Draft YANG Model for IGMP/MLD

draft-guo-pim-igmp-mld-yang-01

Xufeng Liu (Ericsson)
Feng Guo (Huawei Technologies)
Mahesh Sivakumar (Cisco Systems)
Pete McAllister (Metaswitch Networks)
Anish Peter (Juniper Networks)

IETF95

Agenda

- Introduction
- Summary of progress
- Top level structure and critical model design
- Next steps

Introduction

- This draft is still under discussion by the YANG multicast design team
- Archive: http://www.ietf.org/mail-archive/web/yang-multicast/current/maillist.html
- Wiki: http://trac.tools.ietf.org/wg/pim/trac/wiki/yang
- This draft is versioned on github: https://github.com/mcallisterjp/pim-yang/

Summary of Progress

- Scope covers IGMP/MLD protocols and protocol extensions
- High-level structure done
- Configuration attributes done
- Operational state attributes done
- Statistics attributes done
- Still to do:
  - Notifications
  - Further review

Module Position and AF

1. IGMP and MLD module augment options:

*Option A: /rt:routing/rt:routing-instance**/rt:routing-protocols:***

*Option B: /rt:routing/rt:routing-instance**/rt:routing-protocols:/rt:routing-protocol***

Option A chosen.

Option B would allow multiple protocol instances per VRF, which does not make sense for IGMP and MLD.

2. IGMP and MLD belong to the same module ‘ietf-igmp-mld’

- Easier to make it consistent for igmp and mld model

Module Position and AF

3.IGMP and MLD share the same structure, but defined as separate schema branches in the structure, so that:

- implementations may optionally choose to support specific address families
- the names of objects may be different between ipv4 (igmp) and ipv6 (mld) address families.

```
 +--rw igmp
     |   +--rw ...
     +--rw mld
         +--rw ...
```

4.Same features type defined for both IGMP and MLD to make consistent structure between them. implementations have to support them at the same time

```
module: ietf-igmp-mld
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
     +--rw igmp
     |   +--rw global
     |   |   +--rw enable?          boolean {global-admin-enable}?
     |   |   +--rw max-entries?    uint32 {global-max-entries}?
     |   |   +--rw max-groups?     uint32 {global-max-groups}?

     +--rw mld
         +--rw global
         |   +--rw enable?          boolean {global-admin-enable}?
         |   +--rw max-entries?    uint32 {global-max-entries}?
         |   +--rw max-groups?     uint32 {global-max-groups}?
```

Config Structure

Three levels:

- **Global level:** IGMP MLD configuration attributes for the entire routing instance
- **Interface-global:** IGMP MLD configuration attributes applied to interfaces whose interface level attributes are not existing, with same attributes' value for those
- **Interface-level:** IGMP MLD configuration attributes specific to the given interface

•Eg.1: max-groups 100 in **Global level** means 100 groups at most for the entire IGMP instance;Max-groups 100 in **Interface level** means the limitation for each interface is 100, and it may be greater than 100 for the entire IGMP isntance

•Eg.2: Max-groups-per-interface 100 in Interface-global level defined the group number limitation for interfaces, which donot config max-groups themselves.

```
module: ietf-igmp-mld
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw igmp
    |  +--rw global
    |  |  +--rw enable?       boolean {global-admin-enable}?
    |  |  +--rw max-entries?  uint32 {global-max-entries}?
    |  |  +--rw max-groups?   uint32 {global-max-groups}?
    |  +--rw interfaces
    |     +--rw max-groups-per-interface?    uint32 {intf-max-groups}?
    |     +--rw ...
    |     +--rw interface* [interface]
    |        +--rw interface                        if:interface-ref
    |        +--rw enable?                          boolean {intf-admin-enable}?
    |        +--rw max-groups?                      uint32 {intf-max-groups}?
```

Operational State Structure

Three levels:

- **Global level:** IGMP MLD operational state attributes for the entire routing instance
- **Interface-global:** IGMP MLD interface level operational state attributes applied to interfaces whose interface level attributes do not exist, with same attributes' value for those interfaces
- **Interface-specific:** IGMP MLD operational state attributes specific to the given interface.

```
module: ietf-igmp-mld
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
   +--ro igmp
   |  +--ro global
   |  |  +--ro enable?          boolean {global-admin-enable}?
   |  |  +--ro max-entries?     uint32 {global-max-entries}?
   |  |  +--ro max-groups?      uint32 {global-max-groups}?
   |  |  +--ro ...
   |  +--ro interfaces
   |     +--ro last-member-query-interval?   uint16
   |     +--ro max-groups-per-interface?     uint32 {intf-max-groups}?
   |     +--ro ...
   |     +--ro interface* [interface]
   |        +--ro interface                     if:interface-ref
   |        +--ro enable?                       boolean {intf-admin-enable}?
   |        +--ro max-groups?                   uint32 {intf-max-groups}?
```

SSM Mapping Config

Basically two types of definitions: source-addr specified inside policy, or outside policy

Juniper:

```
policy-statement POLICY-ipv4-example1 {
term 1 {
        from {
        route-filter 232.1.1.1/32 exact;
        }
        then {
        ssm-source [ 10.10.10.4 192.168.43.66 ];
        accept;
        }
}
}
```

Cisco:

```
!
access-list 10 permit 232.1.2.10
access-list 11 permit 232.1.2.0 0.0.0.255

ip igmp ssm-map enable
ip igmp ssm-map static 10 172.16.8.10
ip igmp ssm-map static 11 172.16.8.11
```

SSM Mapping Config

Using union type to indicate whether source-addr is inside the group-policy

```
module: ietf-igmp-mld
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw igmp
    |  +--rw interfaces
    |     +--rw interface* [interface]
    |        +--rw interface                           if:interface-ref
    |        +--rw enable?                             boolean {intf-admin-enable}?
    |        +--rw ssm-map* [source-addr group-policy] {intf-ssm-map}?
    |        |  +--rw source-addr     ssm-map-ipv4-addr-type
    |        |  +--rw group-policy    string

typedef ssm-map-ipv4-addr-type {
     type union {
       type enumeration {
         enum 'policy' {
           description
           "Source address is specified in SSM map policy.";
         }
       }
       type inet:ipv4-address;
     }
     description
       "Multicast source IP address type for SSM map.";
   } // source-ipv4-addr-type
```

RPC

IGMP and MLD RPC

```
    rpcs:
    +---x clear-igmp-groups {rpc-clear-groups}?
    |  +---w input
    |     +---w routing-instance?   rt:routing-instance-ref
    |     +---w interface?          leafref
    |     +---w group?              inet:ipv4-address
    +---x clear-mld-groups {rpc-clear-groups}?
       +---w input
          +---w routing-instance?   rt:routing-instance-ref
          +---w interface?          leafref
          +---w group?              inet:ipv4-address
```

Next Steps

- Notifications

- Request further review

- Can this draft be adopted as WG draft?

Appendix: Yang Models

Config Global-level

```
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw igmp
    |  +--rw global
    |  |  +--rw enable?        boolean {global-admin-enable}?
    |  |  +--rw max-entries?   uint32 {global-max-entries}?
    |  |  +--rw max-groups?    uint32 {global-max-groups}?
    +--rw mld
       +--rw global
       |  +--rw enable?        boolean {global-admin-enable}?
       |  +--rw max-entries?   uint32 {global-max-entries}?
       |  +--rw max-groups?    uint32 {global-max-groups}?
```

Config Interface-Global

```
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw igmp
    |  +--rw global
    |  +--rw interfaces
    |     +--rw last-member-query-interval?   uint16
    |     +--rw max-groups-per-interface?     uint32 {intf-max-groups}?
    |     +--rw query-interval?               uint16
    |     +--rw query-max-response-time?      uint16
    |     +--rw require-router-alert?         boolean {intf-require-router-alert}?
    |     +--rw robustness-variable?          uint8
    |     +--rw version?                      Uint8
    |     +--rw interface* [interface]
    +--rw mld
       +--rw global
       +--rw interfaces
          +--rw last-member-query-interval?   uint16
          +--rw max-groups-per-interface?     uint32 {intf-max-groups}?
          +--rw query-interval?               uint16
          +--rw query-max-response-time?      uint16
          +--rw require-router-alert?         boolean {intf-require-router-alert}?
          +--rw robustness-variable?          uint8
          +--rw version?                      uint8
          +--rw interface* [interface]
```

Config Interface-Specific(IGMP)

```
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw igmp
    |  +--rw global
    |  +--rw interfaces
    |     +--rw interface* [interface]
    |        +--rw interface                     if:interface-ref
    |        +--rw enable?                       boolean {intf-admin-enable}?
    |        +--rw group-policy?                 string
    |        +--rw immediate-leave?              empty {intf-immediate-leave}?
    |        +--rw last-member-query-interval?   uint16
    |        +--rw max-groups?                   uint32 {intf-max-groups}?
    |        +--rw max-group-sources?            uint32 {intf-max-group-sources}?
    |        +--rw query-interval?               uint16
    |        +--rw query-max-response-time?      uint16
    |        +--rw require-router-alert?    s     boolean {intf-require-router-alert}?
    |        +--rw robustness-variable?          uint8
    |        +--rw source-policy?                string {intf-source-policy}?
    |        +--rw verify-source-subnet?         empty {intf-verify-source-subnet}?
    |        +--rw version?                      uint8
    |        +--rw join-group*                   inet:ipv4-address {intf-join-group}?
    |        +--rw ssm-map* [source-addr group-policy] {intf-ssm-map}?
    |        |  +--rw source-addr     ssm-map-ipv4-addr-type
    |        |  +--rw group-policy    string
    |        +--rw static-group* [group source-addr] {intf-static-group}?
    |           +--rw group          inet:ipv4-address
    |           +--rw source-addr    source-ipv4-addr-type
    +--rw mld
       +--rw global
```

Config Interface-Specific(MLD)

```
augment /rt:routing/rt:routing-instance/rt:routing-protocols:
    +--rw mld
       +--rw global
       +--rw interfaces
          +--rw interface* [interface]
             +--rw interface                    if:interface-ref
             +--rw enable?                      boolean {intf-admin-enable}?
             +--rw group-policy?                string
             +--rw immediate-leave?             empty {intf-immediate-leave}?
             +--rw last-member-query-interval?   uint16
             +--rw max-groups?                  uint32 {intf-max-groups}?
             +--rw max-group-sources?           uint32 {intf-max-group-sources}?
             +--rw query-interval?              uint16
             +--rw query-max-response-time?     uint16
             +--rw require-router-alert?        boolean {intf-require-router-alert}?
             +--rw robustness-variable?         uint8
             +--rw source-policy?               string {intf-source-policy}?
             +--rw verify-source-subnet?        empty {intf-verify-source-subnet}?
             +--rw version?                     uint8
             +--rw join-group*                  inet:ipv6-address {intf-join-group}?
             +--rw ssm-map* [source-addr group-policy] {intf-ssm-map}?
             |  +--rw source-addr     ssm-map-ipv6-addr-type
             |  +--rw group-policy    string
             +--rw static-group* [group source-addr] {intf-static-group}?
                +--rw group          inet:ipv6-address
                +--rw source-addr    source-ipv6-addr-type
```

Operation Global-level

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
    +--ro igmp
    |  +--ro global
    |  |  +--ro enable?          boolean {global-admin-enable}?
    |  |  +--ro max-entries?     uint32 {global-max-entries}?
    |  |  +--ro max-groups?      uint32 {global-max-groups}?
    |  |  +--ro entries-count?   uint32
    |  |  +--ro groups-count?    Uint32
    +--ro mld
       +--ro global
       |  +--ro enable?          boolean {global-admin-enable}?
       |  +--ro max-entries?     uint32 {global-max-entries}?
       |  +--ro max-groups?      uint32 {global-max-groups}?
       |  +--ro entries-count?   uint32
       |  +--ro groups-count?    uint32
```

Operation Global-level(IGMP)

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
    +--ro igmp
    |  +--ro global
    |  |  +--ro statistics
    |  |     +--ro discontinuity-time?   yang:date-and-time
    |  |     +--ro error
    |  |     |  +--ro total?       yang:counter64
    |  |     |  +--ro query?       yang:counter64
    |  |     |  +--ro report?      yang:counter64
    |  |     |  +--ro leave?       yang:counter64
    |  |     |  +--ro checksum?    yang:counter64
    |  |     |  +--ro too-short?   yang:counter64
    |  |     +--ro received
    |  |     |  +--ro total?    yang:counter64
    |  |     |  +--ro query?    yang:counter64
    |  |     |  +--ro report?   yang:counter64
    |  |     |  +--ro leave?    yang:counter64
    |  |     +--ro sent
    |  |        +--ro total?    yang:counter64
    |  |        +--ro query?    yang:counter64
    |  |        +--ro report?   yang:counter64
    |  |        +--ro leave?    yang:counter64
```

Operation Global-level(MLD)

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
     +--ro mld
       +--ro global
       |  +--ro statistics
       |     +--ro discontinuity-time?   yang:date-and-time
       |     +--ro error
       |     |  +--ro total?       yang:counter64
       |     |  +--ro query?       yang:counter64
       |     |  +--ro report?      yang:counter64
       |     |  +--ro leave?       yang:counter64
       |     |  +--ro checksum?    yang:counter64
       |     |  +--ro too-short?   yang:counter64
       |     +--ro received
       |     |  +--ro total?    yang:counter64
       |     |  +--ro query?    yang:counter64
       |     |  +--ro report?   yang:counter64
       |     |  +--ro leave?    yang:counter64
       |     +--ro sent
       |        +--ro total?    yang:counter64
       |        +--ro query?    yang:counter64
       |        +--ro report?   yang:counter64
       |        +--ro leave?    yang:counter64
```

Operation Interface-Global

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
    +--ro igmp
    |  +--ro global
    |  +--ro interfaces
    |     +--ro last-member-query-interval?   uint16
    |     +--ro max-groups-per-interface?     uint32 {intf-max-groups}?
    |     +--ro query-interval?               uint16
    |     +--ro query-max-response-time?      uint16
    |     +--ro require-router-alert?         boolean {intf-require-router-alert}?
    |     +--ro robustness-variable?          uint8
    |     +--ro version?                      uint8
    |     +--ro interface* [interface]
    +--ro mld
       +--ro global
       +--ro interfaces
          +--ro last-member-query-interval?   uint16
          +--ro max-groups-per-interface?     uint32 {intf-max-groups}?
          +--ro query-interval?               uint16
          +--ro query-max-response-time?      uint16
          +--ro require-router-alert?         boolean {intf-require-router-alert}?
          +--ro robustness-variable?          uint8
          +--ro version?                      uint8
          +--ro interface* [interface]
```

Operation Interface-Specific(IGMP)

```
 augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
    +--ro igmp
    |  +--ro global
    |  +--ro interfaces
    |     +--ro interface* [interface]
    |        +--ro interface                     if:interface-ref
    |        +--ro enable?                       boolean {intf-admin-enable}?
    |        +--ro group-policy?                 string
    |        +--ro immediate-leave?              empty {intf-immediate-leave}?
    |        +--ro last-member-query-interval?   uint16
    |        +--ro max-groups?                   uint32 {intf-max-groups}?
    |        +--ro max-group-sources?            uint32 {intf-max-group-sources}?
    |        +--ro query-interval?               uint16
    |        +--ro query-max-response-time?      uint16
    |        +--ro require-router-alert?         boolean {intf-require-router-
alert}?
    |        +--ro robustness-variable?          uint8
    |        +--ro source-policy?                string {intf-source-policy}?
    |        +--ro verify-source-subnet?         empty {intf-verify-source-subnet}?
    |        +--ro version?                      uint8
    |        +--ro join-group*                   inet:ipv4-address {intf-join-
group}?
    |        +--ro ssm-map* [source-addr group-policy] {intf-ssm-map}?
    |        |  +--ro source-addr     ssm-map-ipv4-addr-type
    |        |  +--ro group-policy    string
```

Operation Interface-Specific(IGMP)

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
   +--ro igmp
   |  +--ro global
   |  +--ro interfaces
   |     +--ro interface* [interface]
   |        +--ro interface                     if:interface-ref
   |        +--ro static-group* [group source-addr] {intf-static-group}?
   |        |  +--ro group          inet:ipv4-address
   |        |  +--ro source-addr    source-ipv4-addr-type
   |        +--ro oper-status?                  enumeration
   |        +--ro dr?                           inet:ipv4-address
   |        +--ro querier?                      inet:ipv4-address
   |        +--ro joined-group*                 inet:ipv4-address {intf-join-group}?
   |        +--ro group* [address]
   |           +--ro address          inet:ipv4-address
   |           +--ro expire?          uint32
   |           +--ro filter-mode?     enumeration
   |           +--ro host-count?      uint32
   |           +--ro up-time?         uint32
   |           +--ro host*            inet:ipv4-address
   |           +--ro last-reporter?   inet:ipv4-address
   |           +--ro source* [address]
   |              +--ro address          inet:ipv4-address
   |              +--ro expire?          uint32
   |              +--ro up-time?         uint32
```

Operation Interface-Specific(MLD)

```
augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
   +--ro mld
      +--ro global
      +--ro interfaces
         +--ro interface* [interface]
            +--ro interface                     if:interface-ref
            +--ro enable?                       boolean {intf-admin-enable}?
            +--ro group-policy?                 string
            +--ro immediate-leave?              empty {intf-immediate-leave}?
            +--ro last-member-query-interval?   uint16
            +--ro max-groups?                   uint32 {intf-max-groups}?
            +--ro max-group-sources?            uint32 {intf-max-group-sources}?
            +--ro query-interval?               uint16
            +--ro query-max-response-time?      uint16
            +--ro require-router-alert?         boolean {intf-require-router-
alert}?
            +--ro robustness-variable?          uint8
            +--ro source-policy?                string {intf-source-policy}?
            +--ro verify-source-subnet?         empty {intf-verify-source-subnet}?
            +--ro version?                      uint8
            +--ro join-group*                   inet:ipv6-address {intf-join-
group}?
            +--ro ssm-map* [source-addr group-policy] {intf-ssm-map}?
            |  +--ro source-addr    ssm-map-ipv6-addr-type
            |  +--ro group-policy   string
```

Operation Interface-Specific(MLD)

```
 augment /rt:routing-state/rt:routing-instance/rt:routing-protocols:
    +--ro mld
       +--ro global
       +--ro interfaces
          +--ro interface* [interface]
             +--ro static-group* [group source-addr] {intf-static-group}?
             |  +--ro group          inet:ipv6-address
             |  +--ro source-addr    source-ipv6-addr-type
             +--ro oper-status?                  enumeration
             +--ro querier?                      inet:ipv6-address
             +--ro joined-group*                 inet:ipv6-address {intf-join-
group}?
             +--ro group* [address]
                +--ro address          inet:ipv6-address
                +--ro expire?          uint32
                +--ro filter-mode?     enumeration
                +--ro host-count?      uint32
                +--ro up-time?         uint32
                +--ro host*            inet:ipv6-address
                +--ro last-reporter?   inet:ipv6-address
                +--ro source* [address]
                   +--ro address          inet:ipv6-address
                   +--ro expire?          uint32
                   +--ro up-time?         uint32
                   +--ro last-reporter?   inet:ipv6-address
```