

PIM Working Group
Internet Draft
Intended status: Standards Track
Expires: June 09, 2019

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

December 10, 2018

A Yang Data Model for IGMP and MLD Snooping
draft-ietf-pim-igmp-mld-snooping-yang-06.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>

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

This Internet-Draft will expire on June 09, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction.....	3
1.1.	Terminology.....	3
1.2.	Tree Diagrams.....	3
2.	Design of Data Model.....	3
2.1.	Overview.....	4
2.2.	IGMP Snooping Instances.....	4
2.3.	MLD Snooping Instances.....	6
2.4.	IGMP and MLD Snooping Instances Reference.....	8
2.5.	IGMP and MLD Snooping RPC.....	9
3.	IGMP and MLD Snooping YANG Module.....	9
4.	Security Considerations.....	31
5.	IANA Considerations.....	33
6.	Normative References.....	34
Appendix A.	Data Tree Example.....	36
A.1	Bridge scenario.....	36
A.2	L2VPN scenario.....	39
	Authors' Addresses.....	43

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

[1.](#) Introduction

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

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

[1.1.](#) Terminology

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

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

[1.2.](#) Tree Diagrams

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

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).

- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

[2. Design of Data Model](#)

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

Zhao & Liu, etc

Expires June 09, 2019

[Page 3]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

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

[2.1. Overview](#)

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

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

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

[2.2. IGMP Snooping Instances](#)

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

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

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

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

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

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

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

```

|   +--rw static-l2vpn-mrouter-interface-ac*   if:interface-ref
|   |       {static-mrouter-interface}?
|   +--rw static-l2vpn-mrouter-interface-pw*   l2vpn-instance-pw-ref
|   |       {static-mrouter-interface}?
|   +--rw version?                             uint8
|   +--rw querier-source?                       inet:ipv4-address
|   +--rw static-l2-multicast-group* [group source-addr]
|   |       {static-l2-multicast-group}?
|   |   +--rw group
|   |   |       rt-types:ipv4-multicast-group-address
|   |   +--rw source-addr
|   |   |       rt-types:ipv4-multicast-source-address
|   |   +--rw bridge-outgoing-interface*   if:interface-ref
|   |   +--rw l2vpn-outgoing-ac*           l2vpn-instance-ac-ref
|   |   +--rw l2vpn-outgoing-pw*           l2vpn-instance-pw-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*         l2vpn-instance-pw-ref
|   +--ro group* [address]
|   |   +--ro address                         rt-types:ipv4-multicast-group-address
|   |   +--ro mac-address?                   yang:phys-address
|   |   +--ro expire                         uint32
|   |   +--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*           l2vpn-instance-ac-ref

```

```

|   |   +--ro l2vpn-outgoing-pw*           l2vpn-instance-pw-ref
|   |   +--ro up-time                       uint32
|   |   +--ro expire                       uint32
|   |   +--ro host-count?                  uint32
|   |   |       {explicit-tracking}?
|   |   +--ro last-reporter?                inet:ipv4-address
|   |   +--ro host* [host-address] {explicit-tracking}?
|   |   |       +--ro host-address           inet:ipv4-address
|   |   |       +--ro host-filter-mode       filter-mode-type
|   +--ro interfaces
|   |   +--ro interface* [name]
|   |   |       +--ro name                   if:interface-ref
|   |   |       +--ro statistics
|   |   |       +--ro received

```

		+++ro num-query?	yang:counter64
		+++ro num-membership-report-v1?	yang:counter64
		+++ro num-membership-report-v2?	yang:counter64
		+++ro num-membership-report-v3?	yang:counter64
		+++ro num-leave?	yang:counter64
		+++ro num-non-member-leave?	yang:counter64
		+++ro num-pim-hello?	yang:counter64
		+++ro sent	
		+++ro num-query?	yang:counter64
		+++ro num-membership-report-v1?	yang:counter64
		+++ro num-membership-report-v2?	yang:counter64
		+++ro num-membership-report-v3?	yang:counter64
		+++ro num-leave?	yang:counter64
		+++ro num-non-member-leave?	yang:counter64
		+++ro num-pim-hello?	yang:counter64

[2.3.](#) MLD Snooping Instances

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

The mld-snooping-instance is the same as IGMP snooping except changing IPV4 addresses to IPV6 addresses.

```

module: ietf-igmp-mld-snooping
  augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
    +--rw igmp-snooping-instance {feature-igmp-snooping}?
    | ...
    +--rw mld-snooping-instance {feature-mld-snooping}?
      +--rw scenario?                               snooping-scenario-type
      +--rw enable?                                  boolean
      +--rw forwarding-mode?                         enumeration
      +--rw explicit-tracking?                       boolean
      | {explicit-tracking}?

```

+++rw exclude-lite?	boolean
{exclude-lite}?	
+++rw send-query?	boolean
+++rw immediate-leave?	empty
{immediate-leave}?	
+++rw last-member-query-interval?	uint16
+++rw query-interval?	uint16

```

+--rw query-max-response-time?          uint16
+--rw require-router-alert?             boolean
|   {require-router-alert}?
+--rw robustness-variable?              uint8
+--rw static-bridge-mrouter-interface*   if:interface-ref
|   {static-mrouter-interface}?
+--rw static-l2vpn-mrouter-interface-ac*  if:interface-ref
|   {static-mrouter-interface}?
+--rw static-l2vpn-mrouter-interface-pw*  l2vpn-instance-pw-ref
|   {static-mrouter-interface}?
+--rw version?                          uint8
+--rw querier-source?                   inet:ipv6-address
+--rw static-l2-multicast-group* [group source-addr]
|   {static-l2-multicast-group}?
|   +--rw group
|   |   rt-types:ipv6-multicast-group-address
|   +--rw source-addr
|   |   rt-types:ipv6-multicast-source-address
|   +--rw bridge-outgoing-interface*     if:interface-ref
|   +--rw l2vpn-outgoing-ac*              l2vpn-instance-ac-ref
|   +--rw l2vpn-outgoing-pw*              l2vpn-instance-pw-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*         l2vpn-instance-pw-ref
+--ro group* [address]
|   +--ro address                        rt-types:ipv6-multicast-group-address
|   +--ro mac-address?                  yang:phys-address
|   +--ro expire                        uint32
|   +--ro up-time                       uint32
|   +--ro last-reporter?                inet:ipv6-address
|   +--ro source* [address]
|   |   +--ro address
|   |   |   rt-types:ipv6-multicast-source-address
|   |   +--ro bridge-outgoing-interface* if:interface-ref
|   |   +--ro l2vpn-outgoing-ac*          l2vpn-instance-ac-ref
|   |   +--ro l2vpn-outgoing-pw*          l2vpn-instance-pw-ref
|   |   +--ro up-time                     uint32
|   |   +--ro expire                     uint32
|   |   +--ro host-count?                 uint32
|   |   |   {explicit-tracking}?
|   |   +--ro last-reporter?              inet:ipv6-address
|   |   +--ro host* [host-address] {explicit-tracking}?
|   |   |   +--ro host-address            inet:ipv6-address

```



```

|          +--ro host-filter-mode      filter-mode-type
+--ro interfaces
  +--ro interface* [name]
    +--ro name                        if:interface-ref
    +--ro statistics
      +--ro received
        |  +--ro num-query?            yang:counter64
        |  +--ro num-report-v1?        yang:counter64
        |  +--ro num-report-v2?        yang:counter64
        |  +--ro num-done?             yang:counter64
        |  +--ro num-pim-hello?        yang:counter64
      +--ro sent
        +--ro num-query?              yang:counter64
        +--ro num-report-v1?          yang:counter64
        +--ro num-report-v2?          yang:counter64
        +--ro num-done?               yang:counter64
        +--ro num-pim-hello?          yang:counter64

```

[2.4.](#) IGMP and MLD Snooping Instances Reference

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

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

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

```

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

```

```

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

```

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

```

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

```

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

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

[2.5.](#) IGMP and MLD Snooping RPC

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

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

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

```
<CODE BEGINS> file ietf-igmp-mld-snooping@2018-11-28.yang
module ietf-igmp-mld-snooping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-igmp-mld-snooping";

  prefix ims;

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

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

  import ietf-interfaces {
    prefix "if";
```

```
}
```

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

```
import ietf-routing-types {
```

Zhao & Liu, etc

Expires June 09, 2019

[Page 9]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

```
    prefix "rt-types";  
}
```

```
import ietf-l2vpn {  
    prefix "l2vpn";  
}
```

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

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

```
organization  
    "IETF PIM Working Group";
```

```
contact
```

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

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

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

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

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

```
              Mahesh Sivakumar
```

<mailto:sivakumar.mahesh@gmail.com>

";

description

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

Copyright (c) 2018 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

Zhao & Liu, etc

Expires June 09, 2019

[Page 10]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

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 2018-11-28 {

description

"Initial revision.";

reference

"RFC XXXX: A YANG Data Model for IGMP and MLD Snooping";

}

/*

* Features

*/

feature feature-igmp-snooping {

description

"Support IGMP snooping protocol.";

reference

"[RFC 4541, Section 1](#)";

}

feature feature-mls-snooping {

description

"Support MLD snooping protocol.";

```

    reference
        "RFC 4541, Section 1";
}

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

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

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

```

```

    reference
        "RFC 4541, Section 2.1";
}

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

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

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

```

```

}

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

/* identities */

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

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

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

```

```

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

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

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

```

```

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

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

/*
 * Typedefs
 */

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

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

typedef igmp-mld-snooping-instance-ref {
    type leafref {
        path "/rt:routing/rt:control-plane-protocols"+
            "/rt:control-plane-protocol/rt:name";
    }
}

```

```

    }
    description
        "This type is used by data models which need to
        reference IGMP & MLD snooping instance.";
}

typedef l2vpn-instance-ac-ref {
    type leafref {
        path "/ni:network-instances/ni:network-instance"+

```

```

        "/l2vpn:endpoint/l2vpn:name";
    }
    description "l2vpn-instance-ac-ref";
}

typedef l2vpn-instance-pw-ref {
    type leafref {
        path "/ni:network-instances/ni:network-instance"+
            "/l2vpn:endpoint/l2vpn:name";
    }
    description "l2vpn-instance-pw-ref";
}

/*
 * Groupings
 */

grouping instance-config-attributes-igmp-snooping {
    description
        "IGMP snooping configuration for each BRIDGE or L2VPN instance.";

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

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

    leaf querier-source {
        type inet:ipv4-address;
        description
            "Use the IGMP snooping querier to support IGMP
            snooping in a VLAN where PIM and IGMP are not configured.
            The IPV4 address is used as source address in messages.";
    }

    list static-l2-multicast-group {
        if-feature static-l2-multicast-group;
        key "group source-addr";
    }
}

```

```

    description
        "A static multicast route, (*,G) or (S,G).";

```



```

leaf group {
    type rt-types:ipv4-multicast-group-address;
    description
        "Multicast group IPV4 address";
}

leaf source-addr {
    type rt-types:ipv4-multicast-source-address;
    description
        "Multicast source IPV4 address.";
}

leaf-list bridge-outgoing-interface {
    when "../../scenario = 'bridge'";
    type if:interface-ref;
    description "Outgoing interface in BRIDGE forwarding";
}

leaf-list l2vpn-outgoing-ac {
    when "../../scenario = 'l2vpn'";
    type l2vpn-instance-ac-ref;
    description "Outgoing AC in L2VPN forwarding";
}

leaf-list l2vpn-outgoing-pw {
    when "../../scenario = 'l2vpn'";
    type l2vpn-instance-pw-ref;
    description "Outgoing PW in L2VPN forwarding";
}
} // static-l2-multicast-group
} // instance-config-attributes-igmp-snooping

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

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

    leaf forwarding-mode {
        type enumeration {
            enum "mac" {
                description
                    "MAC-based lookup mode";
            }
        }
    }
}

```

```
    }
    enum "ip" {
        description
            "IP-based lookup mode";
    }
}
default "ip";
description "The default forwarding mode is ip";
}

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

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

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

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

leaf last-member-query-interval {
```

```
type uint16 {  
    range "1..1023";  
}  
units seconds;
```

```
    default 1;  
    description  
        "Last Member Query Interval, which may be tuned to modify  
        the leave latency of the network.";  
    reference "RFC3376. Sec. 8.8.";  
}  
  
leaf query-interval {  
    type uint16;  
    units seconds;  
    default 125;  
    description  
        "The Query Interval is the interval between General Queries  
        sent by the Querier.";  
    reference "RFC3376. Sec. 4.1.7, 8.2, 8.14.2.";  
}  
  
leaf query-max-response-time {  
    type uint16;  
    units seconds;  
    default 10;  
    description  
        "Query maximum response time specifies the maximum time  
        allowed before sending a responding report.";  
    reference "RFC3376. Sec. 4.1.1, 8.3, 8.14.3.";  
}  
  
leaf require-router-alert {  
    if-feature require-router-alert;  
    type boolean;  
    default false;  
    description  
        "When the value is true, router alert should exist  
        in the IP head of IGMP or MLD packet.";  
}  
  
leaf robustness-variable {  
    type uint8 {  
        range "1..7";  
    }  
}
```

```

    default 2;
    description
        "Querier's Robustness Variable allows tuning for the
        expected packet loss on a network.";
    reference "RFC3376. Sec. 4.1.6, 8.1, 8.14.1.";
}

leaf-list static-bridge-mrouter-interface {
    when "../scenario = '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 "../scenario = '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 "../scenario = 'l2vpn'";
    if-feature static-mrouter-interface;
    type l2vpn-instance-pw-ref;
    description
        "static mrouter interface whose type is PW
        in L2VPN forwarding";
}
} // instance-config-attributes-igmp-mld-snooping

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

    uses instance-config-attributes-igmp-ml-d-snooping;

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

```

```

}

leaf querier-source {
  type inet:ipv6-address;
  description
    "Use the MLD snooping querier to support MLD snooping where
    PIM and MLD are not configured. The IPV6 address is used as
    the source address in messages.";
}

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

  leaf group {
    type rt-types:ipv6-multicast-group-address;

```

```

  description
    "Multicast group IPV6 address";
}

leaf source-addr {
  type rt-types:ipv6-multicast-source-address;
  description
    "Multicast source IPV6 address.";
}

leaf-list bridge-outgoing-interface {
  when "../..scenario = 'bridge'";
  type if:interface-ref;
  description "Outgoing interface in BRIDGE forwarding";
}

leaf-list l2vpn-outgoing-ac {
  when "../..scenario = 'l2vpn'";
  type l2vpn-instance-ac-ref;
  description "Outgoing AC in L2VPN forwarding";
}

leaf-list l2vpn-outgoing-pw {
  when "../..scenario = 'l2vpn'";
  type l2vpn-instance-pw-ref;
  description "Outgoing PW in L2VPN forwarding";
}

```

```

    }
    } // static-l2-multicast-group
} // instance-config-attributes-mld-snooping

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

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

    leaf expire {
        type uint32;
        units seconds;
        mandatory true;
        description
            "The time left before multicast group timeout.";
    }

    leaf up-time {
        type uint32;
        units seconds;
        mandatory true;
    }
}

```

```

        description
            "The time elapsed since L2 multicast record created.";
    }
} // instance-state-group-attributes-igmp-mld-snooping

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

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

    list group {

        key "address";

        config false;

        description "IGMP snooping information";
    }
}

```

```

leaf address {
  type rt-types:ipv4-multicast-group-address;
  description
    "Multicast group IPV4 address";
}

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

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

list source {
  key "address";
  description "Source IPV4 address for multicast stream";

  leaf address {
    type rt-types:ipv4-multicast-source-address;
    description "Source IPV4 address for multicast stream";
  }

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

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

```

```

}

```

```

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

  leaf host-address {
    type inet:ipv4-address;
    description

```

```

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

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

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

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

    leaf entries-count {
        type uint32;
        config false;
        description
            "The number of L2 multicast entries in IGMP & MLD snooping";
    }

    leaf-list bridge-mrouter-interface {
        when "../scenario = 'bridge'";
        type if:interface-ref;
        config false;
        description "mrouter interface in BRIDGE forwarding";
    }

    leaf-list l2vpn-mrouter-interface-ac {
        when "../scenario = 'l2vpn'";
        type if:interface-ref;
        config false;
        description

```

```

        "mrouter interface whose type is interface
         in L2VPN forwarding";

    }

```



```

leaf-list l2vpn-mrouter-interface-pw {
    when "../scenario = 'l2vpn'";
    type l2vpn-instance-pw-ref;
    config false;
    description
        "mrouter interface whose type is PW in L2VPN forwarding";
}
} // instance-config-attributes-igmp-mld-snooping

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

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

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

        leaf address {
            type rt-types:ipv6-multicast-group-address;
            description
                "Multicast group IPV6 address";
        }

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

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

        list source {
            key "address";
            description "Source IPV6 address for multicast stream";

            leaf address {
                type rt-types:ipv6-multicast-source-address;
                description "Source IPV6 address for multicast stream";
            }

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

```

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

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

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

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

  leaf-list bridge-outgoing-interface {
    when "../../scenario = 'bridge'";
    type if:interface-ref;
    description "Outgoing interface in BRIDGE forwarding";
  }

  leaf-list l2vpn-outgoing-ac {
    when "../../scenario = 'l2vpn'";
    type l2vpn-instance-ac-ref;
    description "Outgoing AC in L2VPN forwarding";
  }

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

```
when "../.../scenario = 'l2vpn'";
type l2vpn-instance-pw-ref;
description "Outgoing PW in L2VPN forwarding";
```

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

```
    }

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

    leaf expire {
        type uint32;
        units seconds;
        mandatory true;
        description
            "The time left before multicast group timeout.";
    }

    leaf host-count {
        if-feature explicit-tracking;
        type uint32;
        description
            "The number of host addresses.";
    }
} // instance-state-source-attributes-igmp-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.";
        reference
            "RFC 2236, Section 2.1";
    }

    leaf num-membership-report-v1 {
        type yang:counter64;
        description
            "The number of membership report v1 messages.";
```

```

        reference
            "RFC 3376, Section 4";
    }
    leaf num-membership-report-v2 {
        type yang:counter64;
        description
            "The number of membership report v2 messages.";
        reference
            "RFC 3376, Section 4";
    }

```

```

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

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

```

```

leaf num-query {
    type yang:counter64;
    description
        "The number of Multicast Listener Query messages.";
    reference
        "RFC 3810, Section 5";
}
leaf num-report-v1 {
    type yang:counter64;
    description
        "The number of Version 1 Multicast Listener Report.";
    reference
        "RFC 3810, Section 5";
}
leaf num-report-v2 {
    type yang:counter64;
    description

```

Zhao & Liu, etc

Expires June 09, 2019

[Page 25]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

```

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

grouping igmp-snooping-interface-statistics-attributes {

    description "Interface statistics attributes for IGMP snooping";

    container interfaces {
        config false;

```



```

description
  "Interfaces associated with the MLD snooping instance";

list interface {
  key "name";

  description
    "Interfaces associated with the MLD snooping instance";

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

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

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

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

      uses mld-snooping-statistics;
    }
  }
}

```

```

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

```

```

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

```

```

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

```

```

/*
 * igmp-snooping-instance
 */

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

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

  uses instance-config-attributes-igmp-snooping;

  uses instance-state-attributes-igmp-snooping;

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

/*
 * mld-snooping-instance
 */

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

```

```

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

leaf scenario {
  type snooping-scenario-type;

```



```

        default bridge;
        description
            "The scenario indicates BRIDGE or L2VPN.";
    }

    uses instance-config-attributes-mld-snooping;

    uses instance-state-attributes-mld-snooping;

    uses mld-snooping-interface-statistics-attributes;

} //mld-snooping-instance
}

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

```

```

/*  RPCs  */

```

```

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

    input {

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

        leaf group {
            type rt-types:ipv4-multicast-group-address;

```

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

```
        "Multicast group IPv4 address. If it is not specified,
        all IGMP snooping group tables are cleared.";
    }

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

} // rpc clear-igmp-snooping-groups

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

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

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

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

<CODE ENDS>

4. Security Considerations

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

Zhao & Liu, etc

Expires June 09, 2019

[Page 31]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

is Secure Shell (SSH) [[RFC6242](#)]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [[RFC5246](#)].

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

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

```
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:igmp-snoop
instance
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:mld-snoopi
instance
```

The subtrees under /dot1q:bridges/dot1q:bridge

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

The subtrees under

```
/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan
```

```
/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan/ims:ig
snooping-instance
/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vlan/ims:ml
snooping-instance
```

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

`/ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn/ims:igmp-snoop
instance`

`/ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn/ims:mld-snoopi
instance`

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

Zhao & Liu, etc

Expires June 09, 2019

[Page 32]

Internet-Draft

IGMP & MLD Snooping Yang Module

December 10, 2018

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

`/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:igmp-snoop
instance`

`/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ims:mld-snoopi
instance`

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

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

[5. IANA Considerations](#)

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

This document registers the following namespace URIs in the IETF XML registry [[RFC3688](#)]:

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

Registrant Contact: The IESG.

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

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

Zhao & Liu, etc Expires June 09, 2019 [Page 33]

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

name: ietf-igmp-mld-snooping

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

prefix: ims

reference: RFC XXXX

[6](#). Normative References

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

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

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast

Listener Discovery (MLD) for IPv6", [RFC 2710](#), October 1999.

- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", [RFC 3376](#), October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", [RFC 3810](#), June 2004.
- [RFC4541] M. Christensen, K. Kimball, F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", [RFC 4541](#), May 2006.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", [RFC 4604](#), August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", [RFC 4607](#), August 2006.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6021](#), October 2010.

Zhao & Liu, etc

Expires June 09, 2019

[Page 34]

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), July 2013.
- [RFC8342] M. Bjorklund and J. Schoenwaelder, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), March 2018.
- [RFC8343] M. Bjorklund, "A YANG Data Model for Interface Management", [RFC 8343](#), March 2018.
- [[draft-ietf-pim-igmp-mld-yang-06](#)] X. Liu, F. Guo, M. Sivakumar, P. McAllister, A. Peter, "A YANG data model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", [draft-ietf-pim-igmp-mld-yang-06](#), Oct 20, 2017.
- [[draft-bjorklund-netmod-rfc7223bis-00](#)] M. Bjorklund, "A YANG Data

Model for Interface Management", [draft-bjorklund-netmod-rfc7223bis-00](#), August 21, 2017

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

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

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

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

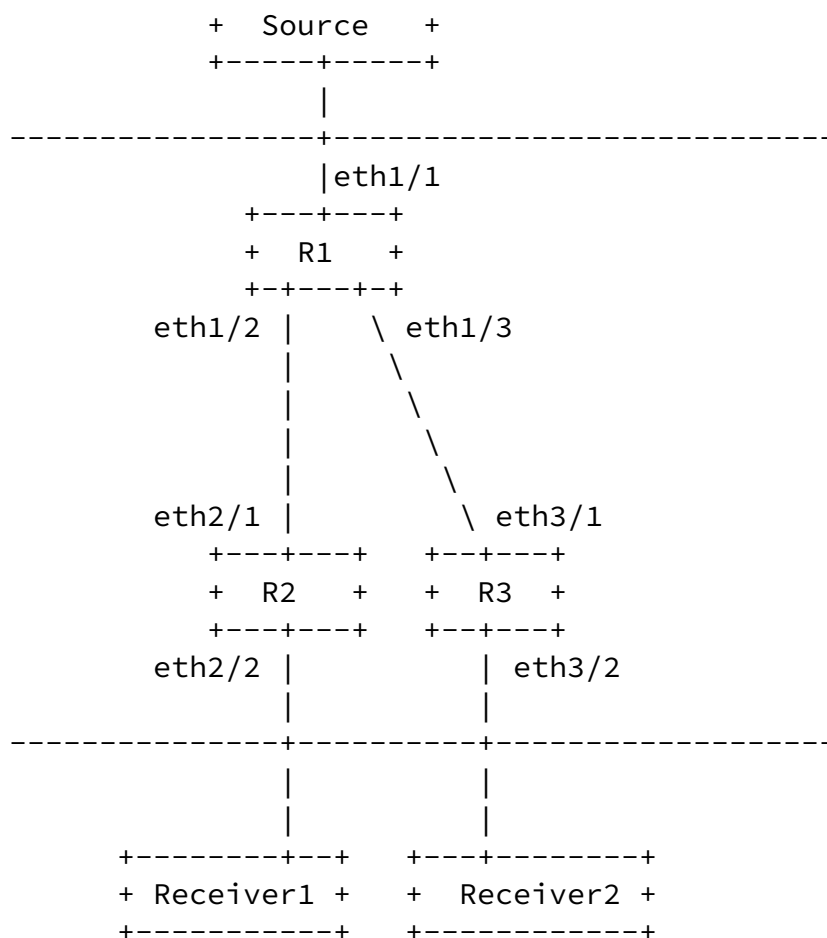
[[draft-ietf-rtgwg-ni-model-12](#)] L. Berger, C. Hopps, A. Lindem, X. Liu, "YANG Model for Network Instances", [draft-ietf-rtgwg-ni-model-12.txt](#), March 19, 2018

[Appendix A](#). Data Tree Example

[A.1](#) Bridge scenario

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

+-----+



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":{

```


Internet-Draft

IGMP & MLD Snooping Yang Module

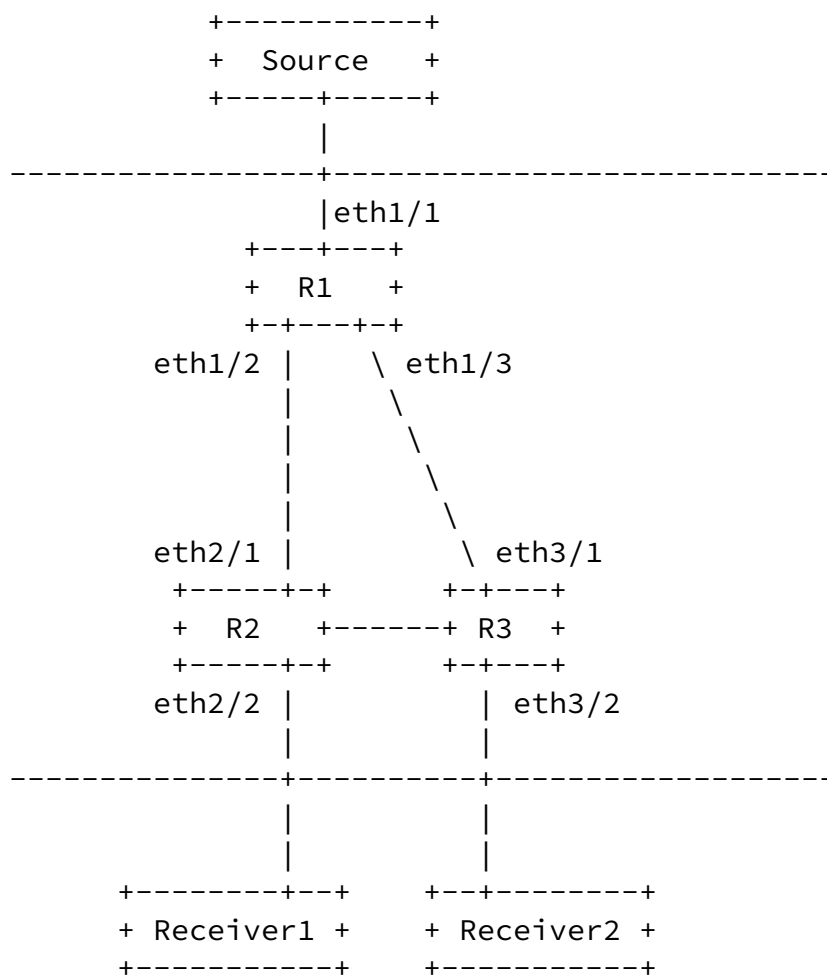
December 10, 2018

```
{
  "type": "ietf-igmp-ml-d-snooping:igmp-snooping",
  "name": "bis1",
  "ietf-igmp-ml-d-snooping:igmp-snooping-instance": {
    "scenario": "ietf-igmp-ml-d-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-ml-d-snooping:igmp-snooping-instance": "bis1"
              }
            ]
          }
        }
      ]
    }
  ]
}
]
```

Internet-Draft IGMP & MLD Snooping Yang Module December 10, 2018

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"
    }
  ]
},
"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": "ac1",
          "primary-ac": {
            "name": "eth1/1"
          }
        },
        {
          "name": "pw2",
          "primary-pw": [
            {
              "name": "pw2"
            }
          ]
        },
        {
          "name": "pw3",
          "primary-pw": [
            {

```

```

        "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": "ac1",
          "primary-ac": {
            "name": "eth1/1"
          }
        },
        {
          "name": "pw2",
          "primary-pw": [
            {
              "name": "pw2"
            }
          ]
        }
      ],
    },
    {

```

```

      "name": "pw3",
      "primary-pw": [
        {
          "name": "pw3"
        }
      ]
    }
  ],
},
"ietf-routing:routing": {
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-igmp-mld-snooping:igmp-snooping",

```

```
    "name": "vis1",
    "ietf-igmp-mld-snooping:igmp-snooping-instance": {
      "scenario": "ietf-igmp-mld-snooping:l2vpn",
      "enable": true
    }
  }
]
}
}
```

Authors' Addresses

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

Email: hongji.zhao@ericsson.com

Xufeng Liu
Volta Networks
USA

EMail: xufeng.liu.ietf@gmail.com

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

Email: liuyisong@huawei.com

Anish Peter
Individual

EMail: anish.ietf@gmail.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale, California
USA

EMail: sivakumar.mahesh@gmail.com