

IGMP & MLD Snooping

YANG Model

draft-zhao-pim-igmp-mld-snooping-yang-01

Hongji Zhao (Ericsson)

Xufeng Liu (Jabil)

Yisong Liu (Huawei)

Mahesh Sivakumar (Cisco)

Anish Peter (Individual)

IETF99

Status

- version 01
 - First version for presentation
 - Structure according to IGMP & MLD model. (We have reached a new agreement about the whole structure just now. The draft needs update.)
 - Effort from multicast yang design team
 - (Ericsson/Jabil/Huawei/Cisco)

Structure

- Define a fully functional schema for IGMP Snooping profile

```
module: ietf-igmp-mld-snooping
  +-rw igmp-snooping-profile* [name]
    +-rw name                      string
    +-rw vlan-id?                  uint16
    +-rw enable?                   boolean {admin-enable}?
    +-rw forwarding-mode?          enumeration
    +-rw explicit-tracking?       empty
    +-rw send-query?              Boolean
    +-rw fast-leave?              empty {fast-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-mrouter-interface* if:interface-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 interface?             if:interface-ref
      +-ro entries-count?         uint32 // operational state data
      +-ro mrouter-interface*     if:interface-ref // operational state data
      +-ro group* [address] // operational state data
        +-ro address               inet:ipv4-address
        +-ro mac-address?          yang:phys-address
        +-ro expire?               uint32
        +-ro up-time?              uint32
        +-ro interface?            if:interface-ref
        +-ro last-reporter?        inet:ipv4-address
        +-ro source* [address]
          +-ro address              inet:ipv4-address
          +-ro up-time?             uint32
          +-ro expire?              uint32
          +-ro last-reporter?       inet:ipv4-address
```

mld-snooping-profile is same as IGMP snooping except changing IPv4 addresses to IPv6 addresses.

Structure

- Prepared to work with other models by profile references

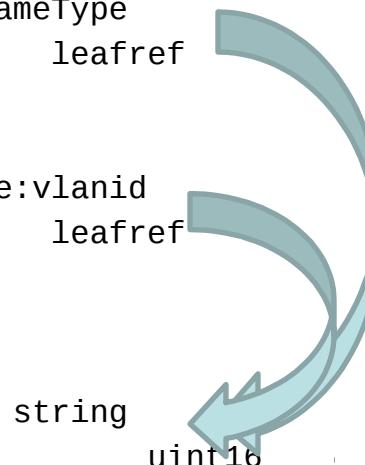
```
ieee 802.1Q bridge model:
```

```
module: ieee-dot1Q-bridge
```

```
  +-rw bridge* [name]
    +-rw name          dot1Qtypes:nameType
    +-rw igmp-snooping-profile      leafref
    +-rw bridge-vlan
      +-rw vlan* [vid]
        | +-rw vid           ieee:vlanid
        +-rw igmp-snooping-profile  leafref
```

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

```
  +-rw igmp-snooping-profile* [name]
    +-rw name          string
    +-rw vlan-id?      uint16
    +-rw enable?       boolean {admin-enable}?
    +-rw forwarding-mode? enumeration
    +-rw explicit-tracking? string
    +-rw send-query?   boolean
    +-rw mrouter-aging-time? uint16
    +-rw fast-leave?   empty {fast-leave}?
  ...
```



- Pros
 - Can be fully, independently defined
 - Can be augmented
 - Easy references to and from other models
 - Can be adjusted to bridge and VLAN cases

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 vlan-id?    uint16
            +---w group?      inet:ipv4-address
            +---w source?     inet:ipv4-address
```

Unsolved problem

- How to extend igmp-snooping-profile to support VPLS?
- How to define operational state data? (Now I have defined some read-only attributes in the igmp-snooping-profile as operational state data . But we haven't reached an agreement yet.)

Next Step

- Welcome more vendors and carriers involved
- Need more comments