not configured. The IPv6 address is used as
         the source address in messages.";
    }

    list static-l2-multicast-group {
      if-feature static-l2-multicast-group;
      key "group source-addr";
      description
        "A static multicast route, (*,G) or (S,G).";

      leaf group {
        type rt-types:ipv6-multicast-group-address;
        description
          "Multicast group IPv6 address";
      }

      leaf source-addr {
        type rt-types:ipv6-multicast-source-address;
        description
          "Multicast source IPv6 address.";
      }

      leaf-list bridge-outgoing-interface {
       when 'derived-from-or-self(../../scenario,"ims:bridge")';
       type if:interface-ref;
       description "Outgoing interface in BRIDGE forwarding";
      }

      leaf-list l2vpn-outgoing-ac {
       when 'derived-from-or-self(../../scenario,"ims:l2vpn")';
       type if:interface-ref;
       description "Outgoing AC in L2VPN forwarding";
      }

      leaf-list l2vpn-outgoing-pw {
       when 'derived-from-or-self(../../scenario,"ims:l2vpn")';
       type pw:pseudowire-ref;
       description "Outgoing PW in L2VPN forwarding";
      }
    } // static-l2-multicast-group
  } // instance-config-attributes-mld-snooping

  grouping instance-state-group-attributes-igmp-mld-snooping {
    description
```

```
      "Attributes for both IGMP and MLD snooping groups.";

    leaf mac-address {
      type yang:phys-address;
      description "Destination MAC address for L2 multicast.";
    }

    leaf expire {
      type rt-types:timer-value-seconds16;
      units seconds;
      description
        "The time left before multicast group timeout.";
    }

    leaf up-time {
      type uint32;
      units seconds;
      mandatory true;
      description
        "The time elapsed since L2 multicast record created.";
    }
  } // instance-state-group-attributes-igmp-mld-snooping


  grouping instance-state-attributes-igmp-snooping {
    description
      "State attributes for IGMP snooping for each instance.";

    uses instance-state-attributes-igmp-mld-snooping;

    list group {

      key "address";

      config false;

      description "IGMP snooping information";

      leaf address {
        type rt-types:ipv4-multicast-group-address;
        description
          "Multicast group IPv4 address";
      }

      uses instance-state-group-attributes-igmp-mld-snooping;

      leaf last-reporter {
        type inet:ipv4-address;
        description
          "Address of the last host which has sent report to join
           the multicast group.";
      }
```

```
      list source {
        key "address";
        description "Source IPv4 address for multicast stream";

        leaf address {
          type rt-types:ipv4-multicast-source-address;
          description "Source IPv4 address for multicast stream";
        }

        uses instance-state-source-attributes-igmp-mld-snooping;

        leaf last-reporter {
          type inet:ipv4-address;
          description
            "Address of the last host which has sent report
             to join the multicast group.";
        }

        list host {
          if-feature explicit-tracking;
          key "host-address";
          description
            "List of multicast membership hosts
             of the specific multicast source-group.";

          leaf host-address {
            type inet:ipv4-address;
            description
              "Multicast membership host address.";
          }
          leaf host-filter-mode {
            type filter-mode-type;
            mandatory true;
            description
              "Filter mode for a multicast membership
               host may be either include or exclude.";
          }
        }// list host

      } // list source
    } // list group
  } // instance-state-attributes-igmp-snooping

  grouping instance-state-attributes-igmp-mld-snooping {

    description
      "State attributes for IGMP & MLD snooping instance.";

    leaf entries-count {
      type uint32;
      config false;
```

```
      description
       "The number of L2 multicast entries in IGMP & MLD snooping";
    }

    leaf-list bridge-mrouter-interface {
      when 'derived-from-or-self(../scenario,"ims:bridge")';
      type if:interface-ref;
      config false;
      description "mrouter interface in BRIDGE forwarding";
    }

    leaf-list l2vpn-mrouter-interface-ac {
      when 'derived-from-or-self(../scenario,"ims:l2vpn")';
      type if:interface-ref;
      config false;
      description
        "mrouter interface whose type is interface
         in L2VPN forwarding";

    }

    leaf-list l2vpn-mrouter-interface-pw {
      when 'derived-from-or-self(../scenario,"ims:l2vpn")';
      type pw:pseudowire-ref;
      config false;
      description
        "mrouter interface whose type is PW in L2VPN forwarding";
    }
  } // instance-config-attributes-igmp-mld-snooping

  grouping instance-state-attributes-mld-snooping {
    description
      "State attributes for MLD snooping of each VLAN.";

    uses instance-state-attributes-igmp-mld-snooping;

    list group {
      key "address";
      config false;
      description "MLD snooping statistics information";

      leaf address {
        type rt-types:ipv6-multicast-group-address;
        description
          "Multicast group IPv6 address";
      }

      uses instance-state-group-attributes-igmp-mld-snooping;

      leaf last-reporter {
        type inet:ipv6-address;
        description
```

```
          "Address of the last host which has sent report
           to join the multicast group.";
      }

      list source {
        key "address";
        description "Source IPv6 address for multicast stream";

        leaf address {
          type rt-types:ipv6-multicast-source-address;
          description "Source IPv6 address for multicast stream";
        }

        uses instance-state-source-attributes-igmp-mld-snooping;

        leaf last-reporter {
         type inet:ipv6-address;
         description
          "Address of the last host which has sent report
           to join the multicast group.";
        }

        list host {
          if-feature explicit-tracking;
          key "host-address";
          description
            "List of multicast membership hosts
             of the specific multicast source-group.";

          leaf host-address {
            type inet:ipv6-address;
            description
              "Multicast membership host address.";
          }
          leaf host-filter-mode {
            type filter-mode-type;
            mandatory true;
            description
              "Filter mode for a multicast membership
               host may be either include or exclude.";
          }
        }// list host
      } // list source
    } // list group
  } // instance-state-attributes-mld-snooping

  grouping instance-state-source-attributes-igmp-mld-snooping {
    description
      "State attributes for IGMP & MLD snooping instance.";

    leaf-list bridge-outgoing-interface {
      when 'derived-from-or-self(../../../scenario,"ims:bridge")';
```

```
      type if:interface-ref;
      description "Outgoing interface in BRIDGE forwarding";
    }

    leaf-list l2vpn-outgoing-ac {
      when 'derived-from-or-self(../../../scenario,"ims:l2vpn")';
      type if:interface-ref;
      description "Outgoing AC in L2VPN forwarding";
    }

    leaf-list l2vpn-outgoing-pw {
      when 'derived-from-or-self(../../../scenario,"ims:l2vpn")';
      type pw:pseudowire-ref;
      description "Outgoing PW in L2VPN forwarding";
    }

    leaf up-time {
      type uint32;
      units seconds;
      mandatory true;
      description
        "The time elapsed since L2 multicast record created";
    }

    leaf expire {
      type rt-types:timer-value-seconds16;
      units seconds;
      description
        "The time left before multicast group timeout.";
    }

    leaf host-count {
      if-feature explicit-tracking;
      type uint32;
      description
        "The number of host addresses.";
    }
  } // instance-state-source-attributes-igmp-mld-snooping

  grouping igmp-snooping-statistics {
    description
      "The statistics attributes for IGMP snooping.";

      leaf num-query {
        type yang:counter64;
        description
          "The number of query messages.";
        reference
          "RFC 2236, Section 2.1";
      }
      leaf num-membership-report-v1 {
        type yang:counter64;
```

```
        description
          "The number of membership report v1 messages.";
        reference
          "RFC 3376, Section 4";
      }
      leaf num-membership-report-v2 {
        type yang:counter64;
        description
          "The number of membership report v2 messages.";
        reference
          "RFC 3376, Section 4";
      }
      leaf num-membership-report-v3 {
        type yang:counter64;
        description
          "The number of membership report v3 messages.";
        reference
          "RFC 3376, Section 4";
      }
      leaf num-leave {
        type yang:counter64;
        description
          "The number of leave messages.";
        reference
          "RFC 3376, Section 4";
      }
      leaf num-non-member-leave {
        type yang:counter64;
        description
          "The number of non member leave messages.";
        reference
          "RFC 3376, Section 4";
      }
      leaf num-pim-hello {
        type yang:counter64;
        description
          "The number of PIM hello messages.";
        reference
          "RFC 7761, Section 4.9";
      }
  } // igmp-snooping-statistics

  grouping mld-snooping-statistics {
    description
      "The statistics attributes for MLD snooping.";

      leaf num-query {
        type yang:counter64;
        description
          "The number of Multicast Listener Query messages.";
        reference
          "RFC 3810, Section 5";
```

```
      }
      leaf num-report-v1 {
        type yang:counter64;
        description
          "The number of Version 1 Multicast Listener Report.";
        reference
          "RFC 3810, Section 5";
      }
      leaf num-report-v2 {
        type yang:counter64;
        description
          "The number of Version 2 Multicast Listener Report.";
        reference
          "RFC 3810, Section 5";
      }
      leaf num-done {
        type yang:counter64;
        description
          "The number of Version 1 Multicast Listener Done.";
        reference
          "RFC 3810, Section 5";
      }
      leaf num-pim-hello {
        type yang:counter64;
        description
          "The number of PIM hello messages.";
        reference
          "RFC 7761, Section 4.9";
      }
  } // mld-snooping-statistics

  grouping igmp-snooping-interface-statistics-attributes {

    description "Interface statistics attributes for IGMP snooping";

    container interfaces {
       config false;

       description
         "Interfaces associated with the IGMP snooping instance";

       list interface {
         key "name";

         description
           "Interfaces associated with the IGMP snooping instance";

         leaf name {
           type if:interface-ref;
           description
             "The name of interface";
```

```
         }

         container statistics {
           description
             "The interface statistics for IGMP snooping";

           container received {
             description
               "Statistics of received IGMP snooping packets.";

             uses igmp-snooping-statistics;
           }
           container sent {
             description
               "Statistics of sent IGMP snooping packets.";

             uses igmp-snooping-statistics;
           }
         }
       }
     }
   }//igmp-snooping-interface-statistics-attributes

   grouping mld-snooping-interface-statistics-attributes {

     description "Interface statistics attributes for MLD snooping";

     container interfaces {
        config false;

        description
          "Interfaces associated with the MLD snooping instance";

        list interface {
          key "name";

          description
            "Interfaces associated with the MLD snooping instance";

          leaf name {
            type if:interface-ref;
            description
              "The name of interface";

          }

          container statistics {
            description
              "The interface statistics for MLD snooping";

            container received {
              description
```

```
                "Statistics of received MLD snooping packets.";

              uses mld-snooping-statistics;
            }
            container sent {
              description
                "Statistics of sent MLD snooping packets.";

              uses mld-snooping-statistics;
            }
          }
        }
    }
  }//mld-snooping-interface-statistics-attributes


  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {

    description
      "IGMP & MLD snooping augmentation to control plane protocol
       configuration and state.";

    /*
      * igmp-snooping-instance
      */

    container igmp-snooping-instance {
      when 'derived-from-or-self(../rt:type, "ims:igmp-snooping")' {
        description
          "This container is only valid for IGMP snooping protocol.";
      }
      if-feature feature-igmp-snooping;
      description
        "IGMP snooping instance to configure the igmp-snooping.";

      leaf scenario {
        type snooping-scenario-type;
        default bridge;
        description
          "The scenario indicates BRIDGE or L2VPN.";
      }

      uses instance-config-attributes-igmp-snooping;

      uses instance-state-attributes-igmp-snooping;

      uses igmp-snooping-interface-statistics-attributes;

    } //igmp-snooping-instance

    /*
```

```
      * mld-snooping-instance
      */

    container mld-snooping-instance {
      when 'derived-from-or-self(../rt:type, "ims:mld-snooping")' {
        description
          "This container is only valid for MLD snooping protocol.";
      }
      if-feature feature-mld-snooping;
      description
        "MLD snooping instance to configure the mld-snooping.";

      leaf scenario {
        type snooping-scenario-type;
        default bridge;
        description
          "The scenario indicates BRIDGE or L2VPN.";
      }

      uses instance-config-attributes-mld-snooping;

      uses instance-state-attributes-mld-snooping;

      uses mld-snooping-interface-statistics-attributes;

    }//mld-snooping-instance
  }



  augment "/dot1q:bridges/dot1q:bridge" {
    description
      "Reference IGMP & MLD snooping instance in BRIDGE scenario";

    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure IGMP snooping instance under bridge view";
    }
    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure MLD snooping instance under bridge view";
    }
  }

  augment "/dot1q:bridges/dot1q:bridge"+
    "/dot1q:component/dot1q:bridge-vlan/dot1q:vlan" {
    description
      "Reference IGMP & MLD snooping instance in BRIDGE scenario";
```

```
    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure IGMP snooping instance under VLAN view";
    }

    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure MLD snooping instance under VLAN view";
    }
  }

  augment "/ni:network-instances/ni:network-instance"+
    "/ni:ni-type/l2vpn:l2vpn" {

    description
      "Reference IGMP & MLD snooping instance in L2VPN scenario";

    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure IGMP snooping instance in L2VPN scenario";
    }
    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;

      description
        "Configure MLD snooping instance in L2VPN scenario";
    }
  }




  /*  RPCs  */

  rpc clear-igmp-snooping-groups {
    if-feature rpc-clear-groups;
    description
      "Clear the specified IGMP snooping cache tables.";

    input {

      leaf name {
        if-feature feature-igmp-snooping;
        type igmp-mld-snooping-instance-ref;
        description
```

```
          "Name of the igmp-snooping-instance";
      }

      leaf group {
        type rt-types:ipv4-multicast-group-address;
        description
          "Multicast group IPv4 address. If it is not specified,
           all IGMP snooping group tables are cleared.";
      }

      leaf source {
        type rt-types:ipv4-multicast-source-address;
        description
          "Multicast source IPv4 address. If it is not specified,
           all IGMP snooping source-group tables are cleared.";
      }
    }
  } // rpc clear-igmp-snooping-groups

  rpc clear-mld-snooping-groups {
    if-feature rpc-clear-groups;
    description
      "Clear the specified MLD snooping cache tables.";

    input {
      leaf name {
        if-feature feature-mld-snooping;
        type igmp-mld-snooping-instance-ref;
        description
          "Name of the mld-snooping-instance";
      }

      leaf group {
        type rt-types:ipv6-multicast-group-address;
        description
          "Multicast group IPv6 address. If it is not specified,
           all MLD snooping group tables are cleared.";
      }

      leaf source {
        type rt-types:ipv6-multicast-source-address;
        description
          "Multicast source IPv6 address. If it is not specified,
           all MLD snooping source-group tables are cleared.";
      }
    }
  } // rpc clear-mld-snooping-groups
}
<CODE ENDS>
```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:igmp-snoopi
ng-
instance
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:mld-snoopin
g-
instance

The subtrees under /dot1q:bridges/dot1q:bridge

/dot1q:bridges/dot1q:bridge/ims:igmp-snooping-instance
/dot1q:bridges/dot1q:bridge/ims:mld-snooping-instance

The subtrees under
/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan

/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan/ims:igm
p-
snooping-instance
/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan/ims:mld
-
snooping-instance

The subtrees under /ni:network-instances/ni:network-instance/ni:ni-
type/l2vpn:l2vpn

/ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn/ims:igmp-snoopi
ng-
instance
/ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn/ims:mld-snoopin
g-
instance

Unauthorized access to any data node of these subtrees can adversely affect the IGMP & MLD Snooping subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:igmp-snoopi
ng-
instance
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:mld-snoopin
g-
instance

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP & MLD Snooping on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The IGMP & MLD Snooping Yang module support the "clear-igmp-snooping-groups" and " clear-mld-snooping-groups" RPCs. If it meets unauthorized RPC operation invocation, the IGMP and MLD Snooping group tables will be cleared unexpectedly.

5. IANA Considerations

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

This document registers the following namespace URIs in the IETF XML

registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

```
--------------------------------------------------------------------

name:         ietf-igmp-mld-snooping

namespace:    urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping

prefix:       ims

reference:    RFC XXXX

--------------------------------------------------------------------
```

6. Normative References

[P802.1Qcp/D2.2] IEEE Approved Draft Standard for Local and Metropolitan Area Networks, "Bridges and Bridged Networks Amendment: YANG Data Model", Mar 2018

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4541] M. Christensen, K. Kimball, F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.

[RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

[RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.

[draft-ietf-pim-igmp-mld-yang-06] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-mld-yang-06, Oct 20, 2017.

[draft-bjorklund-netmod-rfc7223bis-00] M. Bjorklund, "A YANG Data Model for Interface Management", draft-bjorklund-netmod-rfc7223bis-00, August 21, 2017

[draft-bjorklund-netmod-rfc7277bis-00] M. Bjorklund, "A YANG Data Model for IP Management", draft-bjorklund-netmod-rfc7277bis-00, August 21, 2017

[draft-ietf-netmod-revised-datastores-03] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-03, July 3, 2017

[draft-ietf-bess-evpn-yang-02] P.Brissette, A. Sajassi, H. Shah, Z. Li, H. Chen, K. Tiruveedhula, I. Hussain, J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-02, March 13, 2017

[draft-ietf-bess-l2vpn-yang-08] H. Shah, P. Brissette, I. Chen, I. Hussain, B. Wen, K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-06.txt, February 17, 2018

[draft-ietf-rtgwg-ni-model-12] L. Berger, C. Hopps, A. Lindem, X. Liu, "YANG Model for Network Instances", draft-ietf-rtgwg-ni-model-12.txt, March 19, 2018

Appendix A. Data Tree Example

A.1 Bridge scenario

This section contains an example for bridge scenario in the JSON encoding [RFC7951], containing both configuration and state data.

```
              +-----------+
              +  Source   +
              +-----+-----+
                    |
   -----------------+---------------------------
                    |eth1/1
                +---+---+
                +  R1   +
                +-+---+-+
           eth1/2 |     \ eth1/3
                  |      \
                  |       \
                  |        \
                  |         \
           eth2/1 |          \ eth3/1
              +---+---+   +--+---+
              +  R2   +   +  R3  +
              +---+---+   +--+---+
           eth2/2 |           | eth3/2
                  |           |
   ---------------+-----------+-------------------
                  |           |
                  |           |
         +--------+--+   +---+--------+
         + Receiver1 +   +  Receiver2 +
         +-----------+   +------------+
```

The configuration data for R1 in the above figure could be as follows:

```
{
"ietf-interfaces:interfaces":{
  "interface":[
   {
    "name":"eth1/1",
    "type":"iana-if-type:ethernetCsmacd"
   }
  ]
 },
 "ietf-routing:routing":{
  "control-plane-protocols":{
   "control-plane-protocol":[
    {
     "type":"ietf-igmp-mld-snooping:igmp-snooping",
```

```
     "name":"bis1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance":{
       "scenario":"ietf-igmp-mld-snooping:bridge",
       "enable":true
     }
    }
   ]
  }
 },
 "ieee802-dot1q-bridge:bridges":{
  "bridge":[
   {
    "name":"isp1",
    "address":"00-23-ef-a5-77-12",
    "bridge-type":"ieee802-dot1q-bridge:customer-vlan-bridge",
    "component":[
     {
      "name":"comp1",
      "type":"ieee802-dot1q-bridge:c-vlan-component",
      "bridge-vlan":{
       "vlan":[
        {
         "vid":101,
         "ietf-igmp-mld-snooping:igmp-snooping-instance":"bis1"
        }
       ]
      }
     }
    ]
   }
  ]
 }
}
```


The corresponding operational state data for R1 could be as follows:


```
{
 "ietf-interfaces:interfaces": {
  "interface": [
   {
    "name": "eth1/1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
     "discontinuity-time": "2018-05-23T12:34:56-05:00"
    }
   }
  ]
 },
 "ietf-routing:routing": {
  "control-plane-protocols": {
   "control-plane-protocol": [
```

```
    {
     "type": "ietf-igmp-mld-snooping:igmp-snooping",
     "name": "bis1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance": {
      "scenario": "ietf-igmp-mld-snooping:bridge",
      "enable": true
     }
    }
   ]
  }
 },
 "ieee802-dot1q-bridge:bridges": {
  "bridge": [
   {
    "name": "isp1",
    "address": "00-23-ef-a5-77-12",
    "bridge-type": "ieee802-dot1q-bridge:customer-vlan-bridge",
    "component": [
     {
      "name": "comp1",
      "type": "ieee802-dot1q-bridge:c-vlan-component",
      "bridge-vlan": {
       "vlan": [
        {
         "vid": 101,
         "ietf-igmp-mld-snooping:igmp-snooping-instance": "bis1"
        }
       ]
      }
     }
    ]
   }
  ]
 }
}
```

A.2 L2VPN scenario

This section contains an example for L2VPN scenario in the JSON encoding [RFC7951], containing both configuration and state data.

```
              +-----------+
              +  Source   +
              +-----+-----+
                    |
   -----------------+---------------------------
                    |eth1/1
                +---+---+
                +  R1   +
                +-+---+-+
          eth1/2 |     \ eth1/3
                 |      \
                 |       \
                 |        \
                 |         \
          eth2/1 |          \ eth3/1
             +---+---+    +-+---+
             +  R2   +----+ R3  +
             +---+---+    +-+---+
          eth2/2 |           | eth3/2
                 |           |
   ---------------+-----------+-------------------
                 |           |
                 |           |
        +--------+--+   +---+--------+
        + Receiver1 +   +  Receiver2 +
        +-----------+   +------------+
```

The configuration data for R1 in the above figure could be as follows:

```
{
"ietf-interfaces:interfaces":{
  "interface":[
   {
    "name":"eth1/1",
    "type":"iana-if-type:ethernetCsmacd"
   }
  ]
 },
 "ietf-pseudowires:pseudowires": {
  "pseudowire": [
   {
    "name": "pw2"
   },
   {
    "name": "pw3"
```

```
    }
   ]
  },
  "ietf-network-instance:network-instances": {
   "network-instance": [
    {
     "name": "vpls1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance": "vis1",
     "ietf-l2vpn:type": "ietf-l2vpn:vpls-instance-type",
     "ietf-l2vpn:signaling-type": "ietf-l2vpn:ldp-signaling",
     "ietf-l2vpn:endpoint": [
      {
       "name": "acs",
       "ac": [
         {
          "name": "eth1/1"
         }
       ]
      },
      {
       "name": "pws",
       "pw": [
         {
          "name": "pw2"
         },
         {
          "name": "pw3"
         }
       ]
      }
     ]
    }
   ]
  },
  "ietf-routing:routing": {
   "control-plane-protocols": {
    "control-plane-protocol": [
     {
      "type": "ietf-igmp-mld-snooping:igmp-snooping",
      "name": "vis1",
      "ietf-igmp-mld-snooping:igmp-snooping-instance": {
       "scenario": "ietf-igmp-mld-snooping:l2vpn",
       "enable": true
      }
     }
    ]
   }
  }
}
```

The corresponding operational state data for R1 could be as follows:


```
{
"ietf-interfaces:interfaces":{
  "interface":[
   {
    "name":"eth1/1",
    "type":"iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
     "discontinuity-time": "2018-05-23T12:34:56-05:00"
    }
   }
  ]
 },
 "ietf-pseudowires:pseudowires": {
  "pseudowire": [
   {
    "name": "pw2"
   },
   {
    "name": "pw3"
   }
  ]
 },
 "ietf-network-instance:network-instances": {
  "network-instance": [
   {
    "name": "vpls1",
    "ietf-igmp-mld-snooping:igmp-snooping-instance": "vis1",
    "ietf-l2vpn:type": "ietf-l2vpn:vpls-instance-type",
    "ietf-l2vpn:signaling-type": "ietf-l2vpn:ldp-signaling",
    "ietf-l2vpn:endpoint": [
     {
      "name": "acs",
      "ac": [
        {
         "name": "eth1/1"
        }
      ]
     },
     {
      "name": "pws",
      "pw": [
        {
         "name": "pw2"
        },
        {
         "name": "pw3"
        }
      ]
     }
```

```
    ]
   }
  ]
 },
 "ietf-routing:routing": {
  "control-plane-protocols": {
   "control-plane-protocol": [
    {
     "type": "ietf-igmp-mld-snooping:igmp-snooping",
     "name": "vis1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance": {
      "scenario": "ietf-igmp-mld-snooping:l2vpn",
      "enable": true
     }
    }
   ]
  }
 }
}
```

Authors' Addresses

Hongji Zhao
Ericsson (China) Communications Company Ltd.
Ericsson Tower, No. 5 Lize East Street,
Chaoyang District Beijing 100102, P.R. China

Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA

EMail: xufeng.liu.ietf@gmail.com

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

Email: liuyisong@huawei.com

Anish Peter
Individual

EMail: anish.ietf@gmail.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA

EMail: sivakumar.mahesh@gmail.com

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: April 07, 2022

H. Zhao
Ericsson
X. Liu
Volta Networks
Y. Liu
China Mobile
M. Sivakumar
Juniper
A. Peter
Individual

October 08, 2021

A Yang Data Model for IGMP and MLD Snooping
draft-ietf-pim-igmp-mld-snooping-yang-20.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

Status of this Memo

Table of Contents

1. Introduction....3
   1.1. Terminology....3
   1.2. Tree Diagrams....3
   1.3. Prefixes in Data Node Names....4
2. Design of Data Model....4
   2.1. Overview....5
   2.2. Optional Capabilities....5
   2.3. Position of Address Family in Hierarchy....6
3. Module Structure....6
   3.1. IGMP Snooping Instances....6
   3.2. MLD Snooping Instances....8
   3.3. Using IGMP and MLD Snooping Instances....10
   3.4. IGMP and MLD Snooping Actions....11
4. IGMP and MLD Snooping YANG Module....11
5. Security Considerations....31
6. IANA Considerations....33
   6.1. XML Registry....33
   6.2. YANG Module Names Registry....33
7. References....34
   7.1. Normative References....34
   7.2. Informative References....35
Appendix A. Data Tree Example....36
Authors' Addresses....39

1. Introduction

This document defines a YANG [RFC7950] data model for the management of Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping [RFC4541] devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

1.1. Terminology

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

and [RFC7950], including:

* augment

* data model

* data node

* identity

* module

The following terminologies are used in this document:

* mrouter: multicast router, which is a router that has multicast routing enabled [RFC4286].

* mrouter interfaces: snooping switch ports where multicast routers are attached [RFC4541].

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

IGMP: Internet Group Management Protocol [RFC3376].

MLD: Multicast Listener Discovery [RFC3810].

1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in

[RFC8340].

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
yang	ietf-yang-types	[RFC6991]
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
rt-types	ietf-routing-types	[RFC8294]
dot1q	ieee802-dot1q-bridge	[dot1Qcp]

Table 1: Prefixes and Corresponding YANG Modules

2. Design of Data Model

An IGMP/MLD snooping switch [RFC4541] analyzes IGMP/MLD packets and sets up forwarding tables for multicast traffic. If a switch does not run IGMP/MLD snooping, multicast traffic will be flooded in the broadcast domain. If a switch runs IGMP/MLD snooping, multicast traffic will be forwarded based on the forwarding tables to avoid wasting bandwidth. The IGMP/MLD snooping switch does not need to run any of the IGMP/MLD protocols. Because the IGMP/MLD snooping is independent of the IGMP/MLD protocols, the data model defined in this document does not augment, or even require, the IGMP/MLD data model defined in [RFC8652].
The model covers considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches [RFC4541].

IGMP and MLD snooping switches do not adhere to the conceptual model that provides the strict separation of functionality between different

communications layers in the ISO model, and instead utilize information in the upper level protocol headers as factors to be considered in processing at the lower levels [RFC4541].

IGMP Snooping switches utilize IGMP, and could support IGMPv1 [RFC1112], IGMPv2 [RFC2236], and IGMPv3 [RFC3376]. MLD Snooping switches utilize MLD, and could support MLDv1 [RFC2710] and MLDv2 [RFC3810]. The goal of this document is to define a data model that provides a common user interface to IGMP and MLD Snooping.

2.1. Overview

The IGMP and MLD Snooping YANG module defined in this document has all the common building blocks for the IGMP and MLD Snooping switches.

The YANG module includes IGMP and MLD Snooping instance definition, using instance in the L2 service type of BRIDGE [dot1Qcp]. It also includes actions for clearing IGMP and MLD Snooping group tables.

The YANG module doesn’t cover L2VPN, which will be specified in a separated document.

2.2. Optional Capabilities

This model is designed to represent the basic capability subsets of IGMP and MLD Snooping. The main design goals of this document are that the basic capabilities described in the model are supported by any major now-existing implementation, and that the configuration of all implementations meeting the specifications is easy to express through some combination of the optional features in the model and simple vendor augmentations.

There is also value in widely supported features being standardized, to provide a standardized way to access these features, to save work for individual vendors, and so that mapping between different vendors’ configuration is not needlessly complicated. Therefore, this model declares 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 IGMP and MLD Snooping implementations.

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.

2.3. Position of Address Family in Hierarchy

IGMP Snooping only supports IPv4, while MLD Snooping only supports IPv6. The data model defined in this document can be used for both IPv4 and IPv6 address families.

This document defines IGMP Snooping and MLD Snooping as separate schema branches in the structure. The benefits are:

* The model can support IGMP Snooping (IPv4), MLD Snooping (IPv6), or both optionally and independently. Such flexibility cannot be achieved cleanly with a combined branch.

* The structure is consistent with other YANG data models such as [RFC8652], which uses separate branches for IPv4 and IPv6.

* Having separate branches for IGMP Snooping and MLD Snooping allows minor differences in their behavior to be modelled more simply and cleanly. The two branches can better support different features and node types.

3. Module Structure

This model augments the core routing data model specified in [RFC8349].

```
      +--rw routing
         +--rw router-id?
         +--rw control-plane-protocols
         |  +--rw control-plane-protocol* [type name]
         |     +--rw type
         |     +--rw name
         |     +--rw igmp-snooping-instance <= Augmented by this Model
                        ...
         |     +--rw mld-snooping-instance  <= Augmented by this Model
                        ...
```

The "igmp-snooping-instance" container instantiates an IGMP Snooping Instance. The "mld-snooping-instance" container instantiates an MLD Snooping Instance.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407].

3.1. IGMP Snooping Instances

The YANG module ietf-igmp-mld-snooping augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the igmp-snooping-instance container.

All the IGMP Snooping related attributes have been defined in the igmp-snooping-instance. The read-write attributes represent configurable data. The read-only attributes represent state data.

One igmp-snooping-instance could be used in one BRIDGE [dot1Qcp] instance, and it corresponds to one BRIDGE instance.

Currently the value of l2-service-type in igmp-snooping-instance could only be set bridge. After it is set, igmp-snooping-instance could be used in the BRIDGE service.

The values of bridge-mrouter-interface is filled by the snooping device dynamically. It is different from static-bridge-mrouter-interface which is configured.

The attributes under the interfaces show the statistics of IGMP Snooping related packets.

```
augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw igmp-snooping-instance {igmp-snooping}?
       +--rw l2-service-type?                   l2-service-type
       +--rw enable?                            boolean
       +--rw forwarding-table-type?             enumeration
       +--rw explicit-tracking?                 boolean
       |       {explicit-tracking}?
       +--rw lite-exclude-filter?               empty
       |       {lite-exclude-filter}?
       +--rw send-query?                        boolean
       +--rw fast-leave?                        empty {fast-leave}?
       +--rw last-member-query-interval?        uint16
       +--rw query-interval?                    uint16
       +--rw query-max-response-time?           uint16
       +--rw require-router-alert?              boolean
       |       {require-router-alert}?
       +--rw robustness-variable?               uint8
       +--rw static-bridge-mrouter-interface*   if:interface-ref
       |       {static-mrouter-interface}?
       +--rw igmp-version?                      uint8
       +--rw querier-source?                    inet:ipv4-address
       +--rw static-l2-multicast-group* [group source-addr]
       |       {static-l2-multicast-group}?
       |  +--rw group
       |  |       rt-types:ipv4-multicast-group-address
       |  +--rw source-addr
       |  |       rt-types:ipv4-multicast-source-address
       |  +--rw bridge-outgoing-interface*   if:interface-ref
       +--ro entries-count?                     yang:gauge32
       +--ro bridge-mrouter-interface*          if:interface-ref
       +--ro group* [address]
       |  +--ro address
```

```
       |  |       rt-types:ipv4-multicast-group-address
       |  +--ro mac-address?     yang:phys-address
       |  +--ro expire?          rt-types:timer-value-seconds16
       |  +--ro up-time          uint32
       |  +--ro last-reporter?   inet:ipv4-address
       |  +--ro source* [address]
       |     +--ro address
       |     |       rt-types:ipv4-multicast-source-address
       |     +--ro bridge-outgoing-interface*   if:interface-ref
       |     +--ro up-time                      uint32
       |     +--ro expire?
       |     |       rt-types:timer-value-seconds16
       |     +--ro host-count?                  yang:gauge32
       |     |       {explicit-tracking}?
       |     +--ro last-reporter?               inet:ipv4-address
       |     +--ro host* [address] {explicit-tracking}?
       |        +--ro address        inet:ipv4-address
       |        +--ro filter-mode    filter-mode-type
       +--ro interfaces
          +--ro interface* [name]
             +--ro name          if:interface-ref
             +--ro statistics
                +--ro discontinuity-time?   yang:date-and-time
                +--ro received
                |  +--ro query-count?                  yang:counter64
                |  +--ro membership-report-v1-count?   yang:counter64
                |  +--ro membership-report-v2-count?   yang:counter64
                |  +--ro membership-report-v3-count?   yang:counter64
                |  +--ro leave-count?                  yang:counter64
                |  +--ro pim-hello-count?              yang:counter64
                +--ro sent
                   +--ro query-count?                  yang:counter64
                   +--ro membership-report-v1-count?   yang:counter64
                   +--ro membership-report-v2-count?   yang:counter64
                   +--ro membership-report-v3-count?   yang:counter64
                   +--ro leave-count?                  yang:counter64
                   +--ro pim-hello-count?              yang:counter64
```

3.2. MLD Snooping Instances

The YANG module ietf-igmp-mld-snooping augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the mld-snooping-instance container. The mld-snooping-instance could be used in the BRIDGE [dot1Qcp] service to enable MLD Snooping.

All the MLD Snooping related attributes have been defined in the mld-snooping-instance. The read-write attributes represent configurable data. The read-only attributes represent state data.

The mld-snooping-instance has similar structure as IGMP snooping. Some of leaves are protocol related. The mld-snooping-instance uses IPv6 addresses and mld-version, while igmp-snooping-instance uses IPv4 addresses and igmp-version. Statistic counters in each of the above snooping instances are also tailored to the specific protocol type. One mld-snooping-instance could be used in one BRIDGE instance, and it corresponds to one BRIDGE instance.

Currently the value of l2-service-type in mld-snooping-instance could only be set bridge. After it is set, mld-snooping-instance could be used in the BRIDGE service.

The value of bridge-mrouter-interface is filled by the snooping device dynamically. It is different from static-bridge-mrouter-interface which is configured.

The attributes under the interfaces show the statistics of MLD Snooping related packets.

```
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw mld-snooping-instance {mld-snooping}?
       +--rw l2-service-type?                   l2-service-type
       +--rw enable?                            boolean
       +--rw forwarding-table-type?             enumeration
       +--rw explicit-tracking?                 boolean
       |       {explicit-tracking}?
       +--rw lite-exclude-filter?               empty
       |       {lite-exclude-filter}?
       +--rw send-query?                        boolean
       +--rw fast-leave?                        empty {fast-leave}?
       +--rw last-member-query-interval?        uint16
       +--rw query-interval?                    uint16
       +--rw query-max-response-time?           uint16
       +--rw require-router-alert?              boolean
       |       {require-router-alert}?
       +--rw robustness-variable?               uint8
       +--rw static-bridge-mrouter-interface*   if:interface-ref
       |       {static-mrouter-interface}?
       +--rw mld-version?                       uint8
       +--rw querier-source?                    inet:ipv6-address
       +--rw static-l2-multicast-group* [group source-addr]
       |       {static-l2-multicast-group}?
       |  +--rw group
       |  |       rt-types:ipv6-multicast-group-address
       |  +--rw source-addr
       |  |       rt-types:ipv6-multicast-source-address
       |  +--rw bridge-outgoing-interface*   if:interface-ref
       +--ro entries-count?                     yang:gauge32
       +--ro bridge-mrouter-interface*          if:interface-ref
       +--ro group* [address]
```

```
       |  +--ro address
       |  |       rt-types:ipv6-multicast-group-address
       |  +--ro mac-address?     yang:phys-address
       |  +--ro expire?          rt-types:timer-value-seconds16
       |  +--ro up-time          uint32
       |  +--ro last-reporter?   inet:ipv6-address
       |  +--ro source* [address]
       |     +--ro address
       |     |       rt-types:ipv6-multicast-source-address
       |     +--ro bridge-outgoing-interface*   if:interface-ref
       |     +--ro up-time                      uint32
       |     +--ro expire?
       |     |       rt-types:timer-value-seconds16
       |     +--ro host-count?                  yang:gauge32
       |     |       {explicit-tracking}?
       |     +--ro last-reporter?               inet:ipv6-address
       |     +--ro host* [address] {explicit-tracking}?
       |        +--ro address        inet:ipv6-address
       |        +--ro filter-mode    filter-mode-type
       +--ro interfaces
          +--ro interface* [name]
             +--ro name          if:interface-ref
             +--ro statistics
                +--ro discontinuity-time?   yang:date-and-time
                +--ro received
                |  +--ro query-count?       yang:counter64
                |  +--ro report-v1-count?   yang:counter64
                |  +--ro report-v2-count?   yang:counter64
                |  +--ro done-count?        yang:counter64
                |  +--ro pim-hello-count?   yang:counter64
                +--ro sent
                   +--ro query-count?       yang:counter64
                   +--ro report-v1-count?   yang:counter64
                   +--ro report-v2-count?   yang:counter64
                   +--ro done-count?        yang:counter64
                   +--ro pim-hello-count?   yang:counter64
```

3.3. Using IGMP and MLD Snooping Instances

The igmp-snooping-instance could be used in the service of BRIDGE [dot1Qcp] to configure the IGMP Snooping.

For the BRIDGE service this model augments /dot1q:bridges/dot1q:bridge to use igmp-snooping-instance. It means IGMP Snooping is enabled in the whole bridge.

It also augments /dot1q:bridges/dot1q:bridge/dot1q:component/ dot1q:bridge-vlan/dot1q:vlan to use igmp-snooping-instance. It means IGMP Snooping is enabled in the specified VLAN on the bridge.

The mld-snooping-instance could be used in concurrence with igmp-snooping-instance to configure the MLD Snooping.

```
     augment /dot1q:bridges/dot1q:bridge:
       +--rw igmp-snooping-instance?   igmp-mld-snooping-instance-ref
       +--rw mld-snooping-instance?    igmp-mld-snooping-instance-ref

     augment /dot1q:bridges/dot1q:bridge/dot1q:component
               /dot1q:bridge-vlan/dot1q:vlan:
       +--rw igmp-snooping-instance?   igmp-mld-snooping-instance-ref
       +--rw mld-snooping-instance?    igmp-mld-snooping-instance-ref
```

3.4. IGMP and MLD Snooping Actions

IGMP and MLD Snooping actions clear the specified IGMP and MLD Snooping group tables. If both source X and group Y are specified, only source X from group Y in that specific instance will be cleared.

```
augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw igmp-snooping-instance {igmp-snooping}?
      +---x clear-igmp-snooping-groups {action-clear-groups}?
          +---w input
             +---w group     union
             +---w source    rt-types:ipv4-multicast-source-address

augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw mld-snooping-instance {mld-snooping}?
       +---x clear-mld-snooping-groups {action-clear-groups}?
          +---w input
             +---w group     union
             +---w source    rt-types:ipv6-multicast-source-address
```

4. IGMP and MLD Snooping YANG Module

This module references [RFC1112],[RFC2236],[RFC2710],[RFC3376], [RFC3810],[RFC4541],[RFC5790],[RFC6636],[RFC6991],[RFC7761], [RFC8343],[dot1Qcp].

```
<CODE BEGINS> file ietf-igmp-mld-snooping@2021-10-08.yang
module ietf-igmp-mld-snooping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping";

  prefix ims;

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

```
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-yang-types {
    prefix "yang";
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-interfaces {
    prefix "if";
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management (NMDA
       Version)";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  import ieee802-dot1q-bridge {
    prefix "dot1q";
    reference
      "dot1Qcp: IEEE 802.1Qcp-2018 Bridges and Bridged Networks
                - Amendment: YANG Data Model";
  }

  organization
    "IETF PIM Working Group";

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

     Editors:  Hongji Zhao
               <mailto:hongji.zhao@ericsson.com>

               Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

               Yisong Liu
               <mailto:liuyisong@chinamobile.com>
```

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

               Mahesh Sivakumar
               <mailto:sivakumar.mahesh@gmail.com>

    ";

  description
    "The module defines a collection of YANG definitions common for
     all devices that implement Internet Group Management Protocol
     (IGMP) and Multicast Listener Discovery (MLD) Snooping which is
     described in RFC 4541.

     Copyright (c) 2021 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

  revision 2021-10-08 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";
  }

  /*
   * Features
   */

  feature igmp-snooping {
    description
      "Support IGMP snooping.";
    reference
      "RFC 4541";
  }

  feature mld-snooping {
    description
      "Support MLD snooping.";
    reference
      "RFC 4541";
```

```
  }

  feature fast-leave {
    description
      "Support configuration of fast leave. The fast leave feature
       does not send last member query messages to hosts.";
    reference
      "RFC 3376";
  }

  feature static-l2-multicast-group {
    description
      "Support configuration of static L2 multicast group.";
  }

  feature static-mrouter-interface {
    description
      "Support multicast router interface explicitly configured
       by management";
    reference
      "RFC 4541";
  }

  feature action-clear-groups {
    description
      "Support clearing statistics by action for IGMP & MLD snooping.";
  }

  feature require-router-alert {
    description
      "Support configuration of require-router-alert.";
    reference
      "RFC 3376";
  }

  feature lite-exclude-filter {
    description
      "Enable the support of the simplified EXCLUDE filter.";
    reference
      "RFC 5790";
  }

  feature explicit-tracking {
    description
      "Support configuration of per instance explicit-tracking.";
    reference
      "RFC 6636";
  }

  /* identities */
```

```
  identity l2-service-type {
    description
      "Base identity for L2 service type in IGMP & MLD snooping";
  }

  identity bridge {
    base l2-service-type;
    description
      "This identity represents BRIDGE service.";
  }

  identity filter-mode {
    description
      "Base identity for filter mode in IGMP & MLD snooping";
  }

  identity include {
    base filter-mode;
    description
      "This identity represents include mode.";
  }

  identity exclude {
    base filter-mode;
    description
      "This identity represents exclude mode.";
  }

  identity igmp-snooping {
    base rt:control-plane-protocol;
    description
      "IGMP snooping";
  }

  identity mld-snooping {
    base rt:control-plane-protocol;
    description
      "MLD snooping";
  }

  /*
   * Typedefs
   */

  typedef l2-service-type {
    type identityref {
      base "l2-service-type";
    }
    description "The L2 service type used with IGMP & MLD snooping ";
  }
```

```
  typedef filter-mode-type {
    type identityref {
      base "filter-mode";
    }
    description "The host filter mode";
  }

  typedef igmp-mld-snooping-instance-ref {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol/rt:name";
    }
    description
      "This type is used by data models which need to
       reference IGMP & MLD snooping instance.";
  }

  /*
   * Groupings
   */

  grouping instance-config-attributes-igmp-mld-snooping {
    description
      "IGMP and MLD snooping configuration of each VLAN.";

    leaf enable {
      type boolean;
      default false;
        description
          "Set the value to true to enable IGMP & MLD snooping.";
    }

    leaf forwarding-table-type {
      type enumeration {
        enum "mac" {
          description
          "MAC-based lookup mode";
        }
        enum "ip"  {
          description
          "IP-based lookup mode";
        }
      }
      default "ip";
      description "The default forwarding table type is ip";
    }

    leaf explicit-tracking {
      if-feature explicit-tracking;
      type boolean;
      default false;
```

```
      description
        "Track the IGMPv3 and MLDv2 snooping membership reports
         from individual hosts. It contributes to saving network
         resources and shortening leave latency.";
    }

    leaf lite-exclude-filter {
      if-feature lite-exclude-filter;
      type empty;
      description
        "For IGMP Snooping, the presence of this
         leaf enables the support of the simplified EXCLUDE filter
         in the Lightweight IGMPv3 protocol, which simplifies the
         standard versions of IGMPv3.
         For MLD Snooping, the presence of this
         leaf enables the support of the simplified EXCLUDE filter
         in the Lightweight MLDv2 protocol, which simplifies the
         standard versions of MLDv2.";
      reference
        "RFC 5790";
    }

    leaf send-query {
      type boolean;
      default false;
      description
        "When it is true, this switch will send out periodic
         IGMP General Query Message or MLD General Query Message.";
    }

    leaf fast-leave {
      if-feature fast-leave;
      type empty;
      description
        "When immediate leave is enabled, the IGMP software assumes
         that no more than one host is present on each VLAN port.";
    }

    leaf last-member-query-interval {
      type uint16 {
        range "10..10230";
      }
      units deciseconds;
      default 10;
      description
        "Last Member Query Interval, which may be tuned to modify
         the leave latency of the network.
         It is represented in units of 1/10 second.";
      reference "RFC 3376. Sec. 8.8.";
    }
```

```
    leaf query-interval {
      type uint16;
      units seconds;
      default 125;
      description
        "The Query Interval is the interval between General Queries
         sent by the Querier.";
      reference "RFC 3376. Sec. 4.1.7, 8.2, 8.14.2.";
    }

    leaf query-max-response-time {
      type uint16;
      units deciseconds;
      default 100;
      description
        "Query maximum response time specifies the maximum time
         allowed before sending a responding report.
         It is represented in units of 1/10 second.";
      reference "RFC 3376. Sec. 4.1.1, 8.3, 8.14.3.";
    }

    leaf require-router-alert {
      if-feature require-router-alert;
      type boolean;
      default false;
      description
        "When the value is true, router alert should exist
         in the IP header of IGMP or MLD packet. If it doesn't exist,
         the IGMP or MLD packet will be ignored.";
      reference "RFC 3376. Sec. 9.1, 9.2, 9.3.";
    }

    leaf robustness-variable {
      type uint8 {
        range "1..7";
      }
      default 2;
      description
        "Querier's Robustness Variable allows tuning for the
         expected packet loss on a network.";
      reference "RFC 3376. Sec. 4.1.6, 8.1, 8.14.1.";
    }

    leaf-list static-bridge-mrouter-interface {
      when 'derived-from-or-self(../l2-service-type,"ims:bridge")';
      if-feature static-mrouter-interface;
      type if:interface-ref;
      description "static mrouter interface in BRIDGE forwarding";
    }
  } // instance-config-attributes-igmp-mld-snooping
```

```
  grouping instance-state-group-attributes-igmp-mld-snooping {
    description
      "Attributes for both IGMP and MLD snooping groups.";

    leaf mac-address {
      type yang:phys-address;
      description "Destination MAC address for L2 multicast.";
    }

    leaf expire {
      type rt-types:timer-value-seconds16;
      units seconds;
      description
        "The time left before multicast group timeout.";
    }

    leaf up-time {
      type uint32;
      units seconds;
      mandatory true;
      description
        "The time elapsed since L2 multicast record created.";
    }
  } // instance-state-group-attributes-igmp-mld-snooping

  grouping instance-state-attributes-igmp-mld-snooping {

    description
      "State attributes for IGMP & MLD snooping instance.";

    leaf entries-count {
      type yang:gauge32;
      config false;
      description
        "The number of L2 multicast entries in IGMP & MLD snooping";
    }

    leaf-list bridge-mrouter-interface {
      when 'derived-from-or-self(../l2-service-type,"ims:bridge")';
      type if:interface-ref;
      config false;
      description
        "Indicates a list of mrouter interfaces dynamically learned in a
         bridge. When this switch receives IGMP/MLD queries from a
         multicast router on an interface, the interface will become
         mrouter interface for IGMP/MLD snooping.";
    }
  } // instance-config-attributes-igmp-mld-snooping


  grouping instance-state-source-attributes-igmp-mld-snooping {
```

```
    description
      "State attributes for IGMP & MLD snooping instance.";

    leaf-list bridge-outgoing-interface {
      when 'derived-from-or-self(../../../l2-service-
type,"ims:bridge")';
      type if:interface-ref;
      description "Outgoing interface in BRIDGE forwarding";
    }

    leaf up-time {
      type uint32;
      units seconds;
      mandatory true;
      description
        "The time elapsed since L2 multicast record created";
    }

    leaf expire {
      type rt-types:timer-value-seconds16;
      units seconds;
      description
        "The time left before multicast group timeout.";
    }

    leaf host-count {
      if-feature explicit-tracking;
      type yang:gauge32;
      description
        "The number of host addresses.";
    }
  } // instance-state-source-attributes-igmp-mld-snooping

  grouping igmp-snooping-statistics {
    description
      "The statistics attributes for IGMP snooping.";

      leaf query-count {
        type yang:counter64;
        description
          "The number of Membership Query messages.";
        reference
          "RFC 2236";
      }
      leaf membership-report-v1-count {
        type yang:counter64;
        description
          "The number of Version 1 Membership Report messages.";
        reference
          "RFC 1112";
      }
```

```
      leaf membership-report-v2-count {
        type yang:counter64;
        description
          "The number of Version 2 Membership Report messages.";
        reference
          "RFC 2236";
      }
      leaf membership-report-v3-count {
        type yang:counter64;
        description
          "The number of Version 3 Membership Report messages.";
        reference
          "RFC 3376";
      }
      leaf leave-count {
        type yang:counter64;
        description
          "The number of Leave Group messages.";
        reference
          "RFC 2236";
      }
      leaf pim-hello-count {
        type yang:counter64;
        description
          "The number of PIM hello messages.";
        reference
          "RFC 7761";
      }
  } // igmp-snooping-statistics

  grouping mld-snooping-statistics {
    description
      "The statistics attributes for MLD snooping.";

      leaf query-count {
        type yang:counter64;
        description
          "The number of Multicast Listener Query messages.";
        reference
          "RFC 3810";
      }
      leaf report-v1-count {
        type yang:counter64;
        description
          "The number of Version 1 Multicast Listener Report.";
        reference
          "RFC 2710";
      }
      leaf report-v2-count {
        type yang:counter64;
        description
```

```
          "The number of Version 2 Multicast Listener Report.";
        reference
          "RFC 3810";
      }
      leaf done-count {
        type yang:counter64;
        description
          "The number of Version 1 Multicast Listener Done.";
        reference
          "RFC 2710";
      }
      leaf pim-hello-count {
        type yang:counter64;
        description
          "The number of PIM hello messages.";
        reference
          "RFC 7761";
      }
  } // mld-snooping-statistics

  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {
    when 'derived-from-or-self(rt:type, "ims:igmp-snooping")' {
      description
        "This container is only valid for IGMP snooping.";
    }
    description
      "IGMP snooping augmentation to control plane protocol
       configuration and state.";

    container igmp-snooping-instance {
      if-feature igmp-snooping;
      description
        "IGMP snooping instance to configure igmp-snooping.";

      leaf l2-service-type {
        type l2-service-type;
        default bridge;
        description
          "It indicates BRIDGE or other services.";
      }

      uses instance-config-attributes-igmp-mld-snooping;

      leaf igmp-version {
        type uint8 {
          range "1..3";
        }
        default 2;
        description "IGMP version.";
      }
```

```
      leaf querier-source {
        type inet:ipv4-address;
        description
          "The source address of IGMP General Query message,
           which is sent out by this switch.";
      }

      list static-l2-multicast-group {
        if-feature static-l2-multicast-group;
        key "group source-addr";
        description
          "A static multicast route, (*,G) or (S,G).";

        leaf group {
          type rt-types:ipv4-multicast-group-address;
          description
            "Multicast group IPv4 address";
        }

        leaf source-addr {
          type rt-types:ipv4-multicast-source-address;
          description
            "Multicast source IPv4 address.";
        }

        leaf-list bridge-outgoing-interface {
          when 'derived-from-or-self(../../l2-service-
type,"ims:bridge")';
          type if:interface-ref;
          description "Outgoing interface in BRIDGE forwarding";
        }
      } // static-l2-multicast-group

      uses instance-state-attributes-igmp-mld-snooping;

      list group {

        key "address";

        config false;

        description "IGMP snooping information";

        leaf address {
          type rt-types:ipv4-multicast-group-address;
          description
            "Multicast group IPv4 address";
        }

        uses instance-state-group-attributes-igmp-mld-snooping;
```

```
        leaf last-reporter {
          type inet:ipv4-address;
          description
            "Address of the last host which has sent report to join
             the multicast group.";
        }

        list source {
          key "address";
          description "Source IPv4 address for multicast stream";

          leaf address {
            type rt-types:ipv4-multicast-source-address;
            description "Source IPv4 address for multicast stream";
          }

          uses instance-state-source-attributes-igmp-mld-snooping;

          leaf last-reporter {
            type inet:ipv4-address;
            description
              "Address of the last host which has sent report
               to join the multicast group.";
          }

          list host {
            if-feature explicit-tracking;
            key "address";
            description
              "List of multicast membership hosts
               of the specific multicast source-group.";

            leaf address {
              type inet:ipv4-address;
              description
                "Multicast membership host address.";
            }
            leaf filter-mode {
              type filter-mode-type;
              mandatory true;
              description
                "Filter mode for a multicast membership
                 host may be either include or exclude.";
            }
          }// list host
        } // list source
      } // list group

      container interfaces {
         config false;
```

```
         description
           "Contains the interfaces associated with the IGMP snooping
            instance";

         list interface {
           key "name";

           description
             "A list of interfaces associated with the IGMP snooping
              instance";

           leaf name {
             type if:interface-ref;
             description
               "The name of interface";

           }

           container statistics {
             description
               "The interface statistics for IGMP snooping";

             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 received {
               description
                 "Number of received snooped IGMP packets";

               uses igmp-snooping-statistics;
             }
             container sent {
               description
                 "Number of sent snooped IGMP packets";

               uses igmp-snooping-statistics;
             }
           }
         }
      }

      action clear-igmp-snooping-groups {
```

```
        if-feature action-clear-groups;
        description
          "Clear IGMP snooping cache tables.";

        input {
          leaf group {
            type union {
              type enumeration {
                enum 'all-groups' {
                  description
                    "All multicast group addresses.";
                }
              }
              type rt-types:ipv4-multicast-group-address;
            }
            mandatory true;
            description
              "Multicast group IPv4 address. If value 'all-groups' is
               specified, all IGMP snooping group entries are cleared
               for specified source address.";
          }
          leaf source {
            type rt-types:ipv4-multicast-source-address;
            mandatory true;
            description
              "Multicast source IPv4 address. If value '*' is specified,
               all IGMP snooping source-group tables are cleared.";
          }
        }
      } // action clear-igmp-snooping-groups
    } // igmp-snooping-instance
  } // augment

  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {
    when 'derived-from-or-self(rt:type, "ims:mld-snooping")' {
        description
          "This container is only valid for MLD snooping.";
    }
    description
      "MLD snooping augmentation to control plane protocol
       configuration and state.";

    container mld-snooping-instance {
      if-feature mld-snooping;
      description
        "MLD snooping instance to configure mld-snooping.";

      leaf l2-service-type {
        type l2-service-type;
        default bridge;
```

```
        description
          "It indicates BRIDGE or other services.";
      }

      uses instance-config-attributes-igmp-mld-snooping;

      leaf mld-version {
        type uint8 {
          range "1..2";
        }
        default 2;
        description "MLD version.";
      }

      leaf querier-source {
        type inet:ipv6-address;
        description
          "The source address of MLD General Query message,
           which is sent out by this switch.";
      }

      list static-l2-multicast-group {
        if-feature static-l2-multicast-group;
        key "group source-addr";
        description
          "A static multicast route, (*,G) or (S,G).";

        leaf group {
          type rt-types:ipv6-multicast-group-address;
          description
            "Multicast group IPv6 address";
        }

        leaf source-addr {
          type rt-types:ipv6-multicast-source-address;
          description
            "Multicast source IPv6 address.";
        }

        leaf-list bridge-outgoing-interface {
          when 'derived-from-or-self(../../l2-service-
type,"ims:bridge")';
          type if:interface-ref;
          description "Outgoing interface in BRIDGE forwarding";
        }
      } // static-l2-multicast-group

      uses instance-state-attributes-igmp-mld-snooping;

      list group {
        key "address";
```

```
        config false;
        description "MLD snooping statistics information";

        leaf address {
          type rt-types:ipv6-multicast-group-address;
          description
            "Multicast group IPv6 address";
        }

        uses instance-state-group-attributes-igmp-mld-snooping;

        leaf last-reporter {
          type inet:ipv6-address;
          description
            "Address of the last host which has sent report
             to join the multicast group.";
        }

        list source {
          key "address";
          description "Source IPv6 address for multicast stream";

          leaf address {
            type rt-types:ipv6-multicast-source-address;
            description "Source IPv6 address for multicast stream";
          }

          uses instance-state-source-attributes-igmp-mld-snooping;

          leaf last-reporter {
           type inet:ipv6-address;
           description
            "Address of the last host which has sent report
             to join the multicast group.";
          }

          list host {
            if-feature explicit-tracking;
            key "address";
            description
              "List of multicast membership hosts
               of the specific multicast source-group.";

            leaf address {
              type inet:ipv6-address;
              description
                "Multicast membership host address.";
            }
            leaf filter-mode {
              type filter-mode-type;
              mandatory true;
```

```
              description
                "Filter mode for a multicast membership
                 host may be either include or exclude.";
            }
          }// list host
        } // list source
      } // list group

      container interfaces {
         config false;

         description
           "Contains the interfaces associated with the MLD snooping
            instance";

         list interface {
           key "name";

           description
             "A list of interfaces associated with the MLD snooping
              instance";

           leaf name {
             type if:interface-ref;
             description
               "The name of interface";

           }

           container statistics {
             description
               "The interface statistics for MLD snooping";

             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 received {
               description
                 "Number of received snooped MLD packets";

               uses mld-snooping-statistics;
             }
             container sent {
```

```
               description
                 "Number of sent snooped MLD packets";

               uses mld-snooping-statistics;
             }
           }
         }
      }

      action clear-mld-snooping-groups {
        if-feature action-clear-groups;
        description
          "Clear MLD snooping cache tables.";

        input {
          leaf group {
            type union {
              type enumeration {
                enum 'all-groups' {
                  description
                    "All multicast group addresses.";
                }
              }
              type rt-types:ipv6-multicast-group-address;
            }
            mandatory true;
            description
              "Multicast group IPv6 address. If value 'all-groups' is
               specified, all MLD snooping group entries are cleared
               for specified source address.";
          }
          leaf source {
            type rt-types:ipv6-multicast-source-address;
            mandatory true;
            description
              "Multicast source IPv6 address. If value '*' is specified,
               all MLD snooping source-group tables are cleared.";
          }
        }
      } // action clear-mld-snooping-groups
    }// mld-snooping-instance
  } // augment

  augment "/dot1q:bridges/dot1q:bridge" {
    description
      "Use IGMP & MLD snooping instance in BRIDGE.";

    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure IGMP snooping instance under bridge view";
```

```
    }

    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure MLD snooping instance under bridge view";
    }
  }

  augment "/dot1q:bridges/dot1q:bridge"+
    "/dot1q:component/dot1q:bridge-vlan/dot1q:vlan" {
    description
      "Use IGMP & MLD snooping instance in certain VLAN of BRIDGE";

    leaf igmp-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure IGMP snooping instance under VLAN view";
    }

    leaf mld-snooping-instance {
      type igmp-mld-snooping-instance-ref;
      description
        "Configure MLD snooping instance under VLAN view";
    }
  }
}
<CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

ims:igmp-snooping-instance

ims:mld-snooping-instance

The subtrees under /dot1q:bridges/dot1q:bridge

ims:igmp-snooping-instance

ims:mld-snooping-instance

The subtrees under /dot1q:bridges/dot1q:bridge/dot1q:component /dot1q:bridge-vlan/dot1q:vlan

ims:igmp-snooping-instance

ims:mld-snooping-instance

Unauthorized access to any data node of these subtrees can adversely affect the IGMP & MLD Snooping subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

ims:igmp-snooping-instance

ims:mld-snooping-instance

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP & MLD Snooping on this device. The group/source/host information may expose multicast group memberships, and transitively the associations between the user on the host and the contents from the source which could be privately sensitive. Some of the action operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/

ims:igmp-snooping-instance/ims:clear-igmp-snooping-groups

ims:mld-snooping-instance/ims:clear-mld-snooping-groups

Some of the actions in this YANG module may be considered sensitive or vulnerable in some network environments. The IGMP & MLD Snooping YANG module supports the "clear-igmp-snooping-groups" and "clear-mld-snooping-groups" actions. If unauthorized action is invoked, the IGMP and MLD Snooping group tables will be cleared unexpectedly. Especially when using wildcard, all the multicast traffic will be flooded in the broadcast domain. The devices that use this YANG module should heed the Security Considerations in [RFC4541].

6. IANA Considerations

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

6.1. XML Registry

This document registers the following namespace URIs in the IETF XML

registry [RFC3688]:

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping
Registrant Contact: The IETF.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------
```

6.2. YANG Module Names Registry

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

```
---------------------------------------------------------------------
name:         ietf-igmp-mld-snooping
namespace:    urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping
prefix:       ims
reference:    RFC XXXX
---------------------------------------------------------------------
```

7. References

7.1. Normative References

[dot1Qcp] IEEE, "Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 30: YANG Data Model", IEEE Std 802.1Qcp-2018 (Revision of IEEE Std 802.1Q-2014), September 2018, <https://ieeexplore.ieee.org/servlet/opac?punumber=8467505>

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

[RFC2236] W. Fenner, "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4286] B. Haberman and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.

[RFC4541] M. Christensen, K. Kimball, F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC5790] H. Liu, W. Cao, H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] R. Enns, Ed., M. Bjorklund, Ed., J. Schoenwaelder, Ed., A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6636] H. Asaeda, H. Liu, Q. Wu, "Tuning the Behavior of the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) for Routers in Mobile and Wireless Networks", RFC 6636, May 2012.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.

[RFC7761] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 7761, March 2016.

[RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.

[RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.

[RFC8294] X. Liu, Y. Qu, A. Lindem, C. Hopps, L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, December 2017.

[RFC8340] M. Bjorklund, and L. Berger, Ed., "YANG Tree Diagrams", RFC 8340, March 2018.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.

[RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

[RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.

[RFC8349] L. Lhotka, A. Lindem, Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, March 2018.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.

7.2. Informative References

[RFC7951] L. Lhotka, "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.

[RFC8407] A. Bierman, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", RFC 8407, October 2018.

[RFC8652] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", RFC 8652, November 2019.

Appendix A. Data Tree Example

This section contains an example for bridge service in the JSON encoding [RFC7951], containing both configuration and state data.

```
              +-----------+
              +  Source   +
              +-----+-----+
                    |
   -----------------+---------------------------
                    |eth1/1
                +---+---+
                +  R1   +
                +-+---+-+
         eth1/2 |     \ eth1/3
                |      \
                |       \
                |        \
                |         \
         eth2/1 |          \ eth3/1
            +---+---+   +--+---+
            +  R2   +   +  R3  +
            +---+---+   +--+---+
         eth2/2 |           | eth3/2
                |           |
   ---------------+----------+-------------------
                |           |
                |           |
        +--------+--+   +---+--------+
        + Receiver1 +   +  Receiver2 +
        +-----------+   +------------+
```

The configuration data for R1 in the above figure could be as follows:

```
{
"ietf-interfaces:interfaces":{
  "interface":[
   {
    "name":"eth1/1",
    "type":"iana-if-type:ethernetCsmacd"
   }
  ]
```

```
 },
 "ietf-routing:routing":{
  "control-plane-protocols":{
   "control-plane-protocol":[
    {
     "type":"ietf-igmp-mld-snooping:igmp-snooping",
     "name":"bis1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance":{
       "l2-service-type":"ietf-igmp-mld-snooping:bridge",
       "enable":true
     }
    }
   ]
  }
 },
 "ieee802-dot1q-bridge:bridges":{
  "bridge":[
   {
    "name":"isp1",
    "address":"00-23-ef-a5-77-12",
    "bridge-type":"ieee802-dot1q-bridge:customer-vlan-bridge",
    "component":[
     {
      "name":"comp1",
      "type":"ieee802-dot1q-bridge:c-vlan-component",
      "bridge-vlan":{
       "vlan":[
        {
         "vid":101,
         "ietf-igmp-mld-snooping:igmp-snooping-instance":"bis1"
        }
       ]
      }
     }
    ]
   }
  ]
 }
}
```


The corresponding operational state data for R1 could be as follows:


```
{
 "ietf-interfaces:interfaces": {
  "interface": [
   {
    "name": "eth1/1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "statistics": {
     "discontinuity-time": "2018-05-23T12:34:56-05:00"
```

```
    }
   }
  ]
 },
 "ietf-routing:routing": {
  "control-plane-protocols": {
   "control-plane-protocol": [
    {
     "type": "ietf-igmp-mld-snooping:igmp-snooping",
     "name": "bis1",
     "ietf-igmp-mld-snooping:igmp-snooping-instance": {
      "l2-service-type": "ietf-igmp-mld-snooping:bridge",
      "enable": true
     }
    }
   ]
  }
 },
 "ieee802-dot1q-bridge:bridges": {
  "bridge": [
   {
    "name": "isp1",
    "address": "00-23-ef-a5-77-12",
    "bridge-type": "ieee802-dot1q-bridge:customer-vlan-bridge",
    "component": [
     {
      "name": "comp1",
      "type": "ieee802-dot1q-bridge:c-vlan-component",
      "bridge-vlan": {
       "vlan": [
        {
         "vid": 101,
         "ietf-igmp-mld-snooping:igmp-snooping-instance": "bis1"
        }
       ]
      }
     }
    ]
   }
  ]
 }
}
```


The following action is to clear all the entries whose group address is 225.1.1.1 for igmp-snooping-instance bis1.

```
POST /restconf/operations/ietf-routing:routing/control-plane-protocols/\
control-plane-protocol=ietf-igmp-mld-snooping:igmp-snooping,bis1/\
ietf-igmp-mld-snooping:igmp-snooping-instance/\
clear-igmp-snooping-groups HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
```

```
{
  "ietf-igmp-mld-snooping:input" : {
    "group": "225.1.1.1",
    "source": "*"
  }
}
```

Authors' Addresses

Hongji Zhao
Ericsson (China) Communications Company Ltd.
Ericsson Tower, No. 5 Lize East Street,
Chaoyang District Beijing 100102, China

Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA

EMail: xufeng.liu.ietf@gmail.com

Yisong Liu
China Mobile
China

Email: liuyisong@chinamobile.com

Anish Peter
Individual

EMail: anish.ietf@gmail.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA

EMail: sivakumar.mahesh@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

V. Kamath
VMware
R. Chokkanathapuram Sundaram
Cisco Systems, Inc.

March 11, 2019

PIM Null register packing
draft-ietf-pim-null-register-packing-01

Abstract

In PIM-SM networks PIM registers are sent from the first hop router to the RP (Rendezvous Point) to signal the presence of Multicast source in the network. There are periodic PIM Null registers sent from first hop router to the RP to keep the state alive at the RP as long as the source is active. The PIM Null register packet carries information about a single Multicast source and group. This document defines a standard to send multiple Multicast source and group information in a single pim Null register packet and the interoperability between the PIM routers which do not understand the packet format with multiple Multicast source and group details.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Conventions Used in This Document 2
1.2. Terminology 3
2. PIM Register Stop format with capability option 3
3. New PIM Null register message 4
4. New PIM Register Stop message format 4
5. Protocol operation 5
6. PIM Anycast RP considerations 6
7. IANA Considerations 6
8. Acknowledgments 6
9. References 6
9.1. Normative References 7
9.2. Informative References 7
Authors' Addresses 7

1. Introduction

PIM Null registers are sent by First hop routers periodically for Multicast streams to keep the states active on the RP as long as the Multicast source is alive. As the number of multicast sources increases, the number of PIM Null register packets that are sent increases at a given time. This results in more PIM packet processing at RP and FHR. The control plane policing (COPP), monitors the packets that gets processed by the control plane. Due to the high rate at which Null registers are received at the RP, this can lead to COPP drops of Multicast PIM Null register packets. This draft proposes a method to efficiently pack multiple PIM Null registers and register stop into a single message as these packets anyway don't contain data. The draft also proposes interoperability with the routers that do not understand the new packet format.

1.1. Conventions Used in This Document

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

1.2. Terminology

RP: Rendezvous Point

RPF: Reverse Path Forwarding

SPT: Shortest Path Tree

FHR: First Hop Router, directly connected to the source

LHR: Last Hop Router, directly connected to the receiver

2. PIM Register Stop format with capability option

A router (FHR) can decide to pack multiple Null registers based on the capability received from the RP as part of Register Stop. This ensures compatibility with routers that don't support processing of the new format. The capability information can be indicated by the RP via the PIM register stop message sent to the FHR. Thus a FHR will switch to the new format only when it learns RP is capable of handling the packed Null register messages. Conversely, a FHR that doesn't support the new format can continue generating the PIM Null register the current way. To exchange the capability information in the Register Stop message, the "reserved" field can be used to indicate this capability in those register stop messages. One bit of the reserved field is used to indicate the "packing" capability (P bit). The rest of the bits in the "Reserved" field will be retained for future use.

Figure 1: PIM Register Stop message with capability option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |P|  Reserved   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Group Address (Encoded-Group format)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Address (Encoded-Unicast format)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Reserved, Type, Checksum, Group Address, Source Address
Same as RFC 7761 (Section 4.9.4)

P Capability bit used to indicate support for Packed Null Register

3. New PIM Null register message

New PIM Null register message format includes a count to indicate the number of Null register records in the message.

Figure 2: New PIM Null Register message format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |SubType| Rsvd  |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   count       |              Reserved2                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Group Address[1]  (Encoded-Group format)                  |
|     Source Address[1]  (Encoded-Unicast format)               |
.                                                               .
.                                                               .
.                                                               .
.                                                               .
.     Group Address[N]                                          .
|     Source Address[N]                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Reserved, Checksum
: Same as RFC 7761 (Section 4.9.3)

Type, SubType
: The new packed Null Register Type and SubType values TBD

count
: The count of the number of packed Null register records. A record consists of Group and Source Address

Group Address
: IP address of the Multicast Group

Source Address
: IP Address of the Multicast Source

4. New PIM Register Stop message format

The new PIM register stop is message includes a count to indicate the number of records that are present in the message.

Figure 3: New PIM Register Stop message format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |SubType| Rsvd  |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     count     |          Reserved2                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Group Address[1]  (Encoded-Group format)                  |
|     Source Address[1]  (Encoded-Unicast format)               |
.                                                               .
.                                                               .
.                                                               .
.                                                               .
.     Group Address[N]                                          .
|     Source Address[N]                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Reserved, Checksum
: Same as RFC 7761 (Section 4.9.3)

Type
: The new Register Stop Type and SubType values TBD

Record count
: The count of the number of packed register stop records. A record consists of Group and Source Address

Group Address
: IP address of the Multicast Group

Source Address
: IP Address of the Multicast Source

5. Protocol operation

The following combinations exist -
FHR and RP both support the new PIM Register formats -

a. FHR sends the PIM register towards the RP when a new source is detected
b. RP sends a modified register stop towards the FHR that includes capability
information by setting the P bit (Figure 2)
c. Based on the receipt of new Register Stop, FHR will start packing of Null registers using the new packed register format (Figure 1)
d. RP processes the new Null register message and can generate new register Stop messages by packing multiple S,Gs towards the same FHR (Figure 3)

FHR supports but RP doesn’t support new PIM Register formats-

a. FHR sends the PIM register towards the RP
b. RP sends a normal register stop without any capability information
c. FHR then sends Null registers in the old format

RP supports but FHR doesn’t support the new PIM Register formats-

a. FHR sends the PIM register towards the RP
b. RP sends a modified register stop towards the FHR that includes capability information
c. Since FHR doesn’t support the new format, it sends Null registers in the old format

6. PIM Anycast RP considerations

The new PIM register format should be enabled only if its supported by all PIM anycast RP members in the RP set for the RP address.

7. IANA Considerations

This document requires the assignment of 2 new PIM message types for the packed pim register and pim register stop.

8. Acknowledgments

The authors would like to thank Stig Venaas and Umesh Dudani for contributing to the original idea and also their very helpful comments on the draft.

9. References

9.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

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

9.2. Informative References

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

Authors' Addresses

Vikas Ramesh Kamath
VMware
3401 Hillview Ave
Palo Alto CA 94304
USA

Email: vkamath@vmware.com

Ramakrishnan Chokkanathapuram Sundaram
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: ramaksun@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 14 September 2023

V. Kamath
VMware
R. Chokkanathapuram Sundaram
Cisco Systems, Inc.
R. Banthia
Apstra
A. Gopal
Cisco Systems, Inc.

13 March 2023

PIM Null-Register packing
draft-ietf-pim-null-register-packing-16

Abstract

In PIM-SM networks, PIM Null-Register messages are sent by the Designated Router (DR) to the Rendezvous Point (RP) to signal the presence of Multicast sources in the network. There are periodic PIM Null-Registers sent from the DR to the RP to keep the state alive at the RP as long as the source is active. The PIM Null-Register message carries information about a single Multicast source and group.

This document defines a standard to send multiple Multicast source and group information in a single PIM message. This document refers to the new messages as the PIM Packed Null-Register message and PIM Packed Register-Stop message.

Status of This Memo

Table of Contents

1. Introduction 2
  1.1. Conventions used in this document 3
  1.2. Terminology 3
2. Packed Null-Register Packing Capability 3
3. PIM Packed Null-Register message format 3
4. PIM Packed Register-Stop message format 4
5. Protocol operation 5
6. Operational Considerations 6
  6.1. PIM Anycast RP Considerations 6
  6.2. Interoperability between different versions 6
  6.3. Disabling PIM Packed Message Support at RP and/or DR . . 6
7. Fragmentation Considerations 7
8. Security Considerations 7
9. IANA Considerations 7
10. Acknowledgments 8
11. Normative References 8
Authors' Addresses 9

1. Introduction

The DR periodically sends PIM Null-Registers to keep the state of existing multicast sources active on the RP. As the number of multicast sources increases, the number of PIM Null-Register messages that are sent also increases. This results in more PIM packet processing at the RP and the DR.

This document specifies a method to efficiently pack the content of multiple PIM Null-Register and Register-Stop messages [RFC7761] into a single message.

The document also discusses interoperability between PIM routers that support PIM Packed Null-Registers and PIM Packed Register-Stops and PIM routers that do not.

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

RP: Rendezvous Point

DR: Designated Router

2. Packed Null-Register Packing Capability

The RP indicates its ability to receive PIM Packed Null-Register messages (Section 3) and send PIM Packed Register-Stop messages (Section 4) with a Packing Capability bit (P-bit) in the PIM Register-Stop message. The P-bit is allocated in Section 9.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |P|6 5 4 3 2 1 0|           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Group Address (Encoded-Group format)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Address (Encoded-Unicast format)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: PIM Register-Stop message with Packing Capability option

The fields in the PIM Register-Stop message are defined in Section 4.9.4 of [RFC7761], and the common header in [I-D.venaas-pim-rfc8736bis].

Packing Capability bit (P-bit / Flag Bit TBD1): When set, it indicates the ability of the RP to receive PIM Packed Null-Register messages, and send PIM Packed Register-Stop messages.

3. PIM Packed Null-Register message format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |Subtype|  FB   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Group Address[1]   (Encoded-Group format)                 |
|     Source Address[1]  (Encoded-Unicast format)               |
.                                                               .
.                                                               .
.                                                               .
.                                                               .
.     Group Address[N]                                          .
|     Source Address[N]                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: PIM Packed Null-Register message format

The fields in the PIM Packed Null-Register message are defined in Section 4.9.4 of [RFC7761], and the common header in [I-D.venaas-pim-rfc8736bis]

Type, Subtype: The PIM Packed Null-Register Type value TBD2. [I-D.venaas-pim-rfc8736bis]

N: The total number of records; A record consists of a Group Address and Source Address pair.

After parsing the PIM common header, individual records are then parsed one by one until the length of the PIM Packed Null-Register message. This length is inferred from the IP layer.

Sending or receiving a PIM Packed Null-Register message is the equivalent, for all purposes, of sending or receiving an individual Null-Register message for each record represented in the PIM Packed Null-Register message.

4. PIM Packed Register-Stop message format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |Subtype|  FB   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Group Address[1]  (Encoded-Group format)                  |
|     Source Address[1]  (Encoded-Unicast format)               |
.                                                               .
.                                                               .
.                                                               .
.                                                               .
.     Group Address[N]                                          .
|     Source Address[N]                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: PIM Packed Register-Stop message format

The fields in the PIM Packed Register-Stop message are defined in Section 4.9.4 of [RFC7761], and the common header in [I-D.venaas-pim-rfc8736bis]

Type, Subtype: The PIM Packed Register-Stop Type TBD3

N: The total number of records; A record consists of a Group Address and Source Address pair.

After parsing the PIM common header, individual records are then parsed one by one until the length of the PIM Packed Register-Stop message. This length is inferred from the IP layer.

Sending or receiving a PIM Packed Register-Stop message is the equivalent, for all purposes, of sending or receiving an individual Null-Register message for each record represented in the PIM Packed Register-Stop.

5. Protocol operation

As specified in [RFC7761], the DR sends PIM Register messages towards the RP when a new source is detected.

When this feature is enabled/configured, an RP supporting this specification MUST set the P-bit (Flag bit TBD1) in all Register-Stop messages.

When a Register-Stop message with the P-bit set is received, the DR SHOULD send PIM Packed Null-Register messages (Section 3) to the RP instead of multiple Register messages with the N-bit set [RFC7761]. The DR MAY use a mixture of PIM Packed Null-Register messages and

Register messages. The decision is up to the implementation and out of the scope of this document. However, it is RECOMMENDED to stick to the PIM Packed Null-Register and PIM Packed Register-Stop formats as long as the RP and DR have the feature enabled.

The RP, after receiving a PIM Packed Null-Register message, SHOULD start sending PIM Packed Register-Stop messages (Section 4) to the corresponding DR instead of individual Register-Stop messages. The RP MAY use a mixture of PIM Packed Register-Stop messages and individual Register-Stop messages. The decision is up to the implementation and out of the scope of this document. However, it is RECOMMENDED to stick to the PIM Packed Null-Register and PIM Packed Register-Stop formats as long as the RP and DR have the feature enabled.

6. Operational Considerations

6.1. PIM Anycast RP Considerations

The PIM Packed Null-Register packet format should be enabled only if it is supported by all the routers in the Anycast-RP set [RFC4610]. This consideration applies to PIM Anycast RP with MSDP [RFC3446] as well.

6.2. Interoperability between different versions

A router (DR) can decide to use the PIM Packed Null-Register message format based on the Packing Capability received from the RP as part of the PIM Register-Stop. This ensures compatibility with routers that do not support processing of the new packet format. The Packing Capability information MUST be indicated by the RP via the PIM Register-Stop message sent to the DR. Thus, a DR will switch to the new packet format only when it learns that the RP is capable of handling the PIM Packed Null-Register messages.

Conversely, a DR that does not support the packed format can continue generating the PIM Null-Register as defined in [RFC7761] (Section 4.4).

6.3. Disabling PIM Packed Message Support at RP and/or DR

Consider a PIM RP router that supports PIM Packed Null-Registers and PIM Packed Register-Stops. In scenarios where this router now no longer supports this feature, for example, in case of a software downgrade, it will not send a PIM Register-Stop message to the DR in response to a PIM Packed Null-Register message.

When the DR switches to Data Registers from Null-Registers, it MUST start a Packed_Register_Probe_Time timer. If no PIM Packed Register-Stop or Register-Stop with the P-bit set is received within Packed_Register_Probe_Time seconds, the DR can decide that the RP no longer supports PIM Packed Null-Registers. The Packed_Register_Probe_Time timer is configurable; its default value is 60 seconds.

When Packed_Register_Probe_Time expires, The DR MAY also send an unpacked PIM Null-Register and check the PIM Register-Stop to see if the P-bit is set or not. If it is not set then the DR will continue sending unpacked PIM Null-Register messages.

In case the network manager disables the Packing Capability at the RP, or in other words, disables the feature from the RP, the router MUST NOT advertise the Packing Capability. However, an implementation MAY choose to still parse any packed registers if they are received. This may be particularly useful in the transitional period after the network manager disables it.

7. Fragmentation Considerations

As explained in Section 4.4.1 of [RFC7761], the DR may perform Path MTU Discovery to the RP before sending PIM Packed Null-Register messages. Similarly, the RP may perform Path MTU Discovery to the DR before sending PIM Packed Register-Stop messages. In both cases, the number of records in a message should be limited such that it can fit within the Path MTU.

8. Security Considerations

The Security Considerations from [RFC7761] apply to this document. In particular, the effect of forging a PIM Packed Null-Register or Register-Stop message would be amplified to all the records included instead of just one.

By forging a PIM Register-Stop message and setting the P-bit, an attacker can trigger the use of PIM Packed Null-Register messages by a DR thus creating unnecessary churn in the network.

9. IANA Considerations

When this document is published, IANA is asked to assign a Packing Capability bit (TBD1) in the PIM Register-Stop Common Header from the PIM Message Types registry.

When this document is published, IANA is asked to assign a PIM message type (TBD2) for the PIM Packed Null-Register from the PIM Message Types registry. The Flag Bits (0-3) for PIM message type (TBD2) are requested to be "Unassigned".

When this document is published, IANA is asked to assign a PIM message type (TBD3) for the PIM Packed Register-Stop from the PIM Message Types registry. The Flag Bits (0-3) for PIM message type (TBD3) are requested to be "Unassigned".

10. Acknowledgments

The authors would like to thank Stig Venaas, Alvaro Retana, Anish Peter, Zheng Zhang and Umesh Dudani for their helpful comments on the document.

11. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

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

[I-D.venaas-pim-rfc8736bis] Venaas, S. and A. Retana, "PIM Message Type Space Extension and Reserved Bits", Work in Progress, Internet-Draft, draft-venaas-pim-rfc8736bis-00, 1 March 2023, <https://datatracker.ietf.org/doc/html/draft-venaas-pim-rfc8736bis-00>.

[RFC3446] Kim, D., Meyer, D., Kilmer, H., and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)", RFC 3446, DOI 10.17487/RFC3446, January 2003, <https://www.rfc-editor.org/info/rfc3446>.

Authors' Addresses

Vikas Ramesh Kamath
VMware
3401 Hillview Ave
Palo Alto, CA 94304
United States of America
Email: vkamath@vmware.com

Ramakrishnan Chokkanathapuram Sundaram
Cisco Systems, Inc.
Tasman Drive
San Jose, CA 95134
United States of America
Email: ramaksun@cisco.com

Raunak Banthia
Apstra
333 Middlefield Rd STE 200
Menlo Park, CA 94025
United States of America
Email: rbanthia@apstra.com

Ananya Gopal
Cisco Systems, Inc.
Tasman Drive
San Jose, CA 95134
United States of America
Email: ananygop@cisco.com

Network Working Group
Internet-Draft
Updates: 3973, 5015, 6754, 7761, 8364
(if approved)
Intended status: Standards Track
Expires: April 15, 2019

S. Venaas
Cisco Systems, Inc.
A. Retana
Huawei R&D USA
October 12, 2018

PIM reserved bits and type space extension
draft-ietf-pim-reserved-bits-00

Abstract

The currently defined PIM version 2 messages share a common message header format. The common header definition contains eight reserved bits. This document specifies how these bits may be used by individual message types, and creates a registry containing the per message type usage. This document also extends the PIM type space by defining three new message types. For each of the new types, four of the previously reserved bits are used to form an extended type range.

This document Updates RFC7761 and RFC3973 by defining the use of the currently Reserved field in the PIM common header. This document further updates RFC7761 and RFC3973, along with RFC5015, RFC6754 and RFC8364, by specifying the use of the currently Reserved bits for each PIM message.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
3. PIM header common format 3
4. Flag Bit definitions 3
  4.1. Flag Bits for Type 4 (Bootstrap) 4
  4.2. Flag Bits for Type 10 (DF Election) 4
  4.3. Flag Bits for Type 12 (PFM) 4
  4.4. Flag Bits for Type 13 (Type Space Extension) 4
  4.5. Flag Bits for Type 14 (Type Space Extension) 4
  4.6. Flag Bits for Type 15 (Type Space Extension) 4
5. PIM Type Space Extension 5
6. Security Considerations 5
7. IANA considerations 5
8. References 6
  8.1. Normative References 6
  8.2. Informative References 7
Authors' Addresses 7

1. Introduction

The currently defined PIM version 2 messages share a common message header format defined in the PIM Sparse Mode [RFC7761] and Dense Mode [RFC3973] specifications. The common header definition contains eight reserved bits. The message types defined in these documents all use this common header. However, several messages already make use of one or more bits, including the Bootstrap [RFC5059], DF-Election [RFC5015], and PIM Flooding Mechanism (PFM) [RFC8364] messages. There is no document formally specifying that these bits are to be used per message type.

This document refers to the bits specified as Reserved in the common PIM header [RFC7761] [RFC3973] as PIM message type flag bits, or simply flag bits, and it specifies that they are to be separately used on a per message type basis. It creates a registry containing the the per message type usage. For a particular message type, the usage of the flag bits can be defined in the document defining the message type, or a new document that updates that document.

The PIM message types as defined in the PIM Sparse Mode [RFC7761] and Dense Mode [RFC3973] specifications are in the range from 0 to 15. That type space is almost exhausted. Message type 15 was reserved by [RFC6166] for type space extension. In Section 5, this document specifies the use of the flag bits for message types 13, 14 and 15 in order to extend the PIM type space. The registration procedure for the extended type space is the same as for the existing type space, and the existing PIM message type registry is updated to include the extended type space.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. PIM header common format

The common PIM header is defined in section 4.9 of [RFC7761] and section 4.7.1 of [RFC3973]. This document updates the definition of the Reserved field and refers to that field as PIM message type flag bits, or simply flag bits. The new common header format is as below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |   Flags Bits  |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Flags Bits field is defined in Section 4. All other fields remain unchanged.

4. Flag Bit definitions

Unless otherwise specified, all the flag bits for each PIM type are Reserved [RFC8126]. They MUST be set to zero on transmission, and they MUST be ignored upon receipt. The specification of a new PIM type, MUST indicate whether the bits should be treated differently.

Currently for the message types 0 (Hello), 1 (Register), 2 (Register Stop), 3 (Join/Prune), 5 (Assert), 6 (Graft), 7 (Graft-Ack), 8 (Candidate RP Advertisement), 9 (State Refresh) and 11 (ECMP Redirect), all flag bits are Reserved.

When defining flag bits it is helpful to have a well defined way of referring to a particular bit. The most significant of the flag bits, the bit immediately following the type field is referred to as bit 7. The least significant, the bit right in front of the checksum field is referred to as bit 0. This is shown in the diagram below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |7 6 5 4 3 2 1 0|           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.1. Flag Bits for Type 4 (Bootstrap)

PIM message type 4 (Bootstrap) [RFC5059] defines flag bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining flag bits are Reserved.

4.2. Flag Bits for Type 10 (DF Election)

PIM message type 10 (DF Election) [RFC5015] specifies that the four most significant flag bits (bits 4-7) are to be used as a sub-type. The remaining flag bits are currently Reserved.

4.3. Flag Bits for Type 12 (PFM)

PIM message type 12 (PFM) [RFC8364] defines flag bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining flag bits are Reserved.

4.4. Flag Bits for Type 13 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.

4.5. Flag Bits for Type 14 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.

4.6. Flag Bits for Type 15 (Type Space Extension)

This type and the flag bit usage is defined in Section 5.

5. PIM Type Space Extension

The type space defined by the existing PIM specifications is almost exhausted. This document defines types 13, 14 and 15 (Type Space Extension) allowing for 48 additional types by for each of the three types, using the four most significant flag bits (bits 4-7) as a new field to store the extended type. These types are referred to as types 13.0 to 13.15, 14.0 to 14.15 and 15.0 to 15.15 where the last number denotes the value stored in the new field. The remaining four flag bits (bits 0-3) are Reserved to be used by each extended type. The specification of a new PIM extended type MUST indicate whether the bits should be treated differently. The common header for the new types is shown in the diagram below. The "Type" field is set to 13, 14 or 15, and the extended type field "SubType" denotes the value after the dot.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |SubType| Rsvd  |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6. Security Considerations

This document clarifies the use of the flag bits in the common PIM header and it extends the PIM type space. As such, there is no impact on security or changes to the considerations in [RFC7761] and [RFC3973].

7. IANA considerations

This document updates the PIM Message Types registry and also creates a PIM Message Type Flag Bits registry that shows which flag bits are defined for use by each of the PIM message types.

The following changes should be made to the existing PIM Message Types registry. For types 4 (Bootstrap) and 8 (Candidate RP Advertisement) a reference to RFC5059 should be added. For the currently unassigned types 13 and 14, and the reserved type 15, the name should be changed to "Type Space Extension", and reference this document. In addition, right underneath each of the rows for types 13, 14 and 15, there should be a new row where it says "13.0-13.15 Unassigned", "14.0-14.15 Unassigned" and "15.0-15.15 Unassigned", respectively.

A new registry called "PIM Message Type Flag Bits" should be created in the pim-paremeters section with registration procedure "IETF

Review" as defined in [RFC8126] with this document as a reference. The initial content of the registry should be as below.

Type	bit(s)	Name	Reference
0	0-7	Reserved	[RFC3973][RFC7761]
1	0-7	Reserved	[RFC3973][RFC7761]
2	0-7	Reserved	[RFC3973][RFC7761]
3	0-7	Reserved	[RFC3973][RFC7761]
4	0-6	Reserved	[RFC3973][RFC7761]
4	7	No-Forward	[RFC5059]
5	0-7	Reserved	[RFC3973][RFC7761]
6	0-7	Reserved	[RFC3973][RFC7761]
7	0-7	Reserved	[RFC3973][RFC7761]
8	0-7	Reserved	[RFC3973][RFC7761]
9	0-7	Reserved	[RFC3973][RFC7761]
10	0-3	Reserved	[RFC3973][RFC7761]
10	4-7	Sub-type	[RFC5015]
11	0-7	Reserved	[RFC6754]
12	0-6	Reserved	[RFC3973][RFC7761]
12	7	No-Forward	[RFC8364]
13	0-3	N/A (used by 13.0-13.15)	[this document]
13	4-7	Extended type	[this document]
13.0-13.15	0-3	Reserved	[this document]
14	0-3	N/A (used by 14.0-14.15)	[this document]
14	4-7	Extended type	[this document]
14.0-14.15	0-3	Reserved	[this document]
15	0-3	N/A (used by 15.0-15.15)	[this document]
15	4-7	Extended type	[this document]
15.0-15.15	0-3	Reserved	[this document]

8. References

8.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

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

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

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

[RFC6754] Cai, Y., Wei, L., Ou, H., Arya, V., and S. Jethwani, "Protocol Independent Multicast Equal-Cost Multipath (ECMP) Redirect", RFC 6754, DOI 10.17487/RFC6754, October 2012, <https://www.rfc-editor.org/info/rfc6754>.

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <https://www.rfc-editor.org/info/rfc8364>.

8.2. Informative References

[RFC6166] Venaas, S., "A Registry for PIM Message Types", RFC 6166, DOI 10.17487/RFC6166, April 2011, <https://www.rfc-editor.org/info/rfc6166>.

Authors' Addresses

Stig Venaas
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: stig@cisco.com

Alvaro Retana
Huawei R&D USA
2330 Central Expressway
Santa Clara CA 95050
USA

Email: alvaro.retana@huawei.com

Network Working Group
Internet-Draft
Obsoletes: 6166 (if approved)
Updates: 3973, 5015, 5059, 6754, 7761,
8364 (if approved)
Intended status: Standards Track
Expires: March 22, 2020

S. Venaas
Cisco Systems, Inc.
A. Retana
Futurewei Technologies, Inc.
September 19, 2019

PIM Message Type Space Extension and Reserved Bits
draft-ietf-pim-reserved-bits-04

Abstract

The PIM version 2 messages share a common message header format. The common header definition contains eight reserved bits. This document specifies how these bits may be used by individual message types, and creates a registry containing the per-message-type usage. This document also extends the PIM type space by defining three new message types. For each of the new types, four of the previously reserved bits are used to form an extended type range.

This document Updates RFC 7761 and RFC 3973 by defining the use of the currently Reserved field in the PIM common header. This document further updates RFC 7761 and RFC 3973, along with RFC 5015, RFC 5059, RFC 6754 and RFC 8364, by specifying the use of the currently Reserved bits for each PIM message.

This document obsoletes RFC 6166.

Status of This Memo

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
3. PIM header common format 3
4. Flag Bit definitions 3
  4.1. Flag Bits for Type 4 (Bootstrap) 4
  4.2. Flag Bits for Type 10 (DF Election) 4
  4.3. Flag Bits for Type 12 (PFM) 4
  4.4. Flag Bits for Types 13, 14 and 15 (Type Space Extension) 4
5. PIM Type Space Extension 4
6. Security Considerations 5
7. IANA Considerations 5
8. References 6
  8.1. Normative References 6
  8.2. Informative References 7
Authors' Addresses 8

1. Introduction

The PIM version 2 messages share a common message header format defined in the PIM Sparse Mode [RFC7761] specification. The common header definition contains eight Reserved bits. While all message types use this common header, there is no document formally specifying that these bits are to be used per message type.

This document refers to the bits specified as Reserved in the common PIM header [RFC7761] as PIM message type Flag Bits, or simply Flag Bits, and it specifies that they are to be separately used on a per-message-type basis. It creates a registry containing the per-message-type usage.

This document Updates [RFC7761] and [RFC3973] by defining the use of the currently Reserved field in the PIM common header. This document further updates [RFC7761] and [RFC3973], along with [RFC5015], [RFC5059], [RFC6754] and [RFC8364], by specifying the use of the currently Reserved bits for each PIM message.

The currently defined PIM message types are in the range from 0 to 15. That type space is almost exhausted. Message type 15 was reserved by [RFC6166] for type space extension. In Section 5, this document specifies the use of the Flag Bits for message types 13, 14 and 15 in order to extend the PIM type space. This document Obsoletes [RFC6166].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. PIM header common format

The common PIM header is defined in section 4.9 of [RFC7761]. This document updates the definition of the Reserved field and refers to that field as PIM message type Flag Bits, or simply Flag Bits. The new common header format is as below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |   Flag Bits   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: New Common Header

The Flag Bits field is defined in Section 4. All other fields remain unchanged.

4. Flag Bit definitions

Unless otherwise specified, all the Flag Bits for each PIM type are Reserved [RFC8126]. They MUST be set to zero on transmission, and they MUST be ignored upon receipt. The specification of a new PIM type MUST indicate whether the bits should be treated differently.

When defining Flag Bits, it is helpful to have a well-defined way of referring to a particular bit. The most significant of the Flag

Bits, the bit immediately following the type field is referred to as bit 7. The least significant, the bit right in front of the checksum field is referred to as bit 0. This is shown in the diagram below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |7 6 5 4 3 2 1 0|           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Flag Bits

4.1. Flag Bits for Type 4 (Bootstrap)

PIM message type 4 (Bootstrap) [RFC5059] defines Flag Bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining Flag Bits are Reserved.

4.2. Flag Bits for Type 10 (DF Election)

PIM message type 10 (DF Election) [RFC5015] specifies that the four most significant Flag Bits (bits 4-7) are to be used as a Subtype. The usage of those bits is defined in that document. The remaining Flag Bits are Reserved.

4.3. Flag Bits for Type 12 (PFM)

PIM message type 12 (PFM) [RFC8364] defines Flag Bit 7 as No-Forward. The usage of the bit is defined in that document. The remaining Flag Bits are Reserved.

4.4. Flag Bits for Types 13, 14 and 15 (Type Space Extension)

These types and the corresponding Flag Bits are defined in Section 5.

5. PIM Type Space Extension

This document defines types 13, 14 and 15, such that each of these types has 16 subtypes, providing a total of 48 subtypes available for future PIM extensions. This is achieved by defining a new SubType field (see Figure 3) using the four most significant Flag Bits (bits 4-7). The notation type.subtype is used to reference these new extended types. The remaining four Flag Bits (bits 0-3) are Reserved to be used by each extended type (abbreviated as FB below).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |SubType|  FB   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Sub-Types

6. Security Considerations

This document clarifies the use of the Flag Bits in the common PIM header and it extends the PIM type space. As such, there is no impact on security or changes to the considerations in [RFC7761] and [RFC3973].

7. IANA Considerations

This document updates the PIM Message Types registry to indicate which Flag Bits are defined for use by each of the PIM message types. The Registry should now reference this document instead of [RFC6166]. The Registration Policy remains IETF Review [RFC8126]. Assignments into this registry MUST define any non-default usage (see Section 4) of the Flag Bits in addition to defining the Type.

The updated PIM Message Types registry is shown below.

Type	Name	Flag Bits	Reference
0	Hello	0-7: Reserved	[RFC3973][RFC7761]
1	Register	0-7: Reserved	[RFC7761]
2	Register Stop	0-7: Reserved	[RFC7761]
3	Join/Prune	0-7: Reserved	[RFC3973][RFC7761]
4	Bootstrap	0-6: Reserved 7: No-Forward	[RFC5059][RFC7761] [RFC5059]
5	Assert	0-7: Reserved	[RFC3973][RFC7761]
6	Graft	0-7: Reserved	[RFC3973]
7	Graft-Ack	0-7: Reserved	[RFC3973]
8	Candidate RP Advertisement	0-7: Reserved	[RFC7761]
9	State Refresh	0-7: Reserved	[RFC3973]
10	DF Election	0-3: Reserved 4-7: Subtype	[RFC5015] [RFC5015]
11	ECMP Redirect	0-7: Reserved	[RFC6754]
12	PIM Flooding Mechanism	0-6: Reserved 7: No-Forward	[RFC8364] [RFC8364]
13.0-15.15	Unassigned	0-3: Unassigned	[this document]

Table 1: Updated PIM Message Types Registry

The Unassigned types above, as explained in Section 5, use the extended type notation of type.subtype. Each extended type only has 4 Flag Bits available. New extended message types should be assigned conscutively, starting with 13.0, then 13.1, etc.

8. References

8.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

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

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

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

[RFC6166] Venaas, S., "A Registry for PIM Message Types", RFC 6166, DOI 10.17487/RFC6166, April 2011, <https://www.rfc-editor.org/info/rfc6166>.

[RFC6754] Cai, Y., Wei, L., Ou, H., Arya, V., and S. Jethwani, "Protocol Independent Multicast Equal-Cost Multipath (ECMP) Redirect", RFC 6754, DOI 10.17487/RFC6754, October 2012, <https://www.rfc-editor.org/info/rfc6754>.

[RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <https://www.rfc-editor.org/info/rfc8364>.

Authors' Addresses

Stig Venaas
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: stig@cisco.com

Alvaro Retana
Futurewei Technologies, Inc.
2330 Central Expressway
Santa Clara CA 95050
USA

Email: alvaro.retana@futurewei.com

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: September 6, 2019

Yisong Liu
M. McBride
T. Eckert
Huawei Technologies

March 6, 2019

PIM Assert Message Packing
draft-liu-pim-assert-packing-00

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

This Internet-Draft will expire on September 6, 2019.

Copyright Notice

Abstract

In PIM-SM shared networks, there is typically more than one upstream router. When duplicate data packets appear on the LAN from different routers, assert packets are sent from these routers to elect a single forwarder. The PIM assert packets are sent periodically to keep the assert state. The PIM assert packet carries information about a single multicast source and group, along with the metric-preference and metric of the route towards the source or RP. This document defines a standard to send and receive multiple multicast source and group information in a single PIM assert packet in a shared network. This can be particularly helpful when there is traffic for a large number of multicast groups.

Table of Contents

1. Introduction 2
   1.1. Requirements Language 3
   1.2. Terminology 3
2. Use Cases 3
   2.1. Enterprise network 3
   2.2. Video surveillance 3
   2.3. Financial Services 4
   2.4. IPTV broadcast video 4
3. Solution 4
   3.1. PIM Assert Packing Hello Option 5
   3.2. PIM Assert Packing Simple Type 5
   3.3. PIM Assert Packing Aggregation Type 5
4. Packet Format 5
   4.1. PIM Assert Packing Hello Option 6
   4.2. PIM Assert Simple Packing Format 6
   4.3. PIM Assert Aggregation Packing Format 7
5. IANA Considerations 9
6. Security Considerations 9
7. References 9
   7.1. Normative References 9
   7.2. Informative References 10
8. Acknowledgments 10

1. Introduction

In PIM-SM shared networks, there is typically more than one upstream router. When duplicate data packets appear on the LAN, from different upstream routers, assert packets are sent from these routers to elect a single forwarder according to [RFC7761]. The PIM assert packets are sent periodically to keep the assert state. The PIM assert packet carries information about a single multicast

source and group, along with the corresponding metric-preference and metric of the route towards the source or RP.

This document defines a standard to send and receive multiple multicast source and group information in a single PIM assert packet in a shared network. It can efficiently pack multiple PIM assert packets into a single message and reduce the processing pressure of the PIM routers. This can be particularly helpful when there is traffic for a large number of multicast groups.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

RPF: Reverse Path Forwarding

RP: Rendezvous Point

SPT: Shortest Path Tree

RPT: RP Tree

2. Use Cases

PIM Asserts will happen in many services where multicast is used and not limited to the examples described below:

2.1. Enterprise network

When an Enterprise network is connected through a layer-2 network, the intra-enterprise runs layer-3 PIM multicast. The different sites of the enterprise are equivalent to the PIM connection through the shared network. Depending upon the locations and amount of groups there could be many asserts on the first hop routers.

2.2. Video surveillance

Video surveillance deployments have migrated from analog based systems to IP-based systems oftentimes using multicast. In certain deployments, when there are many cameras streaming to many groups, there may be issues with many asserts on first hop routers.

2.3. Financial Services

Financial services extensively rely on IP Multicast to deliver stock market data and its derivatives, and current multicast solution PIM is usually deployed. As the number of multicast flows grows, there are many stock data with many groups may result in many PIM asserts on a shared network from publisher to the subscribers.

2.4. IPTV broadcast video

PIM DR and BDR deployments are often used in host-side network for IPTV broadcast video services. Host-side access network failure scenario may be benefitted by assert packing when many groups are being used. According to [RFC7761] the DR will be elected to forward multicast traffic in the shared access network. When the DR recovers from a failure, the original DR starts to send traffic, and the current DR is still forwarding traffic. In the situation multicast traffic duplication maybe happen in the shared access network and can trigger the assert progress.

In the above scenarios, as the multicast service becomes widely deployed, the number of multicast entries increases, and a large number of assert messages may be sent in a very short period when multicast data packets trigger PIM assert process in the shared networks. The PIM routers need to process a large number of PIM assert small packets in a very short time. As a result, the device load is very large. The assert packet may not be processed in time or even is discarded, thus extending the time of traffic duplication in the network.

Additionally, future backhaul, or fronthaul, networks may want to connect L3 across an L2 underlay supporting Time Sensitive Networks (TSN). The infrastructure may run DetNet over TSN. These transit L2 LANs would have multiple upstreams and downstreams. This draft is taking a proactive approach to prevention of possible future assert issues in these types of environments.

3. Solution

The change to the PIM assert includes two elements: the PIM assert packing hello option and the PIM assert packing method.

There is no change required to the PIM assert state machine. Basically a PIM router can now be the assert winner/loser for multiple packed (S, G)’s in a single assert packet instead of one (S, G) assert at a time. An assert winner is now responsible for forwarding traffic from multiple (S, G)’s out of a particular interface based upon the multiple (S, G)’s packed in a single assert.

3.1. PIM Assert Packing Hello Option

The newly defined Hello Option is used by a router to negotiate the assert packet packing capability. It can only be used when all PIM routers, in the same shared network, support this capability.

This document defines two packing methods. One method is a simple merge of the original messages and the other is to extract the common message fields for aggregation.

3.2. PIM Assert Packing Simple Type

In this type of packing, the original assert message body is used as a record. The newly defined assert message can carry multiple assert records and identify the number of records.

This packing method is simply extended from the original assert packet, but, because the multicast service deployment often uses a small number of sources and RPs, there may be a large number of assert records with the same metric preference or route metric field, which wastes the payload of the transmitted message

3.3. PIM Assert Packing Aggregation Type

When the source or RP addresses, in the actual deployment of the multicast service, are very few, this type of packing will combine the records related to the source address or RP address in the assert message.

* (S, G) assert is aggregated according to the same source address, and all SPT (S, G) entries corresponding to the source address are merged into one assert record.

* (*, G) assert is aggregated according to the same RP address, and all (*, G) and RPT (S, G) entries corresponding to the RP address are merged into one assert record.

This method can optimize the payload of the transmitted message by merging the same field content, but will add the complexity of the packet encapsulation and parsing.

4. Packet Format

This section describes the format of new PIM messages introduced by this document. The messages follow the same transmission order as the messages defined in [RFC7761]

4.1. PIM Assert Packing Hello Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OptionType = TBD         |      OptionLength = 1         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Packing_Type |
+-+-+-+-+-+-+-+-+
```

- OptionType: TBD

- OptionLength: 1

- Packing_Type: The specific packing mode is determined by the value of this field:

  1: indicates simple packing type as described in section 2.2

  2: indicates aggregating packing type as described in section 2.3

  3-255: reserved for future

4.2. PIM Assert Simple Packing Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  |   Reserved    |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |  Number of Assert Records (M) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                        Assert Record [1]                      .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                        Assert Record [2]                      .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      |                               .                               |
      .                               .                               .
      |                               .                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      .                                                               .
      .                        Assert Record [M]                      .
      .                                                               .
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of each record is the same as the PIM assert message body of section 4.9.6 in [RFC7761].

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Group Address (Encoded-Group format)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Source Address (Encoded-Unicast format)            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |R|                      Metric Preference                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             Metric                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3. PIM Assert Aggregation Packing Format

This method also extends PIM assert packets to carry multiple records. The specific assert packet format is the same as section 3.2, but the records are divided into two types.

The (S, G) assert records are organized by the same source address, and the specific message format is:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Source Address (Encoded-Unicast format)            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |0|                      Metric Preference                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             Metric                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |        Number of Groups (N)   |           Reserved            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      |                 Group Address 1 (Encoded-Group format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                 Group Address 2 (Encoded-Group format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             .                                 |
      |                             .                                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                 Group Address N (Encoded-Group format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The (*, G) assert records are organized in the same RP address and are divided into two levels of TLVs. The first level is the group record of the same RP address, and the second level is the source record of the same multicast group address, including (*, G) and RPT (S, G), and the specific message format is:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |             RP Address (Encoded-Unicast format)               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|                     Metric Preference                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             Metric                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   Number of Group Records(O)  |           Reserved            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      .                                                               .
      .                       Group Record [1]                        .
      .                                                               .
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      .                                                               .
      .                       Group Record [2]                        .
      .                                                               .
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                               .                               |
      .                               .                               .
      |                               .                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      .                                                               .
      .                       Group Record [O]                        .
      .                                                               .
```

```
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of each group record is:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Group Address (Encoded-Group format)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |        Number of Sources (P)  |           Reserved            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Source Address 1 (Encoded-Unicast format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Source Address 2 (Encoded-Unicast format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                             .                                 |
      |                             .                                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Source Address P (Encoded-Unicast format)        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5. IANA Considerations

This document requests IANA to assign a registry for PIM assert packing Hello Option in the PIM-Hello Options. The assignment is requested permanent for IANA when this document is published as an RFC. The string TBD should be replaced by the assigned values accordingly.

6. Security Considerations

For general PIM-SM protocol Security Considerations, see [RFC7761].

TBD

7. References

7.1. Normative References

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

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.,Parekh, R., Zhang, Z., and L. Zheng, "Protocol IndependentMulticast - Sparse Mode (PIM-SM): Protocol Specification(Revised)", RFC 7761, March 2016

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017

7.2. Informative References

TBD

8. Acknowledgments

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

TBD

Authors' Addresses

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

Email: liuyisong@huawei.com

Mike McBride
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95055
USA

Email: Michael.mcbride@huawei.com

Toerless Eckert
Huawei Technologies
2330 Central Expy
Santa Clara 95050
USA

Email: tte+ietf@cs.fau.de

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: August 6, 2020

Yisong Liu
China Mobile
M. McBride
T. Eckert
Futurewei
Feb 6, 2020

PIM Assert Message Packing
draft-liu-pim-assert-packing-02

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

This Internet-Draft will expire on August 6, 2020.

Copyright Notice

Abstract

In PIM-SM shared LAN networks, there is typically more than one upstream router. When duplicate data packets appear on the LAN from different routers, assert packets are sent from these routers to elect a single forwarder. The PIM assert packets are sent periodically to keep the assert state. The PIM assert packet carries information about a single multicast source and group, along with the metric-preference and metric of the route towards the source or RP. This document defines a standard to send and receive multiple multicast source and group information in a single PIM assert packet in a shared network. This can be particularly helpful when there is traffic for a large number of multicast groups.

Table of Contents

1. Introduction 3
   1.1. Requirements Language 3
   1.2. Terminology 3
2. Use Cases 3
   2.1. Enterprise network 4
   2.2. Video surveillance 4
   2.3. Financial Services 4
   2.4. IPTV broadcast Video 4
   2.5. Summary 4
3. Solution 5
   3.1. PIM Assert Packing Hello Option 5
   3.2. PIM Assert Packing Simple Type 5
   3.3. PIM Assert Packing Aggregation Type 6
4. Packet Format 6
   4.1. PIM Assert Packing Hello Option 6
   4.2. PIM Assert Simple Packing Format 7
   4.3. PIM Assert Aggregation Packing Format 8
5. IANA Considerations 11
6. Security Considerations 11
7. References 11
   7.1. Normative References 11
   7.2. Informative References 12
8. Acknowledgments 12
Authors' Addresses 13

1. Introduction

In PIM-SM shared LAN networks, there is typically more than one upstream router. When duplicate data packets appear on the LAN, from different upstream routers, assert packets are sent from these routers to elect a single forwarder according to [RFC7761]. The PIM assert packets are sent periodically to keep the assert state. The PIM assert packet carries information about a single multicast source and group, along with the corresponding metric-preference and metric of the route towards the source or RP.

This document defines a standard to send and receive multiple multicast source and group information in a single PIM assert packet in a shared LAN network. It can efficiently pack multiple PIM assert packets into a single message and reduce the processing pressure of the PIM routers. This can be particularly helpful when there is traffic for a large number of multicast groups.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

RPF: Reverse Path Forwarding

RP: Rendezvous Point

SPT: Shortest Path Tree

RPT: RP Tree

DR: Designated Router

BDR: Backup Designated Router

2. Use Cases

PIM Assert will happen in many services where multicast is used and not limited to the examples described below.

2.1. Enterprise network

When an Enterprise network is connected through a layer-2 network, the intra-enterprise runs layer-3 PIM multicast. The different sites of the enterprise are equivalent to the PIM connection through the shared LAN network. Depending upon the locations and amount of groups there could be many asserts on the first hop routers.

2.2. Video surveillance

Video surveillance deployments have migrated from analog based systems to IP-based systems oftentimes using multicast. In the shared LAN network deployments, when there are many cameras streaming to many groups there may be issues with many asserts on first hop routers.

2.3. Financial Services

Financial services extensively rely on IP Multicast to deliver stock market data and its derivatives, and current multicast solution PIM is usually deployed. As the number of multicast flows grows, there are many stock data with many groups may result in many PIM asserts on a shared LAN network from publisher to the subscribers.

2.4. IPTV broadcast Video

PIM DR and BDR deployments are often used in host-side network for IPTV broadcast video services. Host-side access network failure scenario may be benefitted by assert packing when many groups are being used. According to [RFC7761] the DR will be elected to forward multicast traffic in the shared access network. When the DR recovers from a failure, the original DR starts to send traffic, and the current DR is still forwarding traffic. In the situation multicast traffic duplication maybe happen in the shared access network and can trigger the assert progress.

2.5. Summary

In the above scenarios, the existence of PIM assert process depends mainly on the network topology. As long as there is a layer 2 network between PIM neighbors, there may be multiple upstream routers, which can cause duplicate multicast traffic to be forwarded and assert process to occur.

Moreover as the multicast services become widely deployed, the number of multicast entries increases, and a large number of assert messages may be sent in a very short period when multicast data packets trigger PIM assert process in the shared LAN networks. The

PIM routers need to process a large number of PIM assert small packets in a very short time. As a result, the device load is very large. The assert packet may not be processed in time or even is discarded, thus extending the time of traffic duplication in the network.

Additionally, future backhaul, or fronthaul, networks may want to connect L3 across an L2 underlay supporting Time Sensitive Networks (TSN). The infrastructure may run DetNet over TSN. These transit L2 LANs would have multiple upstreams and downstreams. This document is taking a proactive approach to prevention of possible future assert issues in these types of environments.

3. Solution

The change to the PIM assert includes two elements: the PIM assert packing hello option and the PIM assert packing method.

There is no change required to the PIM assert state machine. Basically a PIM router can now be the assert winner or loser for multiple packed (S, G)’s in a single assert packet instead of one (S, G) assert at a time. An assert winner is now responsible for forwarding traffic from multiple (S, G)’s out of a particular interface based upon the multiple (S, G)’s packed in a single assert.

3.1. PIM Assert Packing Hello Option

The newly defined Hello Option is used by a router to negotiate the assert packet packing capability. It can only be used when all PIM routers, in the same shared LAN network, support this capability. This document defines two packing methods. One method is a simple merge of the original messages and the other is to extract the common message fields for aggregation.

3.2. PIM Assert Packing Simple Type

In this type of packing, the original assert message body is used as a record. The newly defined assert message can carry multiple assert records and identify the number of records.

This packing method is simply extended from the original assert packet, but, because the multicast service deployment often uses a small number of sources and RPs, there may be a large number of assert records with the same metric preference or route metric field, which would waste the payload of the transmitted message.

3.3. PIM Assert Packing Aggregation Type

When the source or RP addresses, in the actual deployment of the multicast service, are very few, this type of packing will combine the records related to the source address or RP address in the assert message.

* A (S, G) assert only can contain one SPT (S, G) entry, so it can be aggregated according to the same source address, and then all SPT (S, G) entries corresponding to the same source address are merged into one assert record.

* A (*, G) assert may contain a (*, G) entry or a RPT (S, G) entry, and both entry types actually depend on the route to the RP. So it can be aggregated further according to the same RP address, and then all (*, G) and RPT (S, G) entries corresponding to the same RP address are merged into one assert record.

This method can optimize the payload of the transmitted message by merging the same field content, but will add the complexity of the packet encapsulation and parsing.

4. Packet Format

This section describes the format of new PIM messages introduced by this document. The messages follow the same transmission order as the messages defined in [RFC7761].

4.1. PIM Assert Packing Hello Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OptionType = TBD         |      OptionLength = 1         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Packing_Type |
+-+-+-+-+-+-+-+-+
```

- OptionType: TBD

- OptionLength: 1

- Packing_Type: The specific packing mode is determined by the value of this field:

1: indicates simple packing type as described in section 2.2

2: indicates aggregating packing type as described in section 2.3

3-255: reserved for future

4.2. PIM Assert Simple Packing Format

```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |PIM Ver| Type  |SubType| Rsvd  |           Checksum            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Reserved            |  Number of Assert Records (M) |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                                                               .
 .                        Assert Record [1]                      .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                                                               .
 .                        Assert Record [2]                      .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                               .                               |
 .                               .                               .
 |                               .                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                                                               .
 .                        Assert Record [M]                      .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Reserved, Checksum

Same as [RFC7761] Section 4.9.6

Type

The new Assert Type and SubType values TBD

Number of Assert Records

The number of packed assert records. A record consists of a single assert message body.

The format of each record is the same as the PIM assert message body of section 4.9.6 in [RFC7761].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Group Address (Encoded-Group format)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Address (Encoded-Unicast format)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|                      Metric Preference                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Metric                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3. PIM Assert Aggregation Packing Format

This method also extends PIM assert packets to carry multiple records. The specific assert packet format is the same as section 4.2, but the records are divided into two types.

The (S, G) assert records are organized by the same source address, and the specific message format is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Source Address (Encoded-Unicast format)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|                      Metric Preference                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Metric                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Number of Groups (N)   |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Group Address 1 (Encoded-Group format)        |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Group Address 2 (Encoded-Group format)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             .                                 |
|                             .                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Group Address N (Encoded-Group format)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Source Address, Metric Preference, Metric and Reserved

Same as [RFC7761] Section 4.9.6, but the source address MUST NOT be set to zero.

Number of Groups

The number of group addresses corresponding to the source address field in the (S, G) assert record.

Group Address

Same as [RFC7761] Section 4.9.6, but there are multiple group addresses in the (S, G) assert record

The (*, G) assert records are organized in the same RP address and are divided into two levels of TLVs. The first level is the group record of the same RP address, and the second level is the source record of the same multicast group address, including (*, G) and RPT (S, G), and the specific message format is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             RP Address (Encoded-Unicast format)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|                     Metric Preference                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Metric                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Number of Group Records(O)  |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                       Group Record [1]                        .
```

```
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                                                               .
    .                        Group Record [2]                       .
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               .                               |
    .                               .                               .
    |                               .                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                                                               .
    .                        Group Record [O]                       .
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RP Address

The address of RP corresponding to all of the contained group records. The format for this address is given in the encoded unicast address in [RFC7761] Section 4.9.1

Metric Preference, Metric and Reserved

Same as [RFC7761] Section 4.9.6

Number of Group Records

The number of packed group records. A record consists of a group address and a source address list.

The format of each group record is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Group Address (Encoded-Group format)             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Number of Sources (P)  |           Reserved            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Source Address 1 (Encoded-Unicast format)        |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Source Address 2 (Encoded-Unicast format)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             .                                 |
|                             .                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Source Address P (Encoded-Unicast format)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Group Address and Reserved

Same as [RFC7761] Section 4.9.6

Number of Sources

The number of source addresses corresponding to the group address field in the group record.

Source Address

Same as [RFC7761] Section 4.9.6, but there are multiple source addresses in the group record.

5. IANA Considerations

This document requests IANA to assign a registry for PIM assert packing Hello Option in the PIM-Hello Options and new PIM assert packet type and subtype. The assignment is requested permanent for IANA when this document is published as an RFC. The string TBD should be replaced by the assigned values accordingly.

6. Security Considerations

For general PIM-SM protocol Security Considerations, see [RFC7761].

TBD

7. References

7.1. Normative References

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

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.,Parekh, R., Zhang, Z., and L. Zheng, "Protocol IndependentMulticast - Sparse Mode (PIM-SM): Protocol Specification(Revised)", RFC 7761, March 2016

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017

7.2. Informative References

TBD

8. Acknowledgments

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

TBD

Authors' Addresses

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Toerless Eckert
Futurewei

Email: tte+ietf@cs.fau.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2019

Mankamana. Mishra
Cisco Systems

October 22, 2018

PIM Backup Designated Router Procedure
draft-mankamana-pim-bdr-01

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In this document, we propose a mechanism to elect backup DR on a shared LAN. A backup DR on LAN would be useful for faster convergence. This draft introduces the concept of a Backup Designated Router (BDR) and the procedure to implement it.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Applicability and deviation from draft PIM DR Improvement . . 4
4. Protocol Specification 4
4.1. PIM Backup DR (BDR) election procedure 4
4.2. Existing PIM DR failure 4
4.3. Existing PIM BDR failure 4
4.4. New PIM Router addition in network 4
4.4.1. New PIM router eligible to be PIM DR on shared LAN . 4
4.4.2. New PIM router eligible to be PIM BDR on shared LAN . 5
4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN 5
4.5. Initial case, All new PIM router coming up in shared LAN 5
4.6. Benefit 6
5. Compatibility 6
6. Manageability Considerations 6
7. IANA Considerations 6
8. Security Considerations 6
9. Acknowledgement 6
10. Normative References 6
Author's Address 7

1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR has two roles in the PIM-SM protocol. On the first hop network, the PIM DR is responsible for registering an active source with the Rendezvous Point (RP) if the group is operating in PIM-SM. On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

```
                    ( core networks )
                      |     |     |
                      |     |     |
                     R1    R2    R3
                      |     |     |
                  --(last hop LAN)--
                            |
                            |
                    (many receivers)
```

Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local member. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs.

There are multiple reasons for why network could potentially trigger DR re-election. Some of the reasons are

1. R1 going down

2. Access interface towards shared LAN going down

3. Config changed with lower DR priority

When any of above network event occurs, PIM DR re-election would be triggered. When a new DR is elected in shared LAN, new DR would be responsible to build a multicast tree towards source / RP. There are some cases, where traffic is crucial and the operator wants to have minimum traffic loss with DR failure. To address this requirement, this draft introduces a backup DR election procedure which would minimize traffic loss during PIM DR failure.

2. Terminology

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 [RFC2119] .

BDR - PIM Backup DR

With respect to PIM, this document follows the terminology that has been defined in [RFC4601] .

3. Applicability and deviation from draft PIM DR Improvement

[I-D.ietf-pim-dr-improvement] defines procedure to solve same problem which was stated in the introduction section of this draft. [I-D.ietf-pim-dr-improvement] introduces new PIM Hello options for election of backup PIM DR.

This draft provides mechanism to elect BDR without using any new PIM Hello.

4. Protocol Specification

4.1. PIM Backup DR (BDR) election procedure

[RFC7761] defines procedure for PIM DR election. PIM DR is elected on interface "I" among all PIM routers for which "I" has received PIM Hello. BDR election follows the exact same procedure and the second best PIM DR on shared LAN to be chosen as BDR on interface "I"

BDR would perform each of the responsibility of PIM DR except it would not forward traffic on shared LAN.

4.2. Existing PIM DR failure

When PIM DR fails, PIM DR re-election is triggered on shared LAN. Since BDR is second best DR in LAN, it MUST take over immediately and MUST start forwarding multicast traffic on shared LAN.

Again on a shared LAN, new BDR would be elected. and current BDR would be the new DR.

4.3. Existing PIM BDR failure

When an existing PIM BDR fails, the shared LAN MUST have BDR re-election using the DR election procedure from [RFC7761].

4.4. New PIM Router addition in network

When a new PIM router is added in shared LAN, It could be either one of the below defined roles.

4.4.1. New PIM router eligible to be PIM DR on shared LAN

When a new PIM router is added in a shared LAN and has the highest PIM DR priority configured, if a new router starts propagating its configured DR priority right away, the existing PIM DR would give up its role. Then there would be potential traffic loss till the new DR

learns about membership states and builds a multicast tree to the source or RP.

To avoid any such traffic loss situation, new PIM router SHOULD send a PIM Hello with priority 0. After 2 (default value, SHOULD have way to configure) PIM Hello interval or IGMP Query Interval (Which ever is higher) it SHOULD start propagating its original configured DR priority.

Even though a new PIM router propagating its priority as 0, it MUST start building a multicast tree towards source / RP, This is So that traffic loss could be minimized once it starts sending Hello with configured DR priority.

For a brief amount of time, there would be multiple copies of flows present in the multicast core, but a user SHOULD be able to configure whether to send hello with 0 priority or a configured priority. Depending on the application tolerance (Traffic loss Vs Extra traffic in core) the operator can choose option whichever is suitable for network.

After a PIM Hello or IGMP Query interval, the network would get stable with only one DR and one BDR.

4.4.2. New PIM router eligible to be PIM BDR on shared LAN

It SHOULD follow the exact same procedure defined in the previous section.

4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN

First a PIM Hello MUST be sent with priority 0. Once it has gotten Hello from other PIM neighbors, it knows that it is not eligible to be PIM DR or BDR. It MUST send configured PIM DR priority immediately. It MUST not wait for next hello interval.

4.5. Initial case, All new PIM router coming up in shared LAN

In this case, initially each of the PIM routers would send Hellos with priorities of 0. If a PIM router receives all Hellos with priorities 0, it MUST send out a Hello with a configured PIM DR priority. Since it is initial startup case, it would take up to one Hello interval to converge.

4.6. Benefit

1. Easy to implement as it uses an existing PIM procedure to elect DR.

2. Does not introduce any new Hello option

5. Compatibility

6. Manageability Considerations

7. IANA Considerations

8. Security Considerations

9. Acknowledgement

The author would like to thank Stig Venaas, Tharak Abraham, Anish Kachinthaya, Anvitha Kachinthaya for helping with original idea.

10. Normative References

[I-D.ietf-pim-dr-improvement]
Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.

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

[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, <https://www.rfc-editor.org/info/rfc4601>.

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

Author's Address

Mankamana Mishra
Cisco Systems
821 Alder Drive,
MILPITAS, CALIFORNIA 95035
UNITED STATES

Email: mankamis@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 10, 2021

M. Mishra
S. Santhanam
A. Paramasivam
J. Goh
Cisco Systems
G. Mishra
Verizon Communications Inc. (VZ)

April 8, 2021

PIM Backup Designated Router Procedure
draft-mankamana-pim-bdr-05

Abstract

On a multi-access network, one of the PIM routers is elected as a Designated Router (DR). On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding traffic to these listeners if the group is operating in PIM-SM. In this document, we propose a mechanism to elect backup DR on a shared LAN. A backup DR on LAN would be useful for faster convergence. This draft introduces the concept of a Backup Designated Router (BDR) and the procedure to implement it.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Applicability and deviation from draft PIM DR Improvement . . 4
4. Protocol Specification 4
4.1. PIM Backup DR (BDR) election procedure 4
4.2. Existing PIM DR failure 4
4.3. Existing PIM BDR failure 4
4.4. New PIM Router addition in network 4
4.4.1. New PIM router eligible to be PIM DR on shared LAN . 4
4.4.2. New PIM router eligible to be PIM BDR on shared LAN . 5
4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN 5
4.5. Initial case, All new PIM router coming up in shared LAN 5
4.6. Benefit 6
5. Compatibility 6
6. Manageability Considerations 6
7. IANA Considerations 6
8. Security Considerations 6
9. Acknowledgement 6
10. Normative References 6
Authors' Addresses 7

1. Introduction

On a multi-access LAN such as an Ethernet, one of the PIM routers is elected as a DR. The PIM DR has two roles in the PIM-SM protocol. On the first hop network, the PIM DR is responsible for registering an active source with the Rendezvous Point (RP) if the group is operating in PIM-SM. On the last hop LAN, the PIM DR is responsible for tracking local multicast listeners and forwarding to these listeners if the group is operating in PIM-SM.

Consider the following last hop LAN in Figure 1:

```
                    ( core networks )
                      |     |     |
                      |     |     |
                     R1    R2    R3
                      |     |     |
                  --(last hop LAN)--
                            |
                            |
                    (many receivers)
```

Figure 1: Last Hop LAN

Assume R1 is elected as the Designated Router. According to [RFC4601], R1 will be responsible for forwarding traffic to that LAN on behalf of any local member. In addition to keeping track of IGMP and MLD membership reports, R1 is also responsible for initiating the creation of source and/or shared trees towards the senders or the RPs.

There are multiple reasons for why network could potentially trigger DR re-election. Some of the reasons are

1. R1 going down

2. Access interface towards shared LAN going down

3. Config changed with lower DR priority

When any of above network event occurs, PIM DR re-election would be triggered. When a new DR is elected in shared LAN, new DR would be responsible to build a multicast tree towards source / RP. There are some cases, where traffic is crucial and the operator wants to have minimum traffic loss with DR failure. To address this requirement, this draft introduces a backup DR election procedure which would minimize traffic loss during PIM DR failure.

2. Terminology

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 [RFC2119] .

BDR - PIM Backup DR

With respect to PIM, this document follows the terminology that has been defined in [RFC4601] .

3. Applicability and deviation from draft PIM DR Improvement

[I-D.ietf-pim-dr-improvement] defines procedure to solve same problem which was stated in the introduction section of this draft. [I-D.ietf-pim-dr-improvement] introduces new PIM Hello options for election of backup PIM DR.

This draft provides mechanism to elect BDR without using any new PIM Hello.

4. Protocol Specification

4.1. PIM Backup DR (BDR) election procedure

[RFC7761] defines procedure for PIM DR election. PIM DR is elected on interface "I" among all PIM routers for which "I" has received PIM Hello. BDR election follows the exact same procedure and the second best PIM DR on shared LAN to be chosen as BDR on interface "I"

BDR would perform each of the responsibility of PIM DR except it would not forward traffic on shared LAN.

4.2. Existing PIM DR failure

When PIM DR fails, PIM DR re-election is triggered on shared LAN. Since BDR is second best DR in LAN, it MUST take over immediately and MUST start forwarding multicast traffic on shared LAN.

Again on a shared LAN, new BDR would be elected. and current BDR would be the new DR.

4.3. Existing PIM BDR failure

When an existing PIM BDR fails, the shared LAN MUST have BDR re-election using the DR election procedure from [RFC7761].

4.4. New PIM Router addition in network

When a new PIM router is added in shared LAN, It could be either one of the below defined roles.

4.4.1. New PIM router eligible to be PIM DR on shared LAN

When a new PIM router is added in a shared LAN and has the highest PIM DR priority configured, if a new router starts propagating its configured DR priority right away, the existing PIM DR would give up its role. Then there would be potential traffic loss till the new DR

learns about membership states and builds a multicast tree to the source or RP.

To avoid any such traffic loss situation, new PIM router SHOULD send a PIM Hello with priority 0. After 2 (default value, SHOULD have way to configure) PIM Hello interval or IGMP Query Interval (Which ever is higher) it SHOULD start propagating its original configured DR priority.

Even though a new PIM router propagating its priority as 0, it MUST start building a multicast tree towards source / RP, This is So that traffic loss could be minimized once it starts sending Hello with configured DR priority.

For a brief amount of time, there would be multiple copies of flows present in the multicast core, but a user SHOULD be able to configure whether to send hello with 0 priority or a configured priority. Depending on the application tolerance (Traffic loss Vs Extra traffic in core) the operator can choose option whichever is suitable for network.

After a PIM Hello or IGMP Query interval, the network would get stable with only one DR and one BDR.

4.4.2. New PIM router eligible to be PIM BDR on shared LAN

It SHOULD follow the exact same procedure defined in the previous section.

4.4.3. New PIM router is not eligible to be PIM DR or BDR on shared LAN

First a PIM Hello MUST be sent with priority 0. Once it has gotten Hello from other PIM neighbors, it knows that it is not eligible to be PIM DR or BDR. It MUST send configured PIM DR priority immediately. It MUST not wait for next hello interval.

4.5. Initial case, All new PIM router coming up in shared LAN

In this case, initially each of the PIM routers would send Hellos with priorities of 0. If a PIM router receives all Hellos with priorities 0, it MUST send out a Hello with a configured PIM DR priority. Since it is initial startup case, it would take up to one Hello interval to converge.

4.6. Benefit

1. Easy to implement as it uses an existing PIM procedure to elect DR.

2. Does not introduce any new Hello option

5. Compatibility

6. Manageability Considerations

7. IANA Considerations

8. Security Considerations

9. Acknowledgement

The author would like to thank Stig Venaas, Tharak Abraham, Anish Kachinthaya, Anvitha Kachinthaya for helping with original idea.

10. Normative References

[I-D.ietf-pim-dr-improvement]
Zhang, Z., hu, f., Xu, B., and m. mishra, "PIM DR Improvement", draft-ietf-pim-dr-improvement-04 (work in progress), December 2017.

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

[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, <https://www.rfc-editor.org/info/rfc4601>.

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

Authors' Addresses

Mankamana Mishra
Cisco Systems
821 Alder Drive,
MILPITAS, CALIFORNIA 95035
UNITED STATES

Email: mankamis@cisco.com

Sridhar Santhanam
Cisco Systems
821 Alder Drive,
MILPITAS, CALIFORNIA 95035
UNITED STATES

Email: sridsant@cisco.com

Aravind Paramasivam
Cisco Systems
821 Alder Drive,
MILPITAS, CALIFORNIA 95035
UNITED STATES

Email: arparama@cisco.com

Joseph Goh
Cisco Systems
SINGAPORE

Email: hocgoh@cisco.com

Gyan S. Mishra
Verizon Communications Inc. (VZ)
13101 Columbia Pike FDC1 Rm 304-D
Silver Spring MD 20904
UNITED STATES

Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 12, 2019

S. Venaas
B. Ganesh
K. Thiruvenkatasamy
R. Chokkanathapuram
Cisco Systems, Inc.

March 11, 2019

PIM Flooding Protocol over Reliable Transport
draft-venaas-pim-port-pfm-00

Abstract

The PIM Flooding Protocol (PFM) defined in RFC8364 relies on sending periodic updates as it does not provide for any reliability. If a message is lost, the information will be provided in the next periodic update.

This document extends the Reliable Transport Mechanism for PIM in RFC6559 to allow for sending PFM messages. This significantly reduces the PFM signaling by not requiring frequent periodic updates, and it provides for retransmission, allowing for quick recovery when an IP packet is dropped.

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 https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 12, 2019.

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
3. Protocol specification 3
4. PFM over PORT message definitions 5
4.1. PORT PFM Update 5
4.2. PORT PFM Request 7
4.3. PORT PFM Update Option 8
4.4. PORT PFM Request Option 8
5. Security Considerations 9
6. IANA considerations 9
7. Normative References 9
Authors' Addresses 10

1. Introduction

The PIM Flooding Protocol (PFM) defined in [RFC8364] relies on sending periodic updates as it does not provide for any reliability. If a message is lost, the information will be provided in the next periodic update. With PFM, a router will typically originate a full update every 60 seconds. This ensures that in case of packet drops, one usually will recover in 60 seconds. There is a trade-off between the number of updates and the recovery time.

This document extends the Reliable Transport Mechanism for PIM in [RFC6559] to allow for sending PFM messages. We will refer to it as PORT (PIM Over Reliable Transport). The use of PORT significantly reduces the PFM signaling by not requiring frequent periodic updates, and it provides for retransmission, allowing for quick recovery when an IP packet is dropped. There will still be some full updates, but they can be sent much more rarely. If there is a packet drop, the reliable transport (TCP/SCTP) will ensure retransmission.

The PORT sessions are established as specified in [RFC6559] between PIM neighbors. The sessions may be used to send other PORT messages, or they can be used only for PFM. Unless all the neighbors support PFM over PORT, regular PFM is used. How to signal support and how a router relays a PFM over PORT message as regular PFM and vice versa will be discussed in a later revision.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Protocol specification

PFM messages are sent over PORT by sending PORT PFM Update messages. They contain a PFM message as defined in [RFC8364]. They also contain a Full ID and a Delta ID that together specifies an ID for the update. Some updates are full updates, they contain all the information an originator is announcing. This would be similar to the periodic updates in regular PFM. Full updates over PORT are sent after some a configurable number of deltas have been sent, or whenever information needs to be withdrawn. Delta updates are used for triggered updates, similar to triggered updates in regular PFM. Each time there is some change a delta update can be triggered.

The Full ID is an unsigned 48 bit value and it is assumed that it is always increasing. That is, any Full Update MUST always have a Full ID larger than any previous updates ever sent using the same Originator address. This MUST also be preserved if the router is reloaded. For the protocol to work, it may also be necessary to ensure this if an address used as an originator address is moved to a different router. It is RECOMMENDED that implementations use the number of seconds since 0h UTC on 1 January 1900 as the ID value. This allows for this protocol to be used for about four million years from the time of publication of this document. If for any reason the clock on a router is adjusted to a value back in time, an implementation would have to ensure that values are still increasing. Since Full Updates do not need to be sent every second, one should in this case be able to catch up.

The first time a router originates a PFM message, it sends a Full update, even though it likely is triggered by some event. Full updates always have the Delta ID set to zero. After that it may send several Delta updates. For each Delta update, the Delta ID is incremented, while the Full ID remains the same. After some time it may decide to send a new full update. The Full ID in the full update MUST be larger than the Full ID in the previous update, and Delta ID is reset to zero. A Full update always has Delta ID zero, and a Delta update always has a non-zero Delta ID.

When a router receives an update it performs RPF check as in regular PFM, boundary processing as in regular PFM. For each interface where

the update would have been forwarded in regular PFM, it will be sent over PORT to all PFM PORT neighbors on the interface. If there are any neighbors on the interface not supporting PFM PORT it MAY revert to sending unreliable PFM messages.

When a router receives a Full update it will remove any stored information from the originator and store the information in the new update. When it receives a Delta update it stores the update and keeps all previous information.

Due to routers being restarted, PORT connections going down etc., some routers MAY have missed some updates, potentially not having received any updates when restarting. In order to receive the most recent data from a neighbor it sends a PORT PFM Request message. For each originator the router has stored information from, it will include an option indicating the Full and Delta IDs of the last message received from that originator. A router receiving the Request compares the IDs of the specified originators with the latest data it has for these originators. If it has a more recent full update, it will first send it to the neighbor. Next, if it has more recent delta updates, it will send all the delta updates in the order they were received. This means that the requesting router receives the messages in order. It will first get a full update if a more recent version exists. The ID of this update may be much larger than the previously seen ID. The first Delta update received, if any, will have ID one if a Full update was received, or one larger than the Delta ID in the request, if not. If multiple Delta updates are received, the Delta ID will increment by one for each update. If the router has stored information for any originators not included in the request message, it will also send this information. It will first send the stored Full update, and then the Delta updates. As discussed above, the Delta updates MUST be sent in the order they were received, first sending update one, then update two, and so forth.

The Delta ID is an unsigned 16 bits value. It never wraps around. A router MUST send a new full update if the Delta ID value is reaching its maximum value. It is RECOMMENDED having a configurable limit for how many Delta updates can be sent before sending a new Full update. Sending Full updates often is in some ways wasteful, but it limits how many deltas routers need to store, and they are also used to remove information that no longer is needed.

When a router starts up, it is RECOMMENDED that before it originates any messages, it sends a PORT PFM Request message to receive any updates that neighbors may have stored for the originator address it would use. It could simply not include an option with the originator address it would use, and receive any information neighbors may have,

or it could could include an option, but with the Full ID set to a value smaller than the Full ID it would use for the next Full Update. E.g., if the ID is based on the number of seconds since the epoch, it could send a request based on the current time. It would then normally get no updates from the neighbors with its own ID. If it does, it is RECOMMENDED to log an error, and ensure that the Full IDs of the next future Full Updates are larger than what was received.

In order to handle extra ordinary cases where a router has originated messages with an erroneously large Full ID, it is RECOMMENDED that implementations provides a way for an administrator to clear the stored PFM state on a router, as well as a way to trigger sending of a Full Update on an originator. This means that as a last resort, an administrator could clear the state for an originator on all the routers, and optionally afterwards trigger a full update by the originator.

4. PFM over PORT message definitions

We define a new PORT message for sending a PFM message. This consists of an update version and a new PORT option containing a PFM message as defined in [RFC8364]. We also define a new PORT message for requesting a PFM update from a neighbor. This contains the latest update version that the router has from each originator and requests the neighbor to transmit any information that it is missing.

4.1. PORT PFM Update

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = TBD1          |        Message Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Interface                            |
|                              ID                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Full-update                          |
|                              +-------------------------------|
|              ID              |       Delta-update ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type          |      Option Value Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                               .                               \
/                               .                               /
\                               .                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type          |      Option Value Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Type is TBD.

Message Length: Length in bytes for the value part of the Type/Length/Value encoding. If no PORT Options are included, the length is 12. If n PORT Options with Option Value lengths L1, L2, ..., Ln are included, the message length is 12 + 4*n + L1 + L2 + ... + Ln.

Reserved: Set to zero on transmission and ignored on receipt.

Interface ID: This MUST be the Interface ID of the Interface ID Hello Option contained in the PIM Hello messages that the PIM router is sending to the PIM neighbor. It indicates to the PIM neighbor what interface to associate the update with. This is similar to how the Interface ID is used in [RFC6559]. The

Interface ID allows us to do connection sharing while still allowing the regular PFM RPF neighbor validation.

Full-update ID: If this is a full update, it is the ID of this update. If this is a delta, then this is the ID of the last full update. This is a 48 bit value.

Delta-update ID: If this is a delta update, this is the ID of the delta. Note that the Full-update ID is also used for a delta. If this is a full update, delta-update is set to 0. This is a 16 bit value.

PORT Options: The general format is defined in [RFC6559] section 5.3. This message MUST contain exactly one PFM Update PORT option. The PFM Update PORT option is defined below. It MAY contain other options that are defined for use in a PORT PFM Update message.

4.2. PORT PFM Request

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = TBD2          |        Message Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type           |      Option Value Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                               .                               \
/                               .                               /
\                               .                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type           |      Option Value Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Type is TBD.

Message Length: Length in bytes for the value part of the Type/Length/Value encoding. If no PORT Options are included, the length is 12. If n PORT Options with Option Value lengths L1, L2, ..., Ln are included, the message length is 12 + 4*n + L1 + L2 + ... + Ln.

Reserved: Set to zero on transmission and ignored on receipt.

PORT Options: The general format is defined in [RFC6559] section 5.3. This message MAY contain zero, one or multiple PFM Request PORT options. The options indicate which versions the requesting router has from which originators; one option per originator. No options, means that the requesting router wants a full update for all known originators. The PFM Request PORT option is defined below. It MAY contain other options that are defined for use in a PORT PFM Request message.

4.3. PORT PFM Update Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type = TBD3    |     Option Value Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            PFM Message                        |
|                                .                              |
|                                .                              |
|                                .                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Type is TBD.

Option Value Length: The number of octets that make up the PFM Message.

PFM Message: A PFM Message as defined in [RFC8364].

4.4. PORT PFM Request Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PORT Option Type = TBD4    |     Option Value Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Originator Address (Encoded-Unicast format)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Full-update                         |
|                               +-------------------------------|
|              ID               |       Delta-update ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Type is TBD.

Option Value Length: The length in octets of the originator address plus 6.

Originator Address: The address of an originator as defined in [RFC8364].

Full-update ID: The ID of the last full update that the router has stored. It is requesting getting the most recent newer full update, if any exists. Plus, any deltas after the last full update.

Delta-update ID: The ID of the last delta update that the router has stored. It is requesting getting the most recent newer full update, using the Full-update ID, if it exists plus any deltas after that. If there are no more recent full updates, then it is requesting any delta updates more recent than this ID.

5. Security Considerations

To be completed. Largely similar to the considerations for PIM PORT. One may use TCP/SCTP authentication mechanisms.

6. IANA considerations

To be completed. IANA would need to assign types for the messages and options defined.

7. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6559] Farinacci, D., Wijnands, IJ., Venaas, S., and M. Napierala, "A Reliable Transport Mechanism for PIM", RFC 6559, DOI 10.17487/RFC6559, March 2012, <https://www.rfc-editor.org/info/rfc6559>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <https://www.rfc-editor.org/info/rfc8364>.

Authors' Addresses

Stig Venaas
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: stig@cisco.com

Balaji Ganesh
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: balagane@cisco.com

Kesavan Thiruvenkatasamy
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: kethiruv@cisco.com

Ramakrishnan Chokkanathapuram
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: ramaksun@cisco.com

PIM Working Group
Internet-Draft
Intended status: Informational
Expires: September 9, 2019

M. Mishra
Cisco
T. Eckert
Huawei
H. Asaeda
NICT
A. Peter

O. Komolafe
Arista Networks
S. Babu
Juniper
N. Leymann
DT
R. Josyula
Arris
T. Winters
UNH

March 8, 2019

IGMPv3 and MLDv2 Survey
draft-volunteers-pim-igmp-mld-bis-00

Abstract

The PIM WG intends to progress IGMPv3 and MLDv2 from Proposed Standards to Internet Standards. This document describes the motivation, procedures and questions proposed for a survey of operators, vendors and implementors of IGMPv3 and MLDv2. The objective of the survey is to collate information to help the PIM WG progress these protocols to Internet Standards.

This Internet-Draft will expire on September 9, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. Procedures Followed 3
  2.1. Methodology 3
  2.2. Intended Recipients of Questionnaire 3
  2.3. Processing of Responses 3
3. Questionnaire 3
  3.1. Questionnaire for Vendors or Host Implementors 3
    3.1.1. Implementation Status 4
    3.1.2. Implementation Specifics 4
    3.1.3. Implementation Perspectives 4
  3.2. Questionnaire for Network Operators 5
    3.2.1. Deployment Status 5
    3.2.2. Deployment Specifics 5
    3.2.3. Deployment Perspectives 6
4. Acknowledgments 6
5. References 6
  5.1. Normative References 6
  5.2. Informative References 6
Authors' Addresses 7

1. Introduction

Internet Group Management Protocol Version 3 (IGMPv3) [RFC3376] and Multicast Listener Discovery Version 2 (MLDv2) for IPv6 [RFC3810] are currently Proposed Standards. Given the fact that multiple independent implementations of these protocols exist and they have been successfully and widely used operationally, the PIM WG is keen to progress these protocols to Internet Standards. In order to facilitate this effort, it is critical to establish if there are

features specified in [RFC3376] and [RFC3810] that have not been widely used and also to determine any interoperability issues that have arisen from using the protocols.

Following approach taken for PIM-SM, documented in [RFC7063], the PIM WG has decided that conducting a comprehensive survey on implementations and deployment of IGMPv3 and MLDv2 will provide valuable information to facilitate their progression to Internet Standard.

This document describes the procedures proposed for conducting the survey and introduces the proposed questions.

2. Procedures Followed

2.1. Methodology

The PIM WG Chairs will officially kick off the survey and distribute the questionnaire and pertinent information through appropriate forums, aiming to ensure the survey reaches as wide an audience as possible.

2.2. Intended Recipients of Questionnaire

1. Network operators

2. Router vendors

3. Switch vendors

4. Host implementors

2.3. Processing of Responses

Responses received will remain confidential. Only the aggregated results will be published and so it will be impossible to identify the contributions by individual operators, vendors or implementors. Furthermore, an option to submit the completed questionnaire anonymously will be available.

3. Questionnaire

3.1. Questionnaire for Vendors or Host Implementors

Name:

Affiliation/Organization:

Contact Email:

Do you wish to keep your name and affiliation confidential?: Y/N

3.1.1. Implementation Status

Which of the following have you implemented? And for how long has it been implemented?

1. IGMPv1 [RFC1112] implemented?: Y/N, since:

2. IGMPv2 [RFC2236] implemented?: Y/N, since:

3. IGMPv3 [RFC3376] implemented?: Y/N, since:

4. Lightweight IGMPv3 [RFC5790] Implemented: Y/N, since:

5. MLDv1 [RFC2710] implemented?: Y/N, since:

6. MLDv2 [RFC3810] implemented?: Y/N, since:

7. Lightweight MLDv2 [RFC5790] implemented?: Y/N, since:

3.1.2. Implementation Specifics

1. Which IGMPv3 features have you implemented?

2. Which MLDv2 features have you implemented?

3. Have you carried out IGMPv3 or MLDv2 interoperability tests with other implementations? (What issues arose during these tests?) (How could the standards have help minimize these issues?)

3.1.3. Implementation Perspectives

1. What feature(s) has been deliberately omitted from IGMPv3 or MLDv2 implementations? (Because you think it is sub-optimal or potentially has significant disadvantages/issues?) (Because of insufficient demand/use cases?)

2. Which ambiguities or inconsistencies in RFC 3376 or RFC 3810 made the implementation challenging?

3. What suggestions would you make to the PIM WG as it seeks to progress IGMPv3 and MLDv2 to Internet Standard?

3.2. Questionnaire for Network Operators

Name:

Affiliation/Organization:

Contact Email:

Do you wish to keep your name and affiliation confidential?:

3.2.1. Deployment Status

Which of the following are currently deployed in your network? And for how long has it been deployed?

1. IGMPv1 [RFC1112] deployed?: Y/N, since:

2. IGMPv2 [RFC2236] deployed?: Y/N, since:

3. IGMPv3 [RFC3376] deployed?: Y/N, since:

4. Lightweight IGMPv3 [RFC5790] Implemented: Y/N, since:

5. MLDv1 [RFC2710] deployed?: Y/N, since:

6. MLDv2 [RFC3810] deployed?: Y/N, since:

7. Lightweight MLDv2 [RFC5790] deployed?: Y/N, since:

3.2.2. Deployment Specifics

1. Which IGMPv3 features are in use? (Is Exclude mode with source list in use?)

2. Which MLDv2 features are in use? (Is Exclude mode with source list in use?)

3. Does your network rely on the fallback mechanism between different IGMP versions? (Between which IGMP versions?) (What is your experience with this fallback mechanism?)

4. Are you using equipment with different (multi-vendor) implementations for your deployment? (Have you encountered any inter-operability or backward-compatibility issues amongst differing implementations?) (What are your concerns about these issues?)

3.2.3. Deployment Perspectives

1. What have you found to be the strengths of IGMPv3 or MLDv2?

2. What have you found to be the weaknesses of IGMPv3 or MLDv2?

3. What suggestions would you make to the PIM WG as it seeks to progress IGMPv3 and MLDv2 to Internet Standard?

4. Acknowledgments

The authors would like to thank Stig and Mike for valuable review and feedback.

5. References

5.1. Normative References

[RFC1112] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC5790] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.

5.2. Informative References

[RFC7063] Zheng, L., Zhang, Z., and R. Parekh, "Survey Report on Protocol Independent Multicast - Sparse Mode (PIM-SM) Implementations and Deployments", RFC 7063, December 2013.

Authors' Addresses

Mankamana Mishra
Cisco Systems
821 Alder Drive
Milpitas, CA 95035
USA

Email: mankamis@cisco.com

Toerless Eckert
Huawei Technologies

Email: tte@cs.fau.de

Hitoshi Asaeda
National Institute of Information and Communications Technology

Email: asaeda@nict.go.jp

Anish Peter

Email: anish.ietf@gmail.com

Olufemi Komolafe
Arista Networks

Email: femi@arista.com

Suneesh Babu
Juniper

Email: suneesh@juniper.net

Nicolai Leymann
DT

Email: N.Leymann@telekom.de

Ramakanth Josyula
Arris

Email: ramakanthjosyula@gmail.com

Timothy Winters
UNH

Email: twinters@iol.unh.edu

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: August 11, 2019

H. Zhao
Ericsson
X. Liu
Volta
Y. Liu
Huawei

February 12, 2019

A Yang Data Model for IGMP/MLD Proxy
draft-zhao-pim-igmp-mld-proxy-yang-01.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

Status of this Memo

Table of Contents

1. Introduction.....3
   1.1. Terminology.....3
   1.2. Tree Diagrams.....3
2. Design of Data Model.....3
   2.1. Overview.....4
   2.2. Augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol.....4
3. IGMP/MLD Proxy YANG Module.....5
4. Security Considerations.....10
5. IANA Considerations.....10
6. Normative References.....11
Authors' Addresses.....13

1. Introduction

This document defines a YANG [RFC6020] data model for the management of Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

1.1. Terminology

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

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

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).

- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.

- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying") [RFC4605].

The goal of this document is to define a data model that provides a common user interface to IGMP/MLD proxy. This document provides freedom for vendors to adapt this data model to their product implementations.

2.1. Overview

The IGMP/MLD proxy YANG module defined in this document has all the common building blocks for the IGMP/MLD proxy protocol.

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to enable IGMP/MLD proxy and configure other related parameters.

This YANG module follows the Guidelines for YANG Module Authors (NMDA) [draft-dsdt-nmda-guidelines-01]. This NMDA ("Network Management Datastore Architecture") architecture provides an architectural framework for datastores as they are used by network management protocols such as NETCONF [RFC6241], RESTCONF [RFC8040] and the YANG [RFC7950] data modeling language.

2.2. Augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to configure source lifetime globally and retrieve the IGMP proxy group information for (S,G) or (*,G).

```
module: ietf-igmp-mld-proxy
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw igmp-proxy
       +--rw interfaces
          +--rw interface* [interface-name]
             +--rw interface-name    if:interface-ref
             +--rw version?          uint8
             +--rw enable?           boolean
             +--ro group* [group-address]
                +--ro group-address    inet:ipv4-address
                +--ro up-time?         uint32
                +--ro filter-mode?     enumeration
                +--ro source* [source-address]
                   +--ro source-address          inet:ipv4-address
                   +--ro up-time?                uint32
                   +--ro filter-mode?            enumeration
```

```
                   +--ro downstream-interface* [interface-name]
                      +--ro interface-name    if:interface-ref
                      +--ro filter-mode?      enumeration
```

3. IGMP/MLD Proxy YANG Module

```
<CODE BEGINS> file ietf-igmp-mld-proxy@2019-01-23.yang
module ietf-igmp-mld-proxy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy";
  // replace with IANA namespace when assigned
  prefix imp;

  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>

     Editors:  Hongji Zhao
               <mailto:hongji.zhao@ericsson.com>

               Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

               Yisong Liu
               <mailto:liuyisong@huawei.com>

    ";
```

```
  description
    "The module defines a collection of YANG definitions common for
     all Internet Group Management Protocol (IGMP) and Multicast
     Listener Discovery (MLD) Proxy devices.

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

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

  revision 2019-01-23 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for IGMP and MLD Proxy";
  }


  /*
   * Features
   */


  /*
   * Typedefs
   */


  /*
   * Groupings
   */

  grouping per-interface-config-attributes {

    description "Config attributes under interface view";

    leaf enable {
      type boolean;
      default false;
      description
        "Set the value to true to enable IGMP/MLD proxy";
    }
```

```
  } // per-interface-config-attributes

  grouping state-group-attributes {
    description
      "State group attributes";

    leaf up-time {
      type uint32;
      units seconds;
      description
        "The elapsed time for (S,G) or (*,G).";
    }

    leaf filter-mode {
      type enumeration {
        enum "include" {
          description
            "In include mode, reception of packets sent
             to the specified multicast address is requested
             only from those IP source addresses listed in the
             source-list parameter";
        }
        enum "exclude" {
          description
            "In exclude mode, reception of packets sent
             to the given multicast address is requested
             from all IP source addresses except those
             listed in the source-list parameter.";
        }
      }
      description
        "Filter mode for a multicast group,
         may be either include or exclude.";
    }
  } // state-group-attributes


 /* augments */

  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {

    description
      "IGMP proxy augmentation to routing control plane protocol
       configuration and state.";

    container igmp-proxy {
      description "IGMP proxy";
      container interfaces {
        description
          "Containing a list of upstream interfaces.";
```

```
        list interface {
          key "interface-name";
          description
            "List of upstream interfaces.";

          leaf interface-name {
            type if:interface-ref;
            must "current() != /rt:routing/rt:control-plane-
protocols/pim-base:pim/pim-base:interfaces/pim-base:interface/pim-
base:name" {
              description
                "The upstream interface for IGMP/MLD proxy
                 should not be configured PIM.";
            }
          }

          leaf version {
            type uint8 {
              range "1..3";
            }
            default 2;
            description "IGMP version.";
          }

          uses per-interface-config-attributes;

          list group {
            key "group-address";
            config false;
            description
              "Multicast group membership information
              that joined on the interface.";

            leaf group-address {
              type inet:ipv4-address;
              description
                "Multicast group address.";
            }

            uses state-group-attributes;

            list source {
              key "source-address";
              description
                "List of multicast source information
                 of the multicast group.";
              leaf source-address {
                type inet:ipv4-address;
                description
                  "Multicast source address";
```

```
              }

              uses state-group-attributes;

              list downstream-interface {
                key "interface-name";
                leaf interface-name {
                  type if:interface-ref;
                  description
                    "Downstream interfaces for each upstream-interface";
                }
            leaf filter-mode {
                  type enumeration {
                    enum "include" {
                      description
                        "In include mode, reception of packets sent
                         to the specified multicast address is requested
                         only from those IP source addresses listed in
the source-list parameter";
                    }
                    enum "exclude" {
                      description
                        "In exclude mode, reception of packets sent
                         to the given multicast address is requested
                         from all IP source addresses except those
                         listed in the source-list parameter.";
                    }
                  }
                  description
                    "Filter mode for a multicast group,
                     may be either include or exclude.";
                }
              }
            } // list source
          } // list group
        } // interface
      } // interfaces
    }
  }

 /*  RPCs  */

}
<CODE ENDS>
```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

Unauthorized access to any data node of these subtrees can adversely affect the IGMP/MLD proxy subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP/MLD proxy on this device.

5. IANA Considerations

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

This document registers the following namespace URIs in the IETF XML

registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

---

name: ietf-igmp-mld-proxy

namespace: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy

prefix: imp

reference: RFC XXXX

---

6. Normative References

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4605] B. Fenner, H. He, B. Haberman and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.

[RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

[RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.

[draft-ietf-pim-igmp-mld-yang-06] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-mld-yang-06, Oct 20, 2017.

[draft-dsdt-nmda-guidelines-01] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Guidelines for YANG Module Authors (NMDA)", draft-dsdt-nmda-guidelines-01, May 2017

[draft-ietf-netmod-revised-datastores-03] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-03, July 3, 2017

Authors' Addresses

Hongji Zhao
Ericsson (China) Communications Company Ltd.
Ericsson Tower, No. 5 Lize East Street,
Chaoyang District Beijing 100102, P.R. China

Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA

EMail: Xufeng.liu.ietf@gmail.com

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

Email: liuyisong@huawei.com

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: January 02, 2020

H. Zhao
Ericsson
X. Liu
Volta
Y. Liu
Huawei
M. Panchanathan
Cisco
M. Sivakumar
Juniper

July 03, 2019

A Yang Data Model for IGMP/MLD Proxy
draft-zhao-pim-igmp-mld-proxy-yang-03.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices. The YANG module in this document conforms to Network Management Datastore Architecture (NMDA).

Copyright Notice

Table of Contents

1. Introduction.....3
   1.1. Terminology.....3
   1.2. Tree Diagrams.....3
2. Design of Data Model.....3
   2.1. Overview.....4
   2.2. Augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol.....4
3. IGMP/MLD Proxy YANG Module.....5
4. Security Considerations.....13
5. IANA Considerations.....14
6. Normative References.....15
Authors' Addresses.....17

1. Introduction

This document defines a YANG [RFC6020] data model for the management of Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) proxy devices.

The YANG module in this document conforms to the Network Management Datastore Architecture defined in [RFC8342]. The "Network Management Datastore Architecture" (NMDA) adds the ability to inspect the current operational values for configuration, allowing clients to use identical paths for retrieving the configured values and the operational values.

1.1. Terminology

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

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

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).

- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.

- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of Data Model

The model covers Considerations for Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying") [RFC4605].

The goal of this document is to define a data model that provides a common user interface to IGMP/MLD proxy. This document provides freedom for vendors to adapt this data model to their product implementations.

2.1. Overview

The IGMP/MLD proxy YANG module defined in this document has all the common building blocks for the IGMP/MLD proxy protocol.

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to enable IGMP/MLD proxy and configure other related parameters.

This YANG module follows the Guidelines for YANG Module Authors (NMDA) [draft-dsdt-nmda-guidelines-01]. This NMDA ("Network Management Datastore Architecture") architecture provides an architectural framework for datastores as they are used by network management protocols such as NETCONF [RFC6241], RESTCONF [RFC8040] and the YANG [RFC7950] data modeling language.

2.2. Augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to enable IGMP/MLD proxy under the upstream interface. There is also a constraint to make sure the upstream interface for IGMP/MLD proxy should not be configured PIM.

```
module: ietf-igmp-mld-proxy
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw igmp-proxy {feature-igmp-proxy}?
       +--rw interfaces
          +--rw interface* [interface-name]
             +--rw interface-name    if:interface-ref
             +--rw version?          uint8
             +--rw enable?           boolean
             +--ro group* [group-address]
                +--ro group-address    inet:ipv4-address
                +--ro up-time?         uint32
                +--ro filter-mode?     enumeration
                +--ro source* [source-address]
                   +--ro source-address          inet:ipv4-address
                   +--ro up-time?                uint32
                   +--ro filter-mode?            enumeration
                   +--ro downstream-interface* [interface-name]
                      +--ro interface-name    if:interface-ref
                      +--ro filter-mode?      enumeration
```

```
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw mld-proxy {feature-mld-proxy}?
       +--rw interfaces
          +--rw interface* [interface-name]
             +--rw interface-name    if:interface-ref
             +--rw version?          uint8
             +--rw enable?           boolean
             +--ro group* [group-address]
                +--ro group-address    inet:ipv6-address
                +--ro up-time?         uint32
                +--ro filter-mode?     enumeration
                +--ro source* [source-address]
                   +--ro source-address          inet:ipv6-address
                   +--ro up-time?                uint32
                   +--ro filter-mode?            enumeration
                   +--ro downstream-interface* [interface-name]
                      +--ro interface-name    if:interface-ref
                      +--ro filter-mode?      enumeration
```

3. IGMP/MLD Proxy YANG Module

```
<CODE BEGINS> file ietf-igmp-mld-proxy@2019-07-03.yang
module ietf-igmp-mld-proxy {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy";
  // replace with IANA namespace when assigned
  prefix imp;

  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>

     Editors:  Hongji Zhao
               <mailto:hongji.zhao@ericsson.com>

               Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

               Yisong Liu
               <mailto:liuyisong@huawei.com>

               Mani Panchanathan
               <mailto:mapancha@cisco.com>

               Mahesh Sivakumar
               <mailto:sivakumar.mahesh@gmail.com>

    ";

  description
    "The module defines a collection of YANG definitions common for
     all Internet Group Management Protocol (IGMP) and Multicast
     Listener Discovery (MLD) Proxy devices.

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

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

  revision 2019-07-03 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for IGMP and MLD Proxy";
  }


  /*
   * Features
   */
```

```
  feature feature-igmp-proxy {
    description
      "Support IGMP Proxy protocol.";
    reference
      "RFC 4605";
  }

  feature feature-mld-proxy {
    description
      "Support MLD Proxy protocol.";
    reference
      "RFC 4605";
  }

  /*
   * Identities
   */

  identity igmp-proxy {
    base rt:control-plane-protocol;
    description
      "IGMP Proxy protocol";
  }

  identity mld-proxy {
    base rt:control-plane-protocol;
    description
      "MLD Proxy protocol";
  }


  /*
   * Typedefs
   */


  /*
   * Groupings
   */

  grouping per-interface-config-attributes {

    description "Config attributes under interface view";

    leaf enable {
      type boolean;
      default false;
      description
        "Set the value to true to enable IGMP/MLD proxy";
    }
```

```
  } // per-interface-config-attributes

  grouping state-group-attributes {
    description
      "State group attributes";

    leaf up-time {
      type uint32;
      units seconds;
      description
        "The elapsed time for (S,G) or (*,G).";
    }

    leaf filter-mode {
      type enumeration {
        enum "include" {
          description
            "In include mode, reception of packets sent
             to the specified multicast address is requested
             only from those IP source addresses listed in the
             source-list parameter";
        }
        enum "exclude" {
          description
            "In exclude mode, reception of packets sent
             to the given multicast address is requested
             from all IP source addresses except those
             listed in the source-list parameter.";
        }
      }
      description
        "Filter mode for a multicast group,
         may be either include or exclude.";
    }
  } // state-group-attributes


 /* augments */

  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {

    description
      "IGMP Proxy augmentation to routing control plane protocol
       configuration and state.";

    container igmp-proxy {
     when 'derived-from-or-self(../rt:type, "imp:igmp-proxy")' {
        description
          "This container is only valid for IGMP Proxy protocol.";
      }
```

```
      if-feature feature-igmp-proxy;
      description "IGMP proxy";
      container interfaces {
        description
          "Containing a list of upstream interfaces.";

        list interface {
          key "interface-name";
          description
            "List of upstream interfaces.";


          leaf interface-name {
            type if:interface-ref;
            must "not( current() = /rt:routing"+
           "/rt:control-plane-protocols/pim-base:pim"+
           "/pim-base:interfaces/pim-base:interface"+
           "/pim-base:name )" {

           description
                "The upstream interface for IGMP proxy
                 should not be configured PIM.";
            }
            description "The upstream interface name.";
          }

          leaf version {
            type uint8 {
              range "1..3";
            }
            default 2;
            description "IGMP version.";
          }

          uses per-interface-config-attributes;

          list group {
            key "group-address";
            config false;
            description
              "Multicast group membership information
              that joined on the interface.";

            leaf group-address {
              type inet:ipv4-address;
              description
                "Multicast group address.";
            }

            uses state-group-attributes;
```

```
            list source {
              key "source-address";
              description
                "List of multicast source information
                 of the multicast group.";
              leaf source-address {
                type inet:ipv4-address;
                description
                  "Multicast source address";
              }

              uses state-group-attributes;

              list downstream-interface {
                key "interface-name";
                description "The downstream interfaces list.";
                leaf interface-name {
                  type if:interface-ref;
                  description
                    "Downstream interfaces for each upstream-interface";
                }
            leaf filter-mode {
                  type enumeration {
                    enum "include" {
                      description
                        "In include mode, reception of packets sent
                         to the specified multicast address is requested
                         only from those IP source addresses listed in
the
                         source-list parameter";
                    }
                    enum "exclude" {
                      description
                        "In exclude mode, reception of packets sent
                         to the given multicast address is requested
                         from all IP source addresses except those
                         listed in the source-list parameter.";
                    }
                  }
                  description
                    "Filter mode for a multicast group,
                     may be either include or exclude.";
                }
              }
            } // list source
          } // list group
        } // interface
      } // interfaces
    }
  }
```

```
  augment "/rt:routing/rt:control-plane-protocols"+
        "/rt:control-plane-protocol" {

    description
      "MLD Proxy augmentation to routing control plane protocol
       configuration and state.";

    container mld-proxy {
      when 'derived-from-or-self(../rt:type, "imp:mld-proxy")' {
        description
          "This container is only valid for MLD Proxy protocol.";
      }
      if-feature feature-mld-proxy;
      description "MLD proxy";
      container interfaces {
        description
          "Containing a list of upstream interfaces.";

        list interface {
          key "interface-name";
          description
            "List of upstream interfaces.";

          leaf interface-name {
            type if:interface-ref;
            must "not( current() = /rt:routing"+
           "/rt:control-plane-protocols/pim-base:pim"+
           "/pim-base:interfaces/pim-base:interface"+
           "/pim-base:name )" {

           description
                "The upstream interface for MLD proxy
                 should not be configured PIM.";
            }
           description "The upstream interface name.";
          }

          leaf version {
            type uint8 {
              range "1..2";
            }
            default 2;
            description "MLD version.";
          }

          uses per-interface-config-attributes;

          list group {
            key "group-address";
            config false;
            description
```

```
              "Multicast group membership information
              that joined on the interface.";

            leaf group-address {
              type inet:ipv6-address;
              description
                "Multicast group address.";
            }

            uses state-group-attributes;

            list source {
              key "source-address";
              description
                "List of multicast source information
                 of the multicast group.";
              leaf source-address {
                type inet:ipv6-address;
                description
                  "Multicast source address";
              }

              uses state-group-attributes;

              list downstream-interface {
                key "interface-name";
                description "The downstream interfaces list.";
                leaf interface-name {
                  type if:interface-ref;
                  description
                    "Downstream interfaces for each upstream-interface";
                }
            leaf filter-mode {
                  type enumeration {
                    enum "include" {
                      description
                        "In include mode, reception of packets sent
                         to the specified multicast address is requested
                         only from those IP source addresses listed in
the
                         source-list parameter";
                    }
                    enum "exclude" {
                      description
                        "In exclude mode, reception of packets sent
                         to the given multicast address is requested
                         from all IP source addresses except those
                         listed in the source-list parameter.";
                    }
                  }
                  description
```

```
                    "Filter mode for a multicast group,
                     may be either include or exclude.";
                }
              }
            } // list source
          } // list group
        } // interface
      } // interfaces
    }
  }

 /*  RPCs  */

}
<CODE ENDS>
```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

Unauthorized access to any data node of these subtrees can adversely affect the IGMP/MLD proxy subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus

important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol

Unauthorized access to any data node of these subtrees can disclose the operational state information of IGMP/MLD proxy on this device.

5. IANA Considerations

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

This document registers the following namespace URIs in the IETF XML

registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

---

name: ietf-igmp-mld-proxy

namespace: urn:ietf:params:xml:ns:yang:ietf-igmp-mld-proxy

prefix: imp

reference: RFC XXXX

---

6. Normative References

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4605] B. Fenner, H. He, B. Haberman and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) - Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013.

[RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

[RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", RFC 8343, March 2018.

[draft-ietf-pim-igmp-mld-yang-06] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-mld-yang-06, Oct 20, 2017.

[draft-dsdt-nmda-guidelines-01] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Guidelines for YANG Module Authors (NMDA)", draft-dsdt-nmda-guidelines-01, May 2017

[draft-ietf-netmod-revised-datastores-03] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-03, July 3, 2017

Authors' Addresses

Hongji Zhao
Ericsson (China) Communications Company Ltd.
Ericsson Tower, No. 5 Lize East Street,
Chaoyang District Beijing 100102, P.R. China
Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA
EMail: Xufeng.liu.ietf@gmail.com

Yisong Liu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China
Email: liuyisong@huawei.com

Mani Panchanathan
Cisco
India
Email: mapancha@cisco.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA
EMail: sivakumar.mahesh@gmail.com