IPMC, MVR, and IPMC-LIB Module Changes

Application Note

CONFIDENTIAL

AN1292
Rev. APPL-2024.06
2024-06-28

Table of Contents

1. Introduction

This document describes IPMC, MVR, and IPMC-LIB module changes that have happened between the 2022.06 and 2022.09 release of the WebStaX application software.

It applies to all chip families that run this software.

This document covers changes to all three modules.

2. IPMC Module Changes

2.1. General Changes

2.1.1. `show running-config` CLI command

Previously, when showing `running-config` for IGMP snooping, there were three features:
`show running-config feature ip-igmp-snooping`
`show running-config feature ip-igmp-snooping-port`
`show running-config feature ip-igmp-snooping-vlan`

Now, these are condensed into one single:
`show running-config feature igmp`

Likewise, previously, when showing `running-config` for MLD snooping, there were three features:
`show running-config feature ipv6-mld-snooping`
`show running-config feature ipv6-mld-snooping-port`
`show running-config feature ipv6-mld-snooping-vlan`

These are also condensed into one single:
`show running-config feature mld`

2.2. Changes in ICLI Status Commands and Output

There have been made significant changes to both ICLI status commands and their output.

The following sections attempt to describe these changes in details. Notice that both IGMP and MLD snooping have changed, but for simplicity, only examples of IGMP changes are shown.

2.2.1. `show ip igmp snooping [vlan <vlan_list>] [detail]`

The output before of this command was:

Old VLAN Status Layout

```
# show ip igmp snooping

IGMP Snooping is enabled to start snooping IGMP control plane.

Switch-1 IGMP Interface Status

IGMP snooping VLAN 2 interface is enabled.
Querier status is ACTIVE
RX IGMP Query:0 V1Join:0 V2Join:0 V3Join:0 V2Leave:0
TX IGMP Query:7 / (Source) Specific Query:0
Compatibility:IGMP-Auto / Querier Version:Default / Host Version:Default
```

The old status was not easy readable and combined status and statistics. The new status is in tabular form per VLAN (unless also specifying the `details` argument, as we shall see in a moment).

New VLAN Status Layout

```
# show ip igmp snooping
VLAN Operational State Querier State Active Querier  Next Query/Expiry Time
---- ----------------- ------------- --------------- ----------------------
   1 Admin disabled    -             -                                    -
   2 Active            Active        10.10.137.170                        1
```

The output has one row per VLAN and five columns which are:
`VLAN`:
The VLAN ID the remainder of the row is about.

- `Operational State`:
  Takes one of four different values:
  - `Globally disabled` if IGMP snooping is globally disabled
  - `Admin disabled` if IGMP snooping is disabled on this VLAN, only
  - `Active` if IGMP snooping is enabled on this VLAN
  - `Active (warnings)` if IGMP snooping is enabled on this VLAN, but there are operational warnings, which can be seen with `details`.

The remainder of the columns will hold a dash (`-`) if IGMP snooping is not currently active for this VLAN.

- `Querier State`:
  Shows the switch's current querier state on this VLAN. It takes one of four values:
  - `Disabled` if querier election is not enabled on the VLAN
  - `Init` if querier election is enabled and it is sending the initial queries at faster pace than normal queries
  - `Active` if querier election is enabled and the switch is the current querier
  - `Idle` if querier election is enabled and the switch is *not* the current querier.
- `Active Querier`:
  Shows the current querier on the VLAN.

  - If `Querier State` is `Init` or `Active`, this is the IP address of ourselves. This IP address is also used as Source IP in PDUs sent by the switch.
  - If `Querier State` is `Idle`, this is the IP address of the current querier in this VLAN.
- `Next Query/Expiry Time`:
  - If `Querier State` is `Init` or `Active`, this field shows the number of seconds until the next query is being transmitted by the switch.
  - If `Querier State` is `Idle`, this field shows the number of seconds from now where a remote querier will time out and this switch will take over the querier role (if enabled) unless a query is seen from that switch (or another switch with a smaller IP address) in the meanwhile.

With `detail`, the output of the old command looked like:

Old VLAN Status Layout with Details

```
# show ip igmp snooping detail

IGMP Snooping is enabled to start snooping IGMP control plane.
Multicast streams destined to unregistered IGMP groups will be blocking.

Switch-1 IGMP Interface Status

IGMP snooping VLAN 2 interface is enabled.
Querier status is ACTIVE (Administrative Control: Join Querier-Election)
Querier Up time: 957 seconds; Query Interval: 8 seconds
Querier address is not set and will use system's IP address of this interface.
Active IGMP Querier Address is 10.10.137.170
PRI:0 / RV:2 / QI:20 / QRI:100 / LMQI:10 / URI:1
RX IGMP Query:0 V1Join:0 V2Join:0 V3Join:0 V2Leave:0
TX IGMP Query:50 / (Source) Specific Query:0
IGMP RX Errors:0; Group Registration Count:0
Compatibility:IGMP-Auto / Querier Version:Default / Host Version:Default
```

The new looks like (and notice that the optional keyword is now called `details` instead of just `detail`):

New VLAN Status Layout with Details

```
# show ip igmp snooping details
VLAN:                           1
Operational state:              Admin disabled

 ----------------------------------------------------

VLAN:                           2
Operational state:              Active
Operational warnings:           None
Querier state:                  Active
Active querier:                 10.10.137.170
Querier uptime:                 12151 seconds
Next query:                     17 seconds
Other querier expiry time:      -
Configured compatibility:       Auto
Querier compatibility:          IGMPv2
IGMPv1 querier present timeout: -
IGMPv2 querier present timeout: 297 seconds
Host compatibility:             IGMPv3
IGMPv1 host present timeout:    -
IGMPv2 host present timeout:    -
Number of registered groups:    0
```

Many rows are similar to the output from the tabular format and will not be described again. The new rows are:

- `Operational warnings`:
  Holds one line per operational warning. An operational warning indicates *potential* configuration problems that should be addressed.
  - `MVR is active on the same VLAN. If the MVR VLAN is operationally active, it will win`:
    Indicates that MVR may consume queries on MVR ports marked as source ports. Even without this operational warning, MVR may consume reports received on any VLAN if the report's group address matches a permit rule in MVR's channel filter.
  - `At least one port has a filter profile attached, but IPMC profiles are globally disabled`:
    One or more filter profiles is configured on at least one port. This filter profile is not active, because IPMC profiles are globally disabled, so all group addresses will be permitted on the port(s).
  - `At least one port has attached a filter profile that doesn't exist`:
    At least one of the filter profiles configured on a port does not exist. This means that all group addresses will be denied on the port until the filter profile (IPMC profile) is created.
  - `At least one port has attached a filter profile without any IPv4 ranges attached`:
    At least one of the filter profiles configured on a port hasn't any ranges configured. This could be intentional, or it could be a mistake. The net-effect is that all group addresses will be denied.
    A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.

  - `At least one port has attached a filter profile without any IPv4 permit rules`:
    At least one of the filter profiles configured on a port doesn't have any IPv4 permit rules configured. This could be intentional, or it could be a mistake. The net-effect is that all group addresses will be denied.
    A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.
  - `At least one port has a filter profile attached, where an IPv4 deny rule shadows a permit rule coming later in the profile's rule list`:
    At least one of the filter profiles configured on a port has an IPv4 deny rule coming before an IPv4 permit rule, and the deny rule overlaps the IPv4 range of the permit rule.
    A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.

- `Querier uptime`:
  Shows the number of seconds that this switch has been a querier. Will be a `-` if this switch is not currently the querier or is in the `Init` state.
- `Configured compatibility`:
  Shows the compatibility configured for this VLAN (IGMPv1/IGMPv2/IGMPv3/Auto).
- `Querier compatibility`, `IGMPv1 host present timeout`, and `IGMPv2 host present timeout`:
  The `Querier compatibility` only differs from the configured compatibility if the configured compatibility is `Auto`. If this is the case and an IGMPv1 query is received, this field shows `IGMPv1` while the `IGMPv1 querier present timeout` counts the number of seconds that the querier will still be in IGMPv1 mode. Similarly, if an IGMPv2 query is received, the `IGMPv2 querier present timeout` counts the number of seconds that the querier will still be in IGMPv2 mode. If both IGMP1 and IGMPv2 have timeouts, the querier compatibility will be IGMPv1, in accordance with RFC3376.
- `Host compatibility`, `IGMPv1 host present timeout`, `IGMPv2 host present timeout`:
  The same rules as for the `Querier compatibility` apply to the `Host compatibility`, but instead of received queries, these are set based on received report (Join/Leave) PDUs.
- `Number of registered groups`:
  Shows the number of IGMP groups registered on this VLAN.

2.2.2. `show ip igmp snooping [vlan <vlan_list>] group-database [interface <port_type_list>] [details]`

Previously, in order to see SFM information, the command looked like:

```
show ip igmp snooping [vlan <vlan_list>] group-database [interface <port_type_list>] sfm-information [details]
```

Now, the command always shows sources if available.

Here are examples of outputs of the old command after having received one IGMPv3 TO_IN report with two source addresses (`11.11.11.3` and `11.11.11.4`) and group address `224.1.1.2`.

Old VLAN Group Database Layout with and without details and SFM info

```
# show ip igmp snooping group-database

IGMP Snooping is enabled to start snooping IGMP control plane.

IGMP Group Database

Switch-1 IGMP Group Count: 0
# show ip igmp snooping group-database

IGMP Snooping is enabled to start snooping IGMP control plane.

IGMP Group Database

Switch-1 IGMP Group Table

224.1.1.2 is registered on VLAN 2
Port Members: Gi 1/3

Switch-1 IGMP Group Count: 1
# show ip igmp snooping group-database detail

IGMP Snooping is enabled to start snooping IGMP control plane.
Multicast streams destined to unregistered IGMP groups will be blocking.
Groups in range 232.0.0.0/8 follow IGMP SSM registration service model.

IGMP Group Database

Switch-1 IGMP Group Table

224.1.1.2 is registered on VLAN 2
Port Members: Gi 1/3
Hardware Switch: Yes

Switch-1 IGMP Group Count: 1
# show ip igmp snooping group-database sfm-information

IGMP Snooping is enabled to start snooping IGMP control plane.

IGMP Group Database

Switch-1 IGMP Group Table

224.1.1.2 is registered on VLAN 2
Port Members: Gi 1/3
Gi 1/3 Mode is Include
Allow Source Address  : 11.11.11.3
Allow Source Address  : 11.11.11.4

Switch-1 IGMP Group Count: 1
# show ip igmp snooping group-database sfm-information detail

IGMP Snooping is enabled to start snooping IGMP control plane.
Multicast streams destined to unregistered IGMP groups will be blocking.
Groups in range 232.0.0.0/8 follow IGMP SSM registration service model.

IGMP Group Database

Switch-1 IGMP Group Table

224.1.1.2 is registered on VLAN 2
Port Members: Gi 1/3
Hardware Switch: Yes
Gi 1/3 Mode is Include
Allow Source Address  : 11.11.11.3 (Timer->32)
Hardware Filter: Yes
Allow Source Address  : 11.11.11.4 (Timer->32)
```

```
Hardware Filter: Yes

Switch-1 IGMP Group Count: 1
```

The new layout is in tabular form and if the `details` argument is specified, a few more columns are displayed, as can be seen from the example below, where the same IGMPv3 report is received on Gi 1/3:

New VLAN Group Database Layout with and without details

```
# show ip igmp snooping group-database
VLAN Group Address   Port        Source Address  Filter Mode Fwd
---- --------------- ----------- --------------- ----------- ---
   2 224.1.1.2       Gi 1/3      11.11.11.3      Include     Yes
                                 11.11.11.4      Include     Yes
                                 Other           Include     No
Total group count: 1 (2 sources)

# show ip igmp snooping group-database details
VLAN Group Address   Interface   Source Address  Filter Mode Fwd Grp Timeout Src Timeout In H/W
---- --------------- ----------- --------------- ----------- --- ----------- ----------- ------
   2 224.1.1.2       Gi 1/3      11.11.11.3      Include     Yes           -          34 Yes
                                 11.11.11.4      Include     Yes           -          34 Yes
                                 Other           Include     No            -           - Yes
Total group count: 1 (2 sources)
```

Each group always has a so-called ASM (Any-Source Multicast) entry. If a given <VID, Group, Port> has source addresses attached, these are shown first, followed by the ASM entry, which - in the Source Address field - then says `<Other>`, indicating that if sending multicast data from another source than one of the above-specified, this entry will be used.

The `VLAN`, `Group Address`, and `Port` columns will only be filled in if they differ from the previous line, so in this case, all three lines pertain to VLAN `2`, Group Address `224.1.1.2` and port `Gi 1/3`.

It could also happen that there are no sources in the IGMPv3/MLDv2 reports or a port could receive and process IGMPv1/IGMPv2/MLDv2 joins/reports. In these cases, there will be no source addresses to display and the ASM entry will be shown as `<Any>`, indicating the multicast data sent to this group will be treated according to the `Fwd` field irrespective of the source address.

The remaining columns are:

- `Fwd`:
  This field corresponds to the old layout's `Allow` and `Deny` output. It simply indicates with a `Yes`, that multicast data is forwarded (allowed) to this port or with a `No`, that it is blocked (denied)
- `Filter Mode`:
  This is similar to the old output's `Mode is Include` or `Mode is Exclude`.
- `Grp Timeout`:
  This holds the number of seconds until this group times out. It is only used when `Filter Mode` is `Exclude`.

- `Src Timeout`:
  Holds the number of seconds until this source times out. It is only active when sources are members of the include list.
- `In H/W`:
  This indicates whether the chip has a corresponding entry for this group or <group, source>. If it says `No`, it is because the chip's resources are depleted or because the entry is not needed (the entry is in the so-called "Requested List"), because the ASM entry takes care of the forwarding.

2.2.3. `show ip igmp snooping [vlan <vlan_list>] statistics [details]`

This is a new command, which shows various counters.

These counters used to be embedded in the `show ip igmp snooping detail` command (see above for an example).

Here's an example of the new layout:

New VLAN Statistics Database Layout without details

```
# show ip igmp snooping statistics
VLAN Rx Queries Tx Queries Rx Reports Tx Reports Rx Errors
---- ---------- ---------- ---------- ---------- ----------
   1          0         10         36          0          0
   2          0         54          3          0          0
```

`Rx Queries` and `Tx Queries` include both IGMPv1, IGMPv2, and IGMPv3 queries, and both general, group-specific, and group-and-source-specific queries. They also include both queries that are actually being processed by the switch and queries that are ignored, because of e.g. version incompatibility discards.
`Rx Reports` and `Tx Reports` include both IGMPv1, IGMPv2, and IGMPv3 Joins/ Reports as well as IGMPv2 Leave messages.
`Rx Errors` counts the number of PDUs that are discarded because they contain errors of some sort. That could be e.g. a Join with a non-multicast group-address, but there are myriads of other error possibilities.

Below is a more pinned-out version, shown using the `details` option:

New VLAN Statistics Database Layout with details

```
# show ip igmp snooping statistics details
VLAN 1 Statistics:
Counter                                  Rx Processed Rx Ignored Tx
---------------------------------------- ------------ ---------- ----------
IGMPv1 Joins                                        0          0          0
IGMPv1 Queries                                      0          0          0
IGMPv2 Joins                                       28          0          0
IGMPv2 Leaves                                       0          0          0
IGMPv2 General Queries                              0          0          6
IGMPv2 Group-Specific Queries                       0          0          0
IGMPv3 Reports                                      0          0          0
IGMPv3 General Queries                              0          0          1
IGMPv3 Group-Specific Queries                       0          0          0
IGMPv3 Group-and-Source-Specific Queries            0          0          0
IGMP Error Packets                                             0

VLAN 2 Statistics:
Counter                                  Rx Processed Rx Ignored Tx
---------------------------------------- ------------ ---------- ----------
IGMPv1 Joins                                        0          0          0
IGMPv1 Queries                                      0          0          0
IGMPv2 Joins                                        3          0          0
IGMPv2 Leaves                                       0          0          0
IGMPv2 General Queries                              0          0          6
IGMPv2 Group-Specific Queries                       0          0          0
IGMPv3 Reports                                      0          0          0
IGMPv3 General Queries                              0          0         35
IGMPv3 Group-Specific Queries                       0          0          0
IGMPv3 Group-and-Source-Specific Queries            0          0          0
IGMP Error Packets                                             0
```

There are four columns.

- `Counter`:
  Contains the counter name and is self-explanatory.
- `Rx Processed`:
  Contains the Rx counters for PDUs that are actually processed by the switch software.
- `Rx Ignored`:
  Contains Rx counters for PDUs that are ignored by the switch, e.g. because it is in forced IGMPv1 mode and has received an IGMPv3 report.
- `Tx`:
  Contains Tx counters. It is worth noting that forwarded frames are not counted, only frames generated by the switch software itself.

2.2.4. `show ip igmp snooping mrouter [details]`

Previously the output of this command was something along these lines:

Old Router Status Layout with and without details

```
# show ip igmp snooping mrouter

IGMP Snooping is enabled to start snooping IGMP control plane.

Switch-1 IGMP Router Port Status
Gi 1/3: Dynamic Router Port
Gi 1/4: Static Router Port
Gi 1/5: Static and Dynamic Router Port

# show ip igmp snooping mrouter detail

IGMP Snooping is enabled to start snooping IGMP control plane.
Multicast streams destined to unregistered IGMP groups will be blocking.

Switch-1 IGMP Router Port Status
Gi 1/3: Dynamic Router Port
Gi 1/4: Static Router Port
Gi 1/5: Static and Dynamic Router Port
```

The old command showed some enabledness of IGMP snooping and whether M/C data to unregistered groups were flooded or blocked. This is not part of the new output, because this can be found by `show running-config feature igmp [all-defaults]`.

The new output looks like this:

New Router Status Layout with and without details

```
# show ip igmp snooping mrouter
Interface  Router Status
---------- ------------------
Gi 1/3     Dynamic
Gi 1/4     Static
Gi 1/5     Static and Dynamic

# show ip igmp snooping mrouter details
Interface  Router Status      Dynamic Timeout
---------- ------------------ ---------------
Gi 1/3     Dynamic                        299
Gi 1/4     Static                           -
Gi 1/5     Static and Dynamic             123
```

Only router ports are shown with this command (that is, host ports are not shown).

The only difference between using or not using the `details` argument is a new column that shows the number of seconds until a dynamic entry times out. See Timeout on Dynamically Determined Router Ports for a description.

2.3. JSON/Private MIB Functionality

The JSON-RPC and Private MIB for IPMC are completely changed, and backwards incompatible.

2.4. Query Flood Suppression

Previously, IPMC shared a query suppression counter (to avoid too much query flooding) for both IGMP and MLD.

This is now moved to one for IGMP and one for MLD.

2.5. Timeout on Dynamically Determined Router Ports

Previously, whenever a query was received (and processed) on a port, that port was marked as a Dynamic Router port, and it would be like that until disabling IGMP or rebooting the switch.

With the new implementation, the port is still marked as a Dynamic Router port, but only for the next 300 seconds after the query was received and processed.

This prevents continuous flooding of M/C data to ports on which no router is actually located (anymore).

If this has undesired side-effects, the port can be marked as a static router port, which will never expire.

2.6. Max-Groups

The per-port `ip igmp snooping max-groups <1-10>` configuration option allows for not registering more than a given number of groups on a given port (the `no`-form disables this functionality).

In the previous implementation, if a port had reached its maximum when a report with a new group/new groups arrived, the report was not forwarded to router ports. Now, it gets forwarded anyway, but gets counted as ignored.

A small twist here is IGMPv3 reports, which can hold more than one group. If a report with two or more groups are received and at least one of these groups resulted in a new registration, the report is counted as "Rx Processed", not ignored. Still, it is forwarded to router ports in the VLAN.

2.7. Capabilities

There are functional differences both between chips and between versions of the WebStaX application software. A new debug command outlines the capatilibites on the switch at hand. Let’s see it in action:

```
# platform debug allow

WARNING: The use of 'debug' commands may negatively impact system behavior.
Do not enable unless instructed to. (Use 'platform debug deny' to disable
debug commands.)

NOTE: 'debug' command syntax, semantics and behavior are subject to change
without notice.

# debug show ipmc capabilities
 ---------------
IMPORTANT NOTE:
 ---------------
The resources in hardware used to forward multicast data correctly are of
limited sizes. Moreover, these resources are shared among many different
features, where IPMC is only one of them. Therefore, there is no guarantee that
the maximum number of multicast groups printed below is available. Furthermore,
this number represents the best case for IPv4 groups. Typically, IPv6 groups
take twice the amount of resources, so the best case for IPv6 groups is half of
the displayed number.
The maximum number of source addresses is also only a guiding value and depends
both on other features utilizing the resources in hardware as well as whether
the source addresses are IPv4 or IPv6.

IPMC supported by this implementation:                              Yes
MVR supported by this implementation:                               Yes
Max. number of IPMC VLAN interfaces:                                128
Max. number of MVR VLANs:                                           4
IGMP supported by this implementation:                              Yes
MLD supported by this implementation:                               Yes
H/W support for IPv4 Source Specific Multicast:                     Yes
H/W support for IPv6 Source Specific Multicast:                     Yes
Best case max. number of IPv4 M/C groups (IPv6 is about half of this): 1024
Best case max. number of IPv4 sources (IPv6 is about half of this):    1023
Max. number of source addresses per M/C group:                      8
```

The output is quite self-explanatory and will not be discussed further.

3. MVR Module Changes

3.1. General Changes

3.1.1. `show running-config` CLI command

Previously, when showing `running-config` for MVR, there were two features:
`show running-config feature mvr`
`show running-config feature mvr-port`

Now, these are condensed into one single:
`show running-config feature mvr`

3.1.2. Name

An MVR VLAN may have an associated name for easy identification.

Previously, it could start with a digit or special chars, like a quote. Now it must start with a letter ([a-zA-Z]. This corresponds to isalpha().

Previously, it could contain a ":" or be the word `all`. Now, `all` (case-insensitive) is reserved for referencing all entries in CLI in one go, and characters beyond the first may only be in this range [33; 126] except for 58 (which is a `:`). This corresponds to isgraph(), except for the reserved `:`.

Previously, it was not possible to delete a previously assigned name for an MVR VLAN in ICLI, without first deleting the entire MVR VLAN and then add it again without a name. Now, you can simply call it again without the `name` option:

```
! Start by adding an MVR VLAN with a name
(config)# mvr vlan 17 name hello

! Previously, you had to do this to unassociate the MVR VLAN with a name.
(config)# no mvr vlan 17
(config)# mvr vlan 17

! Now, you can do it by adding it again - this time without a name:
(config)# mvr vlan 17
```

3.1.3. Channel Profile

Previously, if a channel profile was assigned to an MVR VLAN and the profile was afterwards deleted, MVR's configured channel profile would also be deleted.

Now, MVR's configured channel profile stays with it, but an operational warning is raised and the MVR VLAN is made inactive (see later).

3.2. Changes in ICLI Status Commands and Output

There have been made significant changes to both ICLI status commands and their output.

The following sections attempt to describe these changes in details.

3.2.1. `show mvr [vlan <vlan_list>] [details]`

The output before of this command was:

Old MVR VLAN Status Layout

```
# show mvr detail

MVR is now enabled to start group registration.

Switch-1 MVR-IGMP Interface Status

IGMP MVR VLAN 4 (Name is not set) interface is enabled.
Querier status is IDLE ( Forced Non-Querier )
Querier Expiry Time: 255 seconds
IGMP address is not set and will use system's IP address of this interface.
Control frames will be sent as Tagged
PRI:0 / RV:2 / QI:125 / QRI:100 / LMQI:5 / URI:1
RX IGMP Query:16 V1Join:0 V2Join:6 V3Join:0 V2Leave:0
TX IGMP Query:0 / (Source) Specific Query:0
IGMP RX Errors:458; Group Registration Count:0
Port Role Setting:
Source Port  : Gi 1/3
Receiver Port: Gi 1/2
Inactive Port: Gi 1/1,Gi 1/4,Gi 1/5,Gi 1/6,Gi 1/7,Gi 1/8,Gi 1/9,Gi 1/10,Gi 1/11,Gi 1/12,Gi 1/13,Gi 1/
14,Gi 1/15,Gi 1/16,Gi 1/17,Gi 1/18,Gi 1/19,Gi 1/20,Gi 1/21,2.5G 1/1,2.5G 1/2,2.5G 1/3,2.5G 1/4,10G 1/
1,10G 1/2,10G 1/3,10G 1/4
Interface Channel Profile: Example (In IGMP Mode)
Description:
HEAD-> Video (Permit the following range and log the matched entry)
Start Address: 224.1.0.0
End Address  : 225.0.0.0

Switch-1 MVR-MLD Interface Status

MLD MVR VLAN 4 (Name is not set) interface is enabled.
Querier status is IDLE ( Forced Non-Querier )
Querier Expiry Time: 255 seconds
MLD address will use Link-Local address of this interface.
Control frames will be sent as Tagged
PRI:0 / RV:2 / QI:125 / QRI:100 / LMQI:5 / URI:1
RX MLD Query:0 V1Report:0 V2Report:0 V1Done:0
TX MLD Query:0 / (Source) Specific Query:0
MLD RX Errors:0; Group Registration Count:0
Port Role Setting:
Source Port  : Gi 1/3
Receiver Port: Gi 1/2
Inactive Port: Gi 1/1,Gi 1/4,Gi 1/5,Gi 1/6,Gi 1/7,Gi 1/8,Gi 1/9,Gi 1/10,Gi 1/11,Gi 1/12,Gi 1/13,Gi 1/
14,Gi 1/15,Gi 1/16,Gi 1/17,Gi 1/18,Gi 1/19,Gi 1/20,Gi 1/21,2.5G 1/1,2.5G 1/2,2.5G 1/3,2.5G 1/4,10G 1/
1,10G 1/2,10G 1/3,10G 1/4
Interface Channel Profile: Example (In IGMP Mode)
Description:
HEAD-> Video (Permit the following range and log the matched entry)
Start Address: 224.1.0.0
End Address  : 225.0.0.0
```

The old status was not easy readable and combined status, configuration, and statistics. The new status is in tabular form (unless also specifying the `details` argument, as we shall see in a moment).

New VLAN Status Layout

```
# show mvr
VLAN Protocol Operational State   Querier State Active Querier                          Next Query/
Expiry Time
---- -------- ------------------- ------------- ---------------------------------------
----------------------
 100 IGMP     Active              Disabled
-                                                             -
 100 MLD      Inactive (warnings) -
-                                                             -
```

For descriptions of the contents, please refer to the IPMC section’s `show ip igmp snooping [vlan <vlan_list>] [detail]`.

Compared to the IGMP/MLD Snooping output, one new Operational State can be displayed:

- `Inactive (warnings)`:
  Unlike IPMC VLANs, MVR VLANs can be rendered inactive by the application software if not all criteria for activating the MVR VLAN are met. MVR’s IGMP part runs separately from MVR’s MLD part, although both are created when an MVR VLAN is created.

In the output above, we can see that MVR VLAN 100’s IGMP part is active whereas MVR VLAN 100’s MLD part is inactive. Inactive corresponds to administratively disabled.

To see the actual reason for the inactiveness, use the `details` argument in the command:

New VLAN Status Layout

```
# show mvr details
VLAN:                             100
Name:
IGMP:
  Operational state:              Active
  Operational warnings:           None
  Querier state:                  Disabled
  Active querier:                 -
  Querier uptime:                 -
  Next query:                     -
  Other querier expiry time:      255 seconds
  Querier compatibility:          IGMPv3
  IGMPv1 querier present timeout: -
  IGMPv2 querier present timeout: -
  Host compatibility:             IGMPv3
  IGMPv1 host present timeout:    -
  IGMPv2 host present timeout:    -
  Number of registered groups:    0
MLD:
  Operational state:              Inactive
  Operational warnings:           MVR VLAN inactive, because the configured channel profile doesn't
have any IPv6 ranges attached
```

Here, the IGMP part if MVR VLAN 100 is active and it shows the same fields as does the IPMC counter part.

The MLD part is inactive, and the `Operational warnings` row shows why (there can be more than one operational warning, and if so, they are shown on their own rows).

Some operational warnings indicate *potential* configuration problems, whereas others indicate fatal configuration issues, causing the MVR VLAN for that protocol (IGMP/MLD) to be inactive.

All of the operational warnings that may cause an MVR VLAN protocol to become inactive are due to invalid channel profile configuration. A channel profile can - behind the back of MVR - be reconfigured or removed, and MVR takes appropriate action on this by activating or deactivating the MVR VLAN.

Here is a list of possible MVR operational warnings:

- `MVR VLAN inactive, because IPMC profiles are globally disabled`:
  MVR requires the use of channel profiles (IPMC profiles), so if that feature is globally disabled, MVR cannot run. Both IGMP and MLD will be inactive in that case.
- `MVR VLAN inactive, because a channel profile is not configured`:
  MVR requires the use of channel profiles, but no such profile is configured for this MVR VLAN. Both IGMP and MLD will be inactive in that case.
- `MVR VLAN inactive, because the configured channel profile doesn’t exist`:
  The configured channel profile doesn’t exist. Both IGMP and MLD will be inactive in that case.
- `MVR VLAN inactive, because the configured channel profile doesn’t have any IPv4 ranges attached`:
  No IPv4 ranges are included in the configured channel profile. This means that all IPv4 group addresses will be denied, so the MVR VLAN’s IGMP part will be inactive.
  A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.
- `MVR VLAN inactive, because the configured channel profile has no IPv4 permit rules`:
  The channel profile doesn’t specify any IPv4 permit rules, so an IGMP report will never find its way to the core of the IGMP protocol. Therefore the IGMP part is marked inactive. A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.
- `MVR VLAN inactive, because another MVR VLAN instance uses a profile with at least one IPv4 rule that overlaps this one’s`:
  Two MVR VLANs may not have overlapping IPv4 permit address ranges (because which one is then taking care of the IGMP reports?). The MVR VLAN’s IGMP part is inactive if this happens.
  A similar warning can be issued for MLD, where the text `IPv4` is exchanged with `IPv6`.
- `No ports are configured as sources`: To get rid of this, configure at least one port as a source port.
- `No ports are configured as receivers`: To get rid of this, configure at least one port as a receiver port.

- `At least one source port is member of a VLAN interface with same MVR VLAN ID`: In the previous implementation, if you configured an MVR VLAN, `X`, and `X` is also a VLAN interface (one that can be assigned an IP address), and you configure a port, `p`, as a source port and that port is a member of VLAN `X`, then the following trace warning was shown on the console every 5 seconds: `W mvr 15:35:58 60/_mvr_vlan_warning_handler#3885: Warning: Please adjust the management VLAN ports overlapped with MVR source ports!` This trace warning has now been replaced by this operational warnings.

3.2.2. `show mvr [vlan <vlan_list>] group-database [interface <port_type_list>] [details]`

Previously, in order to see SFM information, the command looked like:

```
show mvr [vlan <vlan_list>] group-database [interface <port_type_list>] sfm-information [detail]
```

The output of this command is the same as the output of the corresponding `show ip igmp snooping [vlan <vlan_list>] group-database…` command with the exception that both IGMP and MLD groups are displayed in the same table. Please refer to `show ip igmp snooping [vlan <vlan_list>] group-database [interface <port_type_list>] [details]` for a description.

3.2.3. `show mvr [vlan <vlan_list>] statistics [details]`

This is a new command, which shows various counters.

The output of this command is the same as the output of the corresponding `show ip igmp snooping [vlan <vlan_list>] statistics…` command with the exception that both IGMP and MLD counters are displayed in the same table. Please refer to `show ip igmp snooping [vlan <vlan_list>] statistics [details]` for a description.

3.3. JSON/Private MIB Functionality

The JSON-RPC and Private MIB for MVR are completely changed, and backwards incompatible.

3.4. Query Flood Suppression

Previously, MVR shared a query suppression counter (to avoid too much query flooding) for both IGMP and MLD.

This is now moved to one for IGMP and one for MLD.

4. IPMC Profile Changes

IPMC profiles are part of the IPMC-LIB module.

4.1. General Changes

4.1.1. Profile and Range Names

An IPMC profile is identified by a name and the same goes for profile ranges.

This is still the case, and with the same range of ASCII characters ([33; 126]), but the word 'all' (case-insentitively) is not allowed anymore. It is used in CLI to delete all profiles or ranges in one go.

4.1.2. `show running-config` CLI command

Previously, when showing `running-config` for IPMC profiles, there were two features:
`show running-config feature ipmc-profile`
and
`show running-config feature ipmc-profile-range`

Now, these are condensed into one single:
`show running-config feature ipmc-profile`

4.1.3. Bug Fix when deleting first range from a profile

Fixed bug in profiles when deleting first range from the profile.

Before it was like this:

```
! Create two ranges, a and b.
(config)# ipmc range a 224.0.0.1 224.0.0.255
(config)# ipmc range b 225.0.0.1 225.0.0.255

! Create a profile, p1.
(config)# ipmc profile p1
(config-ipmc-profile)# range a permit
(config-ipmc-profile)# range b permit

! See what came out of it:
(config-ipmc-profile)# do show running-config feature ipmc-profile
Building configuration...
ipmc profile p1
 range a permit
 range b permit
!
!
end

! So far so good. Now try to delete range a
(config-ipmc-profile)# no range a

! And notice that not only "range a" was removed, but also "range b".
(config-ipmc-profile)# do show running-config feature ipmc-profile
Building configuration...
ipmc profile p1
!
!
end

! Now, to emphasize that something is very wrong, try to add range b:
(config-ipmc-profile)# range b permit
(config-ipmc-profile)# do show run fea ipmc-profile
Building configuration...
ipmc profile p1
!
!
end

! Try to add range a again, but before the missing range b:
(config-ipmc-profile)# range a permit next b
% Failed to set range a in profile p1.

! This is different from adding a range before a range that doesn't exist:
(config-ipmc-profile) range a permit next c
% a is not a rule set in profile p1.
```

This is fixed in the new implementation.

4.2. ICLI Warning

Previously, when creating a profile with only deny rules, the following warning was issued:

```
%% Notice that this profile performs deny action for all groups since there is no any permit entry is
included in the profile name <profile-name>
```

This has been removed, and instead it is up to the users of the profile (MVR and IGMP/MLD snooping) to issue operational warnings when this is the case.

4.3. Global Enable

RBNTBD: Rethink this, because in the old implementation, if profiles were globally disabled, ipmc_lib_profile_match() returned "PERMIT".

Previously, there was a global enable of using profiles. Now, profiles are always globally enabled, so they can no longer be disabled.

The code accepts startup-configs containing lines with `no ipmc profile` or `ipmc profile`, but they have no effect. The commands are not available from the console anymore and `show running-config` will never output these lines.

4.4. 'show ipmc profile' CLI Command

The full command is changed from
`show ipmc profile [<word16>] [detail]`
to
`show ipmc profile [name <word1-16>] [details]`

The reason for this change is that it was impossible to show a profile called `detail`, so inserting the `name` keyword makes this possible. I don’t think a lot of customers will notice, because if they use this command, they most likely use it in the short form, like this:
`show ipmc profile`
or
`show ipmc profile detail`
both of which still work.

The output of the command has also changed.

Suppose we have the following profile configuration:

```
ipmc range Audio ff3e::1234 ff3e::2234
ipmc range Video 226.1.2.3 226.2.2.3
ipmc profile Example
 description My profile
 range Video permit
 range Audio deny log
```

Before, the output of `show ipmc profile` was:

```
# show ipmc profile

IPMC Profile is now enabled to start filtering.

Profile: Example (In HYBRID Mode)
Description: My profile
HEAD-> Video (Permit the following range)
Start Address: 226.1.2.3
End Address  : 226.2.2.3
NEXT-> Audio (Deny the following range and log the matched entry)
Start Address: ff3e::1234
End Address  : ff3e::2234
```

Now, it is set up in tabular form, which makes it more readable. There is not room for the profile description in such a format, so this has been omitted. Also, since IPMC profiles are always globally enabled now, the first line (`IPMC Profile is now enabled to start filtering`) is omitted as well.

```
# show ipmc profile
Profile Name     Range Name       Start Address                           End
Address                             Permit Log
---------------- ---------------- ---------------------------------------
--------------------------------------- ------ ---
Example          Video            226.1.2.3
226.2.2.3                               Yes    No
                 Audio            ff3e::1234
ff3e::2234                              No     Yes
```

Before, the output of the `show ipmc profile detail` ICLI command would give:

```
# show ipmc profile detail

IPMC Profile is now enabled to start filtering.

Profile: Example (In HYBRID Mode)
Description: My profile
HEAD-> Video (Permit the following range)
Start Address: 226.1.2.3
End Address  : 226.2.2.3
NEXT-> Audio (Deny the following range and log the matched entry)
Start Address: ff3e::1234
End Address  : ff3e::2234

IGMP will deny matched address between [224.0.0.0 <-> 226.1.2.2]
IGMP will permit matched address between [226.1.2.3 <-> 226.2.2.3]
IGMP will deny matched address between [226.2.2.4 <-> 239.255.255.255]
MLD will deny matched address between [ff00:: <-> ff3e::1233]
MLD will deny and log matched address between [ff3e::1234 <-> ff3e::2234]
MLD will deny matched address between [ff3e::2235 <-> ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]
```

Now, this will be shown in tabular form as well, so there’s not room for the description here either. Unused ranges are listed with a range name of `<Default>`:

```
# show ipmc profile detail
Profile Name     Range Name       Start Address                           End
Address                             Permit Log
---------------- ---------------- ---------------------------------------
--------------------------------------- ------ ---
Example          <Default>        224.0.0.0
226.1.2.2                               No     No
                 Video            226.1.2.3
226.2.2.3                               Yes    No
                 <Default>        226.2.2.4
239.255.255.255                         No     No
                 <Default>        ff00::
ff3e::1233                              No     No
                 Audio            ff3e::1234
ff3e::2234                              No     Yes
                 <Default>        ff3e::2235
ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff No     No
```

4.5. 'show ipmc range' CLI Command

The output of this command is also changed to be in tabular form.

Given the configuration from the previous chapter, the old implementation would output this:

```
# show ipmc range

Range Name   : Audio
Start Address: ff3e::1234
End Address  : ff3e::2234

Range Name   : Video
Start Address: 226.1.2.3
End Address  : 226.2.2.3
```

Now, it will look like this:

```
# show ipmc range
Range Name       Start Address                           End Address
---------------- --------------------------------------- ---------------------------------------
Audio            ff3e::1234                              ff3e::2234
Video            226.1.2.3                               226.2.2.3
```