

Configuring Committed Access Rate

This chapter describes how to configure committed access rate (CAR) and Distributed CAR (DCAR). For a complete description of the commands in this chapter, refer to the *Quality of Service Solutions Command Reference*; the commands are arranged alphabetically in that guide. To locate documentation of specific commands, use the command reference, master index, or search online.

CAR is supported on these platforms:

- Cisco 2600 series
- Cisco 3600 series
- Cisco 4500 series
- Cisco 4700 series
- Cisco 7200 series

Distributed CAR is supported on Cisco 7000 series routers with a Route Switch Processor-based RSP7000 interface processor or Cisco 7500 series routers with a Versatile Interface Processor-based VIP2-40 or greater interface processor. A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.

Note CAR and DCAR can only be used with IP traffic. Non-IP traffic is not rate limited. CAR and DCAR can be configured on an interface or subinterface. However, CAR and DCAR are not supported on the Fast EtherChannel, tunnel, or PRI interfaces, nor on any interface that does not support Cisco Express Forwarding (CEF).

CEF must be enabled on the interface before configuring CAR or DCAR.

Committed Access Rate Configuration Task List

The CAR and DCAR services limit the input or output transmission rate on an interface or subinterface based on a flexible set of criteria. CAR is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

CAR can rate limit traffic based on certain matching criteria, such as incoming interface, IP Precedence, or IP access list. You configure the actions CAR will take when traffic conforms to or exceeds the rate limit.

You can set CAR rate policies that are associated with one of the following:

- All IP traffic
- IP Precedence
- MAC address
- IP access list, both standard and extended. Matching to IP access lists is more processor-intensive than matching based on other criteria.

Each interface can have multiple CAR policies, corresponding to different types of traffic. For example, low priority traffic may be limited to a lower rate than high priority traffic. With multiple rate policies, the router examines each policy in the order entered until the packet matches. If a match is not found, the default action is to transmit.

The rate policies can be independent; each rate policy deals with a different type of traffic. Alternatively, rate policies can be cascading; a packet can be compared to multiple different rate policies in succession. You can configure up to 20 rate policies on a subinterface.

To configure CAR, perform the tasks in the following sections:

- Configure CAR and DCAR for All IP Traffic
- Configure CAR and DCAR Policies
- Configure a Class-Based DCAR Policy
- Monitor CAR and DCAR

See the section “CAR and DCAR Configuration Examples” later in this chapter for ideas of how to configure CAR and DCAR on your network.

Configure CAR and DCAR for All IP Traffic

To configure CAR (or DCAR on Cisco 7000 series with RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor) for all IP traffic, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	interface <i>interface-type interface-number</i>	Specify the interface or subinterface. This command puts the router in interface configuration mode.
2	rate-limit { input output } <i>bps burst-normal burst-max conform-action action exceed-action action</i>	Specify a basic CAR policy for all IP traffic. See Table 4 for a description of conform and exceed <i>action</i> keywords.
3	end	Exit interface configuration mode.

Basic CAR and DCAR functionality requires the following criteria to be defined:

- Packet direction, incoming or outgoing.
- An average rate, determined by a long-term average of the transmission rate. Traffic that falls under this rate will always conform.
- A normal burst size, which determines how large traffic bursts can be before some traffic is considered to exceed the rate limit.

- An excess burst size.

Traffic that falls between the normal burst size and the excess burst size exceeds the rate limit with a probability that increases as the burst size increases. CAR propagates bursts. It does no smoothing or shaping of traffic.

Conform and exceed actions are described in Table 4.

Table 4 Rate-Limit Command Action Keywords

Keyword	Description
continue	Evaluate the next rate-limit command.
drop	Drop the packet.
set-prec-continue <i>new-prec</i>	Set the IP Precedence and evaluate the next rate-limit command.
set-prec-transmit <i>new-prec</i>	Set the IP Precedence and transmit the packet.
transmit	Transmit the packet.

See the sections “Configure CAR and DCAR Policies” and “Configure a Class-Based DCAR Policy” to understand how to configure other CAR and DCAR policy options. See the sections “Subrate IP Services Example” and “Input and Output Rate Limiting on an Interface Example” for examples of how to configure CAR for all IP traffic.

Configure CAR and DCAR Policies

To configure CAR (or DCAR on Cisco 7000 series with the RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor), use the following commands beginning in interface configuration mode (only the first two commands are required):

Step	Command	Purpose
1	interface <i>interface-type interface-number</i>	Specify the interface or subinterface. This command puts the router in interface configuration mode.
2	rate-limit { input output } [access-group [rate-limit] <i>acl-index</i>] <i>bps burst-normal burst-max conform-action</i> <i>action exceed-action action</i>	Specify the rate policy for each particular class of traffic. See Table 4 for a description of conform and exceed <i>action</i> keywords. Repeat this command for each different class of traffic.
3	access-list rate-limit <i>acl-index</i> { <i>precedence</i> <i>mac-address</i> mask <i>prec-mask</i> }	(Optional) Specify a rate-limited access list. Repeat this command if you wish to specify a new access list.
4	access-list <i>acl-index</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] or access-list <i>acl-index</i> { deny permit } <i>protocol source</i> <i>source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [log]	(Optional) Specify a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.
5	end	Exit interface configuration mode.

The following sections describe requirements for specific policies.

IP Precedence or MAC Address

Use the **access-list rate-limit** command to classify packets using either IP Precedence or MAC addresses. You can then apply CAR policies using the **rate-limit** command to individual rate-limited access lists. Packets with different IP Precedences or MAC addresses are treated differently by the CAR service. See the section “Rate Limiting in an IXP Example” for an example of how to configure a CAR policy using MAC addresses.

IP Access List

Use the **access-list** command to define CAR policy based on access list. The *acl-index* argument is an access list number. Use a number from 1 to 99 to classify packets by precedence or precedence mask. Use a number from 100 to 199 to classify by MAC address.

Note If an access list is not present, the **rate-limit** command will act as if no access list is defined and all traffic will be rate limited accordingly.

See the section “Rate Limiting by Access List Example” for an example of how to configure a CAR policy using IP access lists.

Configure a Class-Based DCAR Policy

When you configure DCAR on Cisco 7000 series with RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor, you can classify packets by group, to allow you to partition your network into multiple priority levels or classes of service. This classification is done by setting IP Precedences based on different criteria for use by other QoS features such as Weighted Random Early Detection (WRED) or weighted fair queueing (WFQ).

To configure a class-based DCAR policy, use the following commands beginning in interface configuration mode (the **access-list** command is optional):

Step	Command	Purpose
1	interface <i>interface-type interface-number</i>	Specify the interface or subinterface. This command puts the router in interface configuration mode.
2	rate-limit { input output } [access-group [rate-limit] <i>acl-index</i>] <i>bps burst-normal burst-max conform-action</i> exceed-action <i>action</i>	Specify the rate policy for each particular class of traffic. Repeat this command for each different class of traffic. See Table 4 for policy conform and exceed <i>action</i> keywords.
3	random-detect precedence <i>precedence min-threshold max-threshold mark-prob-denominator</i>	Configure WRED and specify parameters for packets with specific IP Precedence.
4	access-list <i>acl-index</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] or access-list <i>acl-index</i> { deny permit } <i>protocol source source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [log]	(Optional) Specify a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.
5	end	Exit interface configuration mode.

Monitor CAR and DCAR

To monitor CAR and DCAR services in your network, use any the following commands in EXEC mode:

Command	Purpose
<code>show access-lists</code>	Show the contents of current IP and rate-limited access lists.
<code>show access-lists rate-limit [access-list-number]</code>	Show information about rate-limited access lists.
<code>show interfaces [interface-type interface-number] rate-limit</code>	Show information about an interface configured for CAR.

CAR and DCAR Configuration Examples

The following sections provide examples of ways you might use CAR and DCAR to control traffic into and out of your network:

- Subrate IP Services Example
- Input and Output Rate Limiting on an Interface Example
- Rate Limiting in an IXP Example
- Rate Limiting by Access List Example

Subrate IP Services Example

The following example illustrates how to configure a basic CAR policy that allows all IP traffic. In the example, the network operator delivers a physical T3 link to the customer, but offers a less expensive 15 Mbps subrate service. The customer pays only for the subrate bandwidth, which can be upgraded with additional access bandwidth based on demand. The CAR policy limits the traffic rate available to the customer and delivered to the network to the agreed upon rate limit, plus the ability to temporarily burst over the limit.

```
interface hssi 0/0/0
  rate-limit output 15000000 2812500 5625000 conform-action transmit exceed-action drop
  ip address 10.1.0.9 255.255.255.0
```

Input and Output Rate Limiting on an Interface Example

In this example, a customer is connected to an Internet service provider (ISP) by a T3 link. The ISP wants to rate limit the customer's transmissions to 15 Mbps of the 45 Mbps. In addition, the customer is allowed to transmit bursts of 2, 812,500 bytes. All exceeding packets are dropped. The following commands are configured on the ISP's High-Speed Serial Interface (HSSI) connected to the customer:

```
interface Hssi0/0/0
  description 45Mbps to R1
  rate-limit input 15000000 2812500 2812500 conform-action transmit exceed-action drop
  ip address 200.200.14.250 255.255.255.252
  rate-limit output 15000000 2812500 2812500 conform-action transmit exceed-action drop
```

To verify the configuration and monitor CAR statistics, use the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit

Hssi0/0/0 45Mbps to R1
Input
  matches: all traffic
  params: 15000000 bps, 2812500 limit, 2812500 extended limit
  conformed 8 packets, 428 bytes; action: transmit
  exceeded 0 packets, 0 bytes; action: drop
  last packet: 8680ms ago, current burst: 0 bytes
  last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
Output
  matches: all traffic
  params: 15000000 bps, 2812500 limit, 2812500 extended limit
  conformed 0 packets, 0 bytes; action: transmit
  exceeded 0 packets, 0 bytes; action: drop
  last packet: 8680ms ago, current burst: 0 bytes
  last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
```

Rate Limiting in an IXP Example

The following example uses rate limiting to control traffic in an Internet Exchange Point (IXP). Because an IXP comprises many neighbors around an FDDI ring, MAC address rate-limited access lists are used to control traffic from a particular ISP. Traffic from one ISP (at MAC address 00e0.34b0.7777) is compared to a rate limit of 80 Mbps of the 100 Mbps available on the FDDI connection. Traffic that conforms to this rate is transmitted. Nonconforming traffic is dropped.

```
interface Fddi2/1/0
  rate-limit input access-group rate-limit 100 80000000 64000 80000 conform-action
  transmit exceed-action drop
  ip address 200.200.6.1 255.255.255.0
!
access-list rate-limit 100 00e0.34b0.7777
```

To verify the configuration and monitor the CAR statistics, use the **show interfaces rate-limit** command:

```
Router# show interfaces fddi2/1/0 rate-limit

Fddi2/1/0
Input
  matches: access-group rate-limit 100
  params: 800000000 bps, 64000 limit, 80000 extended limit
  conformed 0 packets, 0 bytes; action: transmit
  exceeded 0 packets, 0 bytes; action: drop
  last packet: 4737508ms ago, current burst: 0 bytes
  last cleared 01:05:47 ago, conformed 0 bps, exceeded 0 bps
```

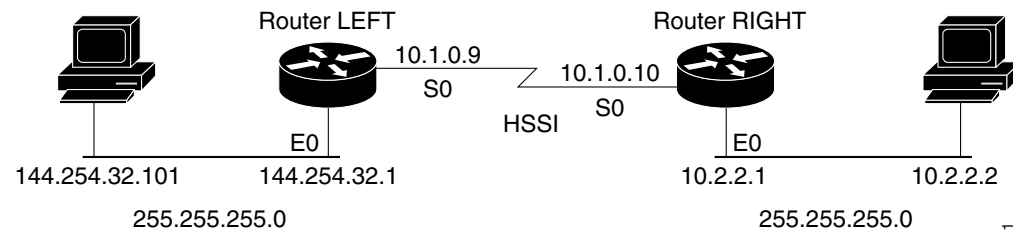
Rate Limiting by Access List Example

The following example shows how CAR can be used to limit the rate by application to ensure capacity for other traffic including mission-critical applications. In the example:

- All World Wide Web traffic is transmitted. However, the IP Precedence for Web traffic that conforms to the first rate policy is set to 5. For nonconforming Web traffic, the IP precedence is set to 0 (best effort).
- FTP traffic is transmitted with an IP Precedence of 5 if it conforms to the second rate policy. If the FTP traffic exceeds the rate policy, it is dropped.
- Any remaining traffic is limited to 8 Mbps, with a normal burst size of 16000 bytes and an excess burst size of 24000 bytes. Traffic that conforms is transmitted with an IP Precedence of 5. Traffic that does not conform is dropped.

Figure 4 illustrates the configuration. Notice that two access lists are created to classify the Web and FTP traffic so that they can be handled separately by CAR.

Figure 4 Rate Limiting by Access List



17191

Configuration Commands for Router LEFT

```
interface Hssi0/0/0
  description 45Mbps to R2
  rate-limit output access-group 101 20000000 24000 32000 conform-action set-prec-
  transmit 5 exceed-action set-prec-transmit 0
  rate-limit output access-group 102 10000000 24000 32000 conform-action
  set-prec-transmit 5 exceed-action drop
  rate-limit output 8000000 16000 24000 conform-action set-prec-transmit 5
  exceed-action drop
  ip address 10.1.0.9 255.255.255.0
  !
access-list 101 permit tcp any any eq www
access-list 102 permit tcp any any eq ftp
```

To verify the configuration and monitor CAR statistics, use the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit

Hssi0/0/0 45Mbps to R2
Input
matches: access-group 101
  params: 20000000 bps, 24000 limit, 32000 extended limit
  conformed 3 packets, 189 bytes; action: set-prec-transmit 5
  exceeded 0 packets, 0 bytes; action: set-prec-transmit 0
  last packet: 309100ms ago, current burst: 0 bytes
  last cleared 00:08:00 ago, conformed 0 bps, exceeded 0 bps
matches: access-group 102
  params: 10000000 bps, 24000 limit, 32000 extended limit
  conformed 0 packets, 0 bytes; action: set-prec-transmit 5
  exceeded 0 packets, 0 bytes; action: drop
  last packet: 19522612ms ago, current burst: 0 bytes
  last cleared 00:07:18 ago, conformed 0 bps, exceeded 0 bps
matches: all traffic
  params: 8000000 bps, 16000 limit, 24000 extended limit
  conformed 5 packets, 315 bytes; action: set-prec-transmit 5
  exceeded 0 packets, 0 bytes; action: drop
  last packet: 9632ms ago, current burst: 0 bytes
  last cleared 00:05:43 ago, conformed 0 bps, exceeded 0 bps
```