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 the Network Management Datastore Architecture (NMDA).
with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

**Table of Contents**

1. Introduction  
   1.1. Terminology  
   1.2. Tree Diagrams  
   1.3. Prefixes in Data Node Names  

2. Design of Data Model  
   2.1. Overview  
   2.2. Optional Features  
   2.3. Position of Address Family in Hierarchy  

3. Module Structure  
   3.1. IGMP Proxy Configuration and Operational State  
   3.2. MLD Proxy Configuration and Operational State  

4. IGMP/MLD Proxy YANG Module  

5. Security Considerations  

6. IANA Considerations  
   6.1. IETF XML Registry  
   6.2. YANG Module Names Registry  

7. References  
   7.1. Normative References  
   7.2. Informative References  

Appendix A. Data Tree Example  

Authors’ Addresses
1. Introduction

This document defines a YANG data model [RFC7950] for the management of Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) Proxy devices [RFC4605]. The YANG module in this document conforms to the Network Management Datastore Architecture as defined in [RFC8342].

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 abbreviations are used in this document and in the defined YANG data model:

IGMP:  Internet Group Management Protocol [RFC3376].
MLD:   Multicast Listener Discovery [RFC3810].
PIM:   Protocol Independent Multicast [RFC7761].

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 and other data model objects are often used without a prefix, as long as the context clearly indicates the YANG module in which each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>YANG Module</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>inet</td>
<td>ietf-inet-types</td>
<td>[RFC6991]</td>
</tr>
<tr>
<td>if</td>
<td>ietf-interfaces</td>
<td>[RFC8343]</td>
</tr>
<tr>
<td>rt</td>
<td>ietf-routing</td>
<td>[RFC8349]</td>
</tr>
</tbody>
</table>
2. Design of Data Model

The model covers forwarding based on IGMP and MLD proxying [RFC4605]. One goal of this document is to define a data model that provides a common user interface for IGMP/MLD Proxy devices.

2.1. Overview

The model defined in this document has all the common building blocks for IGMP/MLD Proxy devices and can be used to configure those devices. The operational state data and statistics can also be retrieved via this model.

2.2. Optional Features

This model is designed to represent the basic capability subsets of IGMP/MLD Proxies. The main design goals of this document are that (1) the basic capabilities described in the model will be supported by any major implementations that exist at the time of this writing and (2) the configuration of all implementations meeting the specifications will be easy to express through some combination of the optional features in the model and simple vendor augmentations.

This model declares two features representing capabilities that not all deployed devices support. One feature is called “igmp-proxy”, and the other feature is called “mld-proxy”. Either or both features could be implemented; this would provide more choices for vendors.

2.3. Position of Address Family in Hierarchy

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

This document defines IGMP Proxies and MLD Proxies in separate schema branches in the structure. The benefits of this technique are as follows:

- The model can support IGMP Proxies (IPv4), MLD Proxies (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 the model defined in [RFC8652], which uses separate branches for IPv4 and IPv6.
• Having separate branches for IGMP Proxies and MLD Proxies allows minor differences in their behavior to be modeled 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-proxy <= Augmented by this model
        |     ...
        |      +--rw mld-proxy <= Augmented by this model
```

The "igmp-proxy" container instantiates an IGMP Proxy. The "mld-proxy" container instantiates an MLD Proxy.

3.1. IGMP Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the igmp-proxy container.

All attributes related to IGMP Proxies are defined in the igmp-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The igmp-version parameter represents the IGMP protocol version; the default value is 2. If the value of the "enabled" parameter is "true", it means that the IGMP Proxy is enabled.

The interface list under igmp-proxy contains upstream interfaces for an IGMP Proxy. A constraint is provided to make sure that the upstream interface for the IGMP Proxy is not configured to use PIM.

To configure a downstream interface for an IGMP Proxy, the ability to enable IGMP on that interface is needed. This is defined in "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)" [RFC8652].

Zhao, et al. Standards Track Page 5
3.2. MLD Proxy Configuration and Operational State

The YANG module augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol to add the mld-proxy container.

All attributes related to MLD Proxies are defined in the mld-proxy container. The read-write attributes represent configurable data. The read-only attributes represent state data.

The mld-version parameter represents the MLD protocol version; the default value is 2. If the value of the "enabled" parameter is "true", it means that the MLD Proxy is enabled.

The interface list under mld-proxy contains upstream interfaces for an MLD Proxy. A constraint is provided to make sure that the upstream interface for the MLD Proxy is not configured to use PIM.

To configure a downstream interface for an MLD Proxy, enable MLD on that interface. This is defined in "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)" [RFC8652].
4. IGMP/MLD Proxy YANG Module

This module references [RFC4605], [RFC6991], [RFC8294], [RFC8343], [RFC8349], and [RFC9128].
prefix pim-base;
reference
  "RFC 9128: YANG Data Model for Protocol Independent Multicast
  (PIM)";
}

organization
  "IETF PIM Working Group";
contact
  "WG Web: <https://datatracker.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>
           Mani Panchanathan <mailto:mapancha@cisco.com>
           Mahesh Sivakumar <mailto:sivakumar.mahesh@gmail.com>"

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

  Copyright (c) 2023 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 Revised BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents

  This version of this YANG module is part of RFC 9398; see the
  RFC itself for full legal notices."

revision 2023-05-30 {
  description
    "Initial revision.";
  reference
    "RFC 9398: A YANG Data Model for Internet Group Management
    Protocol (IGMP) and Multicast Listener Discovery (MLD)
    Proxy Devices";
}

/*
 * Features
 */

feature igmp-proxy {
description "Support for the IGMP Proxy protocol.";
reference
"RFC 4605: Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ('IGMP/MLD Proxying')";
}

feature mld-proxy {
    description "Support for the MLD Proxy protocol.";
    reference "RFC 4605: Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ('IGMP/MLD Proxying')";
}

/*
 * 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.";
}

/*
 * Groupings
 */

grouping per-interface-config-attributes {
    description "'config' attributes as listed under an interface entry.";
    leaf enabled {
        type boolean;
        default "true";
        description "Set the value to 'true' to enable the 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.";
    }
} // state-group-attributes

/* augments */

augment "/rt:routing/rt:control-plane-protocols"
  + "/rt:control-plane-protocol" {
    when "derived-from-or-self(rt:type, "
    + "'igmp-mld-proxy:igmp-proxy')"
    description
        "This augmentation is only valid for IGMP Proxies.";
}

description
    "IGMP Proxy augmentation to routing control plane protocol
configuration and state.";
container igmp-proxy {
    if-feature "igmp-proxy";
    presence "IGMP Proxy configuration.";
    description
        "IGMP Proxy instance configuration.";
    container interfaces {
        description
            "Contains a list of upstream interfaces.";
        list interface {
            key "name";
            description
                "List of upstream interfaces.";
            leaf 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 the IGMP Proxy
must not be configured to use PIM.";
                }
            }
        }
    }
}
"The upstream interface name."

} // list source
} // list group
} // list downstream-interface
} // leaf ifname
} // list ifname
} // leaf igmp-version

} // list group
} // list source
} // leaf igmp-version

} // list group
} // leaf sender-source-address

} // list group
} // leaf group-address

} // list group
} // leaf source-address

} // leaf up-time

} // list downstream-interface
} // leaf ifname

} // list downstream-interface
} // leaf ifname

} // leaf igmp-version
augment "/rt:routing/rt:control-plane-protocols"
  + "/rt:control-plane-protocol" {
    when "derived-from-or-self(rt:type, "
    + "'igmp-mld-proxy:mld-proxy')"
    { description
      "This augmentation is only valid for MLD Proxies.";
    }
    description
    "MLD Proxy augmentation to routing control plane protocol configuration and state.";
    container mld-proxy {
      if-feature "mld-proxy";
      presence "MLD Proxy configuration.";
      description
      "MLD Proxy instance configuration.";
      container interfaces {
        description
        "Contains a list of upstream interfaces.";
        list interface {
          key "name";
          description
          "List of upstream interfaces.";
          leaf 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 the MLD Proxy must not be configured to use PIM.";
            }
            description
            "The upstream interface name.";
          }
          leaf mld-version {
            type uint8 {
              range "1..2";
            }
            default "2";
            description
            "MLD version.";
          }
        }
        uses per-interface-config-attributes;
        leaf sender-source-address {
          type inet:ipv6-address-no-zone;
          description
          "The sender source address of an MLD membership report message or leave message.";
        }
        list group {
          key "group-address";
          config false;
        }
      }
    }
description
"List of the multicast groups in the membership
database built on this upstream interface.";
leaf group-address {
    type rt-types:ipv6-multicast-group-address;
    description
    "Multicast group address.";
}
uses state-group-attributes;
list source {
    key "source-address";
    description
    "Multicast source information
    for the multicast group.";
    leaf source-address {
        type inet:ipv6-address-no-zone;
        description
        "Multicast source address.";
    }
    leaf up-time {
        type uint32;
        units "seconds";
        description
        "The elapsed time for (S,G) or (*,G).";
    }
list downstream-interface {
    key "name";
    description
    "List of downstream interfaces.";
    leaf name {
        type if:interface-ref;
        description
        "Downstream interfaces
        for each upstream interface.";
    }
} // list source
} // list group
} // interface
} // interfaces

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:/ igmp-mld-proxy:igmp-proxy:

igmp-mld-proxy:interfaces
   This subtree specifies the interface list for an IGMP Proxy. Modifying the configuration may cause the IGMP Proxy interface to be deleted or changed.

igmp-mld-proxy:interfaces/interface
   This subtree specifies the configuration for the IGMP Proxy attributes at the interface level. Modifying the configuration may cause the IGMP Proxy to be deleted or changed on a specific interface.

Under /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:/ igmp-mld-proxy:mld-proxy:

igmp-mld-proxy:interfaces
   This subtree specifies the interface list for an MLD Proxy. Modifying the configuration may cause the MLD Proxy interface to be deleted or changed.

igmp-mld-proxy:interfaces/interface
   This subtree specifies the configuration for the MLD Proxy attributes at the interface level. Modifying the configuration may cause the MLD Proxy to be deleted or changed on a specific interface.

Unauthorized access to any data nodes in 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:
Unauthorized access to any data nodes in these subtrees can disclose operational state information about the IGMP/MLD Proxy on this device. Group information or source information may expose multicast group memberships.

6. IANA Considerations

6.1. IETF XML Registry

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

Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

6.2. YANG Module Names Registry

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

Name: ietf-igmp-mld-proxy
Prefix: igmp-mld-proxy
Reference: RFC 9398

7. References

7.1. Normative References


7.2. Informative References


Appendix A. Data Tree Example

This section contains an example for the IGMP Proxy, shown in JSON encoding [RFC7951] and containing both configuration and state data. In the example, the IGMP Proxy is enabled on interface eth1/1.

The ability to enable IGMP on eth1/2 and eth1/3 is also needed. The configuration details are omitted here because this document is focused on IGMP/MLD Proxies.
The configuration data for R1 in the above figure could be as follows:

```json
{
    "ietf-interfaces:interfaces": {
        "interface": [
            {
                "name": "eth1/1",
                "type": "iana-if-type:ipForward",
                "ietf-ip:ipv4": {
                    "address": [
                        {
                            "ip": "203.0.113.1",
                            "prefix-length": 24
                        }
                    ]
                }
            }
        ]
    },
    "ietf-routing:routing": {
        "control-plane-protocols": {
            "control-plane-protocol": [
                {
                    "type": "ietf-igmp-mld-proxy:igmp-proxy",
                    "name": "proxy1",
                    "ietf-igmp-mld-proxy:igmp-proxy": {
                        "interfaces": {
                            "interface": [
                                {
                                    "name": "eth1/1",
                                    "igmp-version": 3,
                                    "enabled": true
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
}
```

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

```json
{
    "ietf-interfaces:interfaces": {
        "interface": [
            {
                "name": "eth1/1",
                "type": "iana-if-type:ipForward",
                "admin-status": "up",
                "oper-status": "up",
                "if-index": 25678136,
                "statistics": {
                    "discontinuity-time": "2021-05-23T10:34:56-06:00"
                }
            }
        ]
    }
}
```
{"ietf-ip:ipv4": {
  "address": [
    {
      "ip": "203.0.113.1",
      "prefix-length": 24
    }
  ]
},
"ietf-routing:routing": {
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-igmp-mld-proxy:igmp-proxy",
        "name": "proxy1",
        "ietf-igmp-mld-proxy:igmp-proxy": {
          "interfaces": {
            "interface": [
              {
                "name": "eth1/1",
                "igmp-version": 3,
                "enabled": true,
                "group": [
                  {
                    "group-address": "233.252.0.23",
                    "filter-mode": "include",
                    "source": [
                      {
                        "source-address": "192.0.2.1",
                        "downstream-interface": [
                          {
                            "name": "eth1/2"
                          },
                          {
                            "name": "eth1/3"
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          },
        }
      }
    ]
  }
}
}
Authors' Addresses

**Hongji Zhao**  
Ericsson (China) Communications Company Ltd.  
Ericsson Tower, No. 5 Lize East Street  
Beijing  
100102  
China  
Email: hongji.zhao@ericsson.com

**Xufeng Liu**  
Alef Edge  
United States of America  
Email: xufeng.liu.ietf@gmail.com

**Yisong Liu**  
China Mobile  
China  
Email: liuyisong@chinamobile.com

**Mani Panchanathan**  
Cisco Systems, Inc.  
3625 Cisco Way  
San Jose, CA  
United States of America  
Email: mapancha@cisco.com

**Mahesh Sivakumar**  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA  
United States of America  
Email: sivakumar.mahesh@gmail.com