Chapter 5
Hardware-Based IP Access Control Lists (ACLs)

This chapter describes hardware-based ACLs, which are ACLs that process traffic in hardware. (This type of ACL is also called rule-based ACLs). ACL Hardware first packet lookup mode programs the ACL into hardware at startup (or as a new ACL is entered).

Table 5.1 lists the products and software releases that support hardware-based ACLs.

<table>
<thead>
<tr>
<th>Product</th>
<th>Software Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devices with 10-Gigabit Ethernet modules</td>
<td>07.6.01 and laterA</td>
</tr>
<tr>
<td>Devices with EP modules</td>
<td>07.6.01 and laterA</td>
</tr>
</tbody>
</table>

a. Starting with release 07.6.01, you can configure an interface to use flow-based or hardware-based ACL mode.

NOTE: For information about flow-based ACLs, see “Software-Based IP Access Control Lists (ACLs)” on page 4-1.

Hardware-based ACLs program the ACL entries you assign to an interface into Content Addressable Memory (CAM) space allocated for the port(s). Devices that use hardware-based ACLs program the ACLs into the CAM entries and use these entries to permit or deny packets in the hardware, without sending the packets to the CPU for processing.

Hardware-based ACLs are supported on physical interfaces, trunk groups, and virtual routing interfaces.

Chapter Overview

The following sections describe hardware-based IP ACLs. For configuration information, see the following sections:

• “Comparison of Flow-Based ACLs and Hardware-Based ACLs” on page 5-2
• “Configuration Considerations” on page 5-4
• “Disabling or Re-Enabling Hardware-Based ACLs” on page 5-5
• “Reapplying ACLs to Interfaces” on page 5-7
• “Specifying the Maximum Number of CAM Entries for ACLs” on page 5-7
• “Enabling ACL Filtering of Fragmented Packets” on page 5-8
• “Configuring and Applying an ACL” on page 5-9
• “ACL Filtering for Traffic Switched Within a Virtual Routing Interface” on page 5-12
• “Displaying ACL Information” on page 5-12
• “Troubleshooting Hardware-Based ACLs” on page 5-12
• “Hardware-Based Policy-Based Routing (PBR)” on page 5-13

Comparison of Flow-Based ACLs and Hardware-Based ACLs

The following sections describe how both ACL modes work. Starting in release 07.6.04, both modes are supported on the EP devices, and 10 Gigabit Ethernet modules, although only one mode at a time can be active on an interface. Hardware-based ACLs are enabled by default on supported devices.

How Flow-Based ACLs Work

Flow-based ACLs work as follows:

When the device receives an IP packet, the device checks the receiving port’s ACL CAM entries for an entry with the same address information as the packet.

• If the CAM contains a matching entry, the device takes the action specified by the entry (permit or deny).

**NOTE:** CAM entries are not programmed when you apply an ACL to an interface. CAM entries are created by the CPU when a packet received by the device matches a CAM entry on the inbound interface, as described below. The Layer 4 CAM entries programmed by the CPU for ACL matches age out if unused for 70 seconds.

**NOTE:** The CAM can contain entries for ACLs with deny actions only if you enable this support by entering the `hw-drop-acl-denied-packet` command.

• If the CAM does not contain a matching entry, the device sends the packet to the CPU for ACL comparison.

  • If the packet matches an ACL applied to inbound traffic on the port and the ACL has the permit action, the CPU programs an ACL permit entry into the Layer 4 CAM for the port that received the packet. The CAM entry contains the packet's address information.

  • If the packet matches an ACL applied to inbound traffic on the port and the ACL has the deny action, the CPU drops the packet but does not program an entry into the Layer 4 CAM, unless you have enabled the CPU to do so by entering the `hw-drop-acl-denied-packet` command.

  • If the packet does not match any of the inbound ACLs on the interface (and therefore matches an implicit **deny ip any any**), the CPU drops the packet. The CPU does not program an entry into the Layer 4 CAM, unless you have enabled the CPU to do so by entering the `hw-drop-acl-denied-packet` command.

    • If the packet's outbound interface has an ACL applied to the outbound traffic direction, the device sends the packet to the CPU for filtering and either drops the packet or forwards the packet on the outbound interface, depending on the results of the ACL comparison.

When the hardware-based ACL mode is in effect, a packet is nevertheless sent to the CPU for processing under certain circumstances. See “How Hardware-Based ACLs Work” on page 5-2.

How Hardware-Based ACLs Work

When you apply an ACL to inbound traffic on an interface, the device programs the Layer 4 CAM with the ACL. Entries for permit and deny ACLs are programmed. Most ACLs require one Layer 4 CAM entry. However, ACLs that match on more than one TCP or UDP application port require a separate CAM entry for each application port.
The Layer 4 CAM entries for ACLs do not age out. They remain in the CAM until you remove the ACL or the ACL mode is changed to flow-based.

- If a packet received on the interface matches an ACL entry in the Layer 4 CAM, the device permits or denies the packet according to the ACL.
- If a packet does not match an ACL, the packet is dropped, since the default action on an interface that has ACLs is to deny the packet.

If an interface has an outbound ACL, the device uses the flow-based ACL mode for all ACLs on that interface.

When the hardware-based ACL mode is in effect, a packet is nevertheless sent to the CPU for processing under the following circumstances:

- The packet does not have any Layer 2 or Layer 3 forwarding information.
- The ACL entry matches on the log option.
- The outbound interface has an outbound ACL. In this case, the device changes the ACL mode on the interface to flow-based ACLs.
- ACL accounting is enabled. In this case, the device changes the ACL mode on all interfaces to flow-based ACLs. ACL accounting is disabled by default on EP devices and Standard devices. The `enable-acl-counter` command at the global CONFIG level enables ACL accounting.

**How Fragmented Packets are Processed**

The descriptions above apply to non-fragmented packets. In 07.6.04 and later, the default processing of fragments by hardware-based ACLs (as well as flow-based ACLs) is as follows:

- The first fragment of a packet is permitted or denied using the ACLs. The first fragment is handled the same way as non-fragmented packets, since the first fragment contains the Layer 4 source and destination application port numbers. The device uses the Layer 4 CAM entry if one is programmed, or applies the interface's ACL entries to the packet and permits or denies the packet according to the first matching ACL.

- For other fragments of the same packet, one of the following occurs:
  - If the device has a CAM entry for the packet (or for previous packets in the same flow), and has not been configured to send the fragments to the CPU, the device uses the CAM entry to forward the fragments in hardware.
    
    The fragments are forwarded even if the first fragment, which contains the Layer 4 information, was denied. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet. However, for stricter fragment control, you can send fragments to the CPU for filtering.
  - If the device is configured to send fragments to the CPU for filtering, the device compares the source and destination IP addresses to the ACL entries that contain Layer 4 information.
    - If the fragment's source and destination addresses exactly match an ACL entry that has Layer 4 information, the device assumes that the ACL entry is applicable to the fragment and permits or denies the fragment according to the ACL entry. The device does not compare the fragment to ACL entries that do not contain Layer 4 information.
    - If both the fragment's source and destination addresses do not exactly match an ACL entry, the device skips the ACL entry and compares the packet to the next ACL entry. This is true even if either the source or destination address (but not both) does exactly match an ACL entry.
    - If the source and destination addresses do not exactly match any ACL entry on the applicable interface, the device drops the fragment.
NOTE: By default, 10 Gigabit Ethernet modules also forward the first fragment instead of using the ACLs to permit or deny the fragment.

You can modify the handling of denied fragments by flow-based ACLs or hardware-based ACLs. In addition, you can throttle the fragment rate on an interface that used hardware-based ACLs. See “Dropping All Fragments That Exactly Match an ACL” on page 4-28 and “Enabling ACL Filtering of Fragmented Packets” on page 5-8.

**Hardware Aging of Layer 4 CAM Entries**

Flow-based ACLs and hardware-based ACLs both use Layer 4 CAM entries. The device permanently programs hardware-based ACLs into the CAM. The entries never age out. In software release 07.6.04 and later, the device does age out Layer 4 CAM entries for flow-based ACLs. A Layer 4 CAM entry for a flow-based ACL ages out if the entry is unused for 70 seconds. The age time is not configurable.

After an entry ages out, its CAM space becomes available for other ACL entries or other features that use the Layer 4 CAM.

**Configuration Considerations**

- Hardware-based ACLs are supported on all EP Ethernet ports and on 10 Gigabit Ethernet ports.
- Hardware-based ACLs are supported on physical interfaces, trunk groups, and virtual routing interfaces.
- Hardware-based ACLs are supported only for inbound traffic.
- Hardware-based ACLs support only one ACL per port. The ACL of course can contain multiple entries (rules). For example, hardware-based ACLs do not support ACLs 101 and 102 on port 1, but hardware-based ACLs do support ACL 101 containing multiple entries.
- By default, the first fragment of a fragmented packet received by an EP device is permitted or denied using the ACLs, but subsequent fragments of the same packet are forwarded in hardware. Generally, denying the first fragment of a packet is sufficient, since a transaction cannot be completed without the entire packet. However, for stricter fragment control, you can send fragments to the CPU for filtering. See “Enabling ACL Filtering of Fragmented Packets” on page 5-8.

NOTE: By default, 10 Gigabit Ethernet modules also forward the first fragment instead of using the ACLs to permit or deny the fragment.

- If you change the content of an ACL (add, change, or delete entries), you must remove and then reapply the ACL to all the ports that use it. Otherwise, the older version of the ACL remains in the CAM and continues to be used. You can easily re-apply ACLs using the `ip rebind-ACL <num> | <name> | all` command. See “Reapplying ACLs to Interfaces” on page 5-7.

NOTE: HP recommends that you also remove and reapply a changed ACL even when you are using the flow-based mode.

- If you or the software changes the ACL mode (between hardware-based and flow-based), you must reapply the ACLs to the ports. Use the `ip rebind-ACL <num> | <name> | all` command.
- ACL statistics are not supported with hardware-based rate limiting. This feature relies on ACL information provided by the CPU, and thus requires flow-based ACLs. If you enable ACL statistics (by entering the `enable-acl-counter` command), the device automatically changes the ACL mode on all ports to flow-based ACLs.
- If you use the `log` option, which generates a Syslog message when a packet matches an ACL and is thus permitted or denied (based on the ACL action), the software changes the ACL mode to flow-based for the traffic flows that match the ACL. Changing the mode to flow-based enables the device to send the matching flows to the CPU for processing. This is required because the CPU is needed to generate the Syslog message.
NOTE: Depending on how many entries have the log option and how often packets match those entries, ACL performance can be affected. Use the log option only when needed.

NOTE: You can globally disable ACL logging without the need to remove the log option from existing ACLs. This enables you to use the ACLs in the hardware-based ACL mode. You also can configure the device to copy traffic that is denied by a hardware-based ACL to an interface. This option allows you to monitor the denied traffic without sending the traffic to the CPU. See “Globally Disabling ACL Logging” on page 5-6.

- If you use the <icmp-type> parameter with an extended ACL, the device uses the CPU to filter packets using the ACL. The CPU is required to filter the ICMP message type.
- For a tagged port that is a member of multiple virtual routing interfaces, you must use the same ACL on all the port's virtual routing interfaces. Alternatively, if you need to use different ACLs, you can use flow-based ACLs instead on all the port's virtual routing interfaces.
- The software automatically disables hardware-based ACLs and enables flow-based ACLs on an interface if one of the following occurs:
  - If you apply an ACL to the outbound traffic direction on the interface.
  - If there is not enough CAM space to hold all the ACL entries in the ACL applied to the port. All ACL entries for a port must fit in the CAM space allocated by the device for the port's ACLs. If all the ACL's entries do not fit into the port's CAM, the device disables hardware-based ACLs on the port and enables flow-based ACLs instead. The device also generates a Syslog message to inform you of the change. See “Syslog Message for Changed ACL Mode” on page 5-6.
    If this occurs, remove the ACL from the port, then either reduce the number of entries in the ACL to fit into the CAM space or adjust the CAM allocations on the device’s ports, to hold the ACL.
  - If you enable any of the following features on the interface. To change back to the hardware-based ACL mode in this case, you must disable the feature that caused the port to change to the flow-based ACL mode.
    - Network Address Translation (NAT)
    - Protection against ICMP or TCP Denial-of-Service (DoS) Attacks
    - ACL-based rate limiting
  - You can use PBR and hardware-based ACLs on the same port. However, HP recommends that you use exactly the same ACL for each feature. Otherwise, it is possible for the ACL's Layer 4 CAM entry to be programmed incorrectly and give unexpected results.

**Disabling or Re-Enabling Hardware-Based ACLs**

By default, the device enables hardware-based ACLs on all ports. The device also can disable hardware-based ACLs based on the conditions described in “Configuration Considerations” on page 5-4. You also can manually change the ACL mode on an interface.

Disabling hardware-based ACLs on some of the ports is useful if some ports have large ACLs (ACLs with many entries) while other ports have few ACL entries. By disabling hardware-based ACLs on the ports that have few or no ACL entries, you can ensure that the ports that do have ACL entries will have enough CAM space for the ACL entries.

You also might want to disable hardware-based ACLs if the ACL entries on a port are used infrequently. In this case, you can conserve CAM entries for other features or other ports with minimal performance impact, since the ACL activity is low.
NOTE: You can determine the ports that have high ACL usage by disabling hardware-based ACLs on all the ports, allowing the device to operate using flow-based ACLs, then displaying ACL accounting information. To enable ACL accounting, enter the `enable-acl-counter` command at the global CONFIG level. To display the ACL accounting information, enter the `show access-list all` command.

To disable hardware-based ACLs on an interface, enter the following command at the configuration level for the port:

```
ProCurveRS(config-if-1/1)# ip access-group flow-mode
```

**Syntax:** `[no] ip access-group flow-mode`

To re-enable hardware-based ACLs on the port, enter the following command:

```
ProCurveRS(config-if-1/1)# no ip access-group flow-mode
```

Syslog Message for Changed ACL Mode

If the device changes the ACL mode from hardware-based to flow-based, the device generates one of the following Syslog messages.

<table>
<thead>
<tr>
<th>Message Level</th>
<th>Message</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification</td>
<td>ACL insufficient L4 cam resource, using flow-based ACL instead</td>
<td>The port does not have a large enough CAM partition for the ACLs. To re-partition the CAM, see the &quot;Changing CAM Partitions&quot; chapter in the Diagnostic Guide for ProCurve 9300/9400 Series Routing Switches.</td>
</tr>
</tbody>
</table>

Globally Disabling ACL Logging

Hardware-based ACLs do not support the `log` option. Even when hardware-based ACLs are enabled, if an ACL entry has the `log` option, traffic that matches that ACL is sent to the CPU for processing.

If your configuration already contains ACLs that you want to use with hardware-based ACLs, but some of the ACLs contain the `log` option, you can globally disable ACL logging without the need to remove the `log` option from each ACL entry. When you globally disable ACL logging, the ACL entries remain unchanged but the `log` option is ignored and the ACL can use the hardware-based ACL mode.

To globally disable ACL logging, enter the following command at the global CONFIG level of the CLI:

```
ProCurveRS(config)# ip access-list disable-log-to-cpu
```

**Syntax:** `[no] ip access-list disable-log-to-cpu`

To re-enable ACL logging, enter the following command:

```
ProCurveRS(config)# no ip access-list disable-log-to-cpu
```

Copying Denied Traffic to a Mirror Port for Monitoring

Although hardware-based ACLs do not support ACL logging, you nonetheless can monitor the traffic denied by hardware-based ACLs. To do so, attach a protocol analyzer to a port and enable the device to redirect traffic denied by ACLs to that port.

To redirect traffic denied by ACLs, enter the following command at the interface configuration level:

```
ProCurveRS(config-if-1/1)# ip access-group redirect-deny-to-interf
```

**Syntax:** `[no] ip access-group redirect-deny-to-interf`
Enter the command on the port to which you want the denied traffic to be copied.

**NOTE:** The software requires that an ACL has already been applied to the interface.

When you enable redirection, the deny action of the ACL entry is still honored. Traffic that matches the ACL is not forwarded.

### Reapplying ACLs to Interfaces

For hardware-based ACLs (and flow-based ACLs), if you make an ACL configuration change, you must reapply the ACLs to their interfaces to place the change into effect. An ACL configuration change includes any of the following:

- Adding, changing, or removing an ACL or an entry in an ACL
- Changing a PBR policy
- Changing the port membership of a VLAN that has an ACL on its virtual routing interface
- Enabling or disabling the TCP strict mode or UDP strict mode (flow-based ACLs only)
- Changing EP ToS-based QoS mappings (since EP QoS uses the Layer 4 CAM)

To reapply all ACLs to their interfaces, enter the following command at the global CONFIG level of the CLI:

```plaintext
ProCurveRS(config)# ip rebind-acl all
```

**Syntax:** [no] ip rebind-acl <num> | <name> | all

This command reappplies all ACLs to their interfaces.

To reapply a specific ACL, enter a command such as the following:

```plaintext
ProCurveRS(config)# ip rebind-acl 101
```

This command reappplies ACL 101 only.

### Specifying the Maximum Number of CAM Entries for ACLs

You can adjust the allocation of Layer 4 CAM space for use by ACLs, on an IPC or IGC basis and on 10 Gigabit Ethernet modules. The new allocation applies to all the ports managed by the IPC or IGC or 10 Gigabit Ethernet module.

Most ACLs require one CAM entry for each ACL entry (rule). The exception is an ACL entry that matches on more than one TCP or UDP application port. In this case, the ACL entry requires a separate Layer 4 CAM entry for each application port on which the ACL entry matches.

Make sure you specify a maximum that is equal to or greater than the largest number of entries required by an ACL applied to any of the ports managed by the same IPC or IGC.

Hardware-based ACLs use CAM partitions 1 and 2. The default number of entries that are allocated in each pool differs depending on the device. For more information about CAM partitions, see the “Changing CAM Partitions” chapter in the *Diagnostic Guide for ProCurve 9300/9400 Series Routing Switches*.

To specify the maximum number of CAM entries the device can allocate for hardware-based ACLs, enter commands such as the following:

```plaintext
ProCurveRS(config)# interface ethernet 1/1
ProCurveRS(config-if-1/1)# ip access-group max-l4-cam 50
```

This command allows up to 50 ACL entries on each port managed by the IPC or IGC that manages port 1/1.

**Syntax:** [no] ip access-group max-l4-cam <num>

The `<num>` parameter specifies the number of CAM entries and can be from 10 – 2048. The default depends on the device.
The command is valid at the interface configuration level. However, the device applies the change to all ports managed by the same IPC or IGC. Regardless of the port number, when you save the change to the startup-config file, the CLI applies the command to the first port managed by the IPC or IGC.

**NOTE:** If you enter the command on more than one port managed by the same IPC or IGC, the CLI uses the value entered with the most-recent command for all the ports on the ICP or IGC.

### Enabling ACL Filtering of Fragmented Packets

By default, when a hardware-based ACL is applied to a port, the port will use the ACL to permit or deny the first fragment of a fragmented packet, but forward subsequent fragments of the same packet in hardware. Since a transaction cannot be completed without the entire packet, filtering the first fragment is generally sufficient for permitting or denying the entire packet.

**NOTE:** The fragmentation support described in this section applies only to EP devices and only to hardware-based ACLs.

**NOTE:** Enhanced fragment handling is not supported on 10 Gigabit Ethernet modules. By default, 10 Gigabit Ethernet modules also forward the first fragment instead of using the ACLs to permit or deny the fragment.

For tighter control, you can enable CPU filtering of all packet fragments on a port. When you enable CPU filtering, the port sends all the fragments of a fragmented packet to the CPU. The CPU then permits or denies each fragment according to the ACL applied to the port. You can enable CPU filtering of fragments on individual ports.

You also can configure the port to drop all packet fragments.

To enable CPU filtering of packet fragments on an individual port, enter commands such as the following:

```bash
ProCurveRS(config)# interface ethernet 1/1
ProCurveRS(config-if-1/1)# ip access-group frag inspect
```

**Syntax:** `[no] ip access-group frag inspect | deny

The `inspect | deny` parameter specifies whether you want fragments to be sent to the CPU or dropped:

- **inspect** – This option sends all fragments to the CPU.
- **deny** – This option begins dropping all fragments received by the port as soon as you enter the command. This option is especially useful if the port is receiving an unusually high rate of fragments, which can indicate a hacker attack.

**NOTE:** Standard devices also support the `ip access-group frag deny` command but the command performs a different service on Standard devices. See "Dropping All Fragments That Exactly Match an ACL".

### Throttling the Fragment Rate

By default, when you enable CPU filtering of packet fragments on an EP device, all fragments are sent to the CPU. Normally, the fragment rate in a typical network does not place enough additional load on the CPU to adversely affect performance. However, performance can be affected if the device receives a very high rate of fragments. For example, a misconfigured server or a hacker can affect the device’s performance by flooding the CPU with fragments.

You can protect against fragment flooding by specifying the maximum number of fragments the device or an individual interface is allowed to send to the CPU in a one-second interval. If the device or an interface receives more than the specified number of fragments in a one-second interval, the device either drops or forwards subsequent fragments in hardware, depending on the action you specify. In addition, the device starts a holdown timer and continues to either drop or forward fragments until the holdown time expires.

The device also generates a Syslog message.
To specify the maximum fragment rate per second, enter commands such as the following:

```
ProCurveRS(config)# ip access-list frag-rate-on-system 15000 exceed-action drop
ProCurveRS(config)# ip access-list frag-rate-on-interface 5000 exceed-action
forward reset-interval 5
```

The first command sets the fragment threshold at 15,000 per second, for the entire device. If the device receives more than 15,000 packet fragments in a one-second interval, the device takes the specified action. The action specified with this command is to drop the excess fragments and continue dropping fragments for a holddown time of ten minutes. After the ten minutes have passed, the device starts sending fragments to the CPU again for processing.

The second command sets the fragment threshold at 5,000 for individual interfaces. If any interface on the device receives more than 5,000 fragments in a one-second interval, the device takes the specified action. In this case, the action is to forward the fragments in hardware without filtering them. The device continues forwarding fragments in hardware for five minutes before beginning to send fragments to the CPU again.

Both thresholds apply to the entire device. Thus, if an individual interface’s fragment threshold is exceeded, the drop or forward action and the holddown time apply to all fragments received by the device.

**Syntax:** [no] ip access-list frag-rate-on-system <num> exceed-action drop | forward reset-interval <mins>

and

**Syntax:** [no] ip access-list frag-rate-on-interface <num> exceed-action drop | forward reset-interval <mins>

The `<num>` parameter specifies the maximum number of fragments the device or an individual interface can receive and send to the CPU in a one-second interval.

- **frag-rate-on-system** – Sets the threshold for the entire device. The device can send to the CPU only the number of fragments you specify per second, regardless of which interfaces the fragments come in on. If the threshold is exceeded, the device takes the exceed action you specify.

- **frag-rate-on-interface** – Sets the threshold for individual interfaces. If an individual interface receives more than the specified maximum number of fragments, the device takes the exceed action you specify.

The `<num>` parameter specifies the maximum number of fragments per second.

- For **frag-rate-on-system**, you can specify from 600 – 12800. The default is 6400.
- For **frag-rate-on-interface**, you can specify from 300 – 8000. The default is 4000.

The **drop | forward** parameter specifies the action to take if the threshold `<num>` parameter is exceeded:

- **drop** – fragments are dropped without filtering by the ACLs
- **forward** – fragments are forwarded in hardware without filtering by the ACLs

The `<mins>` parameter specifies the number of minutes the device will enforce the drop or forward action after a threshold has been exceeded. You can specify from 1 – 30 minutes, for **frag-rate-on-system** or **frag-rate-on-interface**.

**Syslog Messages for Exceeded Fragment Thresholds**

If a fragment threshold is exceeded, the device generates one of the following Syslog messages.

---

**Configuring and Applying an ACL**

The commands for configuring and applying ACLs are the same for the hardware-based and flow-based ACL modes. Use the following commands. For more information about the syntax, see the following:

- “Software-Based IP Access Control Lists (ACLs)” on page 4-1
- “Standard ACL Syntax” on page 4-10
• “Extended ACL Syntax” on page 4-18
• “QoS Options for IP ACLs” on page 5-10

QoS Options for IP ACLs

NOTE: QoS options for IP ACLs are supported in software releases 07.6.0

QoS options enable you to perform QoS for packets that match the ACLs. Using an ACL to perform QoS is an alternative to the following methods:

• Directly setting the internal forwarding priority based on incoming port, VLAN membership, and so on. (This method is described in “Assigning QoS Priorities to Traffic” on page 2-10.)
• Enabling the IP ToS-based QoS feature described in .

NOTE: If you use an ACL on an interface, ToS-based QoS assumes that the ACLs will perform QoS for all packets except the packets that match the permit ip any any ACL.

The following QoS ACL options are supported:

• priority – Assigns traffic that matches the ACL to a hardware forwarding queue. In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority.
• priority-force – Assigns packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Specify one of the following QoS queues:
  • 0 – qosp0
  • 1 – qosp1
  • 2 – qosp2
  • 3 – qosp3
• priority-mapping – Matches on the packet's 802.1p value. This option does not change the packet's forwarding priority through the device or mark the packet.

NOTE: This option is not supported on 10 Gigabit Ethernet modules.

• dscp-mapping – Matches on the packet's DSCP value. This option does not change the packet's forwarding priority through the device or mark the packet.
• dscp-marking – Marks the DSCP value in the outgoing packet with the value you specify.

Using an ACL to Change the Forwarding Queue

The priority option enables you to assign traffic that matches the ACL to a specific hardware forwarding queue (qosp0, qosp1, qosp2, or qosp3).

In addition to changing the internal forwarding priority, if the outgoing interface is an 802.1q interface, this option maps the specified priority to its equivalent 802.1p (CoS) priority and marks the packet with the new 802.1p priority. Table 5.4 lists the default mappings of hardware forwarding queues to 802.1p priorities.

<table>
<thead>
<tr>
<th>Forwarding Queue</th>
<th>qosp0</th>
<th>qosp0</th>
<th>qosp1</th>
<th>qosp1</th>
<th>qosp2</th>
<th>qosp2</th>
<th>qosp3</th>
<th>qosp3</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.1p</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
Here is an example of how to use the **priority** option.

ProCurveRS(config)# access-list 110 permit ip any any priority 2
ProCurveRS(config)# interface 1/1
ProCurveRS(config-if-1/1)# ip access-group 110 out

These commands configure an extended ACL that places all IP traffic that is queued to be sent on port 1/1 into hardware forwarding queue qos2 on that port. In addition, if port 1/1 is tagged, the ACL also marks the packets’ 802.1p value.

The **priority 0 | 1 | 2 | 3** parameter specifies the QoS queue:

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3

**NOTE:** This **priority** option provides the same function as the Layer 4 IP access policies supported on 9300 series Chassis devices. If you configure both a Layer 4 IP access policy and an extended ACL to set the hardware forwarding priority for the same traffic, the device uses the ACL instead of the IP access policy.

The **priority-force** parameter allows you assign packets of outgoing traffic that match the ACL to a specific hardware forwarding queue, even though the incoming packet may be assigned to another queue. Select one of the following QoS queue:

- 0 – qosp0
- 1 – qosp1
- 2 – qosp2
- 3 – qosp3

**Matching on a Packet’s 802.1p Value**

The **priority-mapping** option matches on the packet’s 802.1p value. This option does not change the packet’s forwarding priority through the device or mark the packet.

**NOTE:** This option is not supported on 10 Gigabit Ethernet modules.

To configure an ACL that matches on a packet’s 802.1p priority, enter a command such as the following:

ProCurveRS(config)# access-list 111 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255 priority-mapping 0

**Syntax:** ...**priority-mapping** <8021p-value>

**NOTE:** For complete syntax information, see “Extended ACL Syntax” on page 4-18.

**Matching on a Packet’s DSCP Value**

The **dscp-mapping** option matches on the packet’s DSCP value. This option does not change the packet’s forwarding priority through the device or mark the packet.

To configure an ACL that matches on a packet with DSCP value 29, enter a command such as the following:

ProCurveRS(config)# access-list 112 permit ip 1.1.1.0 0.0.0.255 2.2.2.x 0.0.0.255 dscp-mapping 29

**Syntax:** ...**dscp-mapping** <dscp-value>

**NOTE:** For complete syntax information, see “Extended ACL Syntax” on page 4-18.
Using an IP ACL to Mark ToS Values

The **dscp-marking** option for extended ACLs allows you to configure an ACL that marks matching packets with a specified ToS value.

For example, the following commands configure an ACL that marks all IP packets with DSCP value 5. The ACL is then applied to incoming packets on interface 1/1. Consequently, all inbound packets on interface 1/1 are marked with the specified DSCP value.

```plaintext
ProCurveRS(config)# access-list 120 permit ip any any dscp-marking 5
ProCurveRS(config)# interface 1/1
ProCurveRS(config-if-1/1)# ip access-group 120 in
```

**Syntax**: ...**dscp-marking** <dscp-value> 802.1p-priority-marking <0 – 7> internal-priority-marking <0 – 7>

The **dscp-marking** <dscp-value> parameter maps a DSCP value to an internal forwarding priority. The DSCP value can be from 0 – 63.

ACL Filtering for Traffic Switched Within a Virtual Routing Interface

**NOTE**: This section applies to flow-based ACLs and hardware-based ACLs.

By default, the device does not filter traffic that is switched from one port to another within the same virtual routing interface, even if an ACL is applied to the interface. You can enable the device to filter switched traffic within a virtual routing interface. When you enable the filtering, the device uses the ACLs applied to inbound traffic to filter traffic received by a port from another port in the same virtual routing interface. This feature does not apply to ACLs applied to outbound traffic.

To enable filtering of traffic switched within a virtual routing interface, enter the following command at the configuration level for the interface:

```plaintext
ProCurveRS(config-vif-1)# ip access-group ve-traffic
```

**Syntax**: [no] ip access-group ve-traffic

Displaying ACL Information

To display the number of Layer 4 CAM entries used by each ACL, enter the following command:

```plaintext
ProCurveRS(config)# show access-list all
```

Extended IP access list 100 (Total flows: N/A, Total packets: N/A, Total rule cam use: 3)
permit udp host 192.168.2.169 any (Flows: N/A, Packets: N/A, Rule cam use: 1)
permit icmp any any (Flows: N/A, Packets: N/A, Rule cam use: 1)
deny ip any any (Flows: N/A, Packets: N/A, Rule cam use: 1)

**Syntax**: show access-list <acl-num> | <acl-name> | all

The Rule cam use field lists the number of CAM entries used by the ACL or entry. The number of CAM entries listed for the ACL itself is the total of the CAM entries used by the ACL's entries.

For flow-based ACLs, the Total flows and Flows fields list the number of Layer 4 session table flows in use for the ACL.

The Total packets and Packets fields apply only to flow-based ACLs.

Troubleshooting Hardware-Based ACLs

Use the following methods to troubleshoot a hardware-based ACL:

- To display the number of Layer 4 CAM entries being used by each ACL, enter the **show access-list** <acl-num> | <acl-name> | all command. See “Displaying ACL Information” on page 5-12.
- To view the types of packets being received on an interface, enable ACL statistics using the `enable-acl-counter` command, reapply the ACLs by entering the `ip rebind-acl all` command, then display the statistics by entering the `show ip acl-traffic` command.

- To determine whether an ACL entry is correctly matching packets, add the `log` option to the ACL entry, then reapply the ACL. This forces the device to send packets that match the ACL entry to the CPU for processing. The `log` option also generates a Syslog entry for packets that are permitted or denied by the ACL entry.

- To determine whether the issue is specific to fragmentation, remove the Layer 4 information (TCP or UDP application ports) from the ACL, then reapply the ACL.

If you are using another feature that requires ACLs, either use the same ACL entries for filtering and for the other feature, or change to flow-based ACLs.

## Hardware-Based Policy-Based Routing (PBR)

**NOTE:** This feature is supported on EP devices and on the ProCurve 9408sl running software release 01.0.01 or later.

Hardware-based Policy-Based Routing (PBR) routes traffic in hardware based on policies you define. A PBR policy specifies the next hop for traffic that matches the policy. A PBR policy also can use an ACL to perform QoS mapping and marking for traffic that matches the policy.

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the Layer 4 CAM on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps. You also can map and mark the traffic’s QoS information using the QoS options of the ACLs.

### Next Hop Selection

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. An exception is if you use the `ip policy prefer-direct-route` option. In this case, the policy will instead use a direct route if available. If none of the policy’s direct routes or next hops are available, PBR sends the traffic to the CPU for forwarding.

### Configuration Considerations

- EP supports an unlimited number of PBR policies that contain a single route map instance and a single ACL.

- EP supports up to 64 PBR policies that have more than one route map instance or more than one ACL. In this case, a given policy can have up to six route map instances, with up to six ACLs in each instance, and up to six next hops in each ACL.

- The ACL `log` and `<icmp-type>` options cause PBR to be performed by the CPU instead of in hardware. If you use either of these options in an ACL, no CAM entries are programmed for the ACL.

- PBR ignores explicit or implicit `deny ip any any` ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs.

- PBR always selects the first next hop from the next hop list that is up, unless you use the `ip policy prefer-direct-route` option. If you use this option, PBR selects a direct route instead. If a PBR policy’s next hop goes down, the policy uses another next hop if available. If no next hops are available, the device sends the traffic to the CPU for forwarding.

- For fragmented packets, by default PBR matches a fragment to an ACL if the source and destination addresses in the fragment exactly match an ACL. In this case, PBR uses the next hop that was used for the first fragment, which contains the Layer 4 UDP or TCP application port information. Alternatively, you can configure PBR to select the best next hop on an individual fragment basis.

**NOTE:** PBR is not supported for fragmented packets on 10 Gigabit Ethernet ports if the PBR’s ACL filters on Layer 4 information. For 10 Gigabit Ethernet, the PBR policy sends fragmented packets on the Layer 3 paths.
Configuring a PBR Policy

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR. If you want to map or mark QoS information in the packets, use the QoS options in the ACLs.
- Configure a route map that matches on the ACLs and sets the route information.
- Optionally, enable PBR to use the most direct route if available.
- Apply the route map to an interface.

**NOTE:** If you are using software release 07.6.04 or later, enter the `ip rebind-acl` command at the global CONFIG level of the CLI to place ACL configuration changes into effect.

Configuration Examples

**Basic Example**

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 5.5.5.x/24 through next-hop IP address 1.1.1.1/24 or, if 1.1.1.x is unavailable, through 2.2.2.1/24.

```
ProCurveRS(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq http 5.5.5.0 0.0.0.255
ProCurveRS(config)# route-map net10web permit 101
ProCurveRS(config-route-map net10web)# match ip address 101
ProCurveRS(config-route-map net10web)# set ip next-hop 1.1.1.1
ProCurveRS(config-route-map net10web)# set ip next-hop 2.2.2.2
ProCurveRS(config-route-map net10web)# exit
ProCurveRS(config)# vlan 10
ProCurveRS(config-vlan-10)# tag ethernet 1/1 to 1/4
ProCurveRS(config-vlan-10)# router-interface ve 1
ProCurveRS(config)# interface ve 1
ProCurveRS(config-vif-1)# ip policy route-map net10web
```

**Using the Most Direct Route**

To cause PBR policies to always use the most direct route available, enter the following command at the global CONFIG level of the CLI:

```
ProCurveRS(config)# ip policy prefer-direct-route
```

**Enabling PBR for Fragmented Packets**

By default, PBR policies apply at Layer 3 only. The device matches traffic against the Layer 3 information in a PBR policy's ACLs, and applies the policy if the traffic matches the ACL. The device does not apply a PBR policy to a packet fragment even if the fragment's IP addresses match an ACL in the policy. Instead, the device forwards the fragment using a non-PBR route. This is true even if an ACL in a PBR policy contains Layer 4 information.

To apply a PBR policy to packet fragments:

- Add Layer 4 information to the PBR ACL.

- Enable fragment matching on the interface that has the PBR policy. You can enable fragment matching for the source Layer 4 port, destination Layer 4 port, or both.
  - Enable matching on the destination Layer 4 port in load balancing configurations where you want to ensure that traffic for a particular application is forwarded on the PBR path to the load balancers.
  - Enable matching on the source Layer 4 port if you want to ensure premium service for all traffic from a specific client.
  - Enable matching on both source and destination Layer 4 port if you want to ensure premium service for
fragments in a given traffic flow.

The following example shows how to configure a PBR policy for Network File System (NFS) traffic, which uses UDP application port 2049. In this example, the next hop is selected individually for each fragment that exactly matches the destination IP address in one of the PBR policy's ACLs.

```plaintext
ProCurveRS(config)# access-list 111 permit udp any host 2.3.3.5 eq 2049
ProCurveRS(config)# route-map slbmap permit 1
ProCurveRS(config-route-map slbmap)# match ip address 111
ProCurveRS(config-route-map slbmap)# set next-hop 1.2.3.4
ProCurveRS(config-route-map slbmap)# exit
ProCurveRS(config)# interface ethernet 1/1
ProCurveRS(config-if-1/1)# ip policy route-map slbmap
ProCurveRS(config-if-1/1)# ip policy frag-match-dest
```

**PBR Syntax**

**Route Map Syntax**

**Syntax:** [no] route-map <map-name> permit | deny <num>

The `<map-name>` is a string of characters that names the map. Map names can be up to 32 characters in length. You can define up to 50 route maps on the Routing Switch.

The `permit`/`deny` parameter specifies the action the Routing Switch will take if a route matches a match statement.

- If you specify `deny`, the Routing Switch does not advertise or learn the route.
- If you specify `permit`, the Routing Switch applies the match and set statements associated with this route map instance.

The `<num>` parameter specifies the instance of the route map you are defining. Each route map can have up to 50 instances. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

**Syntax:** [no] match ip address <ACL-num-or-name>

The `<ACL-num-or-name>` parameter specifies a standard or extended ACL number or name.

**Syntax:** [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

---

**NOTE:** The `set ip default` option is not supported.

**NOTE:** The `set interface` option is not supported.

---

**ACL Syntax**

For detailed descriptions of the ACL syntax, see "Configuring Standard Numbered ACLs" on page 4-9 and "Configuring Extended Numbered ACLs" on page 4-13.

**Standard ACL Syntax**

**Syntax:** [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard> [log]

or

**Syntax:** [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname> [log]

**Syntax:** [no] access-list <num> deny | permit host <source-ip> | <hostname> [log]

**Syntax:** [no] access-list <num> deny | permit any [log]

**NOTE:** If you use the `log` option, the ACL entry is sent to the CPU for processing.
**Syntax:** [no] ip access-group <num> [in | out]

**NOTE:** The **out** option is not supported in the hardware-based ACL mode.
**Extended ACL Syntax**

**Syntax:** [no] access-list <num> deny | permit <ip-protocol> <source-ip> | <hostname> <wildcard> 
[<operator> <source-tcp/udp-port>] | <destination-ip> | <hostname> [<icmp-type>] <wildcard> 
[<operator> <destination-tcp/udp-port>] 
[established] 
[precedence <name> | <num>] 
[tos <num>] 
[priority 0 | 1 | 2 | 3] 
[priority-mapping <8021p-value>] 
[dscp-mapping <dscp-value>] 
[dscp-marking <dscp-value>] 
[log]

**NOTE:** The **priority, priority-mapping, dscp-mapping, and dscp-marking** options are supported in 07.6.04 and later and apply only to EP devices and to 10 Gigabit Ethernet modules. See "QoS Options for IP ACLs" on page 5-10.

**Syntax:** [no] access-list <num> deny | permit host <ip-protocol> any any [log]

**NOTE:** If you use the <icmp-type> or log option, the ACL entry is sent to the CPU for processing.

**Syntax:** [no] ip access-group <num> in | out

**NOTE:** The **out** option is not supported in the hardware-based ACL mode.

**PBR Policy Syntax**

**Syntax:** [no] ip policy route-map <map-name>

This command identifies a route map used by the PBR policy.

**Syntax:** [no] ip policy prefer-direct-route

This command configures the PBR policy to prefer a direct route when available.

**Syntax:** [no] ip policy frag-match-dest

This command configures the PBR policy to match on the destination Layer 4 port information as well as on the IP address information in the route map ACLs.

**Syntax:** [no] ip policy frag-match-src

This command configures the PBR policy to match on the source Layer 4 port information as well as on the IP address information in the route map ACLs.

**Syntax:** [no] ip policy frag-match-src-dest

This command configures the PBR policy to match on both the source and destination Layer 4 port information as well as on the IP address information in the route map ACLs.