

# IGMP & MLD Snooping

## YANG Model

draft-zhao-pim-igmp-mld-snooping-yang-03

Hongji Zhao (Ericsson)

Xufeng Liu (Jabil)

Yisong Liu (Huawei)

Mahesh Sivakumar (Cisco)

Anish Peter (Individual)

IETF100

# Status

- version 03
  - Second version for presentation
  - A new agreement about the whole structure is reached. The igmp-snooping-instance is defined to configure the parameters and show the operational state. It also follows the NMDA standard.
  - Effort from multicast yang design team  
( Ericsson/Jabil/Huawei/Cisco )

# Structure

- Define a fully functional schema for IGMP Snooping instance

```
module: ietf-igmp-mld-snooping
  +-rw igmp-snooping-instances
    |  +-rw igmp-snooping-instance* [name]
    |    +-rw name                                string
    |    +-rw id?                                 uint32
    |    +-rw type?                               enumeration
    |    +-rw enable?                            boolean {admin-enable}?
    |    +-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 version?                   uint8
    |    +-rw static-bridge-mrouter-interface* if:interface-ref {static-l2-multicast-group}?
    |    +-rw static-vpls-mrouter-interface* l2vpn-instance-pw-ref {static-l2-multicast-group}?
    |    +-rw querier-source?           inet:ipv4-address
    |    +-rw static-l2-multicast-group* [group source-addr] {static-l2-multicast-group}?
    |      +-rw group                  inet:ipv4-address
    |      +-rw source-addr          source-ipv4-addr-type
    |      +-rw bridge-outgoing-interface* if:interface-ref
    |      +-rw vpls-outgoing-ac*     l2vpn-instance-ac-ref
    |      +-rw vpls-outgoing-pw*     l2vpn-instance-pw-ref
    ...
  ...
```

The igmp-snooping-instance could fit for both bridge and VPLS scenario. It is determined by the value of “type”.

The multicast router interfaces and I2 multicast routing group could be configured manually.

# Structure

- The read-only attributes are the operational state data. There are 3 kinds of outgoing interface which is interface-ref, l2vpn-instance-ac-ref, or l2vpn-instance-pw-ref. We have utilized the existing ietf-interfaces and ietf-l2vpn module to indicate the outgoing interface.

```
module: ietf-igmp-mld-snooping
  +-rw igmp-snooping-instances
    |  +-rw igmp-snooping-instance* [name]
    |
    |  ...
    |  +-ro bridge-mrouter-interface*      if:interface-ref
    |  +-ro vpls-mrouter-interface*       l2vpn-instance-pw-ref
    |  +-ro group* [address]
    |    +-ro address          inet:ipv4-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          inet:ipv4-address
    |      +-ro bridge-outgoing-interface* if:interface-ref
    |      +-ro vpls-outgoing-ac*     l2vpn-instance-ac-ref
    |      +-ro vpls-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 enumeration
```

- For example, l2vpn-instance-pw-ref is the leafref for  
`/l2vpn:l2vpn/l2vpn:instances/l2vpn:instance/l2vpn:endpoint/l2vpn:pw/l2vpn:name`

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

# Bridge scenario

- The igmp-snooping-instance could be referenced in bridge scenario.

```
+--rw bridges
|   +-rw bridge* [name]
|     +-rw name          name-type
|     +-rw igmp-snooping-instance?  igmp-snooping-instance-ref
|     +-rw component* [name]
|       +-rw name        string
|       +-rw bridge-vlan
|         +-rw vlan* [vid]
|           +-rw vid          vlan-index-type
|           +-rw igmp-snooping-instance?  igmp-snooping-instance-ref
|             +-rw interfaces
|               +-rw interface* [name]
|                 +-rw name        string
|                 +-rw igmp-snooping-instance?  igmp-snooping-instance-ref
```

module: ietf-igmp-mld-snooping

```
+--rw igmp-snooping-instance* [name]
|   +-rw name
|   +-rw id?
|   +-rw type?
|   +-rw enable?
```

The diagram illustrates the reference path from a bridge's component to an igmp-snooping-instance. It shows a curved arrow starting from the 'component\*' node under 'bridge' and pointing to the 'igmp-snooping-instance?' node under 'component\*'. The types along this path are: name->name-type, name->igmp-snooping-instance-ref, component\*>string, component\*>igmp-snooping-instance-ref, and interface\*>string. A second curved arrow starts from the 'interface\*' node under 'interfaces' and points to the 'igmp-snooping-instance?' node under 'component\*'. The types along this path are: vid->vlan-index-type, interfaces->igmp-snooping-instance-ref, and interface\*>string. A third curved arrow starts from the 'enable?' node under 'igmp-snooping-instance\*' and points to the 'igmp-snooping-instance?' node under 'component\*'. The types along this path are: enable?->boolean {admin-enable}?, enable?->igmp-snooping-instance-ref, and enable?->igmp-snooping-instance-ref. The final types shown are string, uint32, enumeration, and boolean {admin-enable}?.  
string  
uint32  
enumeration  
boolean {admin-enable}?

# I2vpn scenario

- The igmp-snooping-instance could be referenced in I2vpn scenario.

```
+--rw l2vpn-instances
  +-rw l2vpn-instance* [name]
    +-rw name                      string
    +-rw igmp-snooping-instance?   igmp-snooping-instance-ref
    +-rw endpoint* [name]
      +-rw name                      string
      +-rw igmp-snooping-instance?   igmp-snooping-instance-ref
      +-rw (ac-or-pw-or-redundancy-grp)?
        +-:(ac)
          | +-rw ac* [name]
          |   +-rw name
          |   +-rw igmp-snooping-instance?
        +-:(pw)
          | +-rw pw* [name]
          |   +-rw name
          |   +-rw igmp-snooping-instance?
...
module: ietf-igmp-mld-snooping
+-rw igmp-snooping-instance* [name]
  |   +-rw name
  |   +-rw id?
  |   +-rw type?
  |   +-rw enable?
```

The diagram illustrates a reference loop between two `igmp-snooping-instance-ref` fields. It consists of two curved arrows pointing from one `igmp-snooping-instance-ref` field to another. One arrow originates from the `igmp-snooping-instance-ref` field under the `endpoint*` section and points to the `igmp-snooping-instance-ref` field under the `(ac-or-pw-or-redundancy-grp)?` section. Another arrow originates from the `igmp-snooping-instance-ref` field under the `(ac-or-pw-or-redundancy-grp)?` section and points back to the first `igmp-snooping-instance-ref` field. Both arrows are highlighted with a light blue color.

# IGMP snooping RPC

Clears the specified IGMP Snooping cache tables.

```
module: ietf-igmp-mld-snooping
```

```
rpcs:
```

```
+---x clear-igmp-snooping-groups {rpc-clear-groups}?
| +---w input
|   +---w id?      uint32
|   +---w group?    inet:ipv4-address
|   +---w source?   inet:ipv4-address
```

# Unsolved problem

- The attribute forwarding-mode maybe reconsidered because it only fits for instance, but doesn't fit for interface.
- Counter to non-member leave would also be useful ?

# Next Step

- Apply for WG adoption
- Welcome more vendors and carriers involved
- Need more comments