PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: April 27, 2020

H. Zhao
Ericsson
X. Liu
Volta
Y. Liu
Huawei
M. Panchanathan
Cisco
M. Sivakumar
Juniper

October 28, 2019

A Yang Data Model for IGMP/MLD Proxy
draft-ietf-pim-igmp-mld-proxy-yang-00.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).

This Internet-Draft will expire on April 27, 2020.

Table of Contents

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:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- 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 IGMP/MLD proxy. The interface list under igmp-proxy or mld-proxy contains upstream interfaces for IGMP/MLD proxy. There is also a constraint to make sure the upstream interface for IGMP/MLD proxy should not be configured PIM.

To configure a downstream interface for IGMP/MLD proxy, enable IGMP/MLD on that interface. This is defined in the YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD). In IGMP/MLD proxy model downstream-interface is read-only.

```
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