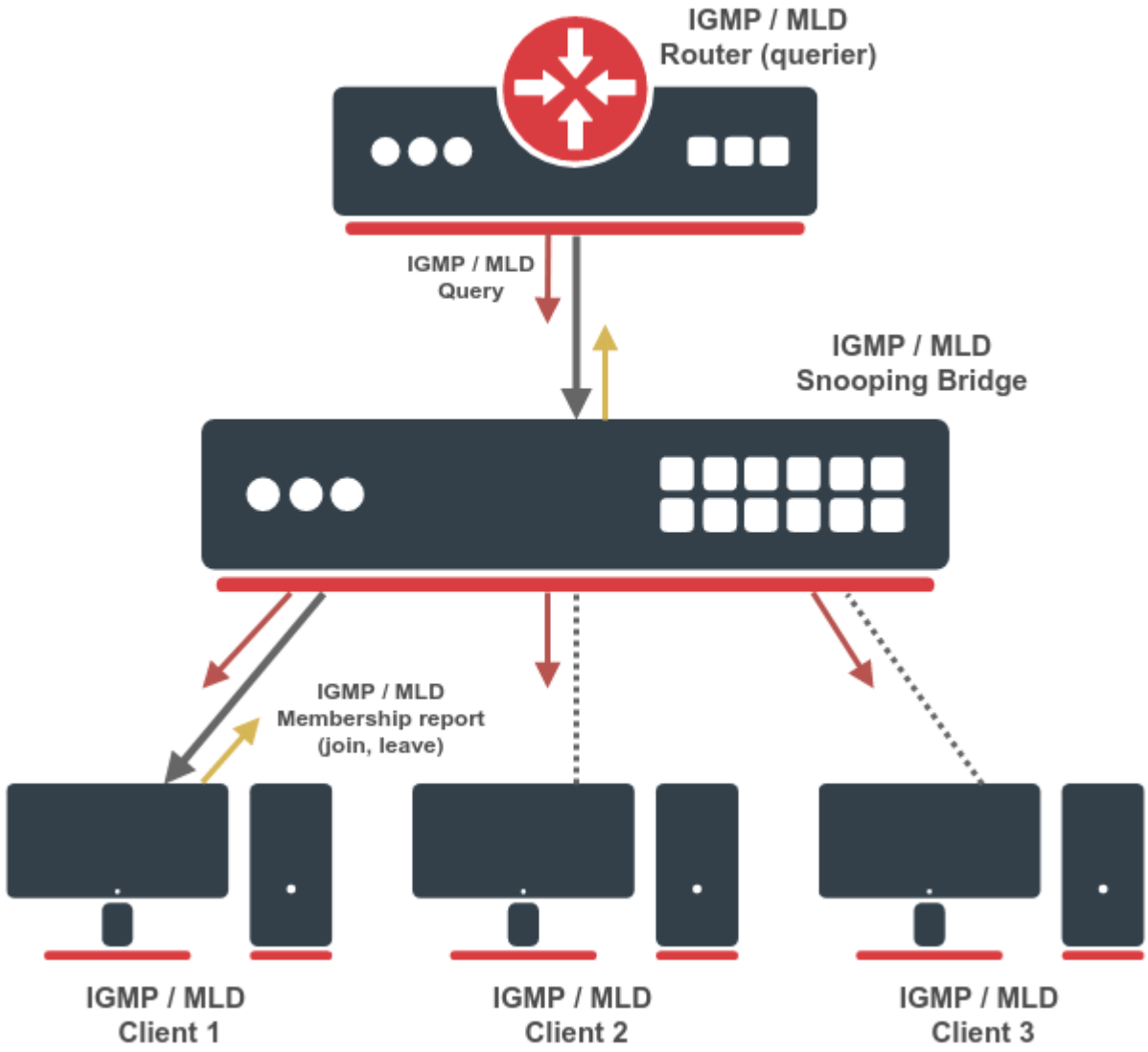
Bridge IGMP/MLD snooping

Introduction

IGMP (Internet Group Management Protocol) and MLD (Multicast Listener Discovery) snooping allow the bridge to listen to IGMP/MLD communication and make forwarding decisions for multicast traffic based on the received information. By default, bridges are flooding multicast traffic to all bridge ports just like broadcast traffic, which might not always be the best scenario (e.g. for multicast video traffic or SDVoE applications). The IGMP/MLD snooping tries to solve the problem by forwarding the multicast traffic only to ports where clients are subscribed to, see an IGMP/MLD network concept below. RouterOS bridge is able to process IGMP v1/v2/v3 and MLD v1/v2 packets. The implemented bridge IGMP/MLD snooping is based on RFC4541, and IGMP/MLD protocols are specified on RFC1112 (IGMPv1) RFC2236 (IGMPv2), RFC3376 (IGMPv3), RFC2710 (MLDv1), RFC3810 (MLDv2).

Source-specific multicast forwarding is not supported for IGMP v3 and MLD v2.



The bridge will process the IGMP/MLD messages only when `igmp-snooping` is enabled. Additionally, the bridge should have an active IPv6 address in order to process MLD packets. At first, the bridge does not restrict the multicast traffic and all multicast packets get flooded. Once IGMP/MLD querier is detected by receiving an IGMP/MLD query message (the query message can be received by an external multicast router or locally by bridge interface with enabled `multicast-querier`), only then the bridge will start to restrict unknown IP multicast traffic and forward the known multicast from the multicast database (MDB). The IGMP and MLD querier detection is independent, which means that detecting only IGMP querier will not affect IPv6 multicast forwarding and vice versa. The querier detection also does not restrict the forwarding of non-IP and link-local multicast groups, like 224.0.0.0/24 and ff02::1.

CRS3xx series devices with Marvell-98DX3236, Marvell-98DX224S or Marvell-98DX226S switch chip are not able to distinguish non-IP/IPv4 /IPv6 multicast packets once IGMP or MLD querier is detected. It means that the switch will stop forwarding all unknown non-IP/IPv4/IPv6 multicast traffic when the querier is detected. This does not apply to certain link-local multicast address ranges, like 224.0.0.0/24 or ff02::1.

Configuration options

This section describes the IGMP and MLD snooping bridge configuration options.

Sub-menu: `/interface bridge`

Property	Description

igmp-snooping (*yes* | *no*; Default: **no**)	Enables IGMP and MLD snooping.
igmp-version (*2* | *3*; Default: **2**)	Selects the IGMP version in which IGMP membership queries will be generated when the bridge interface is acting as an IGMP querier. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.
last-member-interval (*time*; Default: **1s**)	When the last client on the bridge port unsubscribes to a multicast group and the bridge is acting as an active querier, the bridge will send group-specific IGMP/MLD query, to make sure that no other client is still subscribed. The setting changes the response time for these queries. In case no membership reports are received in a certain time period (`last-member-interval` * `last-member-query-count`), the multicast group is removed from the multicast database (MDB). If the bridge port is configured with fast-leave, the multicast group is removed right away without sending any queries. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.
last-member-query-count (*integer: 0.. 4294967295*; Default: **2**)	How many times should `last-member-interval` pass until the IGMP/MLD snooping bridge stops forwarding a certain multicast stream. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.
membership-interval (*time*; Default: **4m20s**)	The amount of time after an entry in the Multicast Database (MDB) is removed if no IGMP/MLD membership reports are received on a bridge port. This property only has an effect when `igmp-snooping` is set to `yes`.
mld-version (*1* | *2*; Default: **1**)	Selects the MLD version in which MLD membership queries will be generated, when the bridge interface is acting as an MLD querier. This property only has an effect when the bridge has an active IPv6 address, `igmp-snooping` and `multicast-querier` is set to `yes`.
multicast-querier (*yes* | *no*; Default: **no**)	Multicast querier generates periodic IGMP/MLD general membership queries to which all IGMP/MLD capable devices respond with an IGMP/MLD membership report, usually a PIM (multicast) router or IGMP proxy generates these queries. By using this property you can make an IGMP/MLD snooping enabled bridge to generate IGMP/MLD general membership queries. This property should be used whenever there is no active querier (PIM router or IGMP proxy) in a Layer2 network. Without a multicast querier in a Layer2 network, the Multicast Database (MDB) is not being updated, the learned entries will timeout and IGMP/MLD snooping will not function properly. Only untagged IGMP/MLD general membership queries are generated, IGMP queries are sent with IPv4 0.0.0.0 source address, MLD queries are sent with IPv6 link-local address of the bridge interface. The bridge will not send queries if an external IGMP /MLD querier is detected (see the monitoring values `igmp-querier` and `mld-querier`). This property only has an effect when `igmp-snooping` is set to `yes`.
multicast-router (*disabled* | *permanent* | *temporary-query*; Default: **temporary-query**)	A multicast router port is a port where a multicast router or querier is connected. On this port, unregistered multicast streams and IGMP/MLD membership reports will be sent. This setting changes the state of the multicast router for a bridge interface itself. This property can be used to send IGMP/MLD membership reports and multicast traffic to the bridge interface for further multicast routing or proxying. This property only has an effect when `igmp-snooping` is set to `yes`. • `disabled` - disabled multicast router state on the bridge interface. Unregistered multicast streams and IGMP/MLD membership reports are not sent to the bridge interface regardless of what is configured on the bridge interface. • `permanent` - enabled multicast router state on the bridge interface. Unregistered multicast streams and IGMP/MLD membership reports are sent to the bridge interface itself regardless of what is configured on the bridge interface. • `temporary-query` - automatically detect multicast router state on the bridge interface using IGMP/MLD queries.
querier-interval (*time* ; Default: **4m15s**)	Changes the timeout period for detected querier and multicast-router ports. This property only has an effect when `igmp-snooping` is set to `yes`.
query-interval (*time*; Default: **2m5s**)	Changes the interval on how often IGMP/MLD general membership queries are sent out when the bridge interface is acting as an IGMP/MLD querier. The interval takes place when the last startup query is sent. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.
query-response-interval (*time*; Default: **10s**)	The setting changes the response time for general IGMP/MLD queries when the bridge is acting as an IGMP/MLD querier. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.

startup-query-count (*integer: 0..4294967295*; Default: **2**)	Specifies how many times general IGMP/MLD queries must be sent when bridge interface is enabled or active querier timeouts. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.
startup-query-interval (*time*; Default: **31s250ms**)	Specifies the interval between startup general IGMP/MLD queries. This property only has an effect when `igmp-snooping` and `multicast-querier` is set to `yes`.

Sub-menu: `/interface bridge` port

Property	Description
fast-leave (*yes* | *no*; Default: **no**)	Enables IGMP/MLD fast leave feature on the bridge port. The bridge will stop forwarding multicast traffic to a bridge port when an IGMP/MLD leave message is received. This property only has an effect when `igmp-snooping` is set to `yes`.
multicast-router (*disabled* | *permanent* | *temporary-query*; Default: **temporary-query**)	A multicast router port is a port where a multicast router or querier is connected. On this port, unregistered multicast streams and IGMP/MLD membership reports will be sent. This setting changes the state of the multicast router for bridge ports. This property can be used to send IGMP/MLD membership reports and multicast streams to certain bridge ports for further multicast routing or proxying. This property only has an effect when `igmp-snooping` is set to `yes`. • `disabled` - disabled multicast router state on the bridge port. Unregistered multicast streams and IGMP/MLD membership reports are not sent to the bridge port regardless of what is connected to it. • `permanent` - enabled multicast router state on the bridge port. Unregistered multicast and IGMP/MLD membership reports are sent to the bridge port regardless of what is connected to it. • `temporary-query` - automatically detect multicast router state on the bridge port using IGMP/MLD queries.
unknown-multicast-flood (*yes* | *no*; Default: **yes**)	Changes the multicast flood option on bridge port, only controls the egress traffic. When enabled, the bridge allows flooding multicast packets to the specified bridge port, but when disabled, the bridge restricts multicast traffic from being flooded to the specified bridge port. The setting affects all multicast traffic, this includes non-IP, IPv4, IPv6 and the link-local multicast ranges (e.g. 224.0.0.0/24 and ff02::1). Note that when `igmp-snooping` is enabled and IGMP/MLD querier is detected, the bridge will automatically restrict unknown IP multicast from being flooded, so the setting is not mandatory for IGMP/MLD snooping setups. When using this setting together with `igmp-snooping`, the only multicast traffic that is allowed on the bridge port is the known multicast from the MDB table.

Sub-menu: `/interface bridge mdb`

Property	Description
bridge (*name*; Default:)	The bridge interface to which the MDB entry is going to be assigned.
disabled (*yes* | *no*; Default: **no**)	Disables or enables static MDB entry.
group (*ipv4* | *ipv6 address*; Default:)	The IPv4 or IPv6 multicast address. Static entries for link-local multicast groups 224.0.0.0/24 and ff02::1 cannot be created, as these packets are always flooded on all ports and VLANs.
ports (*name*; Default:)	The list of bridge ports to which the multicast group will be forwarded.
vid (*integer: 1..4094*; Default:)	The VLAN ID on which the MDB entry will be created, only applies when `vlan-filtering` is enabled. When VLAN ID is not specified, the entry will work in shared-VLAN mode and dynamically apply on all defined VLAN IDs for particular ports.

Monitoring and troubleshooting

This section describes the IGMP/MLD snooping bridge monitoring and troubleshooting options.

To monitor learned multicast database (MDB) entries, use the `print` command.

Sub-menu: `/interface bridge mdb`

Property	Description
bridge (*read-only: name*)	Shows the bridge interface the entry belongs to.
group (*read-only: ipv4 | ipv6 address*)	Shows a multicast group address.
on-ports (*read-only: name*)	Shows the bridge ports which are subscribed to the certain multicast group.
vid (*read-only: integer*)	Shows the VLAN ID for the multicast group, only applies when `vlan-filtering` is enabled.

```
[admin@MikroTik] /interface bridge mdb print
Flags: D - DYNAMIC
Columns: GROUP, VID, ON-PORTS, BRIDGE
 #   GROUP              VID  ON-PORTS  BRIDGE
 0 D ff02::2              1  bridge1   bridge1
 1 D ff02::6a             1  bridge1   bridge1
 2 D ff02::1:ff00:0       1  bridge1   bridge1
 3 D ff02::1:ff01:6a43    1  bridge1   bridge1
 4 D 229.1.1.1           10  ether2    bridge1
 5 D 229.2.2.2           10  ether3    bridge1
                             ether2
 6 D ff02::2             10  ether5    bridge1
                             ether3
                             ether2
                             ether4
```

To monitor the current status of a bridge interface, use the `monitor` command.

Sub-menu: `/interface bridge`

Property	Description
igmp-querier (*none | interface & IPv4 address*)	Shows a bridge port and source IP address from the detected IGMP querier. Only shows detected external IGMP querier, local bridge IGMP querier (including IGMP proxy and PIM) will not be displayed. Monitoring value appears only when `igmp-snooping` is enabled.
mld-querier (*none | interface & IPv6 address*)	Shows a bridge port and source IPv6 address from the detected MLD querier. Only shows detected external MLD querier, local bridge MLD querier will not be displayed. Monitoring value appears only when `igmp-snooping` is enabled and the bridge has an active IPv6 address.
multicast-router (*yes | no*)	Shows if a multicast router is detected on the bridge interface. Monitoring value appears only when `igmp-snooping` is enabled.

```
[admin@MikroTik] /interface bridge monitor bridge1
                  state: enabled
    current-mac-address: 64:D1:54:C7:3A:59
            root-bridge: yes
         root-bridge-id: 0x8000.64:D1:54:C7:3A:59
         root-path-cost: 0
              root-port: none
             port-count: 3
  designated-port-count: 3
          fast-forward: no
      multicast-router: no
          igmp-querier: ether2 192.168.10.10
           mld-querier: ether2 fe80::e68d:8cff:fe39:3824
```

To monitor the current status of bridge ports, use the `monitor` command.

Sub-menu: `/interface bridge port`

Property	Description
multicast-router (*yes | no*)	Shows if a multicast router is detected on the port. Monitoring value appears only when `igmp-snooping` is enabled.

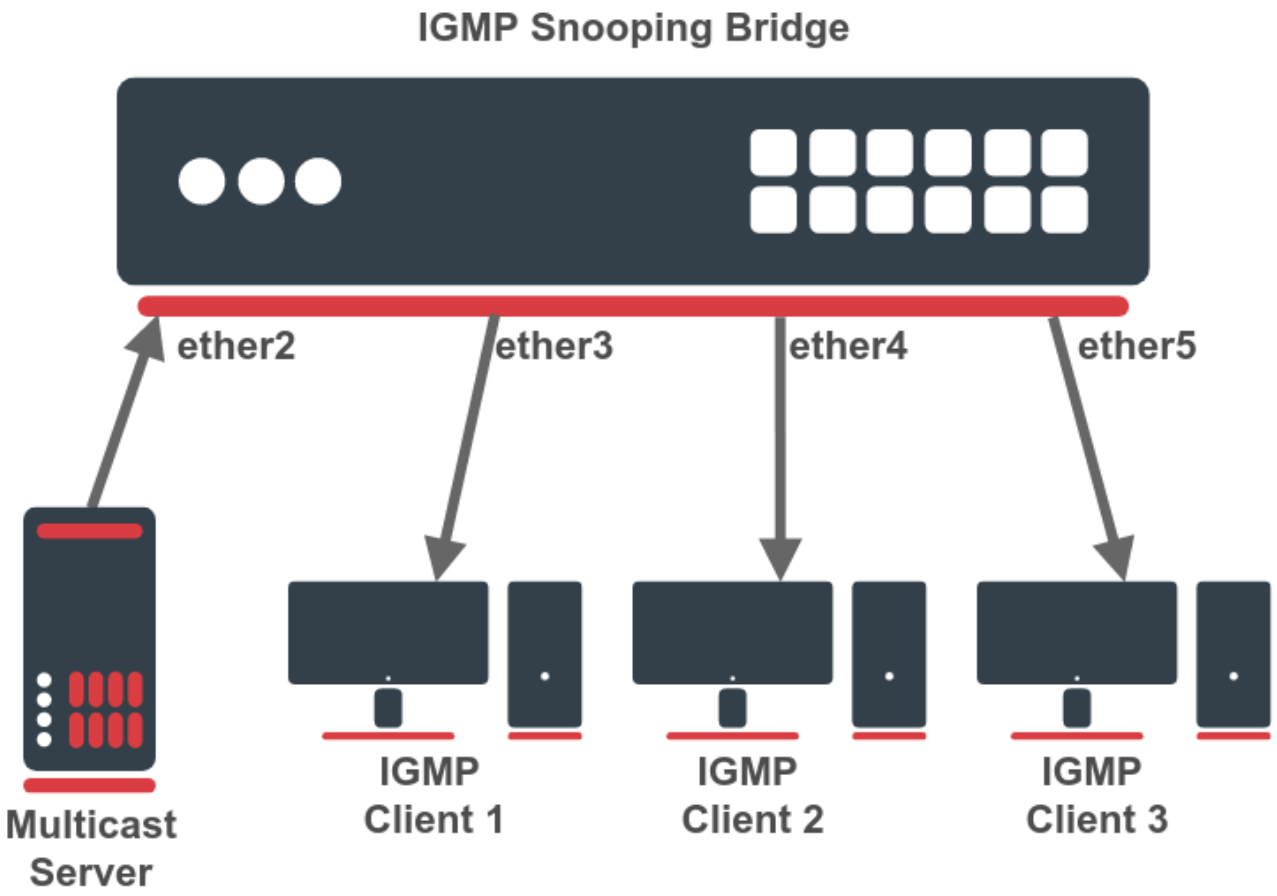
```
[admin@MikroTik] > /interface bridge port monitor [find]
              interface: ether2          ether3          ether4
                 status: in-bridge       in-bridge       in-bridge
            port-number: 1               2               3
                   role: designated-port designated-port designated-port
              edge-port: no              yes             yes
    edge-port-discovery: yes             yes             yes
    point-to-point-port: yes             yes             yes
           external-fdb: no              no              no
           sending-rstp: yes             yes             yes
               learning: yes             yes             yes
             forwarding: yes             yes             yes
       multicast-router: yes             no              no
       hw-offload-group: switch1         switch1         switch1
```

Configuration examples

Below are described the most common configuration examples. Some examples are using a bridge with VLAN filtering, so make sure to understand the filtering principles first - bridge VLAN filtering, bridge VLAN table.

Basic IGMP snooping configuration

The first example consists only of a single IGMP snooping bridge, a single multicast source device, and a couple of multicast client devices. See a network scheme below.



First, create a bridge interface with enabled IGMP snooping. In this example, there is no active IGMP querier (no multicast router or proxy), so a local IGMP querier must be enabled on the same bridge. This can be done with a `multicast-querier` setting. If there is no active IGMP querier in the LAN, the unregistered IP multicast will be flooded and multicast entries will always timeout from the multicast database.

```
/interface bridge
add igmp-snooping=yes multicast-querier=yes name=bridge1
```

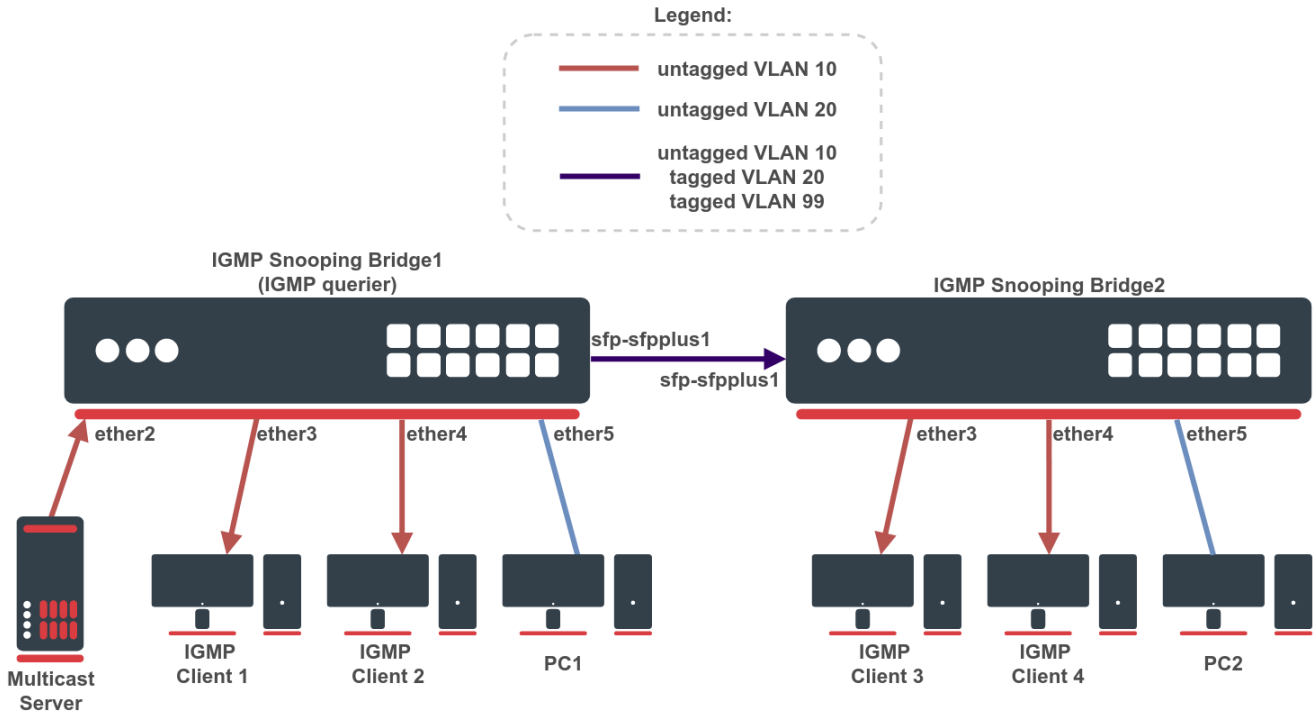
Then add necessary interfaces as bridge ports.

```
/interface bridge port
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
```

The basic IGMP snooping configuration is finished. Use "`/interface bridge mdb print`" command to monitor the active multicast groups. If necessary, you can configure an IP address and DHCP server on the same bridge interface.

IGMP snooping configuration with VLANs

The second example adds some complexity. There are two IGMP snooping bridges and we need to isolate the multicast traffic on a different VLAN. See a network scheme below.



First, create a bridge on both devices and add needed interfaces as bridge ports. To change untagged VLAN for a bridge port, use the `pvid` setting. The Bridge1 will be acting as an IGMP querier. Below are configuration commands for the Bridge1:

```
/interface bridge
add igmp-snooping=yes multicast-querier=yes name=bridge1
/interface bridge port
add bridge=bridge1 interface=ether2 pvid=10
add bridge=bridge1 interface=ether3 pvid=10
add bridge=bridge1 interface=ether4 pvid=10
add bridge=bridge1 interface=ether5 pvid=20
add bridge=bridge1 interface=sfp-sfpplus1 pvid=10
```

And for the Bridge2:

```
/interface bridge
add igmp-snooping=yes name=bridge1
/interface bridge port
add bridge=bridge1 interface=ether3 pvid=10
add bridge=bridge1 interface=ether4 pvid=10
add bridge=bridge1 interface=ether5 pvid=20
add bridge=bridge1 interface=sfp-sfpplus1 pvid=10
```

⚠ Bridge IGMP querier implementation can only send untagged IGMP queries. In case tagged IGMP queries should be sent or IGMP queries should be generated in multiple VLANs, it is possible to install a multicast package, add a VLAN interface and configure a PIM interface on VLAN. The PIM interface can be used as an IGMP querier.

Make sure to configure management access for devices. It is essential when configuring a bridge with VLAN filtering. In this example, a VLAN 99 interface with an IP address is added to the bridge. This VLAN will be allowed on the tagged sfp-sfpplus1 port. Below are configuration commands for the Bridge1:

```
/interface vlan
add interface=bridge1 name=MGMT vlan-id=99
/ip address
add address=192.168.99.1/24 interface=MGMT network=192.168.99.0
/interface bridge vlan
add bridge=bridge1 tagged=bridge1,sfp-sfpplus1 vlan-ids=99
```

And for the Bridge2:

```
/interface vlan
add interface=bridge1 name=MGMT vlan-id=99
/ip address
add address=192.168.99.2/24 interface=MGMT network=192.168.99.0
/interface bridge vlan
add bridge=bridge1 tagged=bridge1,sfp-sfpplus1 vlan-ids=99
```

Add bridge VLAN entries and specify tagged and untagged ports. The VLAN 99 entry was already created when configuring management access, only VLAN 10 and VLAN 20 should be added now. Below are configuration commands for the Bridge1:

```
/interface bridge vlan
add bridge=bridge1 untagged=ether2,ether3,ether4,sfp-sfpplus1 vlan-ids=10
add bridge=bridge1 tagged=sfp-sfpplus1 untagged=ether5 vlan-ids=20
```

And for the Bridge2:

```
/interface bridge vlan
add bridge=bridge1 untagged=ether3,ether4,sfp-sfpplus1 vlan-ids=10
add bridge=bridge1 tagged=sfp-sfpplus1 untagged=ether5 vlan-ids=20
```

Last, enable VLAN filtering. Below is the configuration command for Bridge1 and Bridge2:

```
/interface bridge set [find name=bridge1] vlan-filtering=yes
```

At this point, VLANs and IGMP snooping are configured and devices should be able to communicate through ports. However, it is recommended to go even a step further and apply some additional filtering options. Enable `ingress-filtering` and `frame-types` on bridge ports. Below are configuration commands for the Bridge1:

```
/interface bridge port
set [find interface=ether2] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=ether3] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=ether4] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=ether5] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=sfp-sfpplus1] ingress-filtering=yes
```

And for the Bridge2:

```
/interface bridge port
set [find interface=ether3] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=ether4] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=ether5] ingress-filtering=yes frame-types=admit-only-untagged-and-priority-tagged
set [find interface=sfp-sfpplus1] ingress-filtering=yes
```

Static MDB entries

Since RouterOS version 7.7, it is possible to create static MDB entries for IPv4 and IPv6 multicast groups. For example, to create a static MDB entry for multicast group 229.10.10.10 on ports ether2 and ether3 on VLAN 10, use the command below:

```
/interface bridge mdb
add bridge=bridge1 group=229.10.10.10 ports=ether2,ether3 vid=10
```

Verify the results with the `print` command:

```
[admin@MikroTik] > /interface bridge mdb print where group=229.10.10.10
Columns: GROUP, VID, ON-PORTS, BRIDGE
 # GROUP         VID  ON-PORTS  BRIDGE
12 229.10.10.10   10  ether2    bridge1
                      ether3
```

In case a certain IPv6 multicast group does not need to be snooped and it is desired to be flooded on all ports and VLANs, it is possible to create a static MDB entry on all VLANs and ports, including the bridge interface itself. Use the command below to create a static MDB entry for multicast group ff02::2 on all VLANs and ports (modify the `ports` setting for your particular setup):

```
/interface bridge mdb
add bridge=bridge1 group=ff02::2 ports=bridge1,ether2,ether3,ether4,ether5

[admin@MikroTik] > /interface bridge mdb print where group=ff02::2
Flags: D - DYNAMIC
Columns: GROUP, VID, ON-PORTS, BRIDGE
 #   GROUP    VID  ON-PORTS  BRIDGE
 0   ff02::2                bridge1
15 D ff02::2    1  bridge1   bridge1
16 D ff02::2   10  bridge1   bridge1
                   ether2
                   ether3
                   ether4
                   ether5
17 D ff02::2   20  bridge1   bridge1
                   ether2
                   ether3
18 D ff02::2   30  bridge1   bridge1
                   ether2
                   ether3
```