

Chapter 27

IP Multicasting

Introduction	27-4
References	27-4
IP Multicast Routing	27-5
Interoperability between Multicast Routing Protocols	27-7
Distance Vector Multicast Routing Protocol (DVMRP)	27-7
Configuring DVMRP	27-8
Protocol Independent Multicast (PIM)	27-9
PIM Dense Mode	27-10
PIM Sparse Mode	27-12
Internet Group Management Protocol (IGMP)	27-20
Configuring IGMP	27-20
Static IGMP	27-21
IGMP Proxy	27-23
IGMP Snooping	27-25
IGMP Filtering	27-29
IGMP Throttling	27-31
Multicast Switching	27-32
Multicast Switching Emulation	27-32
Multicast VLAN Registration (MVR)	27-33
Dynamic MVR	27-33
Compatible MVR	27-33
Immediate Leave	27-33
Configuring MVR	27-34
Configuration Examples	27-35
Static IGMP	27-35
Multicasting using DVMRP	27-35
Protocol Independent Multicast (PIM)	27-38
Command Reference	27-47
add dvmrp interface	27-48
add igmp filter	27-49
add igmpsnooping routeraddress	27-50
add igmpsnooping vlan	27-51
add ip igmp destination	27-52
add ip mvr	27-53
add pim bsrcandidate	27-54
add pim interface	27-55
add pim rpcandidate	27-57
create igmp filter	27-59
create ip igmp destination	27-60
create ip mvr	27-61
delete dvmrp interface	27-62

delete igmp filter	27-63
delete igmpsnooping routeraddress	27-63
delete igmpsnooping vlan	27-64
delete ip igmp destination	27-65
delete ip mvr	27-66
delete pim bsrcandidate	27-66
delete pim interface	27-67
delete pim rpcandidate	27-68
destroy igmp filter	27-69
destroy ip igmp destination	27-69
destroy ip mvr	27-70
disable dvmrp	27-70
disable dvmrp debug	27-71
disable igmpsnooping	27-72
disable ip igmp	27-72
disable ip igmp allgroup	27-73
disable ip igmp debug	27-73
disable ip igmp interface	27-74
disable ip multicastswitching	27-74
disable ip mvr	27-75
disable ip mvr debug	27-75
disable pim	27-75
disable pim bsmsecuritycheck	27-76
disable pim debug	27-76
enable dvmrp	27-77
enable dvmrp debug	27-78
enable igmpsnooping	27-79
enable ip igmp	27-79
enable ip igmp allgroup	27-80
enable ip igmp debug	27-80
enable ip igmp interface	27-81
enable ip multicastswitching	27-81
enable ip mvr	27-82
enable ip mvr debug	27-82
enable pim	27-83
enable pim bsmsecuritycheck	27-83
enable pim debug	27-84
purge dvmrp	27-84
purge pim	27-85
reset dvmrp interface	27-85
reset pim interface	27-86
set dvmrp interface	27-87
set igmp filter	27-88
set igmpsnooping vlan	27-89
set igmpsnooping routermode	27-90
set ip igmp	27-91
set ip igmp interface	27-92
set ip mvr	27-93
set pim	27-94
set pim log	27-95
set pim bsrcandidate	27-96
set pim interface	27-97
set pim rpcandidate	27-99
show dvmrp	27-100
show dvmrp counters	27-101
show dvmrp debug	27-102
show dvmrp forwarding	27-103
show dvmrp interface	27-104
show dvmrp neighbour	27-105

show dvmrp route	27-106
show igmp filter	27-107
show igmpsnooping	27-109
show igmpsnooping counter	27-111
show igmpsnooping routeraddress	27-113
show ip igmp	27-114
show ip igmp counter	27-118
show ip igmp debug	27-121
show ip mvr	27-122
show ip mvr counter	27-123
show pim	27-124
show pim bsr candidate	27-125
show pim config	27-126
show pim counters	27-127
show pim debug	27-131
show pim interface	27-132
show pim neighbour	27-134
show pim route	27-135
show pim rpcandidate	27-141
show pim rpset	27-142
show pim staterefresh	27-144
show pim timer	27-145

Introduction

This chapter describes IP multicasting and support for multicasting on the switch.

Most IP packets are sent to a single host—unicast transmission—or to all hosts on a network or subnetwork – broadcast transmission. Multicasting is an alternative where packets are sent to a group of hosts simultaneously on a network or sub-network. Multicasting is also known as *group transmission*.

A multicast environment consists of senders (IP hosts), routers and switches (intermediate forwarding devices) and receivers (IP hosts). A multicast group has a class D IP address (the first number in the IP address – the top four bits – are 1110). Any IP host can send packets to a multicast group, in the same way that they send unicast packets to a particular IP host, by specifying its IP address. A host need not belong to a multicast group in order to send to it. Packets sent to a group address are only received by members of the group.

The switch uses the Internet Group Management Protocol (IGMP) to track multicast group membership, and one or more of the following protocols to route multicast traffic:

- Distance Vector Multicast Routing Protocol (DVMRP)
- Protocol Independent Multicast Sparse Mode (PIM-SM)
- Protocol Independent Multicast Dense Mode (PIM-DM)

DVMRP, PIM Sparse Mode, and PIM Dense Mode must be enabled with a special feature licence. To obtain one, contact an Allied Telesis authorised distributor or reseller.

For simple networks, the switch can be configured to use IGMP proxy instead of DVMRP or PIM, or to emulate multicast switching between VLANs. For networks with an Ethernet ring topology, the switch can use Multicast VLAN Registration (MVR) to route between VLANs instead of DVMRP or PIM.

The multicast routing protocols described in this chapter are dynamic and respond to changes in multicast group membership. Interfaces on the switch can instead be configured statically to send and/or receive multicast packets. Static multicasting is described in [“Static Multicast Forwarding” on page 23-46 of Chapter 23, Internet Protocol \(IP\)](#).

References

Internet Draft *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*, Internet Engineering Task Force, PIM WG, 1 March 2002 (draft-ietf-pim-sm-v2-new-05).

Internet Draft *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)*, Internet Engineering Task Force, PIM WG, 15 February 2002 (draft-ietf-pim-dm-new-v2-01).

RFC 2236, *Internet Group Management Protocol, version 2*.

Internet Draft *Distance Vector Multicast Routing Protocol Version 3*, Internet Engineering Task Force, August 2000 (draft-ietf-idmr-dvmrp-v3-10).

RFC 2715, *Interoperability Rules for Multicast Routing Protocols*.

IP Multicast Routing

For multicasting to succeed, the switch needs to know which of its interfaces are directly connected to members of each multicast group. To establish this, the switch uses IGMP for multicast group management (see [“Internet Group Management Protocol \(IGMP\)” on page 27-20](#)).

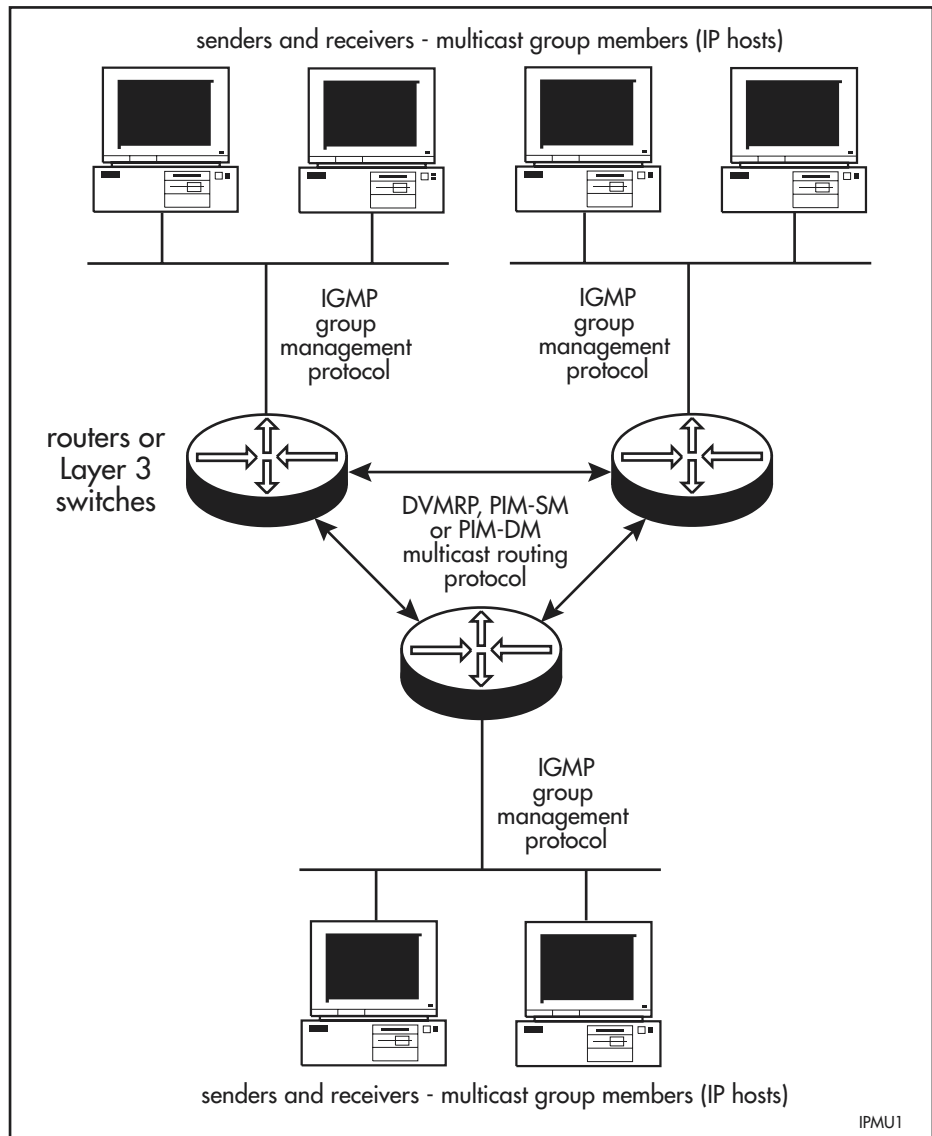
The switch must also know where to send multicast traffic. The switch maintains a routing table for multicast traffic with the following:

- DVMRP (see [“Distance Vector Multicast Routing Protocol \(DVMRP\)” on page 27-7](#))
- PIM-Sparse Mode or PIM-Dense Mode (see [“Protocol Independent Multicast \(PIM\)” on page 27-9](#))

IGMP and one of the multicast routing protocols must be configured before the switch can forward multicast packets.

The relationships between IP hosts, routers, and multicasting protocols are shown in the following figure.

Figure 27-1: Multicast environment



When the switch receives a packet addressed to a multicast group, it forwards it to the interfaces that have group members connected to them, according to IGMP, and out other interfaces specified by the multicast routing protocol. Membership in a multicast group is dynamic; hosts can join and leave at any time. Multicast groups can be long or short lived, and can have relatively stable or constantly changing membership. There is no limit on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

When the switch finds out from IGMP that a new host has joined a multicast group on one of its interfaces, the switch needs to receive the multicast traffic for this group, so that it can forward it to the host. The switch uses the multicast routing protocol (DVMRP, PIM-SM or PIM-DM) to notify routers closer to the sender (upstream) to forward it traffic for the group. Routers running a multicast routing protocol, such as Protocol-Independent Multicast (PIM), maintain forwarding tables to forward multicast packets. DVMRP, PIM Sparse Mode and PIM Dense Mode share a multicast forwarding table.

Interoperability between Multicast Routing Protocols

The switch can be configured as a Multicast Border Router (MBR), as specified in RFC 2715, *Interoperability Rules for Multicast Routing Protocols*. A Multicast Border Router forms the border between two or more multicasting domains that are running different multicast routing protocols (DVMRP, PIM-SM or PIM-DM). The MBR forwards multicast packets across the different domains so that receivers in one domain can receive packets from sources in another domain. Therefore different interfaces on the switch can be configured as DVMRP, PIM-SM or PIM-DM interfaces.

The switch treats sources that are reached via another multicasting domain as if they were directly connected sources.

Distance Vector Multicast Routing Protocol (DVMRP)

DVMRP is an Internet routing protocol that provides an efficient mechanism for connectionless datagram delivery to a group of hosts across an internetwork. It is a distributed protocol that dynamically generates IP Multicast delivery trees by using a technique called Reverse Path Multicasting (RPM). The switch supports DVMRP version 3 as specified in Internet Draft *Distance Vector Multicast Routing Protocol Version 3, September 1999* (draft-ietf-idmr-dvmrp-v3-09).

DVMRP must be enabled with a special feature licence. To obtain a special feature licence contact an Allied Telesis authorised distributor or reseller.

DVMRP maintains its own multicast routing table, and uses this to maintain the shared multicast routing table. It maintains a list of its DVMRP neighbours, to which it sends and receives DVMRP messages.

DVMRP uses a distance vector distributed routing algorithm to build per-source-group multicast delivery trees. Datagrams follow multicast delivery trees from a source to all members of a multicast group, replicating the packet only at necessary branches in the delivery tree. DVMRP calculates and updates the trees dynamically to track the membership of individual groups. When a datagram arrives on an interface, the reverse path to the source of the datagram is determined by examining a DVMRP routing table of known source networks. If the datagram arrives on an interface that would be used to transmit the datagram back to the source, then it is forwarded to the appropriate list of downstream interfaces. Otherwise, it is not on the optimal delivery tree and is discarded. In this way DVMRP filters out duplicate packets arising from loops in the network topology. DVMRP automatically prunes back the delivery tree for each source as group membership changes or switches determine that no group members are present. This keeps the delivery trees to the minimum branches necessary to reach all of the group members. New sections of the tree are added dynamically as new members join the multicast group by grafting the new sections onto the delivery trees.

The switch sends *prune* messages to the DVMRP router (neighbour) next closest to the multicast source when it thinks that it has no group members downstream dependent on it for this multicast group's traffic. If it later finds out that a new group member has joined downstream, it sends a *graft* message

to the upstream router to tell it that it needs to receive this multicast traffic again. These join and graft messages are propagated through the DVMRP network upstream towards the source of the multicast traffic as far as they are needed to keep the traffic flowing as far as it needs to, and no further.

The ports of a DVMRP switch are physical interfaces to a directly attached subnetwork. All interfaces are configured with a metric that specifies the given port's part of the path cost.

An IP interface has to be configured manually to run DVMRP by adding the IP interface to the DVMRP interface list.

Configuring DVMRP

DVMRP is disabled by default, and must be enabled on the switch and on the interfaces it is to operate over to start multicast routing. DVMRP must be enabled on all the interfaces in the DVMRP domain over which the switch sends or receives multicast data, including interfaces to directly connected IP hosts over which IGMP is enabled for group management. To enable or disable DVMRP on the switch, use the commands:

```
enable dvmrp
disable dvmrp
```

To start DVMRP operating over an interface, add the interface to DVMRP by using the command:

```
add dvmrp interface=interface [dlc=1..1024] [metric=1..32]
```

To stop DVMRP from operating over an interface, use the command:

```
delete dvmrp interface=interface [dlc=1..1024]
```

The IP configuration of an interface cannot be changed while DVMRP or PIM is attached to the interface. The DVMRP or PIM interface must first be deleted, and then re-added after the IP changes have been made.

Multicast packets are delivered along the shortest path from one host to another. The distance is the sum of metrics along this path. The switch uses the path cost for routes on each interface to determine which interfaces to send multicast traffic over. When the switch receives a multicast packet over an interface, it adds the metric for the interface to the path cost. A higher metric for an interface causes less traffic to be transmitted across the interface. The default for the metric is 1, and this can be modified with the command:

```
set dvmrp interface=interface [dlc=1..1024] metric=1..32
```

To display information about the DVMRP state, interfaces, neighbours, routes or counters, use the commands:

```
show dvmrp
show dvmrp interface
show dvmrp neighbour
show dvmrp route
show dvmrp counters
```


To reset all DVMRP processes, timers and route information for an interface, that is, to restart all DVMRP processes for an interface as if it has just been added to the DVMRP interface list, use the command:

```
reset dvmrp interface=interface [dlc=1..1024]
```

For experienced users, detailed debugging information can be displayed about the output of neighbours, grafts, prunes, probes and reports. Note that DVMRP debugging can display large amounts of data. To enable or disable DVMRP debugging, use the commands:

```
enable dvmrp debug={all|graft|neighbour|probe|prune|  
report}[,...] interface=interface [dlc=1..1024]  
disable dvmrp debug={all|graft|neighbour|probe|prune|  
report}[,...] interface=interface [dlc=1..1024]
```

Protocol Independent Multicast (PIM)

The two Protocol Independent Multicast routing protocols rely on the presence of an existing unicast routing protocol to adapt to topology changes, but are independent of the mechanisms of the specific unicast routing protocol.

Mode	Description
PIM Dense Mode	Suitable for networks where bandwidth is plentiful, and where members of a multicast group are densely distributed on the network.
PIM Sparse Mode	Suitable when members of the multicast groups are more sparsely distributed over the network because it results in less duplication of data packets over the network.

PIM Sparse Mode and PIM Dense Mode must be enabled with a special feature licence. To obtain one, contact an Allied Telesis authorised distributor or reseller.

The switch can be configured as a Multicast Border Router, with different interfaces connecting to multicast domains that use different multicast routing protocols. Therefore, some PIM interfaces can be configured for PIM-SM and others for PIM-DM. Multicast packets are forwarded between the Sparse Mode and Dense Mode domains as required.

PIM Dense Mode

Unlike PIM Sparse Mode, PIM Dense Mode (PIM-DM) does not use a designated router, bootstrap router, or rendezvous points.

PIM-DM is similar to DVMRP in that it employs the Reverse Path Multicasting (RPM) algorithm. However, there are differences between PIM-DM and DVMRP:

- PIM-DM relies on the presence of an existing unicast routing protocol to provide routing table information to build up information for the multicast forwarding database, but it is independent of the mechanisms of the specific unicast routing protocol. In contrast, DVMRP contains an integrated routing protocol that makes use of its own RIP-like exchanges to compute the required unicast routing information.
- Unlike DVMRP, PIM-DM simply forwards multicast traffic on all downstream interfaces until explicit prune (un-join) messages are received. PIM-DM is willing to accept the overhead of broadcast-and-prune in the interests of simplicity and flexibility, and of eliminating routing protocol dependencies.

PIM-DM assumes that when a source starts sending, all downstream systems want to receive multicast datagrams. Initially, multicast datagrams are flooded to all areas of the network. If some areas of the network do not have group members, dense-mode PIM prunes the forwarding branch by setting up prune state. The prune state has an associated timer, which on expiration turns into forward state, allowing data to go down the branch that was previously in prune state.

The prune state contains source and group address information. When a new member appears in a pruned area, a router can “graft” toward the source for the group, turning the pruned branch into a forwarding branch. The forwarding branches form a tree rooted at the source leading to all members of the group. This tree is called a source rooted tree.

The broadcast of datagrams followed by pruning of unwanted branches is often referred to as a broadcast-and-prune cycle, typical of dense mode protocols. The broadcast-and-prune mechanism in PIM Dense Mode uses a technique called *reverse path forwarding* (RPF), in which a multicast datagram is forwarded only when the receiving interface is the one used to forward unicast datagrams to the source of the datagram.

Configuring PIM Dense Mode

PIM multicasting routing is disabled by default and must be enabled on the switch before any PIM configuration takes effect. However, we recommend that the PIM configuration be completely set up on the switch before PIM is enabled. To enable or disable PIM, use the commands:

```
enable pim
```

```
disable pim
```

For PIM Dense Mode multicast routing to operate on the switch, each interface over which it is to send and receive multicast routing messages and multicast packets must be assigned to PIM-DM.

By default PIM interfaces are set to use Sparse Mode when they are added. To add a PIM-DM interface, use the command:

```
add pim interface=interface mode=dense [other-options...]
```

To delete an interface, use the command:

```
delete pim interface=interface [dlc=1..1024]
```

The IP configuration of an interface cannot be changed while DVMRP or PIM is attached to the interface. The DVMRP or PIM interface must first be deleted, and then re-added after the IP changes have been made.

To modify a PIM interface, use the command:

```
set pim interface=interface [mode={dense|sparse}]
[other-options...]
```

State Refresh messages can be used in a PIM-DM domain to reduce unnecessary multicast traffic. Instead of a source repeatedly flooding downstream routers with multicast packets and repeatedly receiving prune messages, a State Refresh message maintains an existing prune. By default the switch cannot initiate or process State Refresh messages. To enable this functionality on an interface, use one of the commands:

```
add pim interface=interface mode=dense srccapable=yes
[other-options...]

set pim interface=interface srccapable=yes [other-options...]
```

To restart all PIM processes on an interface, resetting the PIM timers, route information and counters for the interface, use the command:

```
reset pim interface=interface [dlc=1..1024]
```

To display information about PIM interfaces, use the command:

```
show pim interface
```

General PIM-DM information

The following commands display general PIM-DM information.

This command...	Shows...
<code>show pim config</code>	CLI commands that make up the switch's PIM configuration.
<code>show pim counters</code>	the number of PIM messages that the switch has received and sent, and the number of bad messages it has received.
<code>show pim neighbour</code>	information about the neighbouring switches that PIM is aware of.
<code>show pim route</code>	the internal PIM routing table.
<code>show pim staterefresh</code>	the internal State Refresh table.

PIM-DM timers

Timers for PIM-DM operations have defaults that suit most networks and should not generally be modified.



Caution Changing these timers to inappropriate values can cause PIM to function in undesirable ways. System administrators should change these timer values based on a sound understanding of their interaction with other devices in the network.

If the timers need to be modified, use the command:

```
set pim [jptime={1..65535|default}]
[keepalivetime={10..65535|default}]
[pruneholdtime={1..65535|default}]
[sourcealivetime={10..65535|default}]
[srinterval={10..255|default}] [other-options...]
```

To list the values of the global PIM timers, use the command:

```
show pim timer
```

PIM-DM debugging To display debugging information about PIM-DM, use the command:

```
enable pim debug={all|assert|bsr|c-rp-adv|graft|hello|join|
register|staterefresh}[,...]
```

To see which debugging options are enabled, use the command:

```
show pim debug
```

PIM Sparse Mode

PIM Sparse Mode (PIM-SM) provides efficient communication between members of sparsely distributed groups - the type of groups that are most common in wide-area internetworks. It is designed on the principle that several hosts wishing to participate in a multicast conference does not justify flooding the entire internetwork with periodic multicast traffic. PIM-SM is designed to limit multicast traffic so that only those routers interested in receiving traffic for a particular group receive the traffic.

The switch supports PIM Sparse Mode as specified in Internet Draft *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*, 1 March 2002 (draft-ietf-pim-sm-v2-new-05). For information about implementation of PIM-SM for IPv6, see [“Protocol Independent Multicast Sparse Mode \(PIM-SM\)” on page 36-9 of Chapter 36, IPv6 Multicasting](#).

Routers with directly attached or downstream members are required to join a Sparse Mode distribution tree by transmitting explicit join messages. If a router does not become part of the predefined distribution tree, it does not receive multicast traffic addressed to the group. In contrast, dense mode multicast routing protocols assume downstream group membership and continue to forward multicast traffic on downstream links until explicit prune messages are received. The default forwarding action of a sparse mode multicast routing protocol is to block traffic unless it is explicitly requested, while the default action of the dense mode multicast routing protocols is to forward traffic.

PIM-SM employs the concept of a rendezvous point (RP) where receivers “meet” sources. The initiator of each multicast group selects a primary RP and a small ordered set of alternative RPs, known as the RP-list. For each multicast group, there is only a single active RP. Each receiver wishing to join a multicast group contacts its directly attached router, which in turn joins the multicast distribution tree by sending an explicit join message to the group’s primary RP. A source uses the RP to announce its presence and to find a path to members that have joined the group. This model requires Sparse Mode routers to maintain some state information (the RP-list) prior to the arrival of data packets. In contrast, Dense Mode multicast routing protocols are data driven, since they do not define any state for a multicast group until the first data packet arrives.

Roles in PIM Sparse Mode

A multicast sender does not need to know the addresses of the members of the group in order to send to them, and the members of the group need not know the address of the sender. Group membership can change at any time. When PIM is enabled on the switch, and before the switch can route multicast traffic, it must establish which of the PIM routers in the network are performing some key roles: *designated router* (DR), *rendezvous point* (RP), and *bootstrap router* (BSR).

Designated router There must be one PIM designated router (DR) in the subnetwork to which the IP hosts are connected. Any PIM-SM interfaces on the subnetwork elect the designated router with the highest DR priority. If there is more than one router with the same priority, or no priority, they choose the interface with the highest IP address number. The DR performs all the PIM functionality for the subnetwork. If the current DR becomes unavailable, the remaining routers elect a new DR on the interface by DR priority or IP address.

Rendezvous point Each multicast group must have a rendezvous point (RP). The RP forms the root of the group's distribution tree. The designated router for a multicast sender sends multicast packets towards the RP. Designated routers with group members connected to them send join messages towards the group's RP. The RP candidate with the lowest priority is elected from all the RP candidates for a group. If the RP becomes unavailable, the remaining RP candidates elect a new RP.

Note that software release versions prior to 2.7.3 did not correctly support the PIM hash mask length option. As a result, the RP selection calculation differs between this release and release versions prior to 2.7.3. If a network contains switches running a mixture of versions, this leads to incorrect forwarding behaviour. To avoid this issue, either ensure that all devices on the network correctly support the hash mask length option (recommended), or ensure that the following **both** hold:

- The hash mask length option on all BSR candidates is configured to 4 bits. This implies that all BSR candidates must be running 2.7.3 or later.
- All RP candidates use a common prefix of 224.0.0.0/240.0.0.0. This has the side effect of collapsing all groups to use a single PIM RP.

Bootstrap router Each PIM-SM network must have at least one bootstrap router (BSR) candidate, unless all routers in the domain are configured statically with information about all RPs in the domain. Every router that is a BSR candidate periodically sends a Bootstrap Candidate Advertisement message to advertise that it is available as a bootstrap router candidate. The BSR candidates in the network elect the router with the highest preference value to be the bootstrap router. The elected bootstrap router listens to PIM Candidate RP Advertisement messages specifying RP candidates for multicast groups. It maintains a list of RP candidates, and sends a bootstrap message every BSM interval, specifying all the multicast groups in the PIM network, and their rendezvous point candidates. Each router uses this information and a standardised hash mechanism to determine the RP for each group.

In summary:

- Each *multicast group* must have at least one rendezvous point candidate
- Each *PIM-SM domain* must have at least one Bootstrap Router candidate, unless all routers in the domain are configured statically with information about all RPs in the domain

- Each *subnetwork* must have at least one Designated Router candidate.

PIM hello messages When PIM is enabled on a switch, it sends out a PIM *Hello* message on all its PIM enabled interfaces, and listens for Hello messages from its PIM neighbours. When a switch receives a Hello message, it records the interface, IP address, priority for becoming a designated router, and the timeout for the neighbour's information. The switch sends Hello messages regularly at the Hello Time interval.

Operation of PIM Sparse Mode

Once roles are established, multicast routing follows specific phases:

1. **Rendezvous point tree**
2. **Register stop**
3. **Shortest path tree**

While multicast routing always begins with phase 1, the designated router for a receiver determines whether and when to move on to phases 2 and 3, depending on the amount of traffic from the source.

Rendezvous point tree Phase 1 establishes and uses a shared tree rooted at the rendezvous point (RP) to forward all multicast data to group members.

When an IP host sends an IGMP join message to the local PIM designated router, which is not the RP for the group, the designated router sends a *PIM join* message towards the RP for the group ("upstream"). The designated router determines which router is the RP for the group from the most recent bootstrap message. Every router the join message passes through records that there is a group member on the incoming interface. Eventually, the join message reaches either the RP, or another router that already knows that it has a group member downstream. If the group has many members, the join messages converge on the RP to form a rendezvous point tree (RPT). This is called a shared tree because multicast data that is sent to the group by any sender shares the tree. The multicast receiver's designated router sends join messages periodically according to the upstream join timer as long as the IP host is a member of the group. When the last receiver on a subnetwork leaves the group, the join messages stop, and their entries timeout on routers that are closer to the RP.

The sender's designated router encapsulates the multicast data in a unicast packet in a process called *registering*, and sends these register packets to the group's RP. When the RP receives the data, it decapsulates them, and forwards them onto the shared tree.

Register stop Phase 2 improves efficiency and performance by using register stop. In this phase the RP joins the shortest path tree between the source and receiver. This allows the original (unencapsulated) packets to be forwarded from the sender, instead of encapsulated packets. It also allows shorter paths to receivers that are close to the sender, making it more efficient in some circumstances.

When the RP for a group receives the first encapsulated data packet from a source, it joins the shortest path tree towards the sender. Once data is able to flow along the shortest path from the sender to the RP, packets do not need to be registered. The RP sends a *register stop* message in reply to the next encapsulated message. When the sender's DR receives the register stop message, it stops registering. The DR sends a *null register* message to the RP to find whether the RP still does not need to receive registered packets. If it receives another register stop message, the DR continues to forward only the

native data packets. If the DR does not receive another register stop message within the register probe time, it resumes registering the data packets and sending them to the RP.

When the RP starts receiving native data packets from the source, it starts to discard the encapsulated packets, and starts forwarding native packets on the shared tree to all the group members. If the path from the source to the RP intersects the shared RP tree for the group, then the packets also take a short-cut onto the shared tree for delivery to the group members down its branches.

Shortest path tree This phase further optimises routing by using shortest path trees (SPT). In phase 3 the receiver joins the shortest path tree between the source and receiver. This allows a multicast group member to receive multicast data by the shortest path from the sender, instead of from the shared RP tree. When the receiver's DR receives multicast data from a particular sender, it sends a *join* message towards the sender. When this message reaches the sender's DR, the DR starts forwarding the multicast data directly towards the receiver. As several receivers all initiate shortest paths to the sender, these paths converge, creating a shortest path tree.

When the multicast packets start arriving from the SPT at the receiver's DR or an upstream router common to the SPT and the RPT, it starts discarding the packets from the RPT, and sends a *prune* message towards the RP. The prune message travels up the RPT until it reaches the RP or a router that still needs to forward multicast packets from this sender to other receivers. Every time a router receives a prune message, it waits a short time (the J/P Override Interval specified in Internet Draft draft-ietf-pim-sm-v2-new-05) before putting the prune into effect, so that other routers on the LAN have the opportunity to override the prune message.

Multi-Access LANs If the PIM-SM network includes multi-access LAN links for transit, as well as point-to-point links, then a mechanism is needed to prevent multiple trees forwarding the same data to the same group member. Two or more routers on a LAN may have different information about how to reach the RP or the multicast sender. They could each send a join message to two different routers closer to the RP for an RPT or the sender for an SPT. This could potentially cause two copies of all the multicast traffic towards the receiver.

When PIM routers notice duplicate data packets on the LAN, they elect a single router to forward the data packets, by each sending PIM *Assert* messages. If one of the upstream routers is on an SPT and the other is on an RPT, the router on the SPT has the shortest path to the sender, and wins the Assert election. If both routers are on RPTs the router with the shortest path to the RP (the lowest sum of metrics to the RP) wins the Assert. If both routers are on an SPT, then the router with the shortest path to the sender (the lowest sum of metrics to the sender's DR) wins the Assert.

The router that won the Assert election forwards these data packets, and acts as the local designated router for any IGMP members on the LAN. The downstream routers on the LAN also receive the Assert messages, and send all their join messages to the Assert winner. The result of an Assert election times out after the Assert Time specified in the Internet Draft draft-ietf-pim-sm-v2-new-05. As long as the situation causing the duplication remains unchanged, the Assert winner sends an Assert message at the Assert time interval, before the previous Assert messages time out. When the last downstream router leaves the SPT, the Assert winner sends an Assert Cancel message saying that it is about to stop forwarding data on the SPT. Any RPT downstream routers then switch back to the RP tree.

Configuring PIM Sparse Mode

PIM multicasting routing is disabled by default and must be enabled on the switch before PIM configuration takes effect. However, we recommend that the PIM configuration be completely set up on the switch before PIM is enabled. To enable or disable PIM, use the commands:

```
enable pim
disable pim
```

For PIM Sparse Mode multicast routing to operate on the switch, each interface over which it is to send and receive multicast routing messages and multicast packets must be assigned to PIM-SM. Each subnetwork must also have at least one designated router candidate, each network must have at least one bootstrap router candidate, and each multicast group must have at least one rendezvous point candidate.

The IP configuration of an interface cannot be changed while DVMRP or PIM is attached to the interface. The DVMRP or PIM interface must first be deleted, and then added again after the IP changes have been made.

PIM-SM interfaces By default PIM interfaces are set to use Sparse Mode when they are added. To add a PIM-SM interface, use the command:

```
add pim interface=interface [drpriority=0..4294967295]
[electby={drpriority|ipaddress}] [mode=sparse]
[other-options...]
```

Each PIM-SM interface has a priority for becoming the designated router (DR) for its subnetwork. The higher the number, the higher the priority. The default designated router priority is 1. If the multicast group must choose a DR from interfaces with the same priority, or no priority, the interface with the highest IP address number is chosen.

The **electby** parameter determines how the switch elects the designated router for this interface. If **drpriority** is specified, the interface transmits its DR priority in its hello messages. If all routers in the subnetwork transmit their DR priorities, routers in the subnetwork can elect the DR by priority. If **ipaddress** is specified, the switch does not transmit its DR priority, which forces the routers in the subnetwork to elect the DR by IP address. The default is **drpriority**.

To delete an interface, use the command:

```
delete pim interface=interface [dlc=1..1024]
```

To modify the mode, designated router priority, or method by which the designated router is elected for a PIM interface, use the command:

```
set pim interface=interface mode={dense|sparse}
[drpriority=0..4294967295] [electby={drpriority|
ipaddress}] [other-options...]
```

To restart all PIM processes on an interface, resetting the PIM timers, route information and counters for the interface, use the command:

```
reset pim interface=interface [dlc=1..1024]
```

To display information about PIM interfaces, use the command:

```
show pim interface
```


Bootstrap router candidates

Each network of PIM-SM routers must have a bootstrap router (BSR). Each PIM-SM connected network must have at least one bootstrap router candidate. The candidate with the highest preference value becomes the bootstrap router. The default preference is 1. The bootstrap router sends a bootstrap message to other PIM-SM routers, containing a list of the RP candidates for multicast groups at BSM interval seconds. To designate the switch as a bootstrap router candidate, use the command:

```
add pim bsrcandidate [preference=0..255]
```

To change the switch's preference of bootstrap router candidate, use the command:

```
set pim bsrcandidate preference=0..255
```

To stop the switch from being as a bootstrap router candidate, use the command:

```
delete pim bsrcandidate
```

To display information about the switch's bootstrap router configuration, use the command:

```
show pim bsrcandidate
```

Rendezvous point

Each multicast group must have a rendezvous point (RP), which is either chosen dynamically from the list of rendezvous point candidates available or is statically configured on each router that processes traffic for that group. For dynamic RP selection, there must be at least one RP candidate in the PIM-SM connected network, but generally there should be several. PIM-SM chooses the RP candidate with lowest preference value to be the RP for the multicast group. The lower the number, the higher its priority. The default priority is 192. The dynamically-chosen RP advertises itself to the current bootstrap router at an interval specified by the **advinterval** parameter in the **set pim** command. The default **advinterval** is 60 seconds.

When an IP host joins a multicast group on a router, the router sends a *join* message to the active rendezvous point. The rendezvous point then knows to send multicast packets for the group to this router. When the last IP host leaves a group, the router sends a *prune* message to the RP, telling it that it no longer needs to receive multicast packets for the group.

To configure the switch to be a dynamic RP candidate, use the command:

```
add pim rpcandidate group=group-address [mask=ipaddress]
[priority=0..255]
```

To modify the switch's RP candidate priority, use the command:

```
set pim rpcandidate group=ipadd [mask=ipadd] priority=0..255
```

The switch has the same values for **priority** for all multicast groups for which it is a rendezvous point candidate, so changing the priority for one group changes it for all groups.

To stop the switch from being an RP candidate, use the command:

```
delete pim rpcandidate group=group-address [mask=ipadd]
```

Static RP mappings can be configured instead of using the bootstrap mechanism. To configure a static rendezvous point on the switch for a

multicast group, specify the IP address of the rendezvous point by using the command:

```
add pim rpcandidate=rp-address group=group-address
[mask=ipaddress]
```

where *rp-address* is the IP address of the router that is the rendezvous point for the multicast group(s) specified. An RP can be statically configured as the RP for multiple groups, but each group can only have one statically-configured RP. Each router in the PIM-SM domain must be configured with the same static RP to group mapping.

Note that if the bootstrap mechanism is also running, a static RP mapping takes precedence.

To delete a static RP, use the command:

```
delete pim rpcandidate=rp-address group=group-address
[mask=ipaddress]
```

To display information about multicast groups for which the switch is a rendezvous point candidate, use the command:

```
show pim rpcandidate
```

To display the static group-to-RP mapping followed by the elected bootstrap router's current set of RP candidates and the groups they are configured for, use the command:

```
show pim rpset
```

General PIM-SM information

The following commands display general PIM-SM information.

This command ...	Shows ...
<code>show pim config</code>	CLI commands that make up the switch's PIM configuration.
<code>show pim counters</code>	the number of PIM messages that the switch has received and sent, and the number of bad messages it has received.
<code>show pim neighbour</code>	information about the neighbouring switches that PIM is aware of.
<code>show pim route</code>	the internal PIM routing table.

PIM-SM timers

Timers for PIM-SM operations have defaults that suit most networks. However, if you need to modify them, use the command:

```
set pim[adinterval={10..15000|default}]
[bsminterval={10..15000|default}] [jpinterval={1..65535|
default}] [keepalivetime={10..65535|default}]
[probetime={1..65535|default}]
[suppressiontime={1..65535|default}] [other-options...]
```



Caution Changing these timers to inappropriate values can cause PIM to function in undesirable ways. System administrators should change these timer values based on a sound understanding of their interaction with other devices in the network.

To list the values of the global PIM timers, use the command:

```
show pim timer
```

PIM-SM debugging To display debugging information about PIM-SM, use the command:

```
enable pim debug={all|assert|bsr|c-rp-adv|hello|join|
register}[,...]
```

To see which debugging options are enabled, use the command:

```
show pim debug
```

Logging and SNMP Traps for PIM Sparse Mode

PIM-SM can be configured to produce log messages in response to status changes and errors, and SNMP traps. This feature does not apply to PIM-DM.

Status messages Events that trigger a status-change log message are:

- PIM interface is disabled
- PIM interface is enabled
- PIM neighbour adjacency has timed out
- PIM neighbour generation ID has changed
- PIM neighbour has changed port
- PIM RP has changed
- PIM DR has changed
- PIM BSR has changed

Error messages Errors that trigger a log message are:

- Invalid PIM packet
- Invalid destination address
- Fragmentation reassembly
- Packet too short
- Bad group address encoding
- Bad source address encoding
- Missing option
- Internal error
- Receive packet—a range of errors that mean the packet was received but cannot be forwarded.

SNMP traps

This trap is generated ...	When ...
PimInterfaceUpTrap	a PIM interface comes up and is active.
PimInterfaceDownTrap	a PIM interface goes down and is inactive.
PimNeighbourLossTrap	a known PIM neighbour has lost adjacency or has timed-out. This trap is part of the experimental PIM MIBs group.
PimNeighbourAddedTrap	a PIM neighbour is added.
PimNeighbourDeletedTrap	a PIM neighbour is deleted.
PimErrorTrap	any one of the PIM error counters is incremented or when a log message of subtype LOG_STY_PIM_ERROR is generated (see list of errors above).

To specify the type of log messages and SNMP traps that the switch generates, use the command:

```
set pim log={none|status|error|all}  
[trap={none|status|error|all}]
```

To display the specified options, use the command:

```
show pim debug
```

Internet Group Management Protocol (IGMP)

IGMP is a protocol used between hosts and multicast routers and switches on a single physical network to establish hosts' membership in particular multicast groups. Multicast routers use this information, in conjunction with a multicast routing protocol, to support IP multicast forwarding across the Internet.

The switch supports Internet Group Management Protocol version 2 (IGMPv2), defined in RFC 2236, *Internet group Management Protocol, version 2*. It can also detect and interoperate with hosts and other designated routers (sometimes called querier routers) running IGMP version 1.

When IGMP is enabled on the switch, and on particular interfaces, it sends out IGMP queries on all IGMP interfaces. If it receives an IGMP message from a router with a lower IP address on an interface, it knows that another switch is acting as the IGMP designated router for that subnetwork. If it receives no IGMP messages with a lower IP address, it takes the role of designated switch for that subnetwork. If it is the designated switch, it continues to send out general IGMP *Host Membership Queries* regularly on this interface.

When an IP host hears a general IGMP Host Membership Query from the switch, it sends an IGMP *Host Membership Report* back to the switch. All the IGMP routers on the subnetwork put an entry into their *local group database*, so that the switches know which interfaces to send packets for this multicast group out of. These entries are updated regularly, as long as the interface has a member of the multicast group connected to it. As hosts join and leave multicast groups dynamically, the switch keeps a list of group memberships for each of its primary interfaces. In the case of multihomed interfaces, the primary interface is the first interface to be configured ([“Multihoming” on page 23-10 of Chapter 23, Internet Protocol \(IP\)](#)).

When an IP host stops belonging to a multicast group, it sends an IGMP *Leave* message to the switch. The switch then sends one or more group-specific IGMP membership queries, and any other IP hosts belonging to the same multicast group reply with a Host Membership Report. IGMP then knows whether there are still any members of this multicast group connected to the interface.

Configuring IGMP

IGMP is disabled by default on the switch, and on all interfaces. To enable or disable IGMP on the switch, use the commands:

```
enable ip igmp  
disable ip igmp
```

IGMP snooping is enabled by default and is independent of IGMP.

IGMP must be enabled on an interface before it can send or receive IGMP messages on the interface. If DVMRP is used for multicast routing, IGMP must also be enabled on interfaces that DVMRP uses. To enable or disable IGMP on an interface, use the commands:

```
enable ip igmp interface=interface
disable ip igmp interface=interface
```

IGMP keeps the local group database up to date with current multicast group members by updating it when it hears IGMP Host Membership Reports on an interface. If the switch is the IGMP designated router for the subnetwork, it sends out IGMP Host Membership Queries at a Query Interval. If it does not receive a Host Membership Report for a multicast group on an interface within the Timeout period, it deletes the multicast group from its local group database. The default of the Query Interval (125 seconds) and of the Timeout ((2*Query Interval) + 10 seconds) suit most networks. These defaults should be changed with caution, and with a sound understanding of how they affect interaction with other devices. To change the intervals, use the command:

```
set ip igmp [lmqi=1..255] [lmqc=1..5]
[queryinterval=1..65535] [queryresponseinterval=1..255]
[robustness=1..5] [timeout=1..65535]
```

IGMP can be configured to monitor the reception of IGMP general query messages on an interface, and generate a log message and an SNMP trap if a general query message is not received within a specified time interval. To configure monitoring on an interface, use the command:

```
set ip igmp interface=interface querytimeout={none|0|
1..65535}
```

To display information about IGMP and multicast group membership, use the command:

```
show ip igmp [interface=interface] [destination=ipadd]
```

If IGMP snooping is enabled, this command also displays the ports listening to the multicast group for each VLAN-based IP interface.

To display IGMP counters, use the command:

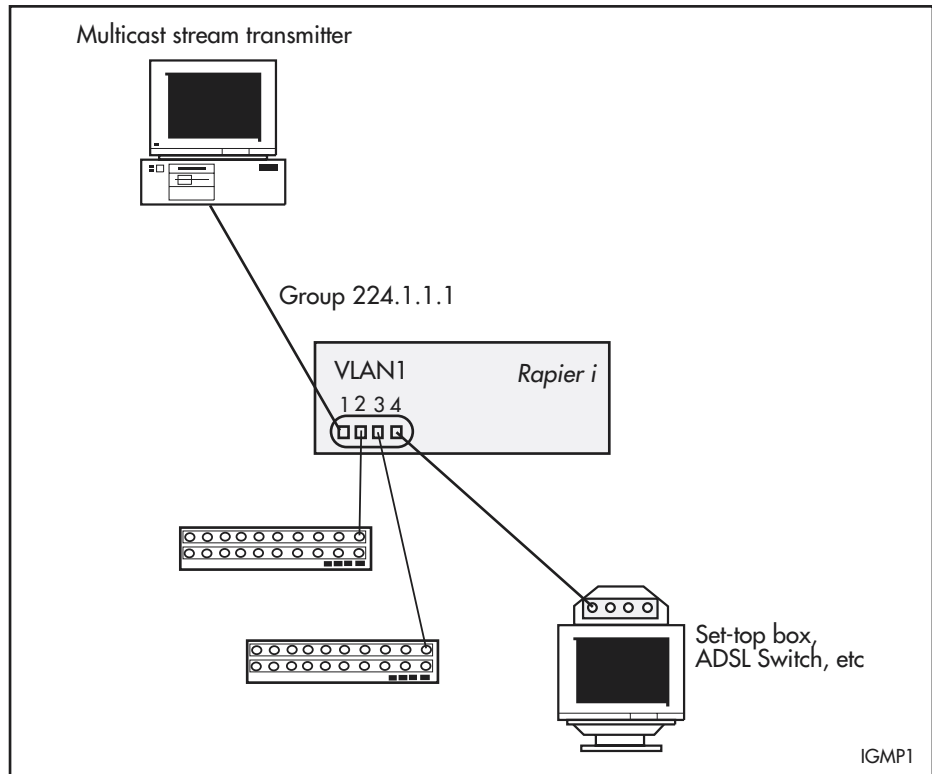
```
show ip igmp counter [interface=interface]
[destination=ipadd]
```

Static IGMP

Static IGMP forwards multicast data over specific interfaces and ports. It is an alternative to dynamic IGMP, and is useful for network segments that have no multicast group members or have hosts that are unable to report group membership with IGMP. A dynamic IGMP configuration does not send multicast traffic to these network segments.

Figure 27-2 shows a switch forwarding the multicast stream to a set-top box after a user specifies that group 224.1.1.1 multicast data should be forwarded out of port 4 of VLAN1. Unlike conventional IGMP membership, this user-specified *static membership* never times out. You can also filter some IGMP debug messages by source IP address and group destination address.

Figure 27-2: Forwarding multicast data over a specific interface and port



To enable IGMP on the switch, use the command:

```
enable ip igmp
```

To enable IGMP on a specific interface, use the command:

```
enable ip igmp interface=interface
```

To create the static IGMP association, use the command:

```
create ip igmp destination=ipaddress interface=interface
[port={all|port-list}]
```

The multicast data for the group specified by the **destination** parameter is forwarded over the ports specified by the **port** parameter. If the **port** parameter is not entered, the association defaults to all ports belonging to the interface.

To display information about the static IGMP association, use the command:

```
show ip igmp [interface=interface] [destination=ipaddress]
```

Any of the four octets of the IP address may be replaced by an asterisk (*) to enable wildcard matches.

To add more ports to an association, use the command:

```
add ip igmp destination=ipaddress interface=interface
port={all|port-list}
```

Unlike dynamic IGMP group membership information, static IGMP associations never time out. If the network configuration changes, they must be manually modified. To delete ports from an association, use the command:

```
delete ip igmp destination=ipaddress interface=interface
port={all|port-list}
```

To remove an association from a switch, use the command:

```
destroy ip igmp destination=ipaddress interface=interface
```

To enable or disable IGMP debugging of destination and source IP addresses, use the commands:

```
enable ip igmp debug [destination={all | ipaddress}]  
[sourceipaddress={all | ipaddress2}]
```

```
disable ip igmp debug
```

Debugging is disabled by default. To display which debugging options are set, use the command:

```
show ip igmp debug
```

IGMP Proxy

In a network with a simple tree topology, you can use IGMP proxy to simplify the configuration of multicast routing. The switch at the root of the tree must run a multicast routing protocol, but all other switches in the network can be configured as IGMP proxy agents.

The IGMP proxy agent must be configured with a single upstream interface and one or more downstream interfaces. An upstream interface is an interface in the direction towards the root of the tree. A downstream interface is an interface in the direction away from the root of the tree.

The IGMP proxy agent periodically transmits IGMP general membership queries to the hosts attached to its downstream interfaces. The proxy agent uses IGMP report and leave messages received on downstream interfaces to build and maintain a database of multicast group memberships, and reports changes to the list of multicast groups in the database on the upstream interface. The following table summarises how the IGMP proxy agent processes each IGMP message type.

When this message...	Is received on this interface...	Then the IGMP proxy agent...
Report	downstream	adds the membership subscription to the multicast group membership database forwards the report message on the upstream interface, if the membership subscription is for a new multicast group
	upstream	discards the message without processing
Leave	downstream	removes the membership subscription from the multicast group membership database forwards the leave message on the upstream interface, if there are no remaining membership subscriptions for the multicast group (no other hosts connected to any of the downstream interfaces have members of the multicast group)
	upstream	discards the message without processing
Group-specific query	downstream	discards the message without processing
	upstream	transmits a report message on the upstream interface, if the multicast group membership database contains at least one member of the multicast group attached to a downstream interface

When this message...	Is received on this interface...	Then the IGMP proxy agent...
General query	downstream	discards the message without processing
	upstream	transmits a report message on the upstream interface for each multicast group in the multicast group membership database with at least one member attached to a downstream interface

The IGMP proxy agent uses the information maintained in the multicast group membership database to forward multicast data packets received on the upstream interface to all downstream interfaces that have members of the multicast group.

Multicast packet forwarding is enabled as long as:

- a multicast routing protocol is not enabled
- an interface is configured with IGMP proxy in the upstream direction
- at least one interface is configured with IGMP proxy in the downstream direction

To add an IP interface and configure IGMP proxying, use the command:

```
add ip interface=interface ipaddress={ipadd|dhcp}
    [igmpproxy={off|upstream|downstream}] [other-options...]
```

To configure IGMP proxy on an existing IP interface, use the command:

```
set ip interface=interface
    igmpproxy={off|upstream|downstream}]
```

IGMP proxy is turned off by default.

IGMP must also be enabled on the switch and on the interface for IGMP proxy to function.

To enable IGMP on the switch, use the command:

```
enable ip igmp
```

To enable IGMP on a specific interface, use the command:

```
enable ip igmp interface=interface
```

You can configure the IGMP proxy agent to monitor the reception of IGMP general query messages on an interface, and to generate a log message and an SNMP trap if an IGMP general query message is not received on the interface within a specified time interval.

To enable monitoring on an interface and set the time interval, use the command:

```
set ip igmp interface=interface querytimeout={none|0|
    1..65535}
```

To display information about IGMP and the IGMP proxy agent, use the command:

```
show ip igmp
```


IGMP Snooping

IGMP snooping lets switches intelligently forward multicast traffic instead of flooding all ports in the VLAN. Because IGMP is an IP-based protocol, multicast group membership for VLAN-aware devices is on a per-VLAN basis. If at least one port in the VLAN is a member of a multicast group and IGMP snooping is not enabled, multicast packets are flooded out all ports in the VLAN.

When IGMP snooping is enabled, the switch listens to IGMP membership reports, queries, and leave messages to identify which ports are members of multicast groups. Multicast traffic is then forwarded only to ports that are members of the multicast group.

IGMP snooping happens automatically at Layer 2 on VLAN interfaces. By default, the switch forwards traffic from ports with multicast listeners, and does not act as a simple hub and flood multicast traffic from all ports. IGMP snooping is independent of the IGMP and Layer 3 configuration, so an IP interface does not have to be attached to the VLAN, and IGMP does not have to be enabled or configured.

IGMP snooping is enabled by default. To disable it, use the command:

```
disable igmpsnooping
```

Disabling IGMP snooping may be useful when filters are used extensively because IGMP snooping uses a Layer 3 filter. When IGMP snooping is disabled, this filter becomes available. See [“Hardware Packet Filters” on page 8-33 of Chapter 8, Switching](#) for information about filters. Note that multicast packets flood the VLAN when IGMP snooping is disabled.

To enable IGMP snooping, use the command:

```
enable igmpsnooping
```

IGMP snooping can be enabled only when a free filter entry is available.

To display information about IGMP snooping, use the command:

```
show igmpsnooping [vlan={vlan-name|1..4094}]
```

To display counters for IGMP snooping, use the command:

```
show igmpsnooping counter [vlan={vlan-name|1..4094}]
```

Downstream routers

IGMP snooping learns which ports have routers attached to them, so it can forward relevant IGMP messages and other IP multicast traffic out those ports. You can specify the following aspects of this:

- the kind of packets that indicate to IGMP snooping that a port has a router downstream
- which specific ports have routers downstream

Packets that indicate routers

By default, IGMP snooping identifies router ports by looking for ports that receive specific multicast packets (such as IGMP queries, PIM messages, OSPF messages, and RIP messages). You can determine what kinds of packets indicate a router is downstream, by using the command:

```
set igmpsnooping routermode={all|default|ip|multicastrouter|none}
```

For each option in this command, the following table lists the addresses that IGMP snooping uses to indicate that a port has a router downstream.

This option...	means that the port is treated as a multicast router port if it receives packets from...
all	any reserved multicast addresses (224.0.0.1 to 224.0.0.255)
multicastrouter	224.0.0.4 (DVMRP routers) 224.0.0.13 (all PIM routers)
default	224.0.0.1 (IGMP Queries) 224.0.0.2 (all routers on this subnet) 224.0.0.4 (DVMRP routers) 224.0.0.5 (all OSPFIGP routers) 224.0.0.6 (OSPF designated routers) 224.0.0.9 (RIP2 routers) 224.0.0.13 (all PIM routers) 224.0.0.15 (all CBT routers)
ip	the current list of addresses, plus addresses specified using the command add igmpsnooping routeraddress and minus addresses specified using the command delete igmpsnooping routeraddress .

If you specify **set igmpsnooping routermode=ip**, then you can add and remove reserved IP multicast addresses to and from the list of router multicast addresses by using the commands:

```
add igmpsnooping routeraddress=ipadd-list
delete igmpsnooping routeraddress=ipadd-list
```

The IP addresses specified must be from 224.0.0.1 to 224.0.0.255.

To display the current mode and list of multicast router addresses, use the command:

```
show igmpsnooping routeraddress
```

Static multicast router ports

In some network configurations, the learning process cannot identify all router ports. For such networks, you can statically configure particular ports as multicast router ports.

To specify the static router ports, use the command:

```
add igmpsnooping vlan={vlan-name|1..4094}
    routerport=port-list
```

To stop ports from being static router ports, use the command:

```
delete igmpsnooping vlan={vlan-name|1..4094}
    routerport=port-list
```

To list the static router ports, use the command:

```
show igmpsnooping
```

Fast Leave

When an IGMP group-specific leave message is received on a port, IGMP Snooping stops the transmission of the group multicast stream after a timeout period. The **lmqi** (Last Member Query Interval) and **lmqc** (Last Member Query Count) parameters of the **set ip igmp** command set the timeout period. This timeout period allows other hosts on the port to register their membership of the multicast group and continue receiving the stream.

The Fast Leave feature allows IGMP Snooping to stop the transmission of a group multicast stream from a port as soon as it receives a leave message, without waiting for the timeout period.

Use the Fast Leave feature to improve bandwidth management on ports that are connected to a single host. Fast Leave should not be configured on a port that has multiple hosts attached because it may adversely affect multicast services to some hosts.

Fast Leave processing is disabled by default. To enable Fast Leave on a specific VLAN, or all VLANs on the switch, use the command:

```
set igmpsnooping vlan={vlan-name|1..4094} fastleave={on|yes|true}
```

To disable Fast Leave on a specific VLAN, or all VLANs on the switch, use the command:

```
set igmpsnooping vlan={vlan-name|1..4094} fastleave={off|no|false}
```

To display the current state of Fast Leave processing on a specific VLAN, or all VLANs on the switch, use the command:

```
show igmpsnooping [vlan={vlan-name|1..4094}]
```

Query Solicitation

Query solicitation minimises loss of multicast data after a topology change on networks that use spanning tree (STP, RSTP, or MSTP) for loop protection.

When IGMP snooping is enabled and Spanning Tree changes the underlying link layer topology, this can interrupt multicast data flow for a significant length of time. Query solicitation prevents this by monitoring for any topology changes. When it detects a change, it generates a special IGMP Leave message known as a Query Solicit, and floods the Query Solicit message to all ports in

every VLAN that query solicitation is enabled on. When the IGMP Querier receives the message, it responds by sending a General Query, which all IGMP listeners respond to. This refreshes snooped group membership information in the network.

Query solicitation functions by default (without you enabling it) on all VLANs on the root bridge in an STP instance. By default, the root bridge always sends a Query Solicit message when the topology changes.

If you have multiple STP instances, query solicitation only sends Query Solicit messages on VLANs in the instance that experienced a topology change.

In switches other than the STP root bridge, query solicitation is disabled by default, but you can enable it by using the command:

```
set igmpsnooping vlan={vlan-name|1..4094|all}
  queriesolicit={on|yes|true}
```

If you enable query solicitation on a switch other than the STP root bridge, both that switch and the root bridge send a Query Solicit message.

Once the Querier receives the Query Solicit message, it sends out a General Query and waits for responses, which update the snooping information throughout the network. If necessary, you can reduce the time this takes by tuning the IGMP timers, especially the **queryresponseinterval** parameter. For more information, see the “IGMP Timers and Counters” section of *How To Configure IGMP on Allied Telesyn Routers and Switches for Multicasting*. This How To Note is available in the Resource Center of the Documentation and Tools CDROM, or from

www.alliedtelesis.co.uk/en-gb/solutions/techdocs.asp?area=howto

On switches other than the STP root bridge, you can disable query solicitation by using the command:

```
set igmpsnooping vlan={vlan-name|1..4094|all}
  queriesolicit={off|no|false}
```

To see whether query solicitation is on or off, check the Query Solicitation field in output of the [show igmpsnooping command on page 27-109](#).

Blocking All-Groups Entries

IGMP snooping all-groups allows you to prevent a port or ports from acting as an all-groups entry.

Sometimes the device cannot differentiate between certain multicast addresses and permanent host groups at Layer 2. For example, this happens with the addresses 239.0.0.2 and 224.0.0.2 where 224.0.0.2 is the all-routers multicast group. If the device receives an IGMP report for the 239.0.0.2 address, which has a MAC address of 01-00-5e-00-00-02, the device creates an all-groups entry in the MARL. All further multicast groups are added to this port, so multicast traffic is forwarded out the port.

By preventing a port or ports from receiving an all-groups entry, you can limit the number of router ports on the device, and therefore the volume of multicast traffic sent over the device's ports. Once disabled with the **disable ip igmp allgroup** command, the port no longer creates MARL entries when the device receives an IGMP report, query, or multicast data over any other port. For example, if port 9 has been disabled as an all-groups port, an all-groups entry will be created for port 9. This will happen when the port receives packets that will create an IGMP router port, such as reserved multicast groups and IGMP

queries. However, a subsequent IGMP report received over port 7 will have an entry made for port 7 only. The IGMP group received on port 7 will not be added to port 9.

The all-groups disabled ports can be viewed in the output of the [show ip igmp](#) and [show igmpsnooping](#) commands.

IGMP Filtering

IGMP filtering lets you manage the distribution of multicast services on each switch port by controlling which multicast groups the hosts attached to a switch port can join.

IGMP filtering is applied to multicast streams forwarded by IGMP, IGMP Snooping, or MVR.

IGMP filtering and throttling can be applied separately, or together, on the same switch port. Filtering is applied first, and any multicast group memberships passed by the filter are further subjected to the limits imposed by throttling. For more information about IGMP throttling, see [“IGMP Throttling” on page 27-31](#).

Static associations of switch ports and multicast groups are not affected by IGMP filtering.

When to use IGMP filters

Use an IGMP filter to:

- limit the multicast groups a downstream port can be a member of, by applying a filter that matches IGMP report messages to the port
- limit the impact of misbehaving devices by applying a filter that matches inappropriate IGMP messages, for example, query messages on a downstream port or leave messages on an upstream port

Filter format

An IGMP filter consists of zero or more entries. An entry consists of:

- A multicast address range to match against. Address ranges in multiple entries can overlap.
- An IGMP message type to match—query, report, or leave.
- An action to take (include or exclude) when an IGMP message is received that matches the multicast address range and message type.

Matching against a filter

When an IGMP filter is applied to a switch port the following happens:

1. IGMP matches incoming IGMP messages from the switch port against each entry in the filter applied to the port.
2. If the message type and group address in the IGMP message matches a filter entry, IGMP takes the action specified by the filter entry:
 - If the action is **include**, IGMP processes the IGMP message as normal.
 - If the action is **exclude**, IGMP excludes the IGMP message from normal IGMP processing and discards the packet.

Filter processing stops when a match is found.

3. If the IGMP message does not match any entry in the filter, but the filter contains at least one entry that matches the message type, then IGMP excludes the IGMP message from normal IGMP processing and discards the packet.

Applying an empty IGMP filter (a filter with no entries) to a switch port allows all incoming IGMP messages to be processed as normal.

Order of entries The order of entries in a filter is important. When IGMP tries to match an IGMP message to a filter, it performs a linear search of the filter to find a matching entry. Each entry is tried in turn, and processing stops at the first match found.

Address ranges can overlap. If the address range of an entry falls entirely within the address range of another entry, the entry with the smaller address range should appear first in the filter. Otherwise it will never be matched against an IGMP message.

Performance can be improved by arranging the entries in a filter to achieve the earliest possible match.

Configuring IGMP filters To configure an IGMP filter, you must create the filter and then apply it to one or more switch ports.

To do this, first create the filter by using the command:

```
create igmp filter=filter-id
```

Then add one or more entries to the filter with the command:

```
add igmp filter=filter-id groupaddress={ipadd|ipadd-ipadd}
[msgtype={query|report|leave}] [action={include|exclude}]
[entry=1..65535]
```

Finally, apply the filter to a switch port with the command:

```
set switch port={port-list|all} igmpfilter=filter-id
[other-options...]
```

You can apply an IGMP filter to more than one switch port, but a single switch port can have only one IGMP filter assigned to it.

To delete or modify an entry in a filter, use the commands:

```
delete igmp filter=filter-id entry=1..65535

set igmp filter=filter-id entry=1..65535
[groupaddress={ipadd|ipadd-ipadd}] [msgtype={query|
report|leave}] [action={include|exclude}]
```

To remove a filter from a switch port, use the command:

```
set switch port={port-list|all} igmpfilter=none
[other-options...]
```

To destroy a filter, first remove the filter from all ports that it is applied to, then use the command:

```
destroy igmp filter=filter-id
```

To display information about IGMP filters, use the command:

```
show igmp filter=filter-id
```

To display the IGMP filter assigned to a switch port, use the command:

```
show switch port[={port-list|all}]
```

IGMP Throttling

IGMP throttling lets you manage the distribution of multicast services on each switch port by limiting the number of multicast groups that a host on a switch port can join.

IGMP throttling is applied to multicast streams forwarded by IGMP, IGMP Snooping, or MVR.

IGMP filtering and throttling can be applied separately, or together, on the same switch port. Filtering is applied first, and any multicast group memberships passed by the filter are further subjected to the limits imposed by throttling. For more information about IGMP filtering, see [“IGMP Filtering” on page 27-29](#).

IGMP throttling controls the maximum number of multicast groups that a port can join. When the number of multicast group memberships associated with a switch port reaches the limit set, further Membership Reports are subject to a throttling action—deny or replace.

If you configure a throttling action of **deny**, when the multicast group membership associated with the port reaches the set limit, additional Membership Reports from that switch port are denied until old membership entries are aged out.

If you configure a throttling action of **replace**, when the multicast group membership associated with the port reaches the set limit, additional Membership Reports from that switch port replace existing membership entries.

Static associations of switch ports and multicast groups are counted in the number of multicast group memberships, but they are not affected by the throttling action.

Configuring IGMP throttling

To enable IGMP throttling on a switch port, set the maximum number of group memberships and the throttling action to take, by using the command:

```
set switch port={port-list|all} igmpmaxgroup=1..65535
  igmpaction={deny|replace} [other-options...]
```

To disable IGMP throttling on a switch port, set the maximum number of group memberships to **none** by using the command:

```
set switch port={port-list|all} igmpmaxgroup=none
  [other-options...]
```

To display the IGMP throttling settings for a switch port, use the command:

```
show switch port [= {port-list|all}]
```

Multicast Switching

On Rapier i Series switches, IP multicast switching (in hardware) between VLANs is automatically enabled when **both** of the following are true:

- a multicast routing protocol (DVMRP, PIM-SM or PIM-DM) is enabled
- an interface is configured for that multicast routing protocol.

VLAN tagging is fully supported, and the Time To Live (TTL) value in the IP header is decremented.

Multicast switching cannot be disabled on Rapier i Series switches. Other Rapier Series switches can emulate multicast switching, but this feature is disabled by default (see [“Multicast Switching Emulation”](#) on page 27-32).

Multicast Switching Emulation

Multicast switching emulation does not apply to Rapier i Series switches. For information about multicast switching on Rapier i Series switches, see [“Multicast Switching”](#) on page 27-32.

The multicast routing protocols PIM and DVMRP require CPU resources. For simple VLAN-based switching topologies with no wide area links, multicast hardware switching emulation can be used to forward IP multicast traffic between VLANs. This reduces the latency of the multicast traffic between VLANs. However, this emulation of multicast switching may cause PIM or DVMRP to behave incorrectly. It should be enabled **only** in simple VLAN-based networks and **not** over wide area networks or internet-based networks. Use it only when the following limitations are acceptable:

- Enabling IP multicast switching emulation without enabling a Layer 3 multicasting protocol may defeat IGMP snooping, by preventing group membership entries from being updated in hardware.
- The switch, and therefore DVMRP and PIM, cannot distinguish between different multicast sources (IP hosts) in the same VLAN. This may result in incorrect operation of DVMRP or PIM multicast routing over WAN connections.
- When the switch hardware forwards IP multicast packets between VLANs, it does not decrement their Time To Live values.
- The switch cannot distinguish between groups whose IP addresses end with the same 23 bits. If another switch can forward multicast packets in parallel to the Rapier, multicast data may loop between the Rapier and this other switch.

The IP multicast switching emulation is disabled by default. To enable or disable it, use the commands:

```
enable ip multicastsitching  
disable ip multicastsitching
```

To see the current state of switching emulation, use the command:

```
show ip
```


Multicast VLAN Registration (MVR)

Multicast VLAN Registration (MVR) is used by applications receiving multicast traffic across an Ethernet ring-based service provider network (for example, broadcasting several television channels).

MVR operates on the underlying mechanism of the Internet Group Management Protocol (IGMP) function (see “[Internet Group Management Protocol \(IGMP\)](#)” on page 27-20), and requires that IGMP be enabled. Receiver ports join and leave multicast streams by sending IGMP messages. With IGMP and MVR both enabled, MVR reacts to join and leave messages from the multicast group configured under MVR. IGMP reacts to all messages.

Once MVR is configured, the CPU sets up a forwarding table in which each entry matches the multicast stream to an associated MAC address. The CPU then intercepts the IGMP messages and modifies the forwarding table to include or remove the receiver port as a receiver of the multicast stream. This selectively allows traffic to cross between different VLANs.

Up to five multicast VLANs can be set up on a switch, and a total of 256 group IP addresses can be added to the switch, divided up to belong to different multicast VLANs.

Whenever MVR is enabled on the switch, multicast routing must be disabled.

Dynamic MVR

In *dynamic mode*, after the switch receives IGMP information packets, it processes the data message, and resends those messages to a multicast router through its source ports. The switch knows which multicast groups exist on which interface when it receives a join message from the interface for the group. It forwards the multicast stream to that interface.

Compatible MVR

In *compatible mode*, IGMP report messages are not sent through source ports. The multicast router must be statically configured so it can forward multicast streams to particular interfaces.

Immediate Leave

The *immediate leave* parameter (**imtleave**) allows a receiver port to leave the multicast group as soon an IGMP Leave message is received by the switch on that port. Immediate leave can be specified in the following commands:

```
create ip mvr vlan=vlan-id sourceport=port-list
receiverport=port-list [imtleave=port-list]
[mode={dynamic|compatible}]

set ip mvr vlan=vlan-id [imtleave=port-list] [mode={dynamic|
compatible}] [receiverport=port-list] [sourceport=port-
list]
```

If immediate leave is not specified, the switch sends an IGMP query on that port and waits for the IGMP group membership reports when it receives an IGMP Leave message. If it receives no reports within the configured IGMP time period (see the [set ip igmp command on page 27-91](#)), the receiver port leaves

the multicast group. The port numbers specified must be members of the receiver port list. If the immediate leave parameter is not specified, the function is disabled on all receiver ports.

Configuring MVR

MVR is disabled by default, and must be enabled on the switch to start multicast VLAN registration. To enable or disable MVR, use the commands:

```
enable ip mvr
disable ip mvr
```

To create MVR on, or remove MVR from the switch, use the commands:

```
create ip mvr vlan=vlan-id sourceport=port-list
receiverport=port-list [other parameters]
destroy ip mvr vlan=vlan-id
```

To add or delete an MVR IP multicast group address or range of addresses, use the command:

```
add ip mvr vlan=vlan-id groupaddress=ipadd[-ipadd]
delete ip mvr vlan=vlan-id groupaddress=ipadd[-ipadd]
```

To set parameters for the MVR mode, receiver ports, source ports, and ports with the immediate leave function, use the command:

```
set ip mvr vlan=vlan-id [imtleave=port-list] [mode={dynamic|
compatible}] [receiverport=port-list] [sourceport=port-
list]
```

To display all information about the MVR configuration, use the command:

```
show ip mvr [vlan=vlan-id]
```

Configuration Examples

This section contains the following multicasting configurations that use IGMP:

- [Static IGMP](#)
- [Multicasting using DVMRP](#)
- [Protocol Independent Multicast \(PIM\)](#)

Static IGMP

The following example shows how to create a static IGMP association. It assumes that *vlan1* has already been configured as an IP interface on the switch.

1. Enable IGMP on the switch.

```
enable ip igmp
```

2. Enable IGMP on vlan1.

This must be done before the static IGMP association is created.

```
enable ip igmp interface=vlan1
```

3. Create the static IGMP association.

The multicast data for the group specified by the **destination** parameter is forwarded over ports specified by the **port** parameter. If the **port** parameter is not entered, the association defaults to all ports belonging to the interface.

```
create ip igmp destination=224.1.2.3 interface=vlan1  
port=1-4
```

4. Check the configuration.

Check that the static IGMP association has been created and IGMP is enabled.

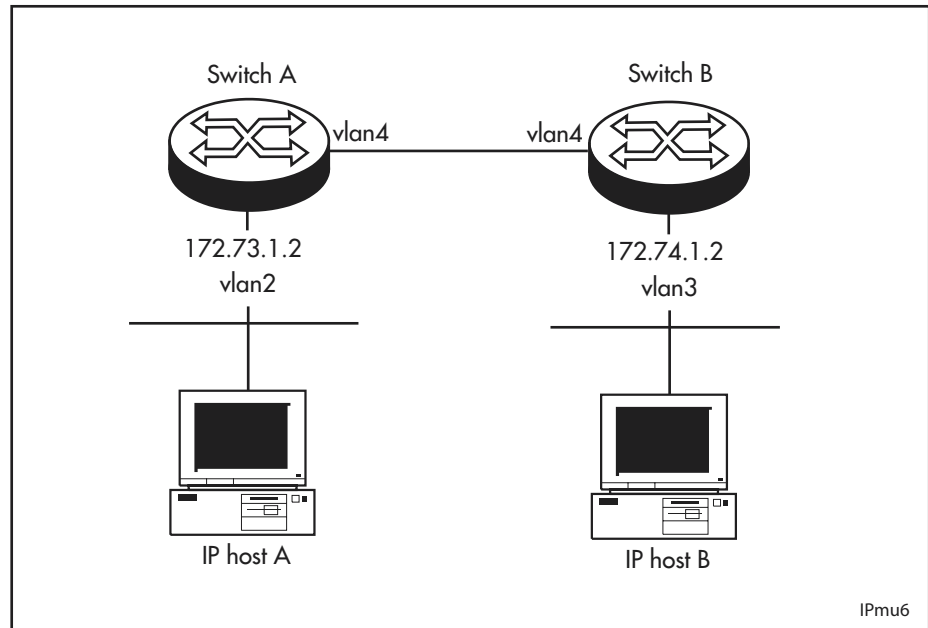
```
show ip igmp destination=224.1.2.3 interface=vlan1
```

Multicasting using DVMRP

This example allows IP hosts to send data to and receive data from the multicast groups. Multicast group management uses IGMP, and multicast routing between the switches uses DVMRP. The example assumes that each switch starts from the default configuration.

Interfaces with DVMRP enabled must also have IGMP enabled.

Figure 27-3: Multicast configuration example using IGMP and DVMRP.

**To configure Switch A****1. Set the system name.**

Set a unique system name for the switch.

```
set sys name=A-dvmrp
```

2. Configure the VLANs.

Configure the admin VLAN including switch port 6, and the uplink VLAN including port 25.

```
create vlan=admin vid=2
create vlan=uplink vid=4
add vlan=admin port=6
add vlan=uplink port=25
```

3. Configure IP.

Enable the IP module, and assign IP addresses to the interfaces.

```
enable ip
add ip interface=vlan2 ipaddress=172.73.1.2
mask=255.255.255.0
add ip interface=vlan4 ipaddress=189.124.7.8
mask=255.255.255.0
```

4. Configure IGMP.

Enable IGMP on the switch for multicast group management.

```
enable ip igmp
```

Enable IGMP on the VLANs.

```
enable ip igmp interface=vlan2
enable ip igmp interface=vlan4
```

IGMP snooping is enabled by default, so does not need to be configured.

5. Configure DVMRP.

Enable DVMRP for multicast routing.

```
enable dvmrp
```

Enable DVMRP on the VLAN.

```
add dvmrp interface=vlan2 metric=1
```

```
add dvmrp interface=vlan4 metric=1
```

To configure Switch B**1. Set the system name.**

Set a unique system name for the switch.

```
set sys name=B-dvmrp
```

2. Configure the VLANs.

Configure the sales VLAN including switch port 1, and the uplink VLAN including port 25.

```
create vlan
```

```
n=sales vid=3
```

```
create vlan=uplink vid=4
```

```
add vlan=sales port=1
```

```
add vlan=uplink port=25
```

3. Configure IP.

Enable IP on the switch, and assign IP addresses to the interfaces used by DVMRP for multicast routing.

```
enable ip
```

```
add ip interface=vlan3 ipaddress=172.74.1.2  
mask=255.255.255.0
```

```
add ip interface=vlan4 ipaddress=189.124.7.9  
mask=255.255.255.0
```

4. Configure IGMP.

Enable IGMP on the switch, and on the interfaces that have IP hosts connected to them, so that the switch can maintain its group membership data.

```
enable ip igmp
```

```
enable ip igmp interface=vlan3
```

```
enable ip igmp interface=vlan4
```

IGMP snooping is enabled by default, so does not need to be configured.

5. Configure DVMRP.

Enable DVMRP on the switch and on each interface over which it is used for multicast routing.

```
enable dvmrp
```

```
add dvmrp interface=vlan3 metric=1
```

```
add dvmrp interface=vlan4 metric=1
```

Confirm multicasting

When the switches have been configured, the DVMRP route exchange takes a few seconds. Then the IP hosts connected to these interfaces can send and receive multicasts.

1. Test multicasting.

Test whether IP multicasting is successful by sending IP multicast data between hosts connected to each of the switches. Check that IGMP report and leave messages are correctly processed by having hosts leave and join groups, and check that IP multicast data forwarding stops and starts correctly.

2. Check the multicast state.

To check each switch, use the commands:

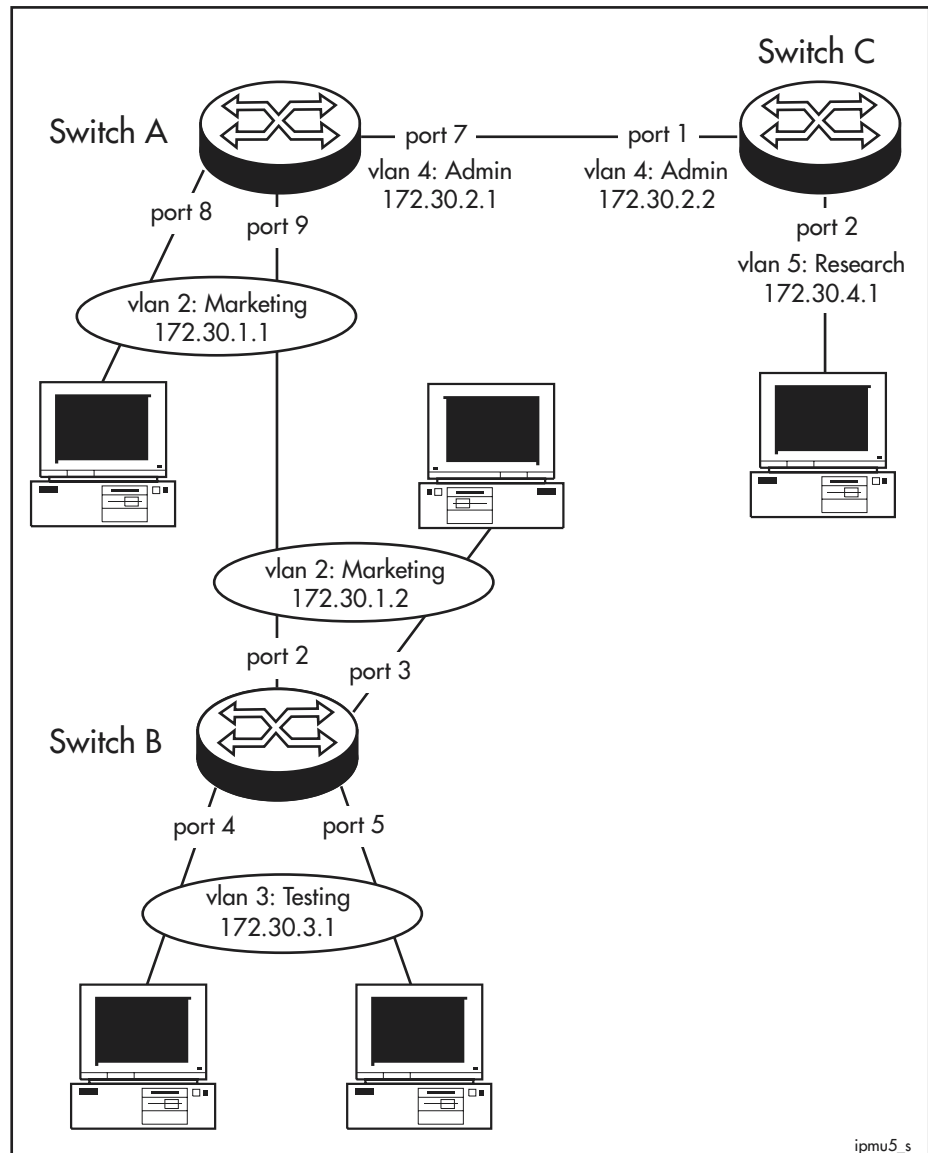
```
show dvmrp
show ip igmp
show ip route multicast
```

Protocol Independent Multicast (PIM)

These examples use PIM-SM or PIM-DM for multicast routing between three switches. The network topology is the same for each example ([Figure 27-4](#)). Multicast group management uses IGMP. The examples assume that each switch starts from the default configuration.

Some interface and port types mentioned in these examples may not be supported on your switch because they vary depending on the model and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the switch.

Figure 27-4: Multicast configuration using PIM sparse or dense mode.



PIM-SM This example uses PIM Sparse Mode and allows IP hosts to send data to and receive data from the multicast groups 225.1.0.0 to 225.1.0.255. The configuration of Switches A, B, and C are very similar, but Switch A is the only switch configured as a PIM Bootstrap Router Candidate and a PIM rendezvous point candidate.

Some interface and port types mentioned in this example may not be supported on your switch. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the switch.

To configure Switch A

1. Set the system name for the switch.

```
set sys name=A-pim-rp
```

2. Configure the VLANs.

Configure the *marketing* VLAN, including ports 8 and 9.

```
create vlan=marketing vid=2
add vlan=2 port=8,9
```

Configure the *admin* VLAN, including port 7.

```
create vlan=admin vid=4
add vlan=4 port=7
```

3. Configure IP.

Enable IP and assign IP addresses for the VLAN interfaces.

```
enable ip

add ip interface=vlan2 ipaddress=172.30.1.1
mask=255.255.255.0

add ip interface=vlan4 ipaddress=172.30.2.1
mask=255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan2
add ip rip int=vlan4
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan2
enable ip igmp interface=vlan4
```

IGMP snooping is enabled by default, so does not need to be configured.

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan2
add pim interface=vlan4
```

The network must have a PIM bootstrap router, so at least one switch in the network must be configured as a Bootstrap Router Candidate. Set this switch to be the Bootstrap Router Candidate.

```
add pim bsr candidate
```

At least one switch in each multicast group must be a PIM rendezvous point (RP) for the multicast group, so at least one switch in each group must be configured as a rendezvous point candidate. Set this switch to be an RP candidate.

```
add pim rpcandidate group=225.1.0.0 mask=255.255.255.0
```

Enable PIM multicast routing.

```
enable pim
```


To configure Switch B

1. Set the system name.

Set a unique system name on the switch.

```
set sys name=B-pim
```

2. Configure the VLANs.

Configure the *marketing* VLAN, including ports 2 and 3.

```
create vlan=marketing vid=2
add vlan=2 port=2,3
```

Configure the *testing* VLAN, including ports 4 and 5.

```
create vlan=testing vid=3
add vlan=3 port=4,5
```

3. Configure IP.

Enable IP, and assign IP addresses to the VLAN interfaces.

```
enable ip
add ip interface=vlan2 ipaddress=172.30.1.2 mask=
255.255.255.0
add ip interface=vlan3 ipaddress=172.30.3.1 mask=
255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan2
add ip rip int=vlan3
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan2
enable ip igmp interface=vlan3
```

IGMP snooping is enabled by default, so does not need to be configured.

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan2
add pim interface=vlan3
```

Enable PIM multicast routing.

```
enable pim
```

To configure Switch C

1. Set the system name.

Set a unique system name on the switch.

```
set sys name=C-pim
```

2. Configure the VLANs.

Configure the *admin* VLAN, including port 1.

```
create vlan=admin vid=4
add vlan=4 port=1
```

Configure the *research* VLAN, including port 2.

```
create vlan=research vid=5
add vlan=5 port=2
```

3. Configure IP.

Enable IP on the switch.

```
enable ip
```

Assign IP addresses to the VLAN interfaces.

```
add ip interface=vlan4 ipaddress=172.30.2.2
mask= 255.255.255.0
add ip interface=vlan5 ipaddress=172.30.4.1
mask= 255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan4
add ip rip int=vlan5
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan4
enable ip igmp interface=vlan5
```

IGMP snooping is enabled by default, so does not need to be configured.

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan5
add pim interface=vlan4
```

Enable PIM multicast routing.

```
enable pim
```

Confirm multicasting

When the three switches have been configured, RIP takes a few seconds to distribute the unicast routing information to all routers. The IP hosts connected to these interfaces can then send and receive multicasts.

1. Test multicasting.

Test whether IP multicasting is successful by sending IP multicast data between hosts connected to each of the switches. Check that IGMP report and leave messages are correctly processed by having hosts leave and join groups.

2. Check the multicast state.

To check each switch, use the commands:

```
show pim
show ip igmp
show ip route multicast
```

PIM-DM This example uses PIM Dense Mode for multicast routing between switches in the same topology as the PIM Sparse Mode example ([Figure 27-4 on page 27-39](#)). Multicast group management uses IGMP. The example assumes that each switch starts from the default configuration.

Some interface and port types mentioned in this example may not be supported on your switch. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the switch.

The configurations of Switch A, B, and C are identical except for names and interfaces.

To configure Switch A

1. Set the system name for the switch.

```
set sys name=A-pim-dm
```

2. Configure the VLANs.

Configure the *marketing* VLAN, including ports 8 and 9.

```
create vlan=marketing vid=2
add vlan=2 port=8,9
```

Configure the *admin* VLAN, including port 7.

```
create vlan=admin vid=4
add vlan=4 port=7
```

3. Configure IP.

Enable IP and assign IP addresses for the VLAN interfaces on the switch.

```
enable ip
add ip interface=vlan2 ipaddress=172.30.1.1
mask=255.255.255.0
add ip interface=vlan4 ipaddress=172.30.2.1
mask=255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan2
add ip rip int=vlan4
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan2
```

```
enable ip igmp interface=vlan4
```

IGMP snooping is enabled by default, so does not need to be configured.

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan2 mode=dense
```

```
add pim interface=vlan4 mode=dense
```

Enable PIM multicast routing.

```
enable pim
```

To configure Switch B

1. Set the system name.

Set a unique system name on the switch.

```
set sys name=B-pim
```

2. Configure the VLANs.

Configure the *marketing* VLAN, including ports 2 and 3.

```
create vlan=marketing vid=2
```

```
add vlan=2 port=2,3
```

Configure the *testing* VLAN, including ports 4 and 5.

```
create vlan=testing vid=3
```

```
add vlan=3 port=4,5
```

3. Configure IP.

Enable IP, and assign IP addresses to the VLAN interfaces.

```
enable ip
```

```
add ip interface=vlan2 ipaddress=172.30.1.2  
mask=255.255.255.0
```

```
add ip interface=vlan3 ipaddress=172.30.3.1  
mask=255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan2
```

```
add ip rip int=vlan3
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan2
```

```
enable ip igmp interface=vlan3
```

IGMP snooping is enabled by default, so does not need to be configured.

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan2 mode=dense
```

```
add pim interface=vlan3 mode=dense
```

Enable PIM multicast routing.

```
enable pim
```

To configure Switch C

1. Set the system name.

Set a unique system name on the switch.

```
set sys name=C-pim
```

2. Configure the VLANs.

Configure the *admin* VLAN, including port 1.

```
create vlan=admin vid=4
```

```
add vlan=4 port=1
```

Configure the *research* VLAN, including port 2.

```
create vlan=research vid=5
```

```
add vlan=5 port=2
```

3. Configure IP.

Enable IP on the switch.

```
enable ip
```

Assign IP addresses to the VLAN interfaces.

```
add ip interface=vlan4 ipaddress=172.30.2.2  
mask= 255.255.255.0
```

```
add ip interface=vlan5 ipaddress=172.30.4.1  
mask= 255.255.255.0
```

4. Configure a unicast routing protocol.

Enable RIP over all interfaces.

```
add ip rip int=vlan4
```

```
add ip rip int=vlan5
```

5. Configure IGMP.

Enable IGMP on the switch for group management.

```
enable ip igmp
```

Enable IGMP on each interface, so that IGMP can find which multicast groups have hosts connected to the interfaces.

```
enable ip igmp interface=vlan4
enable ip igmp interface=vlan5
```

6. Configure PIM.

Define PIM interfaces for the VLAN interfaces.

```
add pim interface=vlan5 mode=dense
add pim interface=vlan4 mode=dense
```

Enable PIM multicast routing.

```
enable pim
```

Confirm multicasting

When the three switches have been configured, RIP takes a few seconds to distribute the unicast routing information to all routers. Then the IP hosts connected to these interfaces can send and receive multicasts.

1. Test multicasting.

Test whether IP multicasting is successful by sending IP multicast data between hosts connected to each of the switches. Check that IGMP report and leave messages are correctly processed by having hosts leave and join groups.

2. Check the multicast state.

To check each switch, use the commands:

```
show pim
show ip igmp
show ip route multicast
```

Command Reference

This section describes the commands available on the switch to configure IGMP for multicast group management, and the multicast routing protocols DVMRP (Distance Vector Multicast Routing Protocol), PIM-SM (Protocol Independent Multicast - Sparse Mode) and PIM-DM (Protocol Independent Multicast - Dense Mode).

Some interface and port types mentioned in this chapter may not be supported on your switch. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the switch.

The shortest valid command is denoted by capital letters in the Syntax section. See [“Conventions” on page lxvi of About this Software Reference](#) in the front of this manual for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of error messages and their meanings.

add dvmrp interface

Syntax ADD DVMrp INTerface=*interface* [DLC=1..1024] [METric=1..32]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command adds the specified interface to the DVMRP interface list, and starts DVMRP processes on the interface.

The **interface** parameter specifies the IP interface. The Layer 2 interface must already be configured. The IP interface must not already be assigned to the DVMRP module. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#). Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

The **metric** parameter specifies the metrics on a DVMRP interface. This metric is added to all routes that are learned via this interface. The default is 1.

Examples To add the interface vlan2 as a DVMRP interface, use the command:

```
add dvm int=vlan2
```

Related Commands

- [delete dvmrp interface](#)
- [enable dvmrp](#)
- [set dvmrp interface](#)
- [show dvmrp](#)

add igmp filter

Syntax ADD IGMP FILTER=*filter-id* GROUPaddress={*ipadd*|*ipadd-ipadd*}
[MSGType={QUERY|REPORT|LEAVE}] [ACTION={INCLUDE|
EXCLUDE}] [ENTRY=1..65535]

where:

- *filter-id* is a decimal number from 1 to 99.
- *ipadd* is an IP address in dotted decimal notation.

Description This command adds an entry to an IGMP filter. IGMP filters control a port's membership of multicast groups by filtering incoming IGMP messages from hosts attached to the port.

To take effect, the filter must be applied to a switch port by using the [set switch port](#) command.

The **filter** parameter specifies the number of the filter to add the entry to. The specified filter must have been created previously by using the [create igmp filter](#) command.

The **groupaddress** parameter specifies an IP multicast group address or a range of IP multicast group addresses to match. Set **groupaddress** to:

- 0.0.0.0 to filter IGMP general query messages
- a multicast address or a range of multicast addresses to filter IGMP group-specific query messages, report messages, and leave messages.

The **msgtype** parameter specifies the type of incoming IGMP message to match. If you specify **query**, the filter will match IGMP general and group-specific query messages. If you specify **report**, the filter will match IGMP report messages. If you specify **leave**, the filter will match IGMP leave messages. The default is **report**.

The **action** parameter specifies the action to take when an IGMP message with a message type matching **msgtype** and a group address matching **groupaddress** is received. If you specify **include**, the message is processed as normal by IGMP. If you specify **exclude**, the message is excluded from processing by IGMP, and the packet is discarded. The default is **include**.

If an IGMP filter contains at least one entry for a particular IGMP message type, then messages of the same type for group addresses that do not match any entries in the filter are implicitly excluded and the packets are discarded.

The **entry** parameter specifies the position of the entry in the filter, and identifies the entry in the filter. The specified entry number must not already be used by another entry. If you do not specify an entry number, the entry is added after the last entry in the filter if there is a free position, or in the last unused position if the last position is already in use.

Examples To add an entry to filter 6 to accept Membership Reports for multicast group addresses in the range 229.1.1.2 to 230.1.2.3, use the command:

```
add igmp fil=6 msgt=rep gro=229.1.1.2-230.1.2.3
```

To add an entry at position 16 in filter 3 to deny Membership Reports for multicast group addresses in the range 231.1.1.20 to 231.1.5.3, use the command:

```
add igmp fil=3 ent=16 msgt=rep gro=231.1.1.20-231.1.5.3
ac=excl
```

To add an entry to filter 1 to exclude all general queries, use the command:

```
add igmp fil=1 msgt=que gro=0.0.0.0 ac=excl
```

Related Commands

- [create igmp filter](#)
- [delete igmp filter](#)
- [destroy igmp filter](#)
- [set igmp filter](#)
- [show igmp filter](#)

add igmpsnooping routeraddress

Syntax ADD IGMPsNooping ROUTERAddress=*ipaddr-list*

where *ipaddr-list* is a reserved IP multicast address in dotted decimal notation, or a comma-separated list of reserved IP multicast addresses

Description This command adds reserved IP multicast addresses to the list of router multicast addresses. The IP address specified must be from 224.0.0.1 to 224.0.0.255. This command is valid when IGMP snooping router mode is set to IP with the **set igmpsnooping routermode** command.

Examples To add addresses 224.0.0.25 and 224.0.0.86 to the router multicast address list, use the command:

```
add igmpsn routera=224.0.0.25,224.0.0.86
```

Related Commands

- [delete igmpsnooping routeraddress](#)
- [set igmpsnooping routermode](#)
- [show igmpsnooping routeraddress](#)

add igmpsnooping vlan

Syntax ADD IGMPSPNooping VLAN={*vlan-name*|1..4094}
ROUTERPort=*port-list*

where

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be **all**.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

Description This command configures ports as multicast router ports. The switch forwards relevant IGMP messages and other IP multicast traffic out these ports. This is useful for network configurations in which the switch's learning process cannot identify all multicast router ports.

The **vlan** parameter specifies a VLAN. The ports are only treated as multicast router ports for that VLAN, not for other VLANs they belong to. There is no default.

The **routerport** parameter specifies the ports in the VLAN that have multicast routers attached to them. There is no default.

Examples To specify that port 3 in vlan2 is a multicast router port, use the command:

```
add igmpsn vlan=2 routerp=3
```

Related Commands

- [delete igmpsnooping routeraddress](#)
- [delete igmpsnooping vlan](#)
- [set igmpsnooping routermode](#)
- [show igmpsnooping](#)
- [show igmpsnooping routeraddress](#)

add ip igmp destination

Syntax ADD IP IGMP DESTination=*ipaddress* INTERface=*interface*
PORT={ALL|*port-list*}

where:

- *ipaddress* is an existing IGMP group destination address.
- *interface* is the name of the interface over which multicast data is forwarded. This must be a VLAN interface.
- *port-list* is a port number, a range of port numbers (specified as a-b), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet port, including uplink ports.

Description This command adds additional ports through which multicast data is forwarded.

The **destination** parameter specifies the IP address to which multicast group address data is forwarded.

The **interface** parameter specifies the interface over which multicast data is forwarded. This must be a VLAN interface, for example VLAN1.

The static IGMP association identified by the **destination** and **interface** parameters must already exist.

The **port** parameter specifies the ports through which multicast data is forwarded. If any of the ports specified in the port list are already part of the association or are not valid for the specified interface, an error message is displayed.

A port may belong to several associations if it belongs to several interfaces (i.e. if there are overlapping VLANs). If one of the ports in the list already has a dynamic IGMP host, it is replaced by the new static entry.

If **all** is specified, all ports belonging to that interface forward multicast data.

Examples To add port 5 to the list of ports through which multicast data for 224.1.2.3 is forwarded over *vlan1*, use the command:

```
add ip igmp des=224.1.2.3 int=vlan1 po=5
```

Related Commands

- [create ip igmp destination](#)
- [delete ip igmp destination](#)
- [destroy ip igmp destination](#)
- [show ip igmp](#)

add ip mvr

Syntax `ADD IP MVR VLAN=1..4094 GROupaddress=ipadd[-ipadd]`

where *ipadd* is an IP address in dotted decimal notation

Description This command adds an MVR IP multicast group address or range of addresses on a switch. All source ports and receiver ports belonging to this multicast group receive multicast data sent to this group address.

The **vlan** parameter specifies the VLAN identifier with which the multicast VLAN is associated.

The **groupaddress** parameter specifies the multicast group IP address or range of IP addresses that belong to the multicast VLAN.

Examples To add a multicast IP group address on a multicast VLAN, use the command:

```
add ip mvr vlan=22 gro=230.1.1.1
```

Related Commands [delete ip mvr](#)
[show ip mvr](#)

add pim bsrcandidate

Syntax ADD PIM BSRCandidate [PREference=0..255]
[HASHmasklength=0..32] [BSMinterval={10..15000|
Default}] [INTERface=*interface*]

where *interface* is the name of the interface over which multicast data is forwarded. The interface can be either a VLAN (e.g. vlan1) or a local interface (e.g. local1).

Description This command configures the switch to be a Bootstrap Router candidate.

The **preference** parameter specifies the preference for the switch to become the bootstrap router. A higher number means a higher priority. The default is 1.

The **hashmasklength** parameter specifies the number of bits of the group number to use when selecting a rendezvous point (RP) candidate if this switch becomes the BSR. A higher number increases the spread of groups across RPs. The default is 30.

Note that software releases prior to 2.7.3 did not correctly support the PIM hash mask length option. As a result, the RP selection calculation differs between this release and release versions prior to 2.7.3. If a network contains switches running a mixture of versions, this leads to incorrect forwarding behaviour. To avoid this issue, either ensure that all devices on the network correctly support the hash mask length option (recommended), or ensure that the following **both** hold:

- The hash mask length option on all BSR candidates is configured to 4 bits. This implies that all BSR candidates must be running 2.7.3 or later.
- All RP candidates use a common prefix of 224.0.0.0/240.0.0.0. This has the side effect of collapsing all groups to use a single PIM RP.

The **bsminterval** parameter can be used to specify the time period in seconds at which the switch sends bootstrap messages when it is elected as the bootstrap router. The default is 60 seconds. This timer is now set with the [set pim](#) command by preference because it applies globally to the PIM-SM domain, but this parameter has been maintained to ensure backwards compatibility.

The **interface** parameter specifies an interface for the switch to use when advertising itself as a candidate bootstrap router. The IP address of this interface is advertised by the router. The interface supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified, the switch instead advertises its first active IP interface.

Examples To add the switch as a Bootstrap Router Candidate to a PIM domain, with a preference of 10 to become the bootstrap router in the domain and a hash mask length of 0, use the command:

```
add pim bsrcandidate preference=10 hasmasklength=0
```

Related Commands

- [delete pim bsrcandidate](#)
- [enable pim](#)
- [set pim bsrcandidate](#)
- [show pim](#)
- [show pim bsrcandidate](#)

add pim interface

Syntax `ADD PIM INTeRface=interface [DLci=1..1024]
[DRPriority=0..4294967295] [ELectby={DRPriority|
IPaddress}] [HEllointerval={10..15000|DEFAult|65535}]
[MODE={Dense|Sparse}] [SRCapable={Yes|No}]`

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command adds the specified IP interface to the PIM interface list so that PIM multicast routing can operate on this interface. Valid interfaces are:

To see a list of current interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#). Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlci** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

The **drpriority** parameter specifies the preference for the switch to become the designated router (DR) on this interface when **electby=drpriority**. A higher value indicates a greater preference. The default is 1.

Note that for compatibility with previous versions, a DR priority of 65535 is treated as **electby=ipaddress** if **electby** has not been specified. If **electby=drpriority** is specified and **drpriority=65535**, then the DR priority is set to 65535.

The **electby** parameter determines how the switch elects the designated router for this interface. If **drpriority** is specified, the interface transmits its DR priority in its hello messages, which allows DR election by priority. If **ipaddress** is specified, the switch does not transmit its DR priority, which causes election by IP address. The default is **drpriority**. Note that a switch with **electby=drpriority** may still elect by IP address when it does not receive DR priority in any one of its neighbours' Hello messages. Election by DR priority is possible only when all routers on the interface supply their DR priority.

The **hellointerval** parameter specifies the interval at which the switch sends Hello messages from this interface. Setting the **hellointerval** parameter to 65535 results in a Hello message being sent with a hold time of 65535, which means "infinity". A router receiving this switch's Hello never expires this switch as a PIM neighbour. This can be useful on point-to-point links. The default is 30 seconds.

The **mode** parameter specifies the PIM operating mode for the interface. The default is **sparse**. All interfaces should have the same mode setting unless the switch is a Multicast Border Router.

The **srcapable** parameter indicates whether this interface originates or processes State Refresh messages. The default is **no**. This parameter applies to dense mode interfaces.

Examples To add interface vlan2 to the PIM-SM interface list, with a priority of 3 to become the designated router for the subnetwork, use the command:

```
add pim int=vlan2 drp=3
```

Related Commands

- delete pim interface
- enable pim
- reset pim interface
- set pim interface
- show pim
- show pim interface

add pim rpcandidate

Syntax ADD PIM RPCandidate[=*rp-address*] GROup=*group-address*
[ADVinterval={10..15000|DEFault}] [INTERface=*interface*]
[MASK=*ipaddress*] [PRIOrity=0..255]

where:

- *group-address* is the IP address of the multicast group in dotted decimal notation.
- *ipaddress* is an IP address in dotted decimal notation.
- *rp-address* is an IP address in dotted decimal notation.
- *interface* is the name of a VLAN (e.g. vlan1) or a local interface (e.g. local1).

Description This command configures the switch to be a rendezvous point candidate for specific multicast groups. There is no limitation on the number of groups or range of groups.

The **rpcandidate** parameter, if specified with a value, is the IP address of the rendezvous point for the multicast group(s). This option can be used to create static RP mappings for networks in which the bootstrap mechanism cannot be used. If the bootstrap mechanism is also running, a static RP mapping takes precedence.

The **group** parameter specifies the multicast group(s) to which the switch is a rendezvous point candidate.

The **advinterval** parameter can be used to specify the time period in seconds at which the switch sends C-RP-Advertisements. The default is 60 seconds. This timer is now set with the **set pim** command by preference because the switch sends C-RP-Advertisements at the same rate for all groups for which it is an RP candidate, but this parameter has been maintained to ensure backwards compatibility. This parameter does not apply to static RP mappings.

The **interface** parameter specifies an interface for the switch to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of the of this interface is advertised by the switch. The **interface** supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified, the switch advertises its first active IP interface instead.

The **mask** parameter specifies the mask for the multicast group address specified in the **group** parameter. This is useful when configuring multiple multicast groups with a common rendezvous point (RP). The default mask is 255.255.255.255.

The **priority** parameter specifies the preference for the switch to become the rendezvous point for the multicast group. A lower value indicates a higher priority. The default is 192. This parameter does not apply to static RP mappings.

Note that the switch has the same values for **priority** for all multicast groups for which it is a rendezvous point candidate, so changing this switch's priority to be the RP for one group changes it for all groups.

Examples To configure the switch to advertise that it is an RP candidate with a priority of 10 to become the RP for the multicast group with address 224.1.1.98, use the command:

```
add pim rpc gro=224.1.1.98 prio=10
```

Related Commands

- [delete pim rpcandidate](#)
- [enable pim](#)
- [set pim rpcandidate](#)
- [show pim](#)
- [show pim rpcandidate](#)

create igmp filter

Syntax `CREate IGMP FILter=filter-id`

where *filter-id* is a decimal number from 1 to 99

Description This command creates an IGMP filter. IGMP filters control a port's membership of multicast groups by filtering incoming IGMP messages from hosts attached to the port.

The **filter** parameter specifies the number of the filter to create, and is used to identify the filter. A filter with the specified number must not already exist.

You can add entries to the filter to match specific multicast groups by using the **add igmp filter** command.

For the filter to take effect, you must apply the filter to a switch port using the [set switch port command on page 8-109 of Chapter 8, Switching](#).

Applying an empty IGMP filter (a filter with no entries) to a switch port allows all incoming IGMP messages to be processed as normal.

Examples To create a filter with a filter ID of 6, use the command:

```
cre igmp fil=6
```

Related Commands

- [add igmp filter](#)
- [delete igmp filter](#)
- [destroy igmp filter](#)
- [set igmp filter](#)
- [show igmp filter](#)

create ip igmp destination

Syntax CREate IP IGMP DEStination=*ipaddress* INTErface=*interface*
[Port={ALL|*port-list*}]

where:

- *ipaddress* is an existing IGMP group destination address.
- *interface* is the name of the interface over which multicast data is forwarded.
- *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet port, including uplink ports.

Description This command creates a static multicast association to forward multicast data from a multicast group to one or more ports.

The **destination** parameter specifies the IP address to which multicast group address data is forwarded.

The **interface** parameter specifies the interface over which multicast data is forwarded.

The static IGMP association identified by the **destination** and **interface** parameters must not already exist.

The **port** parameter specifies the ports through which multicast data is forwarded. If any of the ports specified in the port list are not valid ports for the specified interface, an error message is displayed. An empty port list can be specified by giving no value to the **port** parameter. Ports may be added later with the **add ip igmp destination** command.

If **all** is specified or if the **port** parameter is not entered, all ports that belong to that interface forward multicast data.

Since static IGMP associations are identified by the combination of destination and interface, one destination or interface may belong to several different associations. Also, ports may belong to several associations if there are overlapping VLANs. There is no conflict with existing standard (dynamic) IGMP hosts: if a new static association's port already has a dynamic IGMP host, the new static entry replaces it.

IGMP destinations added with this command never time out. They can be removed with the **destroy ip igmp destination** command.

Examples To forward multicast data to 224.1.2.3 out ports 1 to 4 using *vlan1*, use the command:

```
cre ip igmp des=224.1.2.3 int=vlan1 po=1-4
```

Related Commands

- [add ip igmp destination](#)
- [delete ip igmp destination](#)
- [destroy ip igmp destination](#)
- [show ip igmp](#)

create ip mvr

Syntax CREate IP MVR VLAN=1..4094 SOurceport=*port-list*
RECeiverport=*port-list* [IMTLeave=*port-list*]
[MODE={DYnamic|COMpatible}]

where *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

Description This command creates Multicast VLAN Registration on the switch.

The **vlan** parameter specifies the VLAN Identifier that the multicast VLAN is associated with.

The **sourceport** parameter specifies the uplink (source) ports that send and receive multicast data to and from the multicast VLAN. All source ports must belong to a single multicast VLAN.

The **receiverport** parameter specifies the port(s) that receives multicast data when they are members of the multicast group. The receiver ports cannot belong to the multicast VLAN.

The **imtleave** parameter specifies the ports on which the Immediate Leave feature is enabled. If this parameter is specified, the receiver ports immediately leave the multicast group as soon an IGMP Leave message is received by the switch on that port. Otherwise, the switch sends an IGMP query on the port and waits for the IGMP group membership reports when it receives an IGMP Leave message. If no reports are received within the configured IGMP time period (see the [set ip igmp command on page 27-91](#)), the receiver port leaves the multicast group. The port numbers specified must be members of the **receiverport** list. If the **imtleave** parameter is not specified, the function is disabled on all receiver ports.

The **mode** parameter specifies the mode of the MVR operation. If **dynamic** is specified, IGMP reports are sent out through the source ports of the multicast VLAN. If **compatible** is specified, IGMP reports are not sent out. The default is **dynamic**.

Examples To create MVR, use the command:

```
cre ip mvr vlan=22 so=8,9 rec=1-5, 6-8 imtl=2,4
```

Related Commands [destroy ip mvr](#)
[set ip mvr](#)
[show ip mvr](#)

delete dvmrp interface

Syntax DELEte DVMrp INTerface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command deletes the specified interface from the DVMRP interface list, stops all DVMRP processes for the interface, and deletes all routing information generated by this interface.

The **interface** parameter specifies the IP interface. The IP interface must already be assigned to the DVMRP module. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#). Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To delete interface vlan2 from the DVMRP interface list, use the command:

```
del dvm int=vlan2
```

Related Commands

- [add dvmrp interface](#)
- [disable dvmrp](#)
- [reset dvmrp interface](#)
- [show dvmrp](#)

delete igmp filter

Syntax `DELEte IGMP FILter=filter-id ENTRy={1..65535|ALL}`

where *filter-id* is a decimal number from 1 to 99

Description This command deletes the specified entry or all entries from an IGMP filter.

The **filter** parameter specifies the number of the filter that the entry belongs to. A filter with the specified number must already exist.

The **entry** parameter specifies the entry to delete. The specified entry must exist. If you specify **all**, then all entries are deleted from the filter.

Examples To delete entry 21 from filter 5, use the command:

```
del igmp fil=5 entry=21
```

Related Commands

- [add igmp filter](#)
- [create igmp filter](#)
- [destroy igmp filter](#)
- [set igmp filter](#)
- [show igmp filter](#)

delete igmpsnooping routeraddress

Syntax `DELEte IGMPsNooping ROUTERAddress=ipaddr-list`

where *ipaddr-list* is a reserved IP multicast address in dotted decimal notation, or a comma-separated list of reserved IP multicast addresses

Description This command deletes reserved IP multicast addresses from the list of router multicast addresses. The IP address specified must be from 224.0.0.1 to 224.0.0.255. This command is only valid if IGMP Snooping router mode is set to IP with the **set igmpsnooping routermode** command.

Examples To remove addresses 224.0.0.25 and 224.0.0.86 from the router multicast address list, use the command:

```
del igmpsn routera=224.0.0.25,224.0.0.86
```

Related Commands

- [add igmpsnooping routeraddress](#)
- [set igmpsnooping routermode](#)
- [show igmpsnooping routeraddress](#)

delete igmpsnooping vlan

Syntax `DELEte IGMPsNooping vlan={vlan-name|1..4094}
routerport=port-list`

where

- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.
- *port-list* is a port number, range (specified as *n-m*), or comma-separated list of numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

Description This command stops IGMP snooping from treating ports as multicast router ports. The switch stops forwarding IGMP messages and other IP multicast traffic out these ports.

The **vlan** parameter specifies the VLAN for which the ports are no longer to be treated as multicast router ports. There is no default.

The **routerport** parameter specifies the ports in the VLAN that no longer have multicast routers attached to them. There is no default.

Examples To stop port 3 in vlan2 from being a multicast router port, use the command:

```
del igmpsn vlan=2 routerp=3
```

Related Commands

- [add igmpsnooping routeraddress](#)
- [add igmpsnooping vlan](#)
- [set igmpsnooping routermode](#)
- [show igmpsnooping](#)
- [show igmpsnooping routeraddress](#)

delete ip igmp destination

Syntax DELEte IP IGMP DESTination=*ipaddress* INTerface=*interface*
PORt={ALL|*port-list*}

where:

- *ipaddress* is an existing IGMP group destination address.
- *interface* is the name of the interface over which multicast data is forwarded. This must be a VLAN interface.
- *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet port, including uplink ports.

Description This command deletes ports from a static multicast group. Multicast data from the multicast group are no longer forwarded out the port. The static association identified by the **destination** and **interface** parameters must exist for this command to succeed.

When ports specified in the port list are not assigned to this static association, an error message is displayed. When the last port is removed, the static association still exists, although it has no functionality until ports are added again. To destroy the entire static association, use the **destroy ip igmp destination** command.

Examples To remove ports 1-4 from the list of ports through which multicast data for 224.1.2.3 is forwarded over *vlan1*, use the command:

```
del ip igmp des=224.1.2.3 int=vlan1 po=1-4
```

Related Commands

- [add ip igmp destination](#)
- [create ip igmp destination](#)
- [destroy ip igmp destination](#)
- [show ip igmp](#)

delete ip mvr

Syntax DELEte IP MVR VLAN=1..4094 GROupaddress=*ipadd*- [*ipadd*]

where *ipadd* is an IP address in dotted decimal notation

Description This command deletes an MVR IP multicast group address or range of addresses on a switch.

The **vlan** parameter specifies the VLAN Identifier that the multicast VLAN is associated with.

The **groupaddress** parameter specifies the multicast group IP address or range of IP addresses that belong to the multicast VLAN.

Examples To delete a multicast IP group address from a multicast VLAN, use the command:

```
del ip mvr vlan=22 gro=230.1.1.1
```

Related Commands [add ip mvr](#)
[show ip mvr](#)

delete pim bsr candidate

Syntax DELEte PIM BSRCandidate

Description This command stops the switch from acting as a bootstrap router candidate in the PIM-SM domain.

Examples To stop the switch from acting as a bootstrap router candidate, use the command:

```
del pim bsr
```

Related Commands [add pim bsr candidate](#)
[disable pim](#)
[show pim](#)
[show pim bsr candidate](#)

delete pim interface

Syntax `DELeTe PIM INTeRface=interface [DLCi=1..1024]`

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command deletes the specified interface from the PIM interface list on the switch, stops all PIM processes on the interface, and deletes all routing information generated by the interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the [show pim interface](#) command. Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlci** parameter specifies the DLCI of a Frame Relay interface. The **dlci** parameter is valid for Frame Relay interfaces.

Examples To delete interface vlan2 from PIM interface list, use the command:

```
del pim int=vlan2
```

Related Commands

- [add pim interface](#)
- [disable pim](#)
- [show pim](#)
- [show pim interface](#)

delete pim rpcandidate

Syntax DELEte PIM RPCandidate[=*rp-address*] GROup=*group-address*
[MASK=*ipaddress*]

where:

- *group-address* is the IP address of the multicast group in dotted decimal notation.
- *ipaddress* is an IP address in dotted decimal notation.
- *rp-address* is an IP address in dotted decimal notation.

Description This command deconfigures the switch from acting as a rendezvous point candidate for a multicast group.

The **rpcandidate** parameter is the IP address of the rendezvous point for the multicast group when it is specified with a value. This option can be used to remove a static RP mapping.

The **mask** parameter specifies the mask for the multicast group address specified with the **group** parameter. This is useful when deconfiguring multiple multicast groups with a common rendezvous point (RP). The default mask is 255.255.255.255.

Examples To stop the switch from advertising itself as an RP candidate for multicast group 224.1.1.98, use the command:

```
del pim rpc gro=224.1.1.98
```

Related Commands [add pim rpcandidate](#)
[disable pim](#)
[show pim](#)
[show pim rpcandidate](#)

destroy igmp filter

Syntax DESTroy IGMP FILter=*filter-id*

where *filter-id* is a decimal number from 1 to 99

Description This command destroys an IGMP filter and all entries in the filter. IGMP filters control a port's membership of multicast groups by filtering incoming IGMP messages received from hosts attached to the port.

The **filter** parameter specifies the number of an existing filter to destroy.

You should remove the filter from any ports before you destroy the filter. Use the [show switch port command on page 8-136 of Chapter 8, Switching](#) to see which ports the filter is applied to, and the [set switch port command on page 8-109 of Chapter 8, Switching](#) to remove the filter.

Examples To destroy filter 6, use the command:

```
des igmp fil=6
```

Related Commands

- [add igmp filter](#)
- [create igmp filter](#)
- [delete igmp filter](#)
- [set igmp filter](#)
- [show igmp filter](#)

destroy ip igmp destination

Syntax DESTroy IP IGMP DESTination=*ipaddress* INTERface=*interface*

where:

- *ipaddress* is an existing IGMP group destination address.
- *interface* is the name of the interface over which multicast data is forwarded.

Description This command destroys a static IGMP association. It is not necessary to delete the ports first. The static IGMP association identified by the **destination** and **interface** parameters must already exist for this command to succeed.

Examples To stop the switch forwarding all multicast data for 224.1.2.3 over *vlan1*, use the command:

```
dest ip igmp des=224.1.2.3 int=vlan1
```

Related Commands

- [add ip igmp destination](#)
- [create ip igmp destination](#)
- [delete ip igmp destination](#)
- [show ip igmp](#)

destroy ip mvr

Syntax DESTroy IP MVR VLAN=1..4094

Description This command removes MVR from a switch.

The **vlan** parameter specifies the VLAN Identifier with which the multicast VLAN is associated.

Examples To remove MVR, use the command:

```
dest ip mvr vlan=22
```

Related Commands [create ip mvr](#)
[set ip mvr](#)
[show ip mvr](#)

disable dvmrp

Syntax DISable DVMrp

Description This command disables the DVMRP module thereby stopping the DVMRP routing process. Other DVMRP configuration remains intact. By default, DVMRP is disabled when the switch is started.

Example To disable DVMRP, use the command:

```
dis dvm
```

Related Commands [delete dvmrp interface](#)
[enable dvmrp](#)
[reset dvmrp interface](#)
[show dvmrp](#)

disable dvmrp debug

Syntax `DISable DVMrp DEBug={ALL|GRAft|PRObe|PRUne|REPort}[,...]
INterface=interface [DLC=1..1024]`

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command disables the debugging option for the specified DVMRP interface. The option must currently be enabled. A list of options separated by commas may be specified to disable more than one debugging option at a time. By default all DVMRP debugging is disabled. To see a list of current interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the [show pim interface command on page 27-132](#).

The **debug** parameter specifies debugging options that are disabled. The value of this parameter is a single item or a comma-separated list of items. The debugging that is disabled by each of the options is shown with the [enable dvmrp debug command on page 27-78](#).

The **interface** parameter specifies the DVMRP interface already assigned and configured, which is to be debugged. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the **show dvmrp interface** command.

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To disable all DVMRP debugging on interface vlan2, use the command:

```
dis dvm deb int=vlan2
```

Related Commands [enable dvmrp debug](#)
[show dvmrp](#)
[show dvmrp interface](#)

disable igmpsnooping

Syntax DISable IGMPsNooping

Description This command disables IGMP snooping on the switch. IGMP snooping is enabled by default. Note that multicast packets flood the VLAN when IGMP snooping is disabled.

Disabling IGMP snooping may be useful when filters are used extensively because IGMP snooping uses a Layer 3 filter. When IGMP snooping is disabled, this filter becomes available. See [“Hardware Packet Filters” on page 8-33 of Chapter 8, Switching](#) for information about filters.

Note that IGMP snooping is independent of IGMP, which is disabled by default.

Examples To disable IGMP snooping, use the command:

```
dis igmpsn
```

Related Commands [disable ip igmp interface](#)
[enable igmpsnooping](#)
[enable ip igmp](#)
[enable ip igmp interface](#)
[show ip igmp](#)
[show igmpsnooping](#)

disable ip igmp

Syntax DISable IP IGMP

Description This command disables IGMP on the switch so that multicast routing stops immediately. IGMP is disabled by default. IGMP snooping is enabled by default and is independent of IGMP.

Examples To disable the IGMP module, use the command:

```
dis ip igmp
```

Related Commands [disable ip igmp interface](#)
[enable ip igmp](#)
[show ip igmp](#)

disable ip igmp allgroup

Syntax `DISable IP IGMP ALLGroup=[port-list|ALL]`

where *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

Description This command disables the specified port or ports from acting as a router port. Once disabled, the port no longer receives MARL entries when the device receives an IGMP report, query, or multicast data over any other port.

Example To prevent ports 1, 5, and 7 from acting as an all-group entry, use the command:

```
dis ip igmp allg=1,5,7
```

Related Commands [enable ip igmp allgroup](#)

disable ip igmp debug

Syntax `DISable IP IGMP DEBug`

Description This command disables all IGMP debugging messages and resets the **destination** and **sourceipaddress** parameters set with the **enable ip igmp debug** command to **all**. Debugging is disabled by default.

Examples To disable all IGMP debugging messages and reset the IGMP debug message filters for all, use the command:

```
dis ip igmp deb
```

Related Commands [show ip igmp debug](#)

disable ip igmp interface

Syntax DISable IP IGMP INTerface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command disables IGMP on an IP interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- FR (such as fr0, fr0-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#).

Disabling IGMP on an IP interface or a logical interface will disable IGMP on all logical interfaces associated with the IP interface.

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the data link circuit identifier of a Frame Relay interface.

Examples To disable IGMP on interface vlan2, use the command:

```
dis ip igmp int=vlan2
```

Related Commands

- [disable ip igmp](#)
- [enable ip igmp interface](#)
- [set ip igmp interface](#)
- [show ip igmp](#)

disable ip multicasting

Syntax DISable IP MULTicasting

Description This command disables emulation of hardware switching of IP multicast traffic between VLANs. The default is disabled. Multicast traffic within VLANs continues to be switched by IGMP snooping as long as IGMP snooping is enabled.

This command does not apply to Rapier i Series switches. For other Rapier Series switches, IP multicast packets are routed by software when IP Multicast switching emulation is disabled.

Example To disable hardware switching of multicast traffic between VLANs, use the command:

```
dis ip mult
```

Related Commands [enable ip multicasting](#)

disable ip mvr

Syntax	DISable IP MVR
Description	This command disables MVR. MVR must be currently enabled. The default is disabled.
Examples	To disable MVR, use the command: <pre>dis ip mvr</pre>
Related Commands	enable ip mvr

disable ip mvr debug

Syntax	DISable IP MVR DEBug={ALL JOInt LEAVE MARL}
Description	<p>This command disables debugging messages when specific actions are taken. The current debug option may or may not be disabled. The default is disabled.</p> <p>If all is specified, all debugging on all ports is disabled.</p> <p>If joint is specified, debugging of joint messages is disabled.</p> <p>If leave is specified, debugging of leave messages is disabled.</p> <p>If marl is specified, all debugging of the MARL table is disabled.</p>
Examples	To disable debugging joint messages on the multicast VLAN, use the command: <pre>dis ip mvr deb=joi</pre>
Related Commands	enable ip mvr debug

disable pim

Syntax	DISable PIM
Description	This command disables PIM on the switch. PIM multicast routing stops but PIM configurations remain intact. PIM is disabled by default.
Examples	To disable PIM on the switch, use the command: <pre>dis pim</pre>
Related Commands	delete pim bsrcandidate delete pim interface delete pim rpcandidate enable pim show pim

disable pim bsmsecuritycheck

Syntax DISable PIM BSMSecuritycheck

Description This command disables PIM bootstrap message security checking. The switch stops checking that the source IP address of a bootstrap message is the expected address of the PIM neighbour.

Bootstrap message security checking is enabled by default. You may need to disable it when interoperating with some PIM implementations.

Examples To disable PIM bootstrap message security checking, use the command:

```
dis pim bsms
```

Related Commands [enable pim](#)
[show pim config](#)

disable pim debug

Syntax DISable PIM DEBug={ALL|ASSert|BSR|C-Rp-adv|GRAft|HELlo|JOInt|REGister|STATerefresh}[, ...]

Description This command disables the debugging option. The option must currently be enabled. PIM debugging is disabled by default.

The **debug** parameter specifies which debugging options are to be disabled. The value of this parameter is a single option or a comma-separated list of options. The debugging that results from each of the options is shown in with the [disable pim debug command on page 27-76](#).

Examples To disable all PIM debugging, use the command:

```
dis pim deb=all
```

Related Commands [enable pim debug](#)
[show pim debug](#)
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

enable dvmrp

Syntax ENAbLe DVMrp

Description This command enables DVMRP routing, and activates any existing DVMRP configuration. By default DVMRP is disabled when the switch is started.

Example To enable DVMRP, use the command:

```
ena dvm
```

Related Commands [add dvmrp interface](#)
 [disable dvmrp](#)
 [set dvmrp interface](#)
 [show dvmrp debug](#)

enable dvmrp debug

Syntax ENAbLe DVMrp DEBug={ALL|GRAft|PRObe|PRUnE|REPort}{[,...]
INTErface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from. If a logical interface is not specified, 0 is assumed.

Description This command enables debugging options for the specified DVMRP interface. Debugging may or may not be enabled already. Debugging information is sent to the port or Telnet session from which the command is entered. By default, DVMRP debugging is disabled.

The **debug** parameter specifies the debugging options that are enabled. The value of this parameter is a single option or a comma-separated list of options. The following table describes the debugging options.

Parameter	Description
all	All debug options
graft	DVMRP Graft and Graft Ack packets
probe	DVMRP Router Probe packets
prune	DVMRP Prune packets
report	DVMRP Route Report packets

The **interface** parameter enables DVMRP debugging on the specified interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the **show dvmrp interface** command.

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To enable debugging of DVMRP prune and graft messages on the vlan2 interface, use the command:

```
ena dvm deb=gra,pru int=vlan2
```

Related Commands [disable dvmrp debug](#)
[show dvmrp](#)
[show dvmrp interface](#)

enable igmpsnooping

Syntax ENABle IGMPsNooping

Description This command enables IGMP snooping on the switch. IGMP snooping is enabled by default. IGMP snooping can be enabled only when a free Layer 3 filter is available ([“Hardware Packet Filters” on page 8-33 of Chapter 8, Switching](#)).

Note that IGMP snooping is independent of IGMP, which is disabled by default.

Examples To enable IGMP snooping, use the command:

```
ena igmpsn
```

Related Commands [disable igmpsnooping](#)
[disable ip igmp](#)
[disable ip igmp interface](#)
[enable ip igmp](#)
[enable ip igmp interface](#)
[show ip igmp](#)
[show igmpsnooping](#)

enable ip igmp

Syntax ENABle IP IGMP

Description This command enables IGMP on the switch. IGMP is disabled by default. IGMP snooping is enabled by default and is independent of IGMP.

Examples To enable IGMP, use the command.

```
ena ip igmp
```

Related Commands [disable ip igmp](#)
[enable ip igmp interface](#)
[show ip igmp](#)

enable ip igmp allgroup

Syntax ENABle IP IGMP ALLGroup=[*port-list*|ALL]

where *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port, including uplink ports.

Description This command enables one or more ports to act like a router port. All ports are allowed to be a router port by default, so this command re-enables a port as a router port if it has previously been disabled with the **disable ip igmp allgroup** command.

Example To enable ports 1, 5, and 7 to act as an all-group entry, use the command:

```
ena ip igmp allg=1,5,7
```

Related Commands [disable ip igmp allgroup](#)
[show ip igmp](#)

enable ip igmp debug

Syntax ENABle IP IGMP DEBug [DESTination={ALL|*ipaddress*}]
[SOURCE*ipaddress*={ALL|*ipaddress2*}]

where:

- *ipaddress* is an IGMP group destination address.
- *ipaddress2* is the IP address of a host that responds to IGMP queries.

Description This command enables IGMP debugging of destination and source IP addresses. Debugging is disabled by default.

The **destination** parameter specifies the destination multicast group address for debugging. The default is **all**.

The **sourceipaddress** parameter specifies the host IP address responding to IGMP queries. The default is **all**.

If **destination** and **sourceipaddress** are both specified, debug messages that match both parameters are displayed. Some debug messages are displayed before the packet is fully decoded, and are unable to be filtered.

Examples To enable debugging information relating to IGMP host 10.41.0.22, use the command:

```
ena ip igmp deb source=10.41.0.22
```

To show all IGMP debug messages, use the command:

```
ena ip igmp deb
```

Related Commands [show ip igmp debug](#)

enable ip igmp interface

Syntax ENABle IP IGMP INTerface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface number is not specified, 0 is assumed.

Description This command enables IGMP on an IP interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- FR (such as fr0, fr0-1)
- VLAN (such as vlan1, vlan1-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#). Note that IGMP does not operate on local interfaces.

Enabling IGMP on an IP interface or a logical interface will enable IGMP on all logical interfaces associated with the IP interface.

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To enable IGMP on vlan2 interface, use the command:

```
ena ip igmp int=vlan2
```

Related Commands

- [disable ip igmp interface](#)
- [enable ip igmp](#)
- [set ip igmp interface](#)
- [show ip igmp](#)

enable ip multicastsitching

Syntax ENABle IP MULTicastswitching

Description This command enables hardware switching emulation of IP multicast traffic between VLANs on the same switch. The default is disabled. This command does not apply to Rapier i Series switches. IP multicast switching is automatically enabled on Rapier i Series switches when it is required.

This command enables other Rapier Series switches to emulate multicast switching for DVMRP, PIM-SM and PIM-DM. When it is enabled, IP multicast packets are switched at hardware level. The software has control of how the packets are switched and where they are switched.



Caution If IP multicast switching emulation is enabled and PIM or DVMRP is configured, then IGMP snooping may not operate correctly. Enabling IP multicast switching emulation without enabling a Layer 3 multicasting protocol may defeat IGMP snooping. If IP multicast switching emulation is enabled, VLAN tagging is not fully

supported between VLANs, and the TTL in the IP header may not be decremented correctly. Multicast switching emulation should be used only in simple VLAN-based networks without WAN links or loops.

After enabling IP multicast switching emulation, we recommend that you save your configuration by using the **create config** command and restart the switch to renegotiate IP multicasting group membership entries.

Example To enable hardware switching of multicast traffic between VLANs, use the command:

```
ena ip mult
```

Related Commands [disable ip multicasting](#)

enable ip mvr

Syntax ENAbLe IP MVR

Description This command enables MVR. MVR must be currently disabled. The default is disabled.

Examples To enable MVR, use the command:

```
ena ip mvr
```

Related Commands [disable ip mvr](#)

enable ip mvr debug

Syntax ENAbLe IP MVR DEBug={ALL|JOInt|LEAVe|MARL}

Description This command enables the display of debugging messages when specific actions are taken. The current debug option may or may not be enabled. The default is enabled.

If **all** is specified, debugging on all ports is enabled.

If **joint** is specified, debugging of joint messages is enabled.

If **leave** is specified, debugging of leave messages is enabled.

If **marl** is specified, debugging of the MARL table is enabled.

Examples To enable debugging of joint messages on the multicast VLAN, use the command:

```
ena ip mvr deb=joi
```

Related Commands [disable ip mvr debug](#)

enable pim

Syntax ENABle PIM

Description This command enables PIM routing on the switch. PIM is disabled by default. Any existing PIM configuration is activated after this command has been entered.

Examples To enable PIM routing, use the command:

```
ena pim
```

Related Commands

- [add pim bsrcandidate](#)
- [add pim interface](#)
- [add pim rpcandidate](#)
- [disable pim](#)
- [set pim interface](#)
- [set pim](#)
- [show pim](#)

enable pim bsmsecuritycheck

Syntax ENABle PIM BSMSecuritycheck

Description This command enables PIM bootstrap message security checking, which checks that the source IP address of a bootstrap message is the expected address of the PIM neighbour.

This checking is enabled by default. You may need to disable it when interoperating with some PIM implementations.

Examples To enable PIM bootstrap message security checking, use the command:

```
ena pim bsms
```

Related Commands

- [disable pim bsmsecuritycheck](#)
- [show pim config](#)

enable pim debug

Syntax ENAbLe PIM DEBUg={ALL|ASSErt|BSR|C-Rp-adv|GRAft|HELlo|JOInt|REGister|STATerefresh}[, ...]

Description This command enables debugging options. Debugging may or may not be enabled already. Debugging information is sent to the port or Telnet session from which the command was entered. All PIM debugging is disabled by default.

The **debug** parameter specifies which debugging options are to be enabled. The value of this parameter is a single option or a comma-separated list of options. The following table describes the debugging options.

Parameter	Description
ALL	All debug options.
ASSErt	PIM Assert packets
BSR	PIM Bootstrap packets (Sparse Mode only)
C-Rp-adv	PIM Candidate-RP-Advertisement (Sparse Mode only)
GRAft	PIM Graft packets (Dense Mode only)
HELlo	PIM Hello packets
JOInt	PIM Join/Prune packets
REGister	PIM Register and Register Stop packets (Sparse Mode only)
STATerefresh	PIM State Refresh packets (Dense Mode only)

Examples To enable debugging of PIM hello and join/prune messages, use the command:

```
ena pim deb=hello,joi
```

Related Commands [disable pim debug](#)
[disable debug active](#) in Chapter 4, Configuring and Monitoring the System
[show debug active](#) in Chapter 4, Configuring and Monitoring the System

purge dvmrp

Syntax PURge DVMrp

Description This command disables DVMRP, purges all configuration information relating to the DVMRP multicast routing module, and reinitialises the data structures used by the module. It should be used when first setting up the DVMRP module or when a major change is required.



Caution All current DVMRP configuration information will be lost. Use with extreme caution!

Related Commands

- `delete dvmrp interface`
- `disable dvmrp`
- `reset dvmrp interface`
- `set dvmrp interface`
- `show dvmrp`

purge pim

Syntax PURge PIM

Description This command purges all configuration information relating to the PIM multicast routing module, and reinitialises the data structures used by the module. It also stops the current PIM operation. It should be used when first setting up the PIM module or when a major change is required.



Caution All current PIM configuration information will be lost. Use with extreme caution!

Related Commands

- `delete pim bsr candidate`
- `delete pim interface`
- `delete pim rpcandidate`
- `disable pim`
- `disable pim debug`
- `reset pim interface`
- `show pim`

reset dvmrp interface

Syntax RESET DVMrp INTerface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command resets all DVMRP processes, timers and route information for the specified interface. This effectively restarts all DVMRP processes for this interface as if the interface has just been added to the DVMRP interface list.

The **interface** parameter specifies the IP interface. The IP interface must already have been added to the DVMRP module. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)

- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the **show dvmrp interface** command. Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To reset interface vlan2 in the DVMRP interface list, use the command:

```
reset dvm int=vlan2
```

Related Commands

- [delete dvmrp interface](#)
- [disable dvmrp](#)
- [set dvmrp interface](#)
- [show dvmrp](#)
- [show dvmrp interface](#)

reset pim interface

Syntax RESET PIM INTerface=*interface* [DLC=1..1024]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command resets all timers, route information, and counters associated with the specified interface, and restarts all PIM processes for this interface as if this interface has just been added to PIM interface list. It also disables any enabled PIM debugging on the interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the **show pim interface** command. Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

Examples To reset the ppp0 interface, use the command:

```
reset pim int=ppp0
```

Related Commands

- [set pim interface](#)
- [set pim](#)
- [show pim](#)
- [show pim interface](#)

set dvmrp interface

Syntax SET DVMrp INTerface=*interface* [DLC=1..1024] [METric=1..32]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command modifies the configuration of a DVMRP interface. When any parameter is modified, the DVMRP processes and any routing information generated by this interface is reset.

The **interface** parameter specifies the IP interface. The IP interface must already be assigned to the DVMRP module. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87](#), or the [show dvmrp interface command on page 27-104](#) command. Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlc** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

The **metric** parameter specifies the metrics on a DVMRP interface. This metric is added to all routes that are learned via this interface. The default is 1.

Examples To change the metric for DVMRP interface vlan2 to 4, use the command:

```
set dvm int=vlan2 met=4
```

Related Commands [reset dvmrp interface](#)
[show dvmrp](#)
[show dvmrp interface](#)

set igmp filter

Syntax SET IGMP Filter=*filter-id* ENTry=1..65535
 [GROupaddress={*ipadd*|*ipadd-ipadd*}] [MSGType={QUeRY|
 REPort|LEAVe}] [ACtion={INCLude|EXCLude}]

where:

- *filter-id* is a decimal number from 1 to 99.
- *ipadd* is an IP address in dotted decimal notation.

Description This command modifies an entry in an IGMP filter. IGMP filters control a port's membership of multicast groups by filtering incoming IGMP messages from hosts attached to the port.

The **filter** parameter specifies the number of the filter that the entry belongs to. A filter with the specified number must already exist.

The **entry** parameter specifies the entry to modify. An entry with the specified number must already exist.

The **groupaddress** parameter specifies an IP multicast group address or a range of IP multicast group addresses to match. Set **groupaddress** to:

- 0.0.0.0 to filter IGMP general query messages
- a multicast address or a range of multicast addresses to filter IGMP group-specific query messages, report messages, and leave messages.

The **msgtype** parameter specifies the type of incoming IGMP message to match. If you specify **query**, the filter will match IGMP general and group-specific query messages. If you specify **report**, the filter will match IGMP report messages. If you specify **leave**, the filter will match IGMP leave messages. The default is **report**.

The **action** parameter specifies the action to take when an IGMP message with a message type matching **msgtype** and a group address matching **groupaddress** is received. If you specify **include**, the message is processed as normal by IGMP. If you specify **exclude**, the message is excluded from processing by IGMP, and the packet is discarded. The default is **include**.

If an IGMP filter contains at least one entry for a particular IGMP message type, then messages of the same type for group addresses that do not match any entries in the filter are implicitly excluded and the packets are discarded.

Examples To change the group address for entry 12 in filter 6 to the range 229.1.1.2 to 230.1.2.3, use the command:

```
set igmp fil=6 ent=12 gro=229.1.1.2-230.1.2.3
```

To change entry 1 in filter 2 to accept Membership Reports for multicast group addresses matching the entry's group address range, use the command:

```
set igmp fil=2 ent=1 ac=incl
```

Related Commands

- [add igmp filter](#)
- [create igmp filter](#)
- [delete igmp filter](#)
- [destroy igmp filter](#)
- [show igmp filter](#)

set igmpsnooping vlan

Syntax SET IGMP Snooping VLAN={*vlan-name*|1..4094|ALL}
[Fastleave={ON|OFF|YES|NO|True|False}]
[QUERYSolicit={OFF|NO|False|ON|YES|True}]

where *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits, the underscore, and hyphen. The *vlan-name* cannot be a number or **all**.

Description This command enables or disables Fast Leave processing and query solicitation for IGMP Snooping.

The **vlan** parameter specifies the VLAN on which the specified feature is to be enabled or disabled. The default is **all**.

The **fastleave** parameter specifies whether Fast Leave processing is enabled or disabled. If you specify **on**, **yes** or **true** then Fast Leave processing is enabled on the specified VLAN or all VLANs. If you specify **off**, **no** or **false** then Fast Leave processing is disabled on the specified VLAN or all VLANs. Note that Fast Leave should not be configured on a port that has multiple hosts attached because it may adversely affect multicast services to some hosts. See [“Fast Leave” on page 27-27](#) for more information. The default is **off**.

This command deprecates the following command, which is still valid:

```
set igmpsnooping fastleave={on|yes|true|off|no|false}  
[interface=vlan]
```

The **query solicit** parameter specifies whether query solicitation is enabled on the specified VLANs. Query solicitation minimises loss of multicast data after a topology change on networks that use spanning tree (STP, RSTP, or MSTP) for loop protection. When an STP topology change occurs, IGMP snooping sends a query solicit message out every VLAN that query solicitation is enabled on. When the IGMP Querier receives the message, it responds by sending a General Query, which all IGMP listeners respond to. This refreshes snooped group membership information in the network. See [“Query Solicitation” on page 27-27](#) for more information. The default is **on** for the root bridge in an STP topology and **off** for other switches.

Examples To enable IGMP Snooping Fast Leave processing on VLAN ‘vlan2’, use the command:

```
set igmpsn vlan=vlan2 f=on
```

Related Commands [disable igmpsnooping](#)
[enable igmpsnooping](#)
[set igmpsnooping routermode](#)
[show igmpsnooping](#)

set igmpsnooping routermode

Syntax SET IGMPsNooping ROUTERMode={ALL|DEFault|IP|MULTICAstrouter|NONE}

Description This command determines the kinds of packets that IGMP snooping uses to indicate that a router is attached to a port. For more information, see [“Downstream routers” on page 27-26](#).

The **all** option specifies that all reserved multicast addresses (i.e. 224.0.0.1 to 224.0.0.255) are treated as router multicast addresses.

The **default** option specifies that the following multicast addresses are treated as multicast router addresses:

Router Type	Multicast Address
IGMP Query	224.0.0.1
All routers on this subnet	224.0.0.2
DVMRP Routers	224.0.0.4
All OSPFIGP routers	224.0.0.5
OSPFIGP designated routers	224.0.0.6
RIP2 routers	224.0.0.9
All PIM routers	224.0.0.13
All CBT routers	224.0.0.15

The **ip** option starts with the addresses specified by the currently-set option and lets users add or remove addresses with the **add igmpsnooping routeraddress** and **delete igmpsnooping routeraddress** commands.

The **multicastrouter** option specifies that the following addresses are treated as router multicast addresses:

- DVMRP Routers, 224.0.0.4
- All PIM routers, 224.0.0.13

The **none** option specifies that the switch does not create router ports.

Examples To allow the switch to treat all reserved multicast addresses as router multicast addresses, use the command:

```
set igmpsn routerm=all
```

Related Commands [add igmpsnooping routeraddress](#)
[delete igmpsnooping routeraddress](#)
[show igmpsnooping routeraddress](#)

set ip igmp

Syntax SET IP IGMP [LMQi=1..255] [LMQC=1..5]
[QUERyinterval=1..65535] [QUERYREsponseinterval=1..255]
[ROBustness=1..5] [TIMEOut=1..65535]

Description This command sets operational timers and thresholds for IGMP.



Caution The defaults for these timers suit most networks. Changing them to inappropriate values can cause IGMP to function in undesirable ways. System administrators should change timer values based on a sound understanding of their interaction with other devices in the network.

The **lmqi** parameter specifies the Last Member Query Interval (in 1/10 secs), which is the Max Response Time inserted into Group-Specific Queries sent in response to Leave Group messages. It is also the amount of time between Group-Specific Query messages. The default is 10 (1 second).

The **lmqc** parameter specifies the Last Member Query Count, which is the number of Group-Specific Queries sent before the switch assumes there are no local members. The default is the same as **robustness** value.

The **queryinterval** parameter specifies the seconds of the interval between IGMP Host Membership Queries if this switch is elected the designated router for the LAN. If the switch is not the IGMP designated router, it ignores this parameter. The default is 125.

The **queryresponseinterval** parameter specifies the Max Response Time (in 1/10 second) inserted into the periodic General Queries. The default is 100 (10 seconds).

The **robustness** parameter specifies the Robustness Variable that allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the Robustness Variable may be increased. IGMP is robust to (Robustness Variable - 1) packet losses. The Robustness Variable *must not* be zero and *should not* be 1. The default is 2.

The **timeout** parameter specifies the longest interval in seconds that a group remains in the local multicast group database without the switch (designated router or not) receiving a Host Membership Report for this multicast group. This **timeout** parameter is used by all IGMP routers to maintain their group membership databases. The default is 260. If a value is specified for **queryinterval** without specifying a value for **timeout**, **timeout** is calculated as $(2 * \text{queryinterval}) + 10$. The added 10 seconds is the default **queryresponseinterval** that hosts use when sending Host Membership Reports. When a timeout interval is specified, it overrides a calculated value.

Examples To set the IGMP query interval to 180s (3 minutes), use the command:

```
set ip igmp que=180
```

Related Commands

- [disable ip igmp](#)
- [disable ip igmp interface](#)
- [enable ip igmp](#)
- [enable ip igmp interface](#)
- [set ip igmp interface](#)
- [show ip igmp](#)

set ip igmp interface

Syntax SET IP IGMP INTERface=*interface* QUERYtimeout={NONE|0|1..65535}

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command enables the monitoring of incoming IGMP general query messages on an interface, and generates a log message and an SNMP trap if an IGMP general query message is not received on the interface within a specified time interval.

The **interface** parameter specifies the IP interface to monitor for IGMP general query messages. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- FR (such as fr0, fr0-1)
- VLAN (such as vlan1, vlan1-1)

Modifying IGMP on an IP interface or a logical interface will change the behaviour of IGMP on all logical interfaces associated with the IP interface.

The **querytimeout** parameter specifies the maximum expected time interval, in seconds, between successive IGMP general query messages arriving on the interface. If you specify **none** or **0**, monitoring is disabled. If you specify a non-zero time interval, a log message and an `igmpGeneralQueryNotReceivedEvent` SNMP trap is generated if an IGMP general query message is not received on the interface within the time interval. For more information about trap messages, see [“traps Group” on page C-16 of Appendix C, SNMP MIBs](#). Monitoring is only active when:

- IGMP is enabled globally
- IGMP is enabled on the interface
- the interface is active

The default is **none**.

Examples To set the maximum time period allowed between successive IGMP general query messages on interface vlan2 to 120 seconds, use the command:

```
set ip igmp int=vlan2 query=120
```

Related Commands

- [disable ip igmp](#)
- [disable ip igmp interface](#)
- [enable ip igmp](#)
- [enable ip igmp interface](#)
- [set ip igmp](#)
- [show ip igmp](#)

set ip mvr

Syntax SET IP MVR VLAN=1..4094 [IMTLeave=*port-list*]
[MODE={DYNAMIC|COMPATIBLE}] [RECEIVERport=*port-list*]
[SOURCEport=*port-list*]

where *port-list* is a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at 1 and end at *m*, where *m* is the highest numbered Ethernet switch port.

Description This command modifies MVR on a switch.

The **vlan** parameter specifies the VLAN Identifier with which the multicast VLAN is associated.

The **imtleave** parameter specifies the ports on which the Immediate Leave feature is enabled. If this parameter is specified, the receiver ports immediately leave the multicast group as soon an IGMP Leave message is received by the switch on that port. Otherwise, the switch sends an IGMP query on the port and waits for the IGMP group membership reports when it receives an IGMP Leave message. If no reports are received within the IGMP timeout period (see the [set ip igmp command on page 27-91](#)), the receiver port leaves the multicast group. The port numbers specified must be members of the **receiverport** list.

The **mode** parameter specifies the mode of the MVR operation. If **dynamic** is specified, IGMP reports are sent out through the source ports of the multicast VLAN. If **compatible** is specified, IGMP reports are not sent.

The **receiverport** parameter specifies a receiver port that receives multicast data when it is a member of the multicast group. The receiver ports cannot belong to the multicast VLAN.

The **sourceport** parameter specifies the uplink (source) ports that send and receive multicast data to and from the multicast VLAN. All source ports must belong to a single multicast VLAN.

Examples To modify MVR, use the command:

```
set ip mvr vlan=22 so=8,9 rec=1-5, 6-8 imtl=2,4
```

Related Commands [create ip mvr](#)
[destroy ip mvr](#)
[show ip mvr](#)

set pim

Syntax SET PIM [ADVinterval={10..15000|DEFAULT}]
 [BSMinterval={10..15000|DEFAULT}]
 [JPInterval={1..65535|DEFAULT}]
 [KEEPalivetime={10..65535|DEFAULT}]
 [PRObetime={1..65535|DEFAULT}]
 [PRUNEholdtime={1..65535|DEFAULT}]
 [SOURCEalivetime={10..65535|DEFAULT}]
 [SRInterval={10..255|DEFAULT}]
 [SUPPressiontime={1..65535|DEFAULT}]

Description This command sets timers for PIM operations.



Caution The defaults for these timers suit most networks. Changing them to inappropriate values may cause PIM to function in undesirable ways. System administrators should change these timer values based on a sound understanding of their interaction with other devices in the network.

The **advinterval** parameter specifies the seconds of the interval at which the switch sends C-RP-Advertisements. The default is 60 seconds. This timer applies to PIM-SM only.

The **bsminterval** parameter specifies the seconds of the interval at which the switch sends bootstrap messages when it is the bootstrap switch in the domain. The default is 60 seconds. This timer applies to PIM-SM only.

The **jpinterval** parameter specifies the upstream join timer in seconds. This is the interval at which PIM join/prune messages are sent. For proper operation, a maximum value of 18000 seconds is recommended. The default is 60 seconds.

The **keepalivetime** parameter specifies the seconds that the join state for a particular source and group pair is maintained in the absence of data for that pair. The default is 210 seconds.

The **probetime** interval specifies the register probe time in seconds. This is the time the DR waits for another register stop message after sending a null register message to the RP. If it does not receive a register stop message in this time, it resumes registering data packets to the RP. The default is 5 seconds. This timer applies to PIM-SM only.

The **pruneholdtime** parameter specifies the seconds that the prune state is maintained. This time is used in prune messages to let upstream neighbours know how long to hold the prune state. It is also used as the prune limit timer for suppressing prunes if a prune message has already been sent. The default is 210 seconds. This timer applies to PIM-DM only.

The **sourcealivetime** parameter specifies the seconds that a switch acting as a state refresh originator is active in the absence of data packets from the source. The default is 210 seconds. This timer applies to PIM-DM only.

The **srinterval** parameter specifies the seconds of the interval at which this switch sends state refresh messages, if it is configured to be state refresh capable, and becomes a state refresh originator (in general, this means having a directly connected source). The default is 60 seconds. This timer applies to PIM-DM only.

The **suppressiontime** parameter specifies the register suppression time. This determines the interval at which the sender's DR sends null register messages to the group's RP to tell it to send another register stop message if it still does not need the data to be registered and sent to it. The default is 60 seconds. This timer applies to PIM-SM only.

Examples To set the join/prune message interval to 90 seconds, use the command:

```
set pim jpi=90
```

Related Commands

- [enable pim](#)
- [set pim interface](#)
- [show pim](#)
- [show pim bsrcandidate](#)
- [show pim counters](#)
- [show pim debug](#)
- [show pim interface](#)
- [show pim neighbour](#)
- [show pim route](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)
- [show pim timer](#)

set pim log

Syntax SET PIM LOG=[NONE | STATUS | ERROR | ALL] [TRAP=[NONE | STATUS | ERROR | ALL]]

Description This command applies to PIM-SM only and sets the type of logging for status PIM log messages, error messages and/or sending of SNMP traps for certain error conditions.

The **log** parameter specifies whether status, error, or all log messages should be generated. The default is status.

The **trap** parameter specifies whether status, error, or all traps should be generated.

Related Commands [show pim debug](#)

set pim bsr candidate

Syntax SET PIM BSRCandidate [HASHmasklength=0..32]
[INTERface=*interface*] [PREference=0..255]

where *interface* is the name of the interface over which multicast data is forwarded. The interface can be either a VLAN (e.g. vlan1) or a local interface (e.g. local1).

Description This command sets the switch's Bootstrap Router Candidate preference.

The **hashmasklength** parameter specifies the number of bits of the group number to use when selecting a rendezvous point (RP) candidate if this switch becomes the BSR. A higher number increases the spread of groups across RPs. The default is 30.

Note that software release versions prior to 2.7.3 did not correctly support the PIM hash mask length option. As a result, the RP selection calculation differs between this release and release versions prior to 2.7.3. If a network contains switches running a mixture of versions, this leads to incorrect forwarding behaviour. To avoid this issue, either ensure that all devices on the network correctly support the hash mask length option (recommended), or ensure that the following **both** hold:

- The hash mask length option on all BSR candidates is configured to 4 bits. This implies that all BSR candidates must be running 2.7.3 or later.
- All RP candidates use a common prefix of 224.0.0.0/240.0.0.0. This has the side effect of collapsing all groups to use a single PIM RP.

The **interface** parameter specifies an interface for the switch to use when advertising itself as a candidate bootstrap router. The IP address of this interface is advertised by the switch. The interface supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified, the switch instead advertises its first active IP interface.

The **preference** parameter specifies the preference for this switch to become the bootstrap router for the PIM-SM domain. A higher value indicates a greater preference.

Examples To change the switch's candidate BSR preference to 100 and a hash mask length of 0, use the command:

```
set pim bsr pref=100 hasmasklength=0
```

Related Commands [add pim bsr candidate](#)
[delete pim bsr candidate](#)
[show pim bsr candidate](#)

set pim interface

Syntax SET PIM INTERface=*interface* [DLCi=1..1024]
[DRPriority=0..4294967295] [ELectby={DRPriority|
IPaddress}] [Hellointerval={10..15000|DEfault|65535}]
[MODE={Dense|Sparse}] [SRCapable={Yes|No}]

where *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.

Description This command sets parameters for the specified PIM interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

To see a list of current valid interfaces, use the [show interface command on page 11-87 of Chapter 11, Interfaces](#), or the **show pim interface** command. Note that multihomed interfaces must specify the logical interface number (e.g. ppp1-1).

The **dlci** parameter is valid for Frame Relay interfaces, and specifies the DLCI of a Frame Relay interface.

The **drpriority** parameter specifies the preference for the switch to become the designated router (DR) on this interface when **electby=drpriority**. A higher value indicates a greater preference. The default is 1.

The **electby** parameter determines how the switch elects the designated router for this interface. If **drpriority** is specified, the interface transmits its drpriority in its hello messages, which allows DR election by priority. If **ipaddress** is specified, the switch does not transmit its DR priority, which causes election by IP address. The default is **drpriority**. Note that a switch with **electby=drpriority** may still elect by IP address when it does not receive DR priority in any one of its neighbours' hello messages. Election by DR priority is possible only when all routers on the interface supply their DR priority.

The **hellointerval** parameter specifies the interval at which the switch sends Hello messages from this interface. Setting the **hellointerval** parameter to 65535 results in a Hello message being sent with a hold time of 65535, which means "infinity". A router receiving this switch's Hello never expires this switch as a PIM neighbour. This can be useful on point-to-point links. The default is 30 seconds.

The **mode** parameter specifies the PIM operating mode for the interface. The default is **sparse**.

Ensure that all interfaces have the same mode setting unless the switch is a Multicast Border Router.

The **srcapable** parameter indicates if this interface is able to originate or process State Refresh messages. The default is **no**. This parameter applies to Dense Mode interfaces only.

Examples To set the designated router priority for the interface vlan1 to 100, use the command:

```
set pim int=vlan1 drp=100
```

Related Commands

- [add pim interface](#)
- [delete pim interface](#)
- [enable pim](#)
- [reset pim interface](#)
- [show pim](#)
- [show pim interface](#)

set pim rpcandidate

Syntax SET PIM RPCandidate GROup=*group-address*
[INTerface=*interface*] [MASK=*ipaddress*]
[PRIOrity=0..255]

where:

- *group-address* is the IP address of the multicast group in dotted decimal notation
- *ipaddress* is an IP address in dotted decimal notation
- *interface* is the name of a VLAN (e.g. vlan1) or a local interface (e.g. local1).

Description This command sets the rendezvous point candidate priority for the specified multicast group(s).

The **group** parameter specifies the multicast group or groups to which the switch is a rendezvous point candidate.

The **interface** parameter specifies an interface for the switch to use when advertising itself as the candidate rendezvous point for a multicast group. The IP address of the of this interface is advertised by the switch. The **interface** supplied can be either a configured local interface or a configured VLAN interface. If the parameter is not specified, the switch advertises its first active IP interface instead.

The **mask** parameter specifies the mask for the multicast group address specified in the **group** parameter. This is useful when configuring multiple multicast groups with a common rendezvous point (RP). The default mask is 255.255.255.255. The mask for a group cannot be modified.

The **priority** parameter specifies the preference for the switch to become the rendezvous point for the multicast group. A lower value indicates a higher priority. The default is 192.

Note that the switch has the same values for **priority** for all multicast groups for which it is a rendezvous point candidate, so changing this switch's priority to be the RP for one group changes it for all groups.

Examples To change the switch's RP candidate priority to 10 for the multicast group with address 224.1.1.98, use the command:

```
set pim rpc gro=224.1.1.98 prio=10
```

Related Commands

- [add pim rpcandidate](#)
- [delete pim rpcandidate](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)

show dvmrp

Syntax SHow DVMrp

Description This command displays detailed information about DVMRP routing status on the switch. It is the equivalent of specifying the following commands:

```
show dvmrp counters
show dvmrp debug
show dvmrp interface
show dvmrp neighbour
show dvmrp route
```

Examples To display detailed information about DVMRP, use the command:

```
sh dvm
```

Related Commands [add dvmrp interface](#)
[delete dvmrp interface](#)
[disable dvmrp](#)
[disable dvmrp debug](#)
[enable dvmrp](#)
[enable dvmrp debug](#)
[reset dvmrp interface](#)
[set dvmrp interface](#)

show dvmrp counters

Syntax SHow DVMrp COunters

Description This command displays all DVMRP counters (Figure 27-5, Table 27-1).

Figure 27-5: Example output from the **show dvmrp counters** command

DVMRP Interface Counters			
Interface: vlan1			
	Rcv Pkts	Rcv Bad Pkts	Send Pkts
Probe	0000000280	0000000000	0000000281
Report	0000000042	0000000000	0000000042
Prune	0000000000	0000000000	0000000000
Graft	0000000000	0000000000	0000000000
GraftAck	0000000000	0000000000	0000000000
Total	0000000322	0000000000	0000000323
Interface: vlan2			
	Rcv Pkts	Rcv Bad Pkts	Send Pkts
Probe	0000000000	0000000000	0000000001
Report	0000000000	0000000000	0000000000
Prune	0000000000	0000000000	0000000000
Graft	0000000000	0000000000	0000000000
GraftAck	0000000000	0000000000	0000000000
Total	0000000000	0000000000	0000000001

Table 27-1: Parameters in output of the **show dvmrp counters** command

Parameter	Meaning
Interface	IP interfaces running DVMRP processes.
Rcv Pkts	Number of packets receive in the interface.
Rcv Bad Pkts	Number of DVMRP messages received on the interface by the DVMRP process which were subsequently discarded as invalid (e.g. invalid packet format, or a route report from an unknown neighbour).
Sent Pkts	Number of packets sent to the interface.

Examples To display DVMRP counters, use the command:

```
sh dvm cou
```

Related Commands

- [add dvmrp interface](#)
- [delete dvmrp interface](#)
- [disable dvmrp](#)
- [disable dvmrp debug](#)
- [enable dvmrp](#)
- [enable dvmrp debug](#)
- [reset dvmrp interface](#)
- [set dvmrp interface](#)

show dvmrp debug

Syntax SHow DVMrp DEBug

Description This command displays the interface debugging options ([Figure 27-6](#), [Table 27-2](#)).

Figure 27-6: Example output from the **show dvmrp debug** command

```
DVMRP Debug
Interface      Debug Options
-----
vlan1          None
vlan2          None
-----
```

Table 27-2: Parameters in output of the **show dvmrp debug** command

Parameter	Meaning
Interface	IP interfaces running DVMRP processes.
Debug Options	Debugging options enabled on the specified interface.

Examples To display the debug options that are enabled, use the command:

```
sh dvm deb
```

Related Commands

- [disable dvmrp](#)
- [disable dvmrp debug](#)
- [enable dvmrp](#)
- [enable dvmrp debug](#)
- [reset dvmrp interface](#)

show dvmrp forwarding

Syntax SHow DVMrp FORwarding

Description This command displays the DVMRP forwarding table (Figure 27-7, Table 27-3).

Figure 27-7: Example output from the **show dvmrp forwarding** command

```
DVMRP forwarding table
Source Address      Source Mask      Group           In Port      Pruned Up
Forwarding Ports<DS|Prune|DR|LocalHost>
-----
128.7.6.4          255.255.255.255  224.5.5.5       Vlan1        No
Vlan2<1|0|Yes|No> Fr0-1<0|0|Yes|Yes>
-----
```

Table 27-3: Parameters in output of the **show dvmrp forwarding** command

Parameter	Meaning
Interface	IP interfaces running DVMRP processes.
Source Address	IP address of sources for multicast packets.
Source Mask	Mask for the Source Address.
Group	IP address of the multicast group.
In Port	Interface for incoming packets.
Forwarding Ports	Interface from which packets are forwarded.

Examples To display the DVMRP forwarding table, use the command:

```
sh dvm for
```

Related Commands

- [add dvmrp interface](#)
- [delete dvmrp interface](#)
- [disable dvmrp](#)
- [disable dvmrp debug](#)
- [enable dvmrp](#)
- [enable dvmrp debug](#)
- [reset dvmrp interface](#)
- [set dvmrp interface](#)

show dvmrp interface

Syntax SHow DVMrp INTerface

Description This command displays the DVMRP interface list (Figure 27-8, Table 27-4). Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

Figure 27-8: Example output from the **show dvmrp interface** command

```
DVMRP Interface Table
Interface      Metric
-----
vlan1          001
vlan2          001
-----
```

Table 27-4: Parameters in output of the **show dvmrp interface** command

Parameter	Meaning
Interface	IP interfaces running DVMRP processes.
Metric	Metrics on a DVMRP interface. This metric is added to all routes that are learned via this interface

Examples To display a list of all DVMRP interfaces, use the command:

```
sh dvm int
```

Related Commands

- add dvmrp interface
- delete dvmrp interface
- disable dvmrp
- disable dvmrp debug
- enable dvmrp
- enable dvmrp debug
- reset dvmrp interface
- set dvmrp interface

show dvmrp neighbour

Syntax SHow DVMrp NEighbour

Description This command displays the contents of the DVMRP Neighbour Table (Figure 27-9, Table 27-5).

Figure 27-9: Example output from the **show dvmrp neighbour** command

```
DVMRP Neighbour Table
Interface  IP Address      Two Way
-----
vlan1     192.168.196.2   No
-----
```

Table 27-5: Parameters in output of the **show dvmrp neighbour** command

Parameter	Meaning
Interface	IP interfaces running DVMRP processes.
IP address	IP address of the DVMRP neighbour.
Two Way	Whether the DVMRP neighbour has also recognised this switch as its neighbour.

Examples To display the DVMRP neighbour table, use the command:

```
sh dvm nei
```

Related Commands

- [add dvmrp interface](#)
- [delete dvmrp interface](#)
- [disable dvmrp](#)
- [disable dvmrp debug](#)
- [enable dvmrp](#)
- [enable dvmrp debug](#)
- [reset dvmrp interface](#)
- [set dvmrp interface](#)

show dvmrp route

Syntax SHow DVMrp ROUte

Description This command displays the internal DVMRP routing table ([Figure 27-10](#), [Table 27-6](#)).

Figure 27-10: Example output from the **show dvmrp route** command

DVMRP Routing Table				
Source Address	Source Mask	Metric	Next Hop	Hold Down
Designated Route				
Dependent Neighbours				

128.1.0.0	255.255.0.0	8	PPP0->137.39.3.93	No
vlan1->me	vlan2->me			
vlan1->138.5.8.1	vlan2->139.4.4.0			
128.2.0.0	155.255.0.0	3	vlan1->128.7.5.2	No
vlan2->139.4.4.0				

Table 27-6: Parameters in output of the **show dvmrp route** command

Parameter	Meaning
Source Address	IP address of sources for multicast packets.
Source Mask	Mask for the Source Address.
Metric	Administrative metric assigned to this route.
Next Hop	IP address of the next hop router to get to the source.
Hold Down	Whether the route is held down. If yes, the route is no longer available but has not been deleted from the multicast routing table.
Designated Router	Interface and IP address of the designated router for this interface, or "me" if this switch is the designated router for the interface.
Dependent Neighbours	Interface and IP address of DVMRP neighbours dependent on this switch.

Examples To display the DVMRP routing table, use the command:

```
sh dvm rou
```

Related Commands

- [add dvmrp interface](#)
- [delete dvmrp interface](#)
- [disable dvmrp](#)
- [disable dvmrp debug](#)
- [enable dvmrp](#)
- [enable dvmrp debug](#)
- [reset dvmrp interface](#)
- [set dvmrp interface](#)

show igmp filter

Syntax `SHoW IGMP FILTer [=filter-id]`

where *filter-id* is a decimal number in the range 1 to 99

Description This command displays information about an IGMP filter or all IGMP filters (Figure 27-11, Table 27-7). If a **filter** is specified, only information about that filter is displayed.

Figure 27-11: Example output from the **show igmp filter** command

IGMP Filters							
No.	Entry	Group Address Range	Msg Type	Action	Matches		
1	224	224.1.2.3	Report	Exclude	10		
	229	229.1.1.1	Leave	Include	2		
Reports		- Recd:	80	Passed:	70	Dropped:	10
Queries		- Recd:	0	Passed:	0	Dropped:	0
Leaves		- Recd:	2	Passed:	2	Dropped:	0

Table 27-7: Parameters in the output of the **show igmp filter** command

Parameter	Meaning
No.	The filter number.
Entry	The entry number of an entry in this filter.
Group Address Range	The multicast group address range for this entry.
Msg Type	The type of IGMP message being filtered by this entry; one of "Leave", "Query", or "Report".
Action	The action to take when an IGMP message matching the message type and group address of this entry is received.
Matches	The number of IGMP messages received that were matched by this entry.
Reports, Queries, Leaves	The total number of IGMP messages of the specified type that were received and processed on all the switch ports that this filter is attached to.
Recd	The number of IGMP messages of the specified type that were received on all the switch ports that this filter is attached to.
Passed	The number of IGMP messages of the specified type that were received and accepted on all the switch ports that this filter is attached to.
Dropped	The number of IGMP messages of the specified type that were received and discarded on all the switch ports that this filter is attached to.

Examples To display information about IGMP filter 3, use the command:

```
sh igmp fil=3
```

Related Commands [add igmp filter](#)
[create igmp filter](#)
[delete igmp filter](#)
[destroy igmp filter](#)
[set igmp filter](#)

show igmpsnooping

Syntax SHow IGMPsNooping [VLAN={*vlan-name*|1..4094}]

where *vlan-name* is a unique name for the VLAN 1 to 32 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and the hyphen.

Description This command displays information about IGMP snooping on a VLAN or VLANs (Figure 27-12, Table 27-8).

If a **vlan** is specified, only output for that VLAN is displayed.

Figure 27-12: Example output from the **show igmpsnooping** command

```
IGMP Snooping
-----
Status ..... Enabled
Disabled All-groups ports ..... None

Vlan Name (vlan id) ..... default (1)
Fast Leave ..... On
Static Router Ports ..... None
Query Solicitation ..... Off
Group List .....

    Group. 225.1.2.3                Entry timeout 268 secs
    Ports 16,19

    Group. 239.1.2.3                Entry timeout 180 secs
    Ports 21

Vlan Name (vlan id) ..... vlan2 (2)
Fast Leave ..... On
Group List .....

    All Groups                      Entry timeout 255 secs
    Ports 13

Vlan Name (vlan id) ..... vlan3 (3)
Fast Leave ..... Off
Group List .....

    No group memberships.
-----
```

Table 27-8: Parameters in output of the **show igmpsnooping** command

Parameter	Meaning
Status	Whether IGMP snooping is enabled.
Disabled All-groups ports	A list of ports that are disallowed from acting as an all-groups port.
VLAN Name (vlan id)	The name and VID of the VLAN where IGMP snooping is operating.
Fast Leave	Whether Fast Leave processing is enabled on this VLAN.

Table 27-8: Parameters in output of the **show igmpsnooping** command (cont.)

Parameter	Meaning
Static Router Ports	A list of ports that have been statically configured as multicast router ports. These are in addition to any ports that the switch dynamically determines are multicast router ports.
Query Solicitation	Whether query solicitation is enabled on this VLAN.
Group List	A list of multicast group memberships for this VLAN.
Group	Group multicast address.
All Groups	This entry lists ports that IGMP snooping has identified as members of all groups, for example, ports connected to routers.
Entry timeout	Time in seconds until the group's entry is deleted if no other IGMP messages for the group are seen.
Ports	A list of ports listening to this group.

Examples To display information about IGMP snooping, use the command:

```
sh igmpsn
```

To display information about IGMP snooping on VLAN 2, use the command:

```
sh igmpsn vlan=2
```

Related Commands

- [disable igmpsnooping](#)
- [disable ip igmp](#)
- [enable igmpsnooping](#)
- [enable ip igmp](#)
- [set igmpsnooping vlan](#)
- [set ip igmp](#)
- [show igmpsnooping counter](#)
- [show igmpsnooping routeraddress](#)

show igmpsnooping counter

Syntax SHow IGMPsNooping COUnter [VLAN={*vlan-name*|1..4094}]

where *vlan-name* is a unique name for the VLAN 1 to 32 characters long. Valid characters are uppercase and lowercase letters, digits, the underscore, and the hyphen.

Description This command displays IGMP snooping counters on a VLAN or VLANs (Figure 27-13, Table 27-9).

If a **vlan** is specified, only output for that VLAN is displayed.

Figure 27-13: Example output from the **show igmpsnooping counter** command

```

IGMP Snooping Counters
-----

Vlan Name=default (Vlan Id=1):

  inQuery ..... 817          badQuery ..... 0
  inV1Report ..... 0         badV1Report ..... 0
  inV2Report ..... 1265      badV2Report ..... 0
  inLeave ..... 0             badLeave ..... 0
  inRouterMsg ..... 4488     badRouterMsg ..... 0
  inTotal ..... 6570        badTotal ..... 0
-----

```

Table 27-9: Parameters in output of the **show igmpsnooping counter** command

Parameter	Meaning
inQuery	The number of IGMP membership query messages that were received by the interface.
badQuery	The number of IGMP membership query messages with errors that were received by the interface.
inV1Report	The number of IGMP Version 1 membership report messages that were received by the interface.
badV1Report	The number of IGMP Version 1 membership report messages with errors that were received by the interface.
inV2Report	The number of IGMP Version 2 membership report messages that were received by the interface.
badV2Report	The number of IGMP Version 2 membership report messages with errors that were received by the interface.
inLeave	The number of IGMP Version 2 Leave Group messages that were received by the interface.
badLeave	The number of IGMP Version 2 Leave Group messages with errors that were received by the interface.
inRouterMsg	The number of multicast packets received that were destined for 224.0.0.x. These messages indicate that a router is present on the port.
badRouterMsg	The number of multicast packets received with errors that were destined for 224.0.0.x.

Table 27-9: Parameters in output of the **show igmpsnooping counter** command (cont.)

Parameter	Meaning
inTotal	The total number of IGMP messages that were received by the interface.
badTotal	The total number of IGMP messages with errors that were received by the interface.

Examples To display IGMP snooping counters for all VLANs, use the command:

```
sh igmpsn cou
```

To display IGMP snooping counters for VLAN 2, use the command:

```
sh igmpsn cou vlan=2
```

Related Commands

- [disable igmpsnooping](#)
- [disable ip igmp](#)
- [enable igmpsnooping](#)
- [enable ip igmp](#)
- [set ip igmp](#)
- [show igmpsnooping](#)
- [show igmpsnooping routeraddress](#)

show igmpsnooping routeraddress

Syntax SHow IGMPsNooping ROUTERAddress

Description This command displays the current list of configured IP multicast router addresses configured on the switch.

Figure 27-14: Example output from the **show igmpsnooping routeraddress** command

```
IGMP Snooping Router Address
-----
IGMP Snooping Router Mode ..... default

Router Address List
-----
224.0.0.1      224.0.0.4      224.0.0.6      224.0.0.13
224.0.0.2      224.0.0.5      224.0.0.9      224.0.0.15
-----
```

Table 27-10: Parameters in output of the **show igmpsnooping routeraddress** command

Parameter	Meaning
IGMP Snooping Router Mode	The current IGMP Snooping router mode: all, default, multicastrouter, none, or ip.
Router Address List	A list of configured reserved IP multicast addresses that are treated as multicast router addresses.

Examples To show the current list of configured router multicast addresses, use the command:

```
sh igmpsn routera
```

Related Commands

- [add igmpsnooping routeraddress](#)
- [delete igmpsnooping routeraddress](#)
- [set igmpsnooping routermode](#)
- [show igmpsnooping](#)
- [show igmpsnooping counter](#)

show ip igmp

Syntax SHow IP IGMP [INTerface=*interface*] [DEStination=*ipadd*]

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IGMP multicast group address in dotted decimal notation.

Description This command displays general information about IGMP and the IGMP configuration on each IP interface ([Figure 27-15 on page 27-115](#), [Table 27-11 on page 27-115](#)).

If an **interface** is specified, information is displayed only for that interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- FR (such as fr0, fr0-1)
- VLAN (such as vlan1, vlan1-1)

If a **destination** address is specified, information is displayed only for interfaces that have a multicast group membership matching the **destination** address. Any of the four octets of the IP address may be replaced by an asterix (*) to enable wildcard matches, for example 224.*.*.*

Figure 27-15: Example output from the **show ip igmp** command

```

IGMP Protocol
-----
Status ..... Enabled
Default Query Interval ..... 125 secs
Default Timeout Interval ..... 260 secs

Last Member Query Interval ..... 10 (1/10secs)
Last Member Query Count ..... 2
Robustness Variable ..... 2
Query Response Interval ..... 100 (1/10secs)
Disabled All-groups ports ..... 1,5,7

Interface Name ..... vlan1 (DR)
Status ..... Enabled
Other Querier timeout ..... 164 secs
IGMP Proxy ..... Upstream
General Query Reception Timeout .... None
Group List .....

Group. 224.0.1.22      Last Adv. 10.194.254.254      Refresh time 184 secs
Ports 24

Group. 224.0.1.22      Static association      Refresh time Infinity
Ports 11-14,17,19
Static Ports 17,19

All Groups      Last Adv. 10.116.2.1      Refresh time 254 secs
Ports 24
-----

```

Table 27-11: Parameters in output of the **show ip igmp** command

Parameter	Meaning
General information about IGMP	
Status	Whether IGMP is enabled.
Default Query Interval	The default interval at which Host Membership Queries are sent.
Default Timeout Interval	The default interval after which entries are removed from the group database when no Host Membership Report is received.
Last Member Query Interval	Max Response Time inserted into Group-Specific Queries sent in response to Leave Group messages, and the amount of time between Group-Specific Query messages.
Last Member Query Count	The number of Group-Specific Queries sent before the switch assumes there are no local members.
Robustness Variable	IGMP is robust to (Robustness Variable-1) packet losses.
Query Response Interval	The Max Response Time (in 1/10 secs) inserted into the periodic General Queries.
Disabled All-groups ports	A list of ports that are disabled from acting as an all-groups port.

Table 27-11: Parameters in output of the **show ip igmp** command (cont.)

Parameter	Meaning
Information about each IP interface	
Interface Name	The name of an IP interface.
Status	Whether IGMP is enabled on the interface.
Other Querier timeout	The time that remains before a multicast router decides that there is no longer another multicast router that should be the querier.
IGMP Proxy	The status of IGMP Proxy on this interface; one of "Off", "Upstream", or "Downstream".
General Query Reception Timeout	The maximum expected time interval, in seconds, between successive IGMP general query messages arriving on the interface, or "none" if there is no limit. If a general query message is not received within the time interval, a log message and an SNMP trap are generated.
Group List	A list of multicast group memberships for this interface, or: " No group memberships", if the interface has no multicast group members " No matching group memberships", if the interface has no multicast group members matching the destination address
Group	The group multicast address.
Last Adv.	The last host to advertise the membership report, or "Static association" for static multicast groups.
Refresh time	The time in seconds until the membership group is deleted if another membership report is not received, or "Infinity" for static multicast associations.
Ports	The list of ports listening to this group.
Static Ports	The list of static ports listening to this group. This is a subset of the ports listed in the Ports field, and is only displayed for static groups on a VLAN.

Examples To display general information about IGMP, use the command:

```
sh ip igmp
```

To limit the display to IP interfaces that have multicast group memberships matching 224.*.*, use the command:

```
sh ip igmp des=224.*.*.*
```

To display information about IGMP on interface "vlan1", use the command:

```
sh ip igmp int=vlan1
```

Related Commands

- [add ip igmp destination](#)
- [add ip interface](#)
- [create ip igmp destination](#)
- [delete ip igmp destination](#)
- [destroy ip igmp destination](#)
- [disable ip igmp](#)
- [disable ip igmp interface](#)
- [enable ip igmp](#)
- [enable ip igmp interface](#)
- [set ip igmp](#)
- [set ip interface](#)
- [show ip igmp counter](#)
- [show ip route multicast](#)

show ip igmp counter

Syntax SHow IP IGMP COUnter [INTErface=*interface*]
[DEStination=*ipaddress*]

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IGMP multicast group address in dotted decimal notation.

Description This command displays IGMP counters (see [Figure 27-16 on page 27-118](#), [Table 27-12 on page 27-119](#)).

If an **interface** is specified, counters are displayed only for that interface. Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- FR (such as fr0, fr0-1)
- VLAN (such as vlan1, vlan1-1)

If a **destination** address is specified, counters are displayed only for interfaces that have a multicast group membership matching the **destination** address. Any of the four octets of the IP address may be replaced by an asterisk (*) to enable wildcard matches, for example 224.*.*.*

Figure 27-16: Example output from the **show ip igmp counter** command

```
IGMP Counters
-----

Interface Name: vlan1

inQuery ..... 1          outQuery ..... 5
inV1Report ..... 4
inV2Report ..... 7
inLeave ..... 0
inTotal ..... 12        outTotal ..... 5

badQuery ..... 0
badV1Report ..... 0
badV2Report ..... 0
badLeave ..... 0
badTotal ..... 1
-----
```

Table 27-12: Parameters in output of the **show ip igmp counter** command

Parameter	Meaning
Interface Name	The name of an IP interface.
inQuery	The number of IGMP membership query messages that were received by the interface.
outQuery	The number of IGMP membership query messages that were transmitted by the switch for the interface.
inV1Report	The number of IGMP Version 1 membership report messages that were received by the interface.
inV2Report	The number of IGMP Version 2 membership report messages that were received by the interface.
inLeave	The number of IGMP Leave Group messages that were received by the interface.
inTotal	The total number of IGMP messages that were received by the interface.
outTotal	The total number of IGMP messages that were transmitted by the switch over the interface.
badQuery	The number of IGMP membership query messages with errors that were received by the interface.
badV1Report	The number of IGMP Version 1 membership report messages with errors that were received by the interface.
badV2Report	The number of IGMP Version 2 membership report messages with errors that were received by the interface.
badLeave	The number of IGMP Leave Group messages with errors that were received by the interface.
badLength	The number of IGMP packets received by the interface that were discarded due to an invalid packet length. This field is displayed only when IGMP debugging is enabled.
badChecksum	The number of IGMP packets received by the interface that were discarded due to an invalid packet checksum. This field is displayed only when IGMP debugging is enabled.
badType	The number of IGMP packets received by the interface that were discarded due to an unknown IGMP packet type. This field is displayed only when IGMP debugging is enabled.
badDest	The number of IGMP packets received by the interface that were discarded because they contained IGMP leave or report messages addressed to the all hosts group. This field is displayed only when IGMP debugging is enabled.
badNoReceiver	The number of IGMP packets received by the interface that were discarded because there was no handler for the packet type. This field is displayed only when IGMP debugging is enabled.
badTotal	The total number of IGMP messages with errors that were received by the interface.

Examples To display IGMP counters for all IP interfaces, use the command:

```
sh ip igmp cou
```

To limit the display to IP interfaces that have multicast group memberships matching 224.*.*, use the command:

```
sh ip igmp cou des=224.*.*
```

To display IGMP counters for interface "vlan1", use the command:

```
sh ip igmp cou int=vlan1
```

Related Commands

- [add ip igmp destination](#)
- [add ip interface](#)
- [create ip igmp destination](#)
- [delete ip igmp destination](#)
- [destroy ip igmp destination](#)
- [disable ip igmp](#)
- [disable ip igmp interface](#)
- [enable ip igmp](#)
- [enable ip igmp interface](#)
- [set ip igmp](#)
- [set ip interface](#)
- [show ip igmp](#)
- [show ip route multicast](#)

show ip igmp debug

Syntax SHow IP IGMP DEBug

Description This command shows the IGMP debugging options that have been set.

Figure 27-17: Example output from **show ip igmp debug** command

```
IGMP Debugging Information
-----
IGMP Debugging           Enabled
Filter by group destination 224.1.2.3
Filter by source IP       10.10.1.123
-----
```

Table 27-13: Parameters in output of the **show ip igmp debug** command

Parameter	Meaning
IGMP Debugging	Whether IGMP debugging is enabled.
Filter by group destination	Group Destination Address specified by the destination parameter in the enable ip igmp debug command. When no parameter is given, "No" is displayed instead of the IP address.
Filter by source IP	Source IP address specified by the sourceipaddress parameter in the enable ip igmp debug command. When no parameter is given, "No" is displayed instead of the IP address.

Examples To display IGMP debugging information, use the command:

```
sh ip igmp deb
```

Related Commands [disable ip igmp debug](#)
[enable ip igmp debug](#)

show ip mvr

Syntax SHow IP MVR [VLAN=1..4094]

Description This command displays all information about MVR on the switch (Figure 27-18, Table 27-14).

The **vlan** parameter specifies the VLAN identifier of the multicast VLAN to be displayed. If none is provided, all multicast VLANs on the switch are displayed.

Figure 27-18: Example output from the **show ip mvr** command

Multicast VLAN					
VLAN	Mode	Imtleave	Source Ports	Receiver Ports	
				Current Members	Group Address
22	compatible	3	9,10	1-3, 6-7	
				1,6	235.1.1.1
				2,7	234.1.1.1
3	compatible	8	12,13	4,5,8,9	
				4,8	255.1.1.1

Table 27-14: Parameters in output of the **show ip mvr** command

Parameter	Meaning
VLAN	Multicast VLAN number.
Mode	Mode in which IGMP report packets are either sent out through source ports in the multicast VLAN, ("dynamic") or not sent out ("compatible").s
IMTLEAVE	Port number on which the Immediate Leave feature is enabled.
Source ports	Uplink ports that send or receive multicast data to and from the multicast VLAN.
Receiver ports	Ports that receive multicast data when they are the member of a multicast group.
Current Members	Receiver ports that are currently members of the multicast group.
Group Address	Multicast group IP addresses owned by this multicast VLAN.

Examples To show information for multicast VLAN number 22, use the command:

```
sh ip mvr vlan=22
```

Related Commands [show ip mvr counter](#)

show ip mvr counter

Syntax `SHoW IP MVR [VLAN=1..4094] COUnTer`

Description This command displays the number of times a port has joined and/or left a multicast VLAN ([Figure 27-19](#), [Table 27-15](#)).

The **vlan** parameter specifies the VLAN identifier of the Multicast VLAN to be displayed. If none is provided, all multicast VLANs on the switch are displayed.

Figure 27-19: Example output from the **show ip mvr counter** command

Multicast VLAN			
VLAN	Group Address	Joins	Leaves
23	235.1.1.1	16	15
	215.1.1.1	9	8
3	225.1.1.1	3	2

Table 27-15: Parameters in output of the **show ip mvr counter** command

Parameter	Meaning
VLAN	Multicast VLAN number.
Group Address	Multicast group IP address owned by this multicast VLAN.
Joins	Number of times the receiver port has become a member of the IP multicast stream.
Leaves	Number of times the receiver port has left the IP multicast stream.

Examples To display multicast VLAN counter information, use the command:

```
sh ip mvr vlan=22 cou
```

Related Commands [show ip mvr](#)

show pim

Syntax SHow PIM

Description This command displays detailed information about the PIM routing status on the switch, and is equivalent to specifying all of the following commands in the following order:

1. show PIM interface
2. show PIM route
3. show PIM neighbour
4. show PIM counters
5. show PIM debug
6. show PIM rpcandidate
7. show PIM bsrcandidate
8. show PIM rpset
9. show PIM timer
10. show PIM config

Examples To display detailed PIM routing status information, use the command:

```
sh pim
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show pim bsrcandidate](#)
- [show pim counters](#)
- [show pim debug](#)
- [show pim interface](#)
- [show pim neighbour](#)
- [show pim route](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)
- [show pim timer](#)

show pim bsr candidate

Syntax SHow PIM BSRCandidate

Description This command displays information about the switch as a BSR candidate for PIM-SM (Figure 27-20, Figure 27-21, Table 27-16).

Figure 27-20: Example output from the **show pim bsr candidate** command for an elected BSR

```
PIM BSR Candidate
-----
Preference ..... 1
BSR State ..... Elected BSR
  Elected BSR IP address ..... 101.202.101.202
  Elected BSR preference ..... 1
```

Figure 27-21: Example output from the **show pim bsr candidate** command for an unelected BSR candidate

```
PIM BSR Candidate
-----
BSR State ..... Accepts Preferred BSM
  Elected BSR IP address ..... 101.202.101.202
  Elected BSR preference ..... 1
```

Table 27-16: Parameters in output of the **show pim bsr candidate** command

Parameter	Meaning
Preference	The preference value for the switch to be a candidate bootstrap router. The higher the number, the higher the priority. This parameter is present when the switch is the elected BSR.
BSR State	Current status of the BSR; one of "Accepts Preferred BSM" (the switch is available to become the BSR), or "Elected BSR" (the switch is the BSR).
Elected BSR IP address	IP address of the BSR. If the switch is the BSR, this address is one of the switch's addresses.
Elected BSR preference	The preference of the BSR. When the switch is the BSR, this is its preference.

Examples To display information about the switch as a BSR candidate, use the command:

```
sh pim bsr
```

Related Commands

- [add pim bsr candidate](#)
- [delete pim bsr candidate](#)
- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [set pim bsr candidate](#)
- [show ip](#)
- [show pim](#)

show pim config

Syntax SHow PIM CONFig

Description This command lists the command line interface commands that make up the PIM configuration ([Figure 27-22](#)).

Figure 27-22: Example output from the **show pim config** command

```
#PIM4 configuration
#
add pim interface=vlan1
add pim interface=vlan2 drpriority=100
enable pim
```

Examples To display the PIM configuration, use the command:

```
sh pim conf
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show config](#)

show pim counters

Syntax SHow PIM COunters

Description This command displays information about PIM counters (Figure 27-23, Figure 27-24, Table 27-17 on page 27-128).

Figure 27-23: Example output from the **show pim counters** command for PIM Sparse Mode

```
PIM4 Counters
-----
Sparse Mode
-----
vlan1:
  inHello ..... 14      outHello ..... 15
  inRegister ..... 0     outRegister ..... 0
  inRegisterStop ..... 0 outRegisterStop ..... 0
  inJP ..... 0          outJP ..... 0
  inAssert ..... 0       outAssert ..... 0
  inBSM ..... 8         outBSM ..... 3
  inCRPAdv ..... 0       outCRPAdv ..... 0
  inTotal ..... 22      outTotal ..... 18

vlan1 Bad:
  badHello ..... 0
  badRegister ..... 0
  badRegisterStop ..... 0
  badJP ..... 0
  badAssert ..... 0
  badBSM ..... 0
  badCRPAdv ..... 0
  badTotal ..... 0
```

Figure 27-24: Example output from the **show pim counters** command for PIM Dense Mode

```
PIM4 Counters
-----
Dense Mode
-----
vlan1:
  inHello ..... 25      outHello ..... 26
  inGraft ..... 0       outGraft ..... 0
  inGraftAck ..... 0     outGraftAck ..... 0
  inJP ..... 0          outJP ..... 0
  inAssert ..... 0       outAssert ..... 0
  inSRM ..... 0         outSRM ..... 0
  inTotal ..... 25      outTotal ..... 26

vlan1 Bad:
  badHello ..... 0
  badGraft ..... 0
  badGraftAck ..... 0
  badJP ..... 0
  badAssert ..... 0
  badTotal ..... 0
```

Table 27-17: Parameters in output of the **show pim counters** command

Parameter	Meaning
inHello	The number of PIM hello messages received by the interface.
inRegister	The number of PIM register messages that were received by the interface. This parameter is displayed for PIM-SM interfaces only.
inRegisterStop	The number of PIM register stop messages received by the interface. This parameter is displayed for PIM-SM interfaces only.
inGraft	The number of PIM graft messages received by the interface. This parameter is displayed for PIM-DM interfaces only.
inGrackAck	The number of PIM graft acknowledgement messages that were received by the interface. This parameter is displayed for PIM-DM interfaces only.
inJP	The number of PIM join and prune messages received by the interface.
inAssert	The number of PIM assert messages received by the interface.
inBSM	The number of PIM bootstrap messages received by the interface. This parameter is displayed for PIM-SM interfaces only.
inCRPAdv	The number of PIM candidate RP advertisement messages received by the interface. This parameter is displayed for PIM-SM interfaces only.
inSRM	The number of PIM state refresh messages received by the interface. This parameter is displayed for PIM-DM interfaces only.
inTotal	The total number of PIM messages received by the interface.
outHello	The number of PIM hello messages transmitted by the interface.
outRegister	The number of PIM register messages transmitted by the interface. This parameter is displayed for PIM-SM interfaces only.
outRegisterStop	The number of PIM register stop messages transmitted by the interface. This parameter is displayed for PIM-SM interfaces only.
outGraft	The number of PIM graft messages transmitted by the interface. This parameter is displayed for PIM-DM interfaces only.
outGrackAck	The number of PIM graft acknowledgement messages transmitted by the interface. This parameter is displayed for PIM-DM interfaces only.
outJP	The number of PIM join and prune messages transmitted by the interface.
outAssert	The number of PIM assert messages transmitted by the interface.

Table 27-17: Parameters in output of the **show pim counters** command (cont.)

Parameter	Meaning
outBSM	The number of PIM bootstrap messages transmitted by the interface. This parameter is displayed for PIM-SM interfaces only.
outCRPAdv	The number of PIM candidate RP advertisement messages transmitted by the interface. This parameter is displayed for PIM-SM interfaces only.
outSRM	The number of PIM state refresh messages transmitted by the interface. This parameter is displayed for PIM-DM interfaces only.
outTotal	The total number of PIM messages that were transmitted by the interface.
badHello	The number of PIM hello messages with errors that were received by the interface.
badRegister	The number of PIM register messages with errors received by the interface. This parameter is displayed for PIM-SM interfaces only.
badRegisterStop	The number of PIM register stop messages with errors received by the interface. This parameter is displayed for PIM-SM interfaces only.
badGraft	The number of PIM graft messages with errors received by the interface. This parameter is displayed for PIM-DM interfaces only.
badGrackAck	The number of PIM graft acknowledgement messages with errors received by the interface. This parameter is displayed for PIM-DM interfaces only.
badJP	The number of PIM join and prune messages with errors received by the switch.
badAssert	The number of PIM assert messages with errors received by the interface.
badBSM	The number of PIM bootstrap messages with errors received by the interface. This parameter is displayed for PIM-SM interfaces only.
badCRPAdv	The number of PIM candidate RP advertisement messages with errors received by the interface. This parameter is displayed for PIM-SM interfaces only.
badSRM	The number of PIM state refresh messages with errors received by the interface. This parameter is displayed for PIM-DM interfaces only.
badTotal	The total number of PIM messages with errors received by the interface.

Examples To display information about PIM counters, use the command:

```
sh pim cou
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show pim](#)

show pim bsrcandidate
show pim debug
show pim interface
show pim neighbour
show pim route
show pim rpcandidate
show pim rpset
show pim timer

show pim debug

Syntax SHow PIM DEBug

Description This command displays the list of PIM interface debugging options (Figure 27-25, Table 27-18).

Figure 27-25: Example output from the **show pim debug** command

```
PIM Debug Options
-----
Debug Options Enabled: Join, Assert

Logging Options Enabled : status

Trapping Options Enabled: none

Info (1097049): The PIM module is not enabled.
```

Table 27-18: Parameters in output of the **show pim debug** command

Parameter	Meaning
Debug Options Enabled	A comma-separated list of the PIM debugging options that are enabled, or "None" if debugging is disabled, or "All" if all debugging is enabled. Options are listed with the enable pim debug command on page 27-84.
Logging Options Enabled	The logging options that are enabled. See set pim log command on page 27-95.
Trapping Options Enabled	The trapping options that are enabled. See set pim log command on page 27-95.

Examples To display a list of enabled PIM interface debugging options, use the command:

```
sh pim deb
```

Related Commands

- [disable pim](#)
- [disable pim debug](#)
- [disable debug active](#) in Chapter 4, Configuring and Monitoring the System
- [enable pim](#)
- [enable pim debug](#)
- [set pim log](#)
- [show ip](#)
- [show pim](#)
- [show pim bsrcandidate](#)
- [show pim counters](#)
- [show debug active](#) in Chapter 4, Configuring and Monitoring the System
- [show pim interface](#)
- [show pim neighbour](#)
- [show pim route](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)
- [show pim timer](#)

show pim interface

Syntax SHOW PIM INTERface

Description This command displays information about all PIM interfaces and their designated router status (Figure 27-26, Figure 27-27, Table 27-19). Valid interfaces are:

- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1, vlan1-1)
- FR (such as fr0, fr0-1)

Figure 27-26: Example output from the **show pim interface** command for PIM Sparse Mode.

```
PIM4 Sparse mode Interface Table
-----
Interface ..... vlan1
  IP address ..... 172.128.71.25
    DR election by ..... DR priority
    DR priority ..... 10
    DR winner ..... Me
  Hello interval ..... 30

Interface ..... vlan2
  IP address ..... 172.128.72.33
    DR election by ..... DR priority
    DR priority ..... 1
    DR winner ..... 172.128.72.14
  Hello interval ..... 30
```

Figure 27-27: Example output from the **show pim interface** command for PIM Dense Mode

```
PIM4 Dense mode Interface Table
-----
Interface ..... vlan2
  IP address ..... 192.168.0.111
  State refresh capable ..... No
  Hello interval ..... 30
```

Table 27-19: Parameters in output of the **show pim interface** command

Parameter	Meaning
Interface	IP interfaces running PIM processes.
Dci	The data link circuit identifier for a Frame Relay interface.
IP Address	The IP address on this interface.
DR election by	How this interface elects a DR; one of "DR priority" (the DR priority is transmitted in Hello messages and election is by priority), or "IP address" (the DR priority is not transmitted in Hello messages so election is by IP address).
DR priority	The priority for the DR candidate to become the PIM designated router. A candidate with a higher priority is more likely to become the DR.

Table 27-19: Parameters in output of the **show pim interface** command (cont.)

Parameter	Meaning
DR Winner	The IP address of the PIM designated router for the interface, or "me" when this switch is the designated router.
State refresh capable	Whether this interface originates and processes State Refresh messages for PIM-DM.
Hello interval	The interval, in seconds, at which the switch sends PIM Hello messages on this interface. The value 65535 indicates that the Hello message never expires.

Examples To display information about all PIM interfaces, use the command:

```
sh pim int
```

Related Commands

- [add pim interface](#)
- [delete pim interface](#)
- [disable pim](#)
- [enable pim](#)
- [reset pim interface](#)
- [set pim interface](#)
- [set pim](#)
- [show ip](#) in Chapter 23, Internet Protocol (IP)
- [show pim](#)
- [show pim bsrcandidate](#)
- [show pim counters](#)
- [show pim debug](#)
- [show pim neighbour](#)
- [show pim route](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)
- [show pim timer](#)

show pim neighbour

Syntax SHow pim NEighbour

Description This command displays information about the PIM Neighbour Table (Figure 27-28, Figure 27-29, Table 27-20).

Figure 27-28: Example output from the **show pim neighbour** command for PIM Sparse Mode

```
PIM4 Sparse mode Neighbour Table
-----
Interface ..... vlan1
  IP Address ..... 137.39.3.93
    DR Priority ..... 1
      Neighbour Liveness Timer ..... 82
```

Figure 27-29: Example output from the **show pim neighbour** command for PIM Dense Mode

```
PIM4 Dense mode Neighbour Table
-----
Interface ..... vlan1
  IP Address ..... 192.168.57.2
    Neighbour Liveness Timer ..... 105
      Is state refresh capable ..... No
```

Table 27-20: Parameters in output of the **show pim neighbour** command

Parameter	Meaning
Interface	Interface to which the PIM neighbour is connected.
IP Address	IP address of the neighbour.
DR Priority	Priority for this neighbour to become the designated router for the subnetwork.
Neighbour Liveness Timer	Time in seconds until the neighbour is removed from the neighbour table.
Is state refresh capable	Whether the neighbour originates and processes State Refresh messages for PIM-DM.

Examples To display information about the PIM Neighbour Table, use the command:

```
sh pim nei
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)

show pim route

Syntax SHow PIM ROUTe

Description This command displays information about the internal PIM routing table, for PIM Sparse Mode (Figure 27-30, Figure 27-31 on page 27-136, Table 27-21 on page 27-136) and/or Dense Mode (Figure 27-32 on page 27-139, Table 27-22 on page 27-139).

Figure 27-30: Example output from the **show pim route** command for PIM Sparse Mode, when the switch is the RP

```
PIM4 Sparse Mode Tree Information Base
-----
Group ..... 224.1.1.1
Type ..... (*,G)
  RP Address ..... I am the RP
  Expiry time ..... 630
  Join/prune time ..... 0
  Immediate output interfaces .. vlan3

Type ..... (S,G)
  Source ..... 192.168.0.1
  RPF Neighbour to Src ..... 192.168.1.1
  RPF Interface to Src ..... vlan1
  Expiry time ..... 180
  Keepalive time ..... 160
  Join/prune time ..... 0
  Register time ..... 0
  SPT bit ..... Unset
  Inherited output interfaces .. vlan3
  Immediate output interfaces .. None

Type ..... (S,G,rpt)
  Source ..... 192.168.0.1
  RP Address ..... I am the RP
  Expiry time ..... 180
  Override time ..... 0
  Inherited output interfaces .. vlan3

Type ..... (*,*,RP)
  RP Address ..... I am the RP
  Expiry time ..... 210
  Join/prune time ..... 0
  Immediate output interfaces .. None
```

Figure 27-31: Example output from the **show pim route** command for PIM Sparse Mode, when the switch is not the RP

```

PIM4 Sparse Mode Tree Information Base
-----
Group ..... 224.1.1.1
  Type ..... (*,G)
    RP Address ..... 192.168.1.1
    RPF Neighbour to RP ..... 192.168.2.1
    RPF Interface to RP ..... vlan1
    Expiry time ..... 630
    Join/prune time ..... 0
    Immediate output interfaces .. vlan3

  Type ..... (S,G)
    Source ..... 192.168.0.1
    RPF Neighbour to Src ..... Directly connected
    RPF Interface to Src ..... vlan2
    Expiry time ..... 230
    Keepalive time ..... 210
    Join/prune time ..... 0
    Register time ..... 21
    SPT bit ..... Unset
    Inherited output interfaces .. vlan3
    Immediate output interfaces .. vlan3

  Type ..... (S,G,rpt)
    Source ..... 192.168.0.1
    RP Address ..... 192.168.1.1
    Expiry time ..... 230
    Override time ..... 0
    Inherited output interfaces .. vlan3

  Type ..... (*,*,RP)
    RP Address ..... 192.168.1.1
    Next hop to RP ..... 192.168.2.1
    RPF Interface to RP ..... vlan1
    Expiry time ..... 210
    Join/prune time ..... 0
    Immediate output interfaces .. None

```

Table 27-21: Parameters in output of the **show pim route** command for PIM Sparse Mode

Entry Parameter for			
Parameter	Type	Entry Type	Meaning
Group			The IP address of the multicast group.
Type			The type of entry in the Tree Information Base.
		(*,G)	The entry for traffic from any source to a particular group.
		RP Address	The IP address of the rendezvous point for the group.
		RPF Neighbour to RP	The address of the PIM neighbour to the RP, taking into account any PIM assert messages. Packets from the RP would be received from this neighbour.
		RPF Interface to RP	The interface on which packets from the RP would be received.

Table 27-21: Parameters in output of the **show pim route** command for PIM Sparse Mode (cont.)

Parameter	Entry Type	Parameter for Entry Type	Meaning
		Expiry time	The time remaining until this entry is deleted, in seconds. A zero value indicates that the timer is not running. This timer decrements when there are no (S,G) entries.
		Join/prune time	The join/prune timer in seconds. When the switch sees a prune message on the correct upstream interface, and it still needs to receive traffic via that rp tree, it sends a join message when this timer expires. A zero value indicates that the timer is not running.
		Immediate output interfaces	The interfaces with downstream routers or IGMP hosts that are interested in this (*,G) entry.
	(S,G)		The entry for traffic from a particular source to a particular group.
		Source	The IP address of the multicast sender.
		RPF Neighbour to Src	The address of the PIM neighbour to the source, taking into account any PIM assert messages. Packets from the source would be received from this neighbour. "Directly connected" indicates that the source is directly connected to the switch.
		RPF Interface to Src	The interface on which packets from the source would be received, if the source is in this multicasting domain.
		Expiry time	The time remaining until this entry is deleted, in seconds. A zero value indicates that the timer is not running. The expiry time is 20 seconds longer than the keepalive time.
		Keepalive time	The Keepalive timer in seconds. A zero value indicates that the timer is not running because no data is being received. When the switch is forwarding multicast data in hardware, the data stream is checked every 60 seconds to verify data is still flowing. If it is, the timer is reset.
		Join/prune time	The join/prune timer in seconds. When the switch sees a prune message on the correct upstream interface, and it still needs to receive traffic via that sp tree, it sends a join message when this timer expires. A zero value indicates that the timer is not running.
		Register time	The register suppression time, in seconds. When this timer reaches the register probe time, a null register message is sent to the RP.
		SPT bit	Whether forwarding is set on the shortest path tree.
		Inherited output interfaces	The interfaces to forward (S,G) data to.
		Immediate output interfaces	The interfaces with downstream routers or IGMP hosts that are interested in this (S,G) data.

Table 27-21: Parameters in output of the **show pim route** command for PIM Sparse Mode (cont.)

Parameter	Entry Type	Parameter for Entry Type	Meaning
(S,G, rpt)			The entry that is used for suppressing traffic on the RP tree from a particular source to a particular group. This entry applies when the traffic is known to be flowing down the shortest path tree, so the traffic is no longer needed via the RP tree.
		Source	The IP address of the multicast sender.
		RP Address	The IP address of the rendezvous point for the group.
		Expiry time	The time remaining until this entry is deleted, in seconds. The expiry time is 20 seconds longer than the (S,G) Keepalive time.
		Override time	The override timer in seconds. When the switch sees a prune message on the correct upstream interface, and it still needs to receive traffic via that rp tree, it sends a join message when this timer expires. A zero value indicates that the timer is not running.
		Inherited output interfaces	The interfaces that still require (S,G) data via the RP tree.
(*,*, RP)			The entry for handling multicast traffic to and from a network that is running a different multicast protocol. This entry applies when the switch is a PIM multicast border router (PMBR).
		RP Address	The IP address of the rendezvous point for the group.
		Next hop to RP	The address of the next routing device on the best unicast routing path to the RP.
		RPF Interface to RP	The interface on which packets from the RP would be received.
		Expiry time	The time remaining until this entry is deleted, in seconds.
		Join/Prune time	The join/prune timer in seconds. When the switch sees a prune message on the correct upstream interface, and it still needs to receive traffic via that rp tree, it sends a join message when this timer expires. A zero value indicates that the timer is not running.
		Immediate output interfaces	The interfaces with downstream routers that are interested in this (*,*,RP) entry.

Figure 27-32: Example output from the **show pim route** command for PIM Dense Mode

```

PIM4 Dense Mode Tree Information Base
-----
Source ..... 172.95.1.1
Group ..... 238.1.2.3
  RPF Neighbour to Src ..... Directly connected
  RPF Interface to Src ..... vlan1
  Source Alive time ..... 200
  Expiry time ..... 220
  Prune override time ..... 0
  Prune limit time ..... 0
  Immediate output interfaces .. vlan2

Source ..... 172.96.2.1
Group ..... 238.1.2.3
  RPF Neighbour to Src ..... 192.168.57.1
  RPF Interface to Src ..... vlan2
  Keep Alive time ..... 200
  Expiry time ..... 220
  Prune override time ..... 0
  Prune limit time ..... 50
  Immediate output interfaces .. vlan2

```

Table 27-22: Parameters in output of the **show pim route** command for PIM Dense Mode

Parameter	Meaning
Source	The IP address of the multicast sender.
Group	The IP address of the multicast group.
RPF Neighbour to Src	The address of the PIM neighbour to the source, taking into account any PIM assert messages. Packets from the source would be received from this neighbour. "Directly connected" indicates that the source is directly connected to the switch.
RPF Interface to Src	The interface on which the switch expects to receive traffic from the source.
Keep Alive time	The Keepalive timer in seconds. A zero value indicates that the timer is not running because no data is being received. When the switch is forwarding multicast data in hardware, the data stream is checked every 60 seconds to verify data is still flowing. If it is, the timer is reset.
Source Alive time	An alive timer in seconds that is the equivalent of the Keepalive timer but applies to directly connected sources. A zero value indicates that the timer is not running because no data is being received. When the switch is forwarding multicast data in hardware, the data stream is checked every 60 seconds to verify data is still flowing. If it is, the timer is reset.
Expiry time	The time remaining until this entry is deleted, in seconds. The expiry time is 20 seconds longer than the (S,G) Keepalive or Sourcealive time.
Prune override time	The prune override timer, in seconds. When the switch sees a prune message on the correct upstream interface, and it still needs to receive traffic, it sends a join message when this timer expires. A zero value indicates that the timer is not running.

Table 27-22: Parameters in output of the **show pim route** command for PIM Dense Mode (cont.)

Parameter	Meaning
Prune limit time	The prune limit, in seconds. A zero value indicates that the timer is not running. The switch cannot send a data-triggered prune until this timer expires.
Immediate output interfaces	The interfaces with routers or IGMP hosts that are interested in this (S,G) data.

Examples To display information about the internal PIM routing table, use the command:

```
sh pim rou
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show ip route multicast](#)
- [show pim](#)
- [show pim bsrcandidate](#)
- [show pim counters](#)
- [show pim debug](#)
- [show pim interface](#)
- [show pim neighbour](#)
- [show pim rpcandidate](#)
- [show pim rpset](#)
- [show pim timer](#)

show pim rpcandidate

Syntax SHow PIM RPCandidate

Description This command displays information about multicast groups for which the switch is a PIM-SM rendezvous point candidate ([Figure 27-33](#), [Table 27-23](#)).

Figure 27-33: Example output from the **show pim rpcandidate** command

```
PIM4 RP Candidate
-----
Priority ..... 192
Interface .....vlan1
  Group address/Mask ..... 224.1.1.1 / 255.255.255.255
  Group address/Mask ..... 224.2.2.0 / 255.255.255.0
```

Table 27-23: Parameters in output of the **show pim rpcandidate** command

Parameter	Meaning
Priority	Priority for the switch to become the rendezvous point for any multicast groups.
Group Address	Multicast groups associated with the specified rendezvous point.
Mask	Mask for the address.
Interface	Interface the switch advertises itself as when advertising as a rendezvous point for multicast groups.

Examples To display information about multicast groups for which the switch is a rendezvous point candidate, use the command:

```
sh pim rpc
```

Related Commands

- [add pim rpcandidate](#)
- [delete pim rpcandidate](#)
- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [set pim rpcandidate](#)
- [show ip](#)
- [show pim](#)

show pim rpset

Syntax SHow PIM RPSet

Description This command displays the static group-to-RP mapping (Figure 27-34, Table 27-24), followed by the elected bootstrap router's current set of RP candidates and the groups they are configured for (Figure 27-35, Table 27-25). It applies to PIM-SM only.

Figure 27-34: Example output from the **show pim rpset** command when the RP is statically configured

```
PIM4 Static RP Mapping
-----
RP Address ..... 192.168.2.1
  Group address/Mask ..... 239.1.0.0 /
255.255.0.0
```

Table 27-24: Parameters in output of the **show pim rpset** command when the RP is statically configured

Parameter	Meaning
RP address	IP address of the router that is statically configured as the RP for the following group(s).
Group address	IP address of the multicast group.
Mask	Mask for the multicast group address.

Figure 27-35: Example output from the **show pim rpset** command when the RP is determined using the bootstrap mechanism

```
PIM4 RP Set Information
-----
Group address/Mask ..... 224.1.1.1 / 255.255.255.255
  RP Candidate address .. 192.168.1.1
    Priority ..... 192
    Holdtime ..... 120
  RP Candidate address .. 192.168.2.1
    Priority ..... 180
    Holdtime ..... 120
```

Table 27-25: Parameters in output of the **show pim rpset** command when the RP is determined using the bootstrap mechanism

Parameter	Meaning
Group address	IP address of the multicast group.
Mask	Mask for the multicast group address.
RP Candidate address	IP addresses of each RP candidate for the multicast group and mask pair.
Priority	Priority for the RP candidate to become the RP. A candidate with a lower priority is more likely to become the RP.

Table 27-25: Parameters in output of the **show pim rpset** command when the RP is determined using the bootstrap mechanism

Parameter	Meaning
Holdtime	The time in seconds for which this RP candidate is valid. Unless the RP advertisement is refreshed, the RP candidate is deleted when this time has elapsed.

Examples To display information about multicast group and mask pairs, use the command:

```
sh pim rps
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show pim](#)

show pim staterefresh

Syntax SHow PIM STATerefresh

Description This command displays the internal State Refresh table for PIM-DM (Figure 27-36, Table 27-26).

Figure 27-36: Example output from the **show pim staterefresh** command

```
PIM4 Dense Mode State Refresh
-----
Source ..... 172.95.2.1
Group ..... 238.1.2.3
  Originator state ..... Originator
    Direct Connect to source on .... vlan2
    Source alive timer ..... 200
    State refresh timer ..... 50

Source ..... 172.96.2.1
Group ..... 238.1.2.3
  Originator state ..... Not Originator
```

Table 27-26: Parameters in output of the **show pim staterefresh** command

Parameter	Meaning
Source	IP address of the multicast sender.
Group	IP address of the multicast group.
Originator state	Whether the switch can act as a state refresh message originator. A switch acts as an originator when the source is directly connected.
Direct Connect to source on	Interface to which the source is connected.
Source alive timer	An alive timer in seconds for directly connected sources. A zero value indicates that the timer is not running because no data is being received. When the switch is forwarding multicast data in hardware, the data stream is checked every 60 seconds to verify data is still flowing. If it is, the timer is reset.
State refresh timer	Time in seconds before the next state refresh message is sent.

Examples To display the internal State Refresh table, use the command:

```
sh pim stat
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show pim](#)

show pim timer

Syntax SHow PIM TIMer

Description This command displays information about timer intervals for PIM operations (Figure 27-37, Table 27-27).

Figure 27-37: Example output from the **show pim timer** command

```
PIM Timers
-----
Join/Prune interval ..... 60
Register probe time ..... 5
Register suppression time ..... 60
Keep Alive time ..... 210
BSM interval ..... 60
RP adv interval ..... 60
Prune hold time ..... 210
Source Alive time ..... 210
State refresh interval ..... 60
```

Table 27-27: Parameters in output of the **show pim timer** command

Parameter	Meaning
Join/Prune Interval	Time interval in seconds at which the switch sends join/prune messages.
Register Probe time	Time interval in seconds that the DR waits for another register stop message after sending a null register message to the RP.
Register Suppression time	Time interval in seconds at which the sender's DR sends null register messages to the group's RP.
Keep Alive time	Length in seconds that the join state for a particular source and group pair is maintained in the absence of data for that pair.
BSM interval	Length in seconds that the switch sends bootstrap messages when it is the bootstrap router in the domain.
RP adv interval	Length in seconds that the switch sends C-RP-advertisements.
Prune hold time	Length in seconds that upstream routers maintain the prune state.
Source Alive time	Length in seconds that a switch acting as a state refresh originator is active in the absence of data packets from the source.
State refresh interval	Length in seconds that a switch sends state refresh messages.

Examples To display information about timer intervals for PIM operations, use the command:

```
sh pim tim
```

Related Commands

- [disable pim](#)
- [enable pim](#)
- [set pim](#)
- [show ip](#)
- [show pim](#)

