

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: November 29, 2020

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

May 30, 2020

A Yang Data Model for IGMP and MLD Snooping
draft-ietf-pim-igmp-ml-d-snooping-yang-11.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

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/lid-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 November 29, 2020.

Internet-Draft IGMP & MLD Snooping Yang Module

May 30, 2020

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction.....	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.....	7
3.2.	MLD Snooping Instances.....	9
3.3.	Using IGMP and MLD Snooping Instances.....	11
3.4.	IGMP and MLD Snooping RPC.....	12
4.	IGMP and MLD Snooping YANG Module.....	12
5.	Security Considerations.....	34
6.	IANA Considerations.....	35
7.	References.....	36
7.1.	Normative References.....	36
7.2.	Informative References.....	38
Appendix A.	Data Tree Example.....	39
A.1	Bridge scenario.....	39
A.2	L2VPN scenario.....	42

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

[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 [[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 means nodes attached to a switch have 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](#)].

AC: Attachment Circuit [[RFC3916](#)].

PW: Pseudo Wire [[RFC3916](#)].

[1.2.](#) Tree Diagrams

Tree diagrams used in this document follow the notation defined in [[RFC8340](#)].

Zhao & Liu, etc

Expires November 29, 2020

[Page 3]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

[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]
ni	ietf-network-instance	[RFC8529]
pw	ietf-pseudowires	[draft-ietf-bess-l2vpn-yang]
l2vpn	ietf-l2vpn	[draft-ietf-bess-l2vpn-yang]
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 bandwidth waste. 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

Zhao & Liu, etc

Expires November 29, 2020

[Page 4]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

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 scenario of BRIDGE [dot1Qcp] and L2VPN [draft-ietf-bess-l2vpn-yang]. 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.](#) 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.

Zhao & Liu, etc

Expires November 29, 2020

[Page 5]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

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.

* The separate branches for IGMP Snooping and MLD Snooping can accommodate their differences better and cleaner. 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](#)].

A configuration data node is marked as mandatory only when its value must be provided by the user. Where nodes are not essential to protocol operation, they are marked as optional. Some other nodes are essential but have a default specified, so that they are also optional and need not be configured explicitly.

[3.1.](#) 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 represents configurable

data. The read-only attribute represents state data.

One igmp-snooping-instance could be used in one BRIDGE [[dot10cp](#)] instance or L2VPN [[draft-ietf-bess-l2vpn-yang](#)] instance. One igmp-snooping-instance corresponds to one BRIDGE instance or one L2VPN instance.

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

The value of bridge-mrouter-interface, l2vpn-mrouter-interface-ac, l2vpn-mrouter-interface-pw are filled by the 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.

```
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-table-type?                  enumeration
      | +--rw explicit-tracking?                       boolean
      | |       {feature-explicit-tracking}?
      | +--rw exclude-lite?                           boolean
      | |       {feature-exclude-lite}?
      | +--rw send-query?                             boolean
      | +--rw immediate-leave?                        empty
```

Zhao & Liu, etc

Expires November 29, 2020

[Page 7]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```
      | |       {feature-immediate-leave}?
      | +--rw last-member-query-interval?            uint16
      | +--rw query-interval?                        uint16
      | +--rw query-max-response-time?               uint16
      | +--rw require-router-alert?                  boolean
      | |       {feature-require-router-alert}?
      | +--rw robustness-variable?                   uint8
      | +--rw static-bridge-mrouter-interface*       if:interface-ref
      | |       {static-mrouter-interface}?
      | +--rw static-l2vpn-mrouter-interface-ac*     if:interface-ref
      | |       {static-mrouter-interface}?
```



```

|   +---rw static-l2vpn-mrouter-interface-pw*   pw:pseudowire-ref
|   |       {static-mrouter-interface}?
|   +---rw igmp-version?                        uint8
|   +---rw querier-source?                      inet:ipv4-address
|   +---rw static-l2-multicast-group* [group source-addr]
|   |       {feature-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
|   |   +---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-address
|   |   |   +---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
|   |   |   |       {feature-explicit-tracking}?
|   |   |   +---ro last-reporter?                inet:ipv4-address
|   |   |   +---ro host* [host-address] {feature-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	

[3.2.](#) MLD Snooping Instances

The YANG module defines mld-snooping-instance which could be used in the BRIDGE [[dot1Qcp](#)] or L2VPN [[draft-ietf-bess-l2vpn-yang](#)] scenario to enable MLD Snooping.

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

The mld-snooping-instance is the same as IGMP snooping except changing IPv4 addresses to IPv6 addresses. One mld-snooping-instance could be used in one BRIDGE instance or L2VPN instance. One mld-snooping-instance corresponds to one BRIDGE instance or L2VPN instance.

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

The value of bridge-mrouter-interface, l2vpn-mrouter-interface-ac, l2vpn-mrouter-interface-pw are filled by the 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 MLD Snooping related packets.

```
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-table-type? enumeration
  +--rw explicit-tracking? boolean
  |   {feature-explicit-tracking}?
  +--rw exclude-lite? boolean
  |   {feature-exclude-lite}?
  +--rw send-query? boolean
  +--rw immediate-leave? empty
  |   {feature-immediate-leave}?
  +--rw last-member-query-interval? uint16
  +--rw query-interval? uint16
  +--rw query-max-response-time? uint16
  +--rw require-router-alert? boolean
  |   {feature-require-router-alert}?
  +--rw robustness-variable? uint8
  +--rw static-bridge-mrouter-interface* if:interface-ref
  |   {static-mrouter-interface}?
  +--rw static-l2vpn-mrouter-interface-ac* if:interface-ref
  |   {static-mrouter-interface}?
  +--rw static-l2vpn-mrouter-interface-pw* pw:pseudowire-ref
  |   {static-mrouter-interface}?
  +--rw mld-version? uint8
  +--rw querier-source? inet:ipv6-address
  +--rw static-l2-multicast-group* [group source-addr]
  |   {feature-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
  |   +--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

```

```

|         +--ro up-time                               uint32
|         +--ro expire?
|         |         rt-types:timer-value-seconds16
|         +--ro host-count?                           uint32
|         |         {feature-explicit-tracking}?
|         +--ro last-reporter?                         inet:ipv6-address
|         +--ro host* [host-address] {feature-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

```

3.3. Using IGMP and MLD Snooping Instances

The igmp-snooping-instance could be used in the scenario of BRIDGE [[dot1Qcp](#)] or L2VPN [[draft-ietf-bess-l2vpn-yang](#)] to configure the IGMP Snooping.

For the BRIDGE scenario 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.

```
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

```

For the L2VPN scenario this model augments /ni:network-instances/
ni:network-instance/ni:ni-type/l2vpn:l2vpn [[RFC8529](#)] to use igmp-
Zhao & Liu, etc Expires November 29, 2020 [Page 11]

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

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-mld-snooping-instance-ref
+---rw mld-snooping-instance?     igmp-mld-snooping-instance-ref

```

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

[3.4.](#) 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

```

[4.](#) IGMP and MLD Snooping YANG Module

This module references [[RFC3376](#)], [[RFC4541](#)], [[RFC6636](#)], [[RFC6991](#)], [[RFC8343](#)], [[RFC8529](#)], [[dot1Qcp](#)], and [[draft-ietf-bess-l2vpn-yang](#)].

```
<CODE BEGINS> file ietf-igmp-ml-d-snooping@2020-04-29.yang
module ietf-igmp-ml-d-snooping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-ml-d-snooping";

  prefix ims;

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

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

Zhao & Liu, etc Expires November 29, 2020

[Page 12]

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

```
    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 ietf-l2vpn {
    prefix "l2vpn";
    reference
      "draft-ietf-bess-l2vpn-yang: YANG Data Model for MPLS-based
```

```

L2VPN";
}

import ietf-network-instance {
    prefix "ni";
    reference
        "RFC 8529: YANG Data Model for Network Instances";
}

import ietf-pseudowires {
    prefix "pw";
    reference
        "draft-ietf-bess-l2vpn-yang: YANG Data Model for MPLS-based
L2VPN";
}

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";

```

Zhao & Liu, etc

Expires November 29, 2020

[Page 13]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

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 [[RFC 4541](#)].

Copyright (c) 2020 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 2020-04-29 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";
}
```

```
/*
 * Features
 */
```

```
feature feature-igmp-snooping {
  description
    "Support IGMP snooping.";
}

feature feature-mld-snooping {
  description
    "Support MLD snooping.";
}

feature feature-immediate-leave {
  description
    "Support configuration of immediate-leave.";
```



```

}

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

feature static-mrouter-interface {
    description
        "Support configuration of mrouter interface.";
}

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

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

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

feature feature-explicit-tracking {
    description
        "Support configuration of per instance explicit-tracking.";
    reference
        "RFC 6636, Section 3";
}

```

/* identities */

Zhao & Liu, etc

Expires November 29, 2020

[Page 15]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```

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";
}

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

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

```

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

```
    if-feature 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 Attachment Circuit (AC) in L2VPN
forwarding";
    }

    leaf-list l2vpn-outgoing-pw {
        when 'derived-from-or-self(..../scenario,"ims:l2vpn")';
        type pw:pseudowire-ref;
        description "Outgoing Pseudo Wire (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-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 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 exclude-lite {
    if-feature feature-exclude-lite;
    type boolean;
    default false;
    description
        "Track the Lightweight IGMPv3 and MLDv2 protocol report";
    reference "RFC 5790";
}

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 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 "10..10230";
    }
}
```

Zhao & Liu, etc

Expires November 29, 2020

[Page 19]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```
    units one-tenth-second;
    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 one-tenth-second;
    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 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.";
}

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(..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 mld-version {
        type uint8 {
            range "1..2";
        }
        default 2;
        description "MLD 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 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 Attachment Circuit (AC) in L2VPN
forwarding";
    }

    leaf-list l2vpn-outgoing-pw {
        when 'derived-from-or-self(..../scenario,"ims:l2vpn")';
        type pw:pseudowire-ref;
        description "Outgoing Pseudo Wire (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 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-ml-d-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

```

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

```
}

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-ml-d-snooping {
  description
    "State attributes for MLD snooping of each VLAN.";

  uses instance-state-attributes-igmp-ml-d-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-ml-d-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
```

Zhao & Liu, etc

Expires November 29, 2020

[Page 25]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

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

```
list host {
  if-feature 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 Attachment Circuit (AC) in L2VPN
forwarding";
}

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

```

leaf up-time {
Zhao & Liu, etc

Expires November 29, 2020

[Page 26]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```

    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 feature-explicit-tracking;
    type uint32;
    description
        "The number of host addresses.";
}
} // instance-state-source-attributes-igmp-ml-d-snooping

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

    leaf num-query {
        type yang:counter64;
        description
            "The number of query messages.";
    }
}

```

```

}
leaf num-membership-report-v1 {
    type yang:counter64;
    description
        "The number of membership report v1 messages.";
}
leaf num-membership-report-v2 {
    type yang:counter64;
    description
        "The number of membership report v2 messages.";
}
leaf num-membership-report-v3 {
    type yang:counter64;
    description
        "The number of membership report v3 messages.";
}
leaf num-leave {
    type yang:counter64;
    description
        "The number of leave messages.";
}
leaf num-non-member-leave {

```

```

    type yang:counter64;
    description
        "The number of non member leave messages.";
}
leaf num-pim-hello {
    type yang:counter64;
    description
        "The number of PIM hello messages.";
}
} // 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.";
    }
    leaf num-report-v1 {
        type yang:counter64;
        description

```

```

        "The number of Version 1 Multicast Listener Report.";
    }
    leaf num-report-v2 {
        type yang:counter64;
        description
            "The number of Version 2 Multicast Listener Report.";
    }
    leaf num-done {
        type yang:counter64;
        description
            "The number of Version 1 Multicast Listener Done.";
    }
    leaf num-pim-hello {
        type yang:counter64;
        description
            "The number of PIM hello messages.";
    }
} // 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 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.";
        }
        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.";
    }
    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
        "Use 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
    "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";
  }
}

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

  description
    "Use 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 */

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```
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;
    }

```

Zhao & Liu, etc Expires November 29, 2020 [Page 33]

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

```

        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>

```

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

Zhao & Liu, etc

Expires November 29, 2020

[Page 34]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

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.

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. These are the operations and their sensitivity/vulnerability:

ims:clear-igmp-snooping-groups

ims:clear-mld-snooping-groups

The IGMP & MLD Snooping YANG module supports 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.

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

Zhao & Liu, etc Expires November 29, 2020 [Page 35]

Internet-Draft IGMP & MLD Snooping Yang Module May 30, 2020

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.

Zhao & Liu, etc

Expires November 29, 2020

[Page 36]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

[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.

[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.

- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), July 2013.
- [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.
- [RFC8407] A. Bierman, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", [RFC 8407](#), October 2018.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), August 2018.

Zhao & Liu, etc

Expires November 29, 2020

[Page 37]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

- [RFC8529] L. Berger, C. Hopps, A. Lindem, D. Bogdanovic, X. Liu, "YANG Data Model for Network Instances", [RFC 8529](#), March 2019.
- [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.
- [[draft-ietf-bess-l2vpn-yang](#)] Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model

for MPLS-based L2VPN", [draft-ietf-bess-l2vpn-yang-10](#) (work in progress), July 2019.

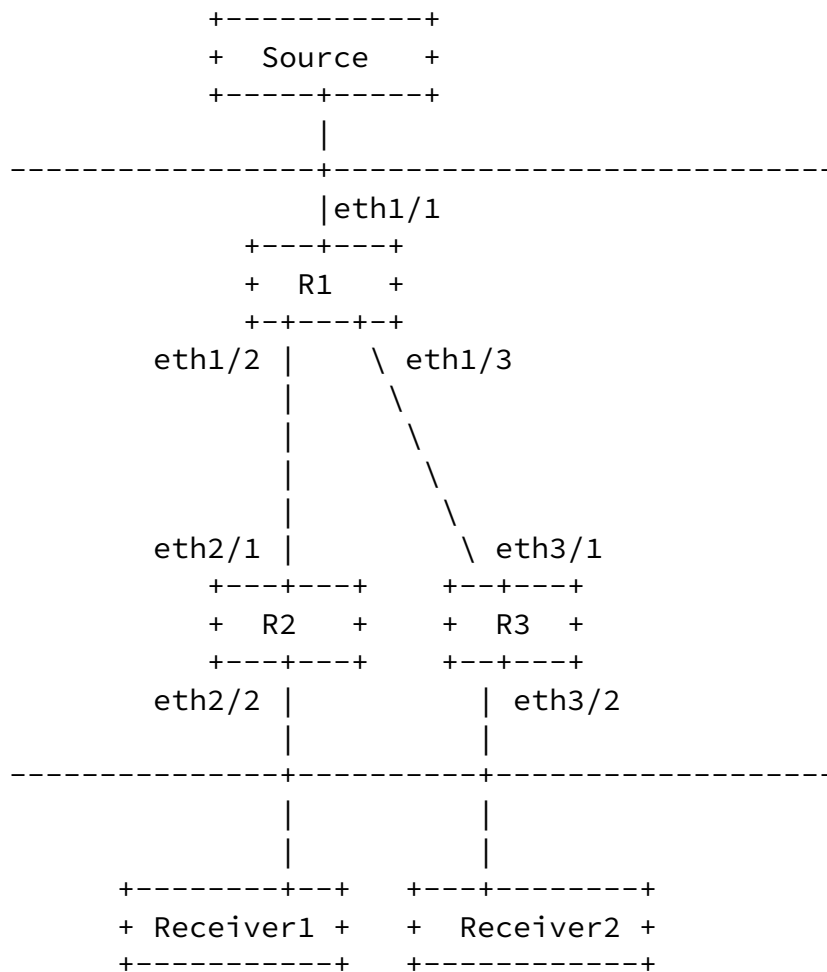
[7.2](#). Informative References

- [RFC3916] X. Xiao, Ed., D. McPherson, Ed., P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", [RFC 3916](#), September 2004.
- [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.
- [RFC7951] L. Lhotka, "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), August 2016.

[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.



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",

```

Zhao & Liu, etc Expires November 29, 2020

[Page 40]

Internet-Draft

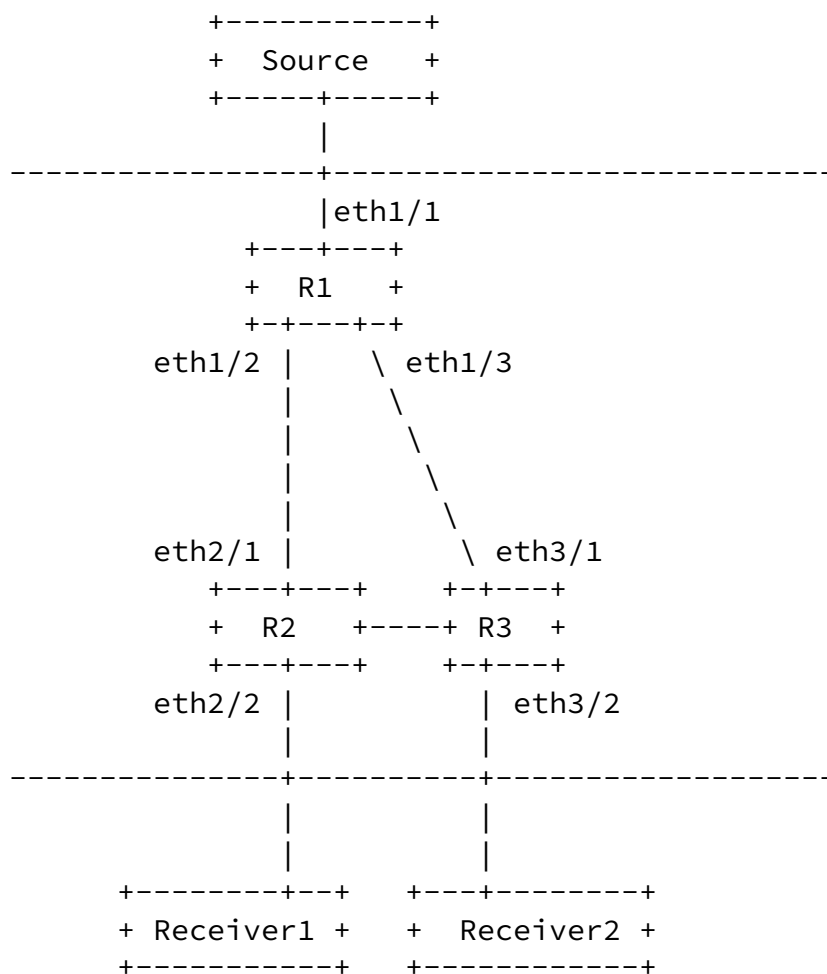
IGMP & MLD Snooping Yang Module

May 30, 2020

```
    "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.



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"
      }
    ]
  }
}
```

Zhao & Liu, etc

Expires November 29, 2020

[Page 42]

Internet-Draft

IGMP & MLD Snooping Yang Module

May 30, 2020

```
    }
  ]
},
"ietf-network-instance:network-instances": {
  "network-instance": [
    {
      "name": "vpls1",
      "ietf-igmp-mlld-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
        }
      }
    ]
  }
}

```

```
}  
}  
]  
}  
}  
}
```

Zhao & Liu, etc	Expires November 29, 2020	[Page 45]
-----------------	---------------------------	-----------

Internet-Draft	IGMP & MLD Snooping Yang Module	May 30, 2020
----------------	---------------------------------	--------------

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
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