Implementing MLD Snooping

This module describes how to implement MLD snooping on the Cisco ASR 9000 Series Router.

Feature History for MLD Snooping

Release	Modification
Release 4.3.0	This feature was introduced.

MLD Snooping

Multicast Listener Discovery (MLD) snooping provides a way to constrain multicast traffic at Layer 2. By snooping the MLD membership reports sent by hosts in the bridge domain, the MLD snooping application can set up Layer 2 multicast forwarding tables to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

MLD snooping uses the information in MLD membership report messages to build corresponding information in the forwarding tables to restrict IPv6 multicast traffic at Layer 2. The forwarding table entries are in the form <Route, OIF List>, where:

- Route is a <*, G> route or <S, G> route.
- OIF List comprises all bridge ports that have sent MLD membership reports for the specified route plus all multicast router (mrouter) ports in the bridge domain.

For more information regarding MLD snooping, refer the *Multicast Configuration Guide for Cisco ASR 9000 Series Routers*.

Prerequisites for MLD Snooping

- The network must be configured with a layer2 VPN.
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for MLD Snooping

Following are the restrictions (features that are not supported):

- MLD Snooping is supported only on L2VPN bridge domains.
- Explicit host tracking.
- Multicast Admission Control.
- Security filtering.
- Report rate limiting.
- Multicast router discovery.

Advantages of MLD Snooping

Advantages of MLD Snooping

- In its basic form, it reduces bandwidth consumption by reducing multicast traffic that would otherwise flood an entire VPLS bridge domain.

- With the use of some optional configurations, it provides security between bridge domains by filtering the MLD reports received from hosts on one bridge port and preventing leakage towards the hosts on other bridge ports.

High Availability (HA) features for MLD

MLD supports the following HA features:

- Process restarts
- RP Failover
- Stateful Switch-Over (SSO)
- Non-Stop Forwarding (NSF)—Forwarding continues unaffected while the control plane is restored following a process restart or route processor (RP) failover.
- Line card online insertion and removal (OIR)

Bridge Domain Support for MLD

MLD snooping operates at the bridge domain level. When MLD snooping is enabled on a bridge domain, the snooping functionality applies to all ports under the bridge domain, including:

- Physical ports under the bridge domain.
- Ethernet flow points (EFPs)—An EFP can be a VLAN, VLAN range, list of VLANs, or an entire interface port.
- Pseudowires (PWs) in VPLS bridge domains.
- Ethernet bundles—Ethernet bundles include IEEE 802.3ad link bundles and Cisco EtherChannel bundles. From the perspective of the MLD snooping application, an Ethernet bundle is just another EFP. The forwarding application in the Cisco ASR 9000 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

Multicast Router and Host Ports

MLD snooping classifies each port as one of the following:

- Multicast router ports (mrouter ports)—These are ports to which a multicast-enabled router is connected. Mrouter ports are usually dynamically discovered, but may also be statically configured. Multicast traffic is always forwarded to all mrouter ports, except when an mrouter port is the ingress port.
- Host ports—Any port that is not an mrouter port is a host port.

Multicast Router Discovery for MLD

MLD snooping discovers mrouter ports dynamically. You can also explicitly configure a port as an emrouter port.

- Discovery- MLD snooping identifies upstream mrouter ports in the bridge domain by snooping mld query messages and Protocol Independent Multicast Version 2 (PIMv2) hello messages. Snooping PIMv2 hello messages identifies mld nonqueriers in the bridge domain.
- Static configuration—You can statically configure a port as an mrouter port with the **mrouter** command in a profile attached to the port. Static configuration can help in situations when incompatibilities with non-Cisco equipment prevent dynamic discovery.

Multicast Traffic Handling for MLD

The following tables describe the traffic handling behavior by MLD mrouters and host ports.

Table 1: Multicast Traffic Handling for a MLDv1 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—
MLD group-specific queries	Forwards to all other mrouter ports.	Dropped
MLDv1 joins	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.
MLDv2 reports	Ignores	Ignores
MLDv1 leaves	Invokes last member query processing.	Invokes last member query processing.

Table 2: Multicast Traffic Handling for a MLDv2 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—

Traffic Type	Received on MRouter Ports	Received on Host Ports
MLD group-specific queries	If received on the querier port floods on all ports.	—
MLDv1 joins	Handles as MLDv2 IS_EX{} reports.	Handles as MLDv2 IS_EX{} reports.
MLDv2 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
MLDv1 leaves	Handles as MLDv2 IS_IN{} reports.	Handles as MLDv2 IS_IN{} reports.

Creating a MLD Snooping Profile

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. Optionally, add commands to override default configuration values.
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile default-bd-profile`	Enters MLD snooping profile configuration mode and creates a named profile. The default profile enables MLD snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.
Step 3	Optionally, add commands to override default configuration values.	If you are creating a bridge domain profile, consider the following: • An empty profile is appropriate for attaching to a bridge domain. An empty profile enables MLD snooping with default configuration values. • You can optionally add more commands to the profile to override default configuration values.

	Command or Action	Purpose
		• If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port. If you are creating a port-specific profile, consider the following: • While an empty profile could be attached to a port, it would have no effect on the port configuration. • When you attach a profile to a port, MLD snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations. You can detach a profile, change it, and reattach it to add commands to a profile at a later time.
Step 4	**commit**	

Activating MLD Snooping on a Bridge Domain

To activate MLD snooping on a bridge domain, attach a MLD snooping profile to the desired bridge domain as explained here.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **mld snooping profile** *profile-name*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.

	Command or Action	Purpose
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# mld snooping profile default-bd-profile`	Attaches the named MLD snooping profile to the bridge domain, enabling MLD snooping on the bridge domain.
Step 6	**commit**	

Deactivating MLD Snooping on a Bridge Domain

To deactivate MLD snooping from a bridge domain, remove the profile from the bridge domain using the following steps:

Note A bridge domain can have only one profile attached to it at a time.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **no mld snooping**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:**	Enters Layer 2 VPN configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# l2vpn`	
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**no mld snooping** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# no mld snooping`	Detaches the MLD snooping profile from the bridge domain, disabling MLD snooping on that bridge domain. **Note** Only one profile can be attached to a bridge domain at a time. If a profile is attached, MLD snooping is enabled. If a profile is not attached, MLD snooping is disabled.
Step 6	**commit**	

Configuring Static Mrouter Ports (MLD)

Before you begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Note Static mrouter port configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add mrouter port configuration to a profile intended for bridge domains.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **mrouter**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile mrouter-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**mrouter** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# mrouter`	Configures a port as a static mrouter port.
Step 4	**commit**	

Configuring Router Guard (MLD)

To prevent multicast routing protocol messages from being received on a port and, therefore, prevent a port from being a dynamic mrouter port, follow these steps. Note that both router guard and static mrouter commands may be configured on the same port.

Before you begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Note Router guard configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add router guard configuration to a profile intended for bridge domains. To do so would prevent all mrouters, including MLD queriers, from being discovered in the bridge domain.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **router-guard**
4. **commit**
5. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile host-port-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.
Step 3	**router-guard** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# router-guard`	Protects the port from dynamic discovery.
Step 4	**commit**	
Step 5	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Immediate-leave for MLD

To add the MLD snooping immediate-leave option to an MLD snooping profile, follow these steps.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **immediate-leave**
4. **commit**
5. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.

	Command or Action	Purpose
	`host-port-profile`	
Step 3	**immediate-leave** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# immediate-leave`	Enables the immediate-leave option. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	**commit**	
Step 5	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile host-port-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Internal Querier for MLD

Before you begin

MLD snooping must be enabled on the bridge domain for this procedure to take effect.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **system-ip-address** *ip-addr*
4. **internal-querier**
5. **commit**
6. **show mld snooping profile** *profile-name* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config)# mld snooping profile internal-querier-profile`	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.

	Command or Action	Purpose
Step 3	**system-ip-address** *ip-addr* **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# system-ip-address 10.1.1.1`	Configures an IP address for internal querier use. The default system-ip-address value (0.0.0.0) is not valid for the internal querier. You must explicitly configure an IP address.
Step 4	**internal-querier** **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# internal-querier`	Enables an internal querier with default values for all options.
Step 5	**commit**	
Step 6	**show mld snooping profile** *profile-name* **detail** **Example:** `RP/0/RSP0/CPU0:router# show mld snooping profile internal-querier-profile detail`	(Optional) Displays the configuration settings in the named profile.

Configuring Static Groups for MLD

To add one or more static groups or MLDv2 source groups to an MLD snooping profile, follow these steps.

Before you begin

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

SUMMARY STEPS

1. **configure**
2. **mld snooping profile** *profile-name*
3. **static-group** *group-addr* [**source** *source-addr*]
4. Repeat the previous step, as needed, to add more static groups.
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**mld snooping profile** *profile-name* **Example:**	Enters MLD snooping profile configuration mode and creates a new profile or accesses an existing profile.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# mld snooping profile host-port-profile`	
Step 3	**static-group** *group-addr* [**source** *source-addr*] **Example:** `RP/0/RSP0/CPU0:router(config-mld-snooping-profile)# static-group 239.1.1.1 source 10.0.1.1`	Configures a static group. • If you add this option to a profile attached to a bridge domain, it applies to all ports under the bridge. • If you add this option to a profile attached to a port, it applies to the port.
Step 4	Repeat the previous step, as needed, to add more static groups.	(Optional) Adds additional static groups.
Step 5	**commit**	

Configuring MLD Snooping

1. Create two profiles:

```
mld snooping profile bridge_profile
!
mld snooping profile port_profile
   mrouter
!
```

2. Configure two physical interfaces for L2 support.

```
interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   !
!
interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   no shut
   !
!
```

3. Add interfaces to the bridge domain. Attach bridge_profile to the bridge domain and port_profile to one of the Ethernet interfaces. The second Ethernet interface inherits MLD snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     mld snooping profile bridge_profile
     interface GigabitEthernet0/8/0/38
       mld snooping profile port_profile
```

```
        interface GigabitEthernet0/8/0/39
        !
    !
!
```

4. Verify the configured bridge ports.

```
show mld snooping port
```

Multicast Listener Discovery over BVI

MLDv2 support over BVI enables implementing IPv6 multicast routing over a L2 segment of the network that is using an IPv6 VLAN. The multicast routes are bridged through BVI interface from L3 segment to the L2 segment of the network.

Note

- As per MLDv2 RFC recommendation the MLDv2 reports should carry the Hop-by-Hop options header for the reports to get punted up.
- MLDv2 is supported over BVI only when BVI is configured as a forwarding interface.
- This feature is supported only on 64-bit Linux-based IOS XR ASR 9000 operating system.

MLD and BVI Overview

Routers use the Internet Group Management Protocol (IGMP) (IPv4) and Multicast Listener Discovery (MLD) (IPv6) to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending IGMP or MLD report messages.

MLDv2 shares feature parity with IGMPv3 with respect to all supported interface types with the exception of PPoE and subinterfaces. MLDv2 enables a node to report interest in listening to packets only from specific multicast source addresses.

A BVI interface is a routed interface representing a set of interfaces (bridged) in the same L2 broadcast domain. MLD join messages coming in or out of this broadcast domain passes through the BVI interface.

Configure MLD Over BVI

This sample configuration shows how to configure BVI interface to join a multicast group and statically forward multicast traffic using MLDv2:

```
router# configure terminal
router (config)# router mld
router (config-mld)# vrf BVI
router (config-mld-vrf)# interface BVI100
router (config-mld-vrf-int)# join-group fe32::1 192::4
router (config-mld-vrf-int)# static-group fe32::2 192::4
router (config-mld-vrf-int)# commit
router (config-mld-vrf-int)# exit
router (config-mld-vrf)# exit
```

```
router (config-mld)# exit
router (config)# exit
```

Verification

Use the command **show mld bvi stats** and **show mld group bvi** < *num*> to verify the MLDv2 over BVI configuration:

```
router# show mld bvi stats
Thu Nov 22 13:58:34.474 UTC
AIPC buffers received                     : 8365
AIPC buffer released                      : 8365
AIPC messages send blocked                : 0
AIPC buffer release failed                : 0
AIPC NULL buffer handles                  : 0
AIPC open notifications received          : 0
AIPC close notifications received         : 0
AIPC error notifications received         : 0
AIPC LWM notifications received           : 0
AIPC input waiting notifications received : 8308
AIPC send status notifications received   : 2485
AIPC publish notifications received       : 0
AIPC queue full notifications received    : 0
AIPC output notifications received        : 0
AIPC connect notifications received       : 1
IGMP protocol messages received           : 8365
IGMP Mrouter Add messages received        : 0
IGMP Mrouter Delete messages received     : 0
IGMP Mrouter Sweep messages received      : 1
IGMP Mrouter Add messages transmitted     : 13
IGMP Mrouter Delete messages transmitted  : 22
IGMP Mrouter Sweep messages transmitted   : 0
IGMP Mrouter Unknown messages received    : 0
IGMP Mrouter Unknown messages transmitted : 0
AIPC transmission error                   : 0
AIPC buffers transmited                   : 0
IGMP protocol buffers transmitted         : 2482
IGMP Mrouter buffers transmitted          : 3
IGMP Unknown buffers transmited           : 0
IGMP WTX Msgs received                    : 0
IGMP WTX Msgs sent                        : 0
IGMP WTX Msgs sent to protocol            : 0
IGMP WTX Msgs dropped due DC              : 99264
IGMP WTX Msgs dropped no memory           : 0
IGMP WTX Msgs freed                       : 0


router# show mld group bvi 100
Thu Nov 22 13:58:52.055 UTC
MLD Connected Group Membership

BVI100

Group Address : ff02::2
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 03:31:07
      Expires : never
Group Address : ff02::d
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 03:31:07
show mld group bvi 100
Thu Nov 22 13:58:52.055 UTC
MLD Connected Group Membership
```

```
BVI100

Group Address : ff02::2
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 03:31:07
      Expires : never
Group Address : ff02::d
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 03:31:07
      Expires : never
Group Address : ff02::16
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 03:31:07
      Expires : never
Group Address : ff02::1:ff01:1
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 01:59:20
      Expires : 00:04:01
Group Address : ff02::1:ff3d:b73f
Last Reporter : fe80::1a33:9dff:fe3d:b73f
       Uptime : 01:59:20
      Expires : 00:04:01
Group Address : ff33::2:52:1:1
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:2
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:3
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:4
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:5
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:6
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
Group Address : ff33::2:52:1:7
Last Reporter : fe80::5869
       Uptime : 03:30:49
      Expires : not used
```

IPv6 Multicast Listener Discovery Snooping over BVI

Multicast Listener Discovery (MLD) snooping provides a way to constrain multicast traffic at L2. By snooping the MLD membership reports sent by hosts in the bridge domain, the MLD snooping application can set up L2 multicast forwarding tables. This table is later used to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

MLDv2 support over BVI enables implementing IPv6 multicast routing over a L2 segment of the network that is using an IPv6 VLAN. The multicast routes are bridged via BVI interface from L3 segment to L2 segment of the network.

MLDv2 snooping over BVI enables forwarding MLDv2 membership reports received over the L2 domain to MLD snooping instead of MLD.

Configuring Internal Querier for MLD Snooping

This configuration enables a multicast router acting as a MLD querier to send out group-and-source-specific query:

```
router# config
RP0/0/RP0/CPU0:router(config)# mld snooping profile grp1
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# system-ip-address fe80::1 link-local
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# internal-querier
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# commit
```

Verification

Use the **show mld snooping profile detail** command to verify the MLD snooping configuration:

```
router# show mld snooping profile detail
Thu Nov 22 13:58:18.844 UTC
MLD  Snoop Profile grp1:
  System IP Address:                    fe80::1
  Bridge Domain References:             2
  Port References:                      12

MLD  Snoop Profile grp10:
  System IP Address:                    fe80::5610
  Bridge Domain References:             0
  Port References:                      0
```

MLD Snooping Synchronization for EVPN Multi-Homing

In an EVPN multi-homing network, where customer edge devices (CEs) are multi-homed to more than one PE device, the MLD snooping sync feature enables routers to accurately track multicast group membership information and forward multicast traffic only to the interested receivers.

In an All-Active redundancy mode, the CEs can send an MLD message to any one of the multi-homed PEs, either DF or non-DF. Only the EVPN DF forwards traffic for the bridge domain (BD) for any group. Therefore, all PEs attached to a given EVPN Segment (ES) must coordinate MLD Join and Leave Group (x, G) state, where x may be either '*' or a particular source S, for each [EVI, broadcast domain (BD)] on that ES. This allows the DF for that ES, EVI, or BD to correctly advertise or withdraw a Selective Multicast Ethernet Tag route for that (x, G) group in that EVI or BD when needed.

In Single-Active redundancy mode, the PEs attached to a multi-homed ES coordinate the MLD Join (x, G) state. MLD join messages are received by the DF PE and distributed to the non-DF PEs for faster convergence. The non-DF PE also receives traffic by building the distribution tree toward the Rendezvous Point (RP) or multicast source, but doesn't forward it to the receivers in a multicast group. When a non-DF PE becomes the DF PE, it starts forwarding traffic to the CE.

Some benefits of the MLD state synchronization feature are as follows:

- **Seamless Mobility Support**—It ensures smooth mobility support for multicast listeners. When listeners move between different network devices or ports, the synchronized MLD snooping state helps maintain consistent multicast group membership information. The DF intelligently updates the forwarding information, ensuring uninterrupted multicast service delivery to mobile listeners.
- **Reduced Control Plane Overhead**—By synchronizing the MLD snooping state, we have reduced signaling messages overhead in the control plane for routing. The DF processes and propagates multicast control messages, such as MLD join and leave messages, only to the relevant ports based on the synchronized group membership information. This minimizes unnecessary control plane processing and improves network scalability.
- **Enhanced Network Stability**:—It contributes to network stability by maintaining consistent multicast group membership information across PE devices. This ensures reliable multicast service delivery and prevents disruptions or inconsistencies that could impact the network's overall performance.
- **Efficient Resource Utilization**—It uses a DF to optimize resource utilization by forwarding multicast traffic only to the ports where receivers are present. This prevents unnecessary multicast data replication and conserves network bandwidth, improving overall network efficiency.

MLD Snooping Synchronization with Proxy Querier

Each subnet has one of the two roles:

- Querier—the router with the lowest IP address in a subnet. Querier is responsible for sending the MLD or IGMP queries to know which multicast groups are active on the subnet.
- Non-Querier—the router that listens for MLD or IGMP queries and forwards them to the entire VLAN.

Initially, all multicast routers start up as a Querier on each attached network. If a router hears a Query message from a lower IP address, it becomes a Non-Querier. If a router doesn't hear a Query message for a certain period, it becomes the Querier again. The Querier router regularly sends a General Query on each attached network to gather multicast group membership information.

In this feature, two peer PEs in EVPN can both act as Queriers for the same BD. The first PE receiving the MLD join from CE sends an EVPN Join sync message to the second peer PE, which, upon receipt, sets the "learnt via EVPN " flag on the group. The group is not expired for lack of a direct MLD Join response to the initiated query (by second PE) as long as the 'learnt via EVPN' flag is set.

Note The MLD queries are not sent over the MPLS core. ACL filter is applied on the core facing interface to drop all the MLD queries.

Usage Guidelines and Limitations

A BD can have a mix of MLDv1 and MLDv2 receivers in it. Additionally, MLDv2 Join messages could either be in the include or exclude mode, where a multicast receiver can specify to either listen only for packets from some list of source addresses (include) or only for packets that don't come from some list of source addresses (exclude).

In the following table, use the supported and unsupported scenarios for the MLDv1 and MLDv2 Joins at PE as guidelines for using the MLD snooping synchronization feature:

Table 3: MLDv1 and MLDv2 Mix Joins—Supported and Unsupported Scenarios

Before Join Received State	After Join Received State			
	V1 (*, G)	**V2 Include (S, G)**	**V2 Exclude (*, G)**	**V2 Exclude (S, G)**
No state	Accepted	Accepted	Accepted	Drop
V1 (*, G)	Accepted	Drop	Accepted	Drop
V1 Include (S, G)	Drop	Accepted	Drop	Drop
V2 Exclude (*, G)	Accepted	Drop	Accepted	Drop

This feature has the following limitations:

- If the source is directly connected to the PE where the MLD Join is received, no MLD sync route is generated.
- Any router behind an All-Active multi-homed network is not supported.
- Configuring different MLD snooping profiles on peer PEs in an All-Active multi-homed network is not supported.
- An mrouter port behind CE is not supported.
- To prevent convergence issues, per multicast route DF election is not supported.
- The IGMP and MLD snooping profiles must be enabled together.

Configure MLD Snooping Synchronization for EVPN Multi-Homing

To configure MLD Snooping Synchronization for EVPN Multi-Homing, use the following example configuration:

/* Configure the EVPN EVI */

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether34
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 23.23.23.11.FF.11.11.11.11
Router(config-evpn-ac-es)# exit
Router(config-evpn-ac)# exit
Router(config-evpn)# evi 5
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance)# exit
Router(config-evpn)# exit
```

/* Configure the L2VPN BD with MLD snooping profile and EVI */

```
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
```

```
Router(config-l2vpn-bg)# bridge-domain bd5
Router(config-l2vpn-bg-bd)# mld snooping profile prof1
Router(config-l2vpn-bg-bd-mld-snooping-profile)# exit
Router(config-l2vpn-bg-bd)# igmp snooping profile prof2
Router(config-l2vpn-bg-bd-igmp-snooping-profile)# exit
Router(config-l2vpn-bg-bd)# interface Bundle-Ether34.5
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# routed interface BVI5
Router(config-l2vpn-bg-bd-bvi)# exit
Router(config-l2vpn-bg-bd)# evi 5
Router(config-l2vpn-bg-bd-evpn-instance)# exit
```

/* Configure the MLD snooping profile

```
Router(config)# mld snooping profile prof1
Router(config-mld-snooping-profile)# internal-querier
Router(config-mld-snooping-profile)# internal-querier query-interval 5
Router(config-mld-snooping-profile)# commit
```

Running Configuration

```
/*EVPN EVI*/
evpn
    interface Bundle-Ether34
      ethernet-segment
        identifier type 0 23.23.23.11.FF.11.11.11.11
    !
    evi 5
      advertise-mac
      !
    !

/* Configure the L2VPN BD with MLD snooping profile and EVI */
l2vpn
 bridge group bg1
   bridge-domain bd5
     mld snooping profile prof1
     igmp snooping profile prof2
     interface Bundle-Ether34.5
     !
     routed interface BVI5
     !
     evi 5
     !

/*MLD Snooping Profile*/
mld snooping profile prof1
     internal-querier
     internal-querier query-interval 5
```

Verify MLD Snooping Synchronization for EVPN Multi-Homing

To verify the configuration for this feature, use the following example commands.

/*Verify MLD Snooping Synchronization*/

```
RP/0/RP1/CPU0:tb11-r8#show mld  snooping  group
Fri Oct  6 17:53:42.640 UTC
```

```
Key: GM=Group Filter Mode, PM=Port Filter Mode
Flags Key: S=Static, D=Dynamic, E=Explicit Tracking, R=Replicated

                          Bridge Domain bg-1:bd-1001

Ver GM PM Port                       Exp   Flgs Group,Source
--- -- -- ----                       ---   ---- ------------
*/B indicates MLD snooping sync through BGP*/
V2  IN IN BE1.1001                   never B    ff03::1,1108:101::100

*/D indicates MLD snooping state is locally learned through EVPN*/
V2  IN IN BE2.1001                   223   D    ff03::1,1108:101::100
```

/*Verify DF Election*/

```
RP/0/RP1/CPU0:tb11-r8#show evpn ethernet-segment carving detail

Tue Oct 17 18:14:56.607 UTC
Legend:
  B   - No Forwarders EVPN-enabled,
  C   - MAC missing (Backbone S-MAC PBB-EVPN / Grouping ES-MAC vES),
  RT  - ES-Import Route Target missing,
  E   - ESI missing,
  H   - Interface handle missing,
  I   - Name (Interface or Virtual Access) missing,
  M   - Interface in Down state,
  O   - BGP End of Download missing,
  P   - Interface already Access Protected,
  Pf  - Interface forced single-homed,
  R   - BGP RID not received,
  S   - Interface in redundancy standby state,
  X   - ESI-extracted MAC Conflict
  SHG - No local split-horizon-group label allocated
  Hp  - Interface blocked on peering complete during HA event
  Rc  - Recovery timer running during peering sequence

Ethernet Segment Id      Interface                          Nexthops
------------------------ ---------------------------------- ----------
0000.0100.ac00.0001.0a00 BE1                                7.7.7.7
                                                            8.8.8.8
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : b402.1657.e485
     IfHandle       : 0x2000a164
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0000.0100.ac00.0001.0a00
  ES Import RT      : 0001.00ac.0000 (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
     7.7.7.7 [MOD:P:00:T]
     8.8.8.8 [MOD:P:00:T]
  Service Carving Synchronization:
     Mode           : NONE
```

```
   Peer Updates   :
               7.7.7.7 [SCT: N/A]
               8.8.8.8 [SCT: N/A]
Service Carving Results:
   Forwarders     : 999
   Elected        : 500
          EVI E   :     1001,    1003,    1005,    1007,    1009,    1011
          EVI E   :     1013,    1015,    1017,    1019,    1021,    1023,
          …..
          EVI E   :     1999,    2001
   Not Elected    : 499
          EVI NE  :     1002,    1004,    1006,    1008,    1010,    1012
         …
          EVI NE  :     1990,    1992,    1994,    1996,    1998,    2000,
          EVI NE  :     2002
…..
   Main port         :
   Interface name : Bundle-Ether2
   Interface MAC  : b402.1657.e484
   IfHandle       : 0x2000a16c
   State          : Up
   Redundancy     : Not Defined
ESI ID            : 1
ESI type          : 0
   Value          : 0011.0200.ac00.0001.0a00
ES Import RT      : 1102.00ac.0000 (from ESI)
Source MAC        : 0000.0000.0000 (N/A)
Topology          :
   Operational    : MH, All-active
   Configured     : All-active (AApF) (default)
Service Carving   : Auto-selection
   Multicast      : Disabled
Convergence       :
Peering Details   : 2 Nexthops
   7.7.7.7 [MOD:P:00:T]
   8.8.8.8 [MOD:P:00:T]
Service Carving Synchronization:
   Mode           : NONE
   Peer Updates   :
               7.7.7.7 [SCT: N/A]
               8.8.8.8 [SCT: N/A]
Service Carving Results:
   Forwarders     : 998
   Elected        : 500
          EVI E   :     1001,    1003,    1005,    1007,    1009,    1011
         …
          EVI E   :     1987,    1989,    1991,    1993,    1995,    1997,
          EVI E   :     1999,    2001
   Not Elected    : 498
          EVI NE  :     1002,    1004,    1006,    1008,    1010,    1012
          EVI NE  :     1980,    1982,    1984,    1986,    1988,    1990,
          EVI NE  :     1992,    1994,    1996,    1998,    2000,    2002
EVPN-VPWS Service Carving Results:
   Primary        : 0
   Backup         : 0
   Non-DF         : 0
MAC Flushing mode : STP-TCN
Peering timer     : 3 sec [not running]
Recovery timer    : 30 sec [not running]
Carving timer     : 0 sec [not running]
Revert timer      : 0 sec [not running]
HRW Reset timer   : 5 sec [not running]
Local SHG label   : 27051
Remote SHG labels : 1
```

```
             27051 : nexthop 7.7.7.7
  Access signal mode: Bundle OOS

N/A                      Te0/1/0/4/0                          8.8.8.8
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : TenGigE0/1/0/4/0
     Interface MAC  : b402.1657.e0a0
     IfHandle       : 0x020040c8
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 0
  ESI type          : Invalid
  ES Import RT      : 0000.0000.0000 (Incomplete Configuration)
  Source MAC        : b402.1657.e480 (PBB BSA, no ESI)
  Topology          :
     Operational    : SH
     Configured     : Single-active (AApS) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 1 Nexthops
     8.8.8.8 [MOD:P:00]
  Service Carving Synchronization:
     Mode           : NONE
     Peer Updates   :
                 8.8.8.8 [SCT: N/A]
  Service Carving Results:
     Forwarders     : 10
     Elected        : 10
            EVI E   :     1001,    1002,    1003,    1004,    1005,    1006
            EVI E   :     1007,    1008,    1009,    1010
     Not Elected    : 0
  EVPN-VPWS Service Carving Results:
     Primary        : 0
     Backup         : 0
     Non-DF         : 0
  MAC Flushing mode : STP-TCN
  Peering timer     : 0 sec [not running]
  Recovery timer    : 0 sec [not running]
  Carving timer     : 0 sec [not running]
  Revert timer      : 0 sec [not running]
  HRW Reset timer   : 5 sec [not running]
  Local SHG label   : None
  Remote SHG labels : 0
  Access signal mode: Unsupported
```

/*Verify EVPN IGMP Snooping*/

```
RP/0/RSP0/CPU0:tb8-r3-AVA2#show evpn igmp
Mon Nov  6 11:18:19.497 UTC

EVI   Ethernet Segment         (S,G)                                        Source
                Type
----- ------------------------ -------------------------------------------
----------------------------- ------
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::1)                  Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::1)                  Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::1:2)                Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::1:2)                Bundle-Ether2.1001
```

```
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:1::1)             Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:1::1)            Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:1::1)            Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (2205:101::23,ff03:13:1::1)              Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (2205:101::23,ff03:13:1::1)              Bundle-Ether2.1001
            JOIN
1002  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::2)                  Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::2)                  Bundle-Ether2.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:2::1)            Bundle-Ether2.1002
            JOIN
1002  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:2::1)           Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:2::1)           Bundle-Ether2.1002
            JOIN
1002  0000.0100.ac00.0001.0a00 (2205:101::23,ff03:14:1::1)             Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (2205:102::441,ff03:14:1::1)            Bundle-Ether2.1002
            JOIN
1003  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::3)                  Bundle-Ether1.1003
            JOIN
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::3)                  Bundle-Ether2.1003
            JOIN
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:3::1)            Bundle-Ether2.1003
            JOIN
1003  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:3::1)           Bundle-Ether1.1003
            JOIN
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:3::1)           Bundle-Ether2.1003
            JOIN
1004  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::4)                  Bundle-Ether1.1004
            JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::4)                  Bundle-Ether2.1004
            JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:4::1)            Bundle-Ether2.1004
            JOIN
1004  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:4::1)           Bundle-Ether1.1004
            JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:4::1)           Bundle-Ether2.1004
            JOIN
1005  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::5)                  Bundle-Ether1.1005
            JOIN
1005   0011.0200.ac00.0001.0a00 (1108:101::100,ff03::5)                    7.7.7.7
               JOIN
1005  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:5::1)            Bundle-Ether2.1005
            JOIN
1005  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:5::1)           Bundle-Ether1.1005
            JOIN
1005  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:5::1)           Bundle-Ether2.1005
            JOIN
1006  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::6)                  Bundle-Ether1.1006
            JOIN
1006  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::6)                  Bundle-Ether2.1006
            JOIN
1006  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:6::1)            Bundle-Ether2.1006
            JOIN
1006  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:6::1)           Bundle-Ether1.1006
```

```
            JOIN
1006  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:6::1)              7.7.7.7
              JOIN
1007  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::7)                  Bundle-Ether1.1007
            JOIN
1007  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::7)                  Bundle-Ether2.1007
            JOIN
1007  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:7::1)              7.7.7.7
              JOIN
1007  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:7::1)            Bundle-Ether1.1007
            JOIN
1007  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:7::1)            Bundle-Ether2.1007
            JOIN
1008  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::8)                  Bundle-Ether1.1008
            JOIN
1008  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::8)                     7.7.7.7
              JOIN
1008  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:8::1)            Bundle-Ether2.1008
            JOIN
1008  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:8::1)            Bundle-Ether1.1008
            JOIN
1008  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:8::1)            Bundle-Ether2.1008
            JOIN
1009  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::9)                  Bundle-Ether1.1009
            JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::9)                  Bundle-Ether2.1009
            JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:9::1)            Bundle-Ether2.1009
            JOIN
1009  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:9::1)            Bundle-Ether1.1009
            JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:9::1)            Bundle-Ether2.1009
            JOIN
1010  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::a)                  Bundle-Ether1.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::a)                  Bundle-Ether2.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:a::1)              7.7.7.7
              JOIN
1010  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:a::1)            Bundle-Ether1.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:a::1)            Bundle-Ether2.1010
            JOIN
```

Configuring MLD Snooping on Ethernet Bundles

1. This example assumes that the front-ends of the bundles are preconfigured. For example, a bundle configuration might consist of three switch interfaces, as follows:

```
    interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
    interface GigabitEthernet0/0/0/3
```

```
   channel-group 1 mode on
!
```

2. Configure two MLD snooping profiles.

```
mld snooping profile bridge_profile
!
mld snooping profile port_profile
   mrouter
!
```

3. Configure interfaces as bundle member links.

```
interface GigabitEthernet0/0/0/0
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/1
  bundle id 1 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/2
  bundle id 2 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/3
  bundle id 2 mode on
  negotiation auto
!
```

4. Configure the bundle interfaces for L2 transport.

```
interface Bundle-Ether 1
          l2transport
          !
       !
       interface Bundle-Ether 2
          l2transport
          !
       !
```

5. Add the interfaces to the bridge domain and attach MLD snooping profiles.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     mld snooping profile bridge_profile
     interface bundle-Ether 1
       mld snooping profile port_profile
     interface bundle-Ether 2
     !
  !
!
```

6. Verify the configured bridge ports.

```
show mld snooping port
```