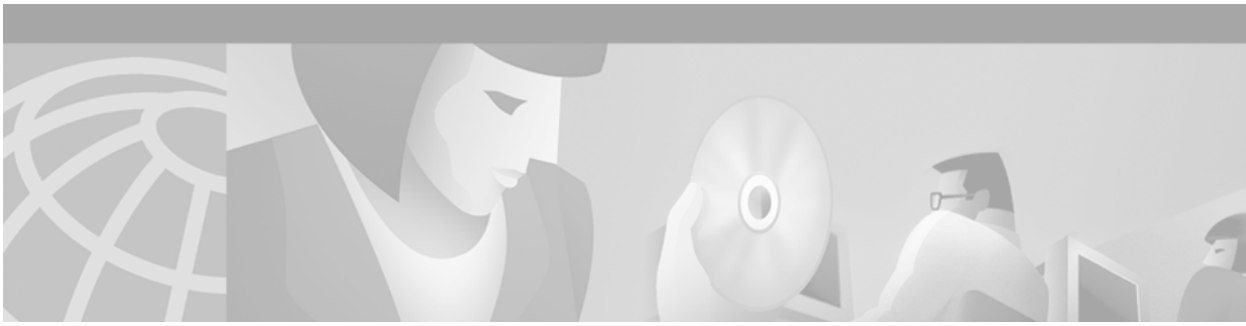


Policing and Shaping Overview

Cisco IOS QoS offers two kinds of traffic regulation mechanisms—policing and shaping.

The rate-limiting features of committed access rate (CAR) and the Traffic Policing feature provide the functionality for policing traffic. The features of Generic Traffic Shaping (GTS), Class-Based Shaping, Distributed Traffic Shaping (DTS), and Frame Relay Traffic Shaping (FRTS) provide the functionality for shaping traffic.

Note To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com. You can access Feature Navigator at http://www.cisco.com/go/fn.

You can deploy these features throughout your network to ensure that a packet, or data source, adheres to a stipulated contract and to determine the QoS to render the packet. Both policing and shaping mechanisms use the traffic descriptor for a packet—indicated by the classification of the packet—to ensure adherence and service. (See the chapter "Classification Overview" in this book for a description of a traffic descriptor.)

Policers and shapers usually identify traffic descriptor violations in an identical manner. They usually differ, however, in the way they respond to violations, for example:

- A policer typically drops traffic. (For example, the CAR rate-limiting policer will either drop the packet or rewrite its IP precedence, resetting the type of service bits in the packet header.)
- A shaper typically delays excess traffic using a buffer, or queueing mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected. (For example, GTS and Class-Based Shaping use a weighted fair queue to delay packets in order to shape the flow, and DTS and FRTS use either a priority queue, a custom queue, or a FIFO queue for the same, depending on how you configure it.)

Traffic shaping and policing can work in tandem. For example, a good traffic shaping scheme should make it easy for nodes inside the network to detect misbehaving flows. This activity is sometimes called policing the traffic of the flow.

This chapter gives a brief description of the Cisco IOS QoS traffic policing and shaping mechanisms. Because policing and shaping all use the token bucket mechanism, this chapter first explains how a token bucket works. This chapter includes the following sections:

What Is a Token Bucket?

A token bucket is a formal definition of a rate of transfer. It has three components: a burst size, a mean rate, and a time interval (Tc). Although the mean rate is generally represented as bits per second, any two values may be derived from the third by the relation shown as follows:

```
mean rate = burst size / time interval
```

Here are some definitions of these terms:

- Mean rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the Committed Burst (Bc) size, it specifies in bits (or bytes) per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For a shaper, such as GTS, it specifies bits per burst; for a policer, such as CAR, it specifies bytes per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

By definition, over any integral multiple of the interval, the bit rate of the interface will not exceed the mean rate. The bit rate, however, may be arbitrarily fast within the interval.

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer, such as CAR, or a traffic shaper, such as FRTS or GTS. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator. (Neither CAR nor FRTS and GTS implement either a true token bucket or true leaky bucket.)

In the token bucket metaphor, tokens are put into the bucket at a certain rate. The bucket itself has a specified capacity. If the bucket fills to capacity, newly arriving tokens are discarded. Each token is permission for the source to send a certain number of bits into the network. To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens (in the case of GTS) or the packet is discarded or marked down (in the case of CAR). If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst a source can send into the network is roughly proportional to the size of the bucket.

Note that the token bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue; if it did not have a data buffer, it would be a policer. For traffic shaping, packets that arrive that cannot be sent immediately are delayed in the data buffer.

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket's capacity, divided by the time interval, plus the established rate at which tokens are placed in the token bucket. See the following formula:

```
(token bucket capacity in bits / time interval in seconds) + established rate in bps =
maximum flow speed in bps
```

This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Policing with CAR

CAR embodies a rate-limiting feature for policing traffic, in addition to its packet classification feature discussed in the chapter "Classification Overview" in this book. The rate-limiting feature of CAR manages the access bandwidth policy for a network by ensuring that traffic falling within specified rate parameters is sent, while dropping packets that exceed the acceptable amount of traffic or sending them with a different priority. The exceed action for CAR is to drop or mark down packets.

The rate-limiting function of CAR does the following:

- Allows you to control the maximum rate of traffic sent or received on an interface.
- Gives you the ability to define Layer 3 aggregate or granular incoming or outgoing (ingress or egress) bandwidth rate limits and to specify traffic handling policies when the traffic either conforms to or exceeds the specified rate limits.

  Aggregate bandwidth rate limits match all of the packets on an interface or subinterface. Granular bandwidth rate limits match a particular type of traffic based on precedence, MAC address, or other parameters.

CAR is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

How It Works

CAR examines traffic received on an interface or a subset of that traffic selected by access list criteria. It then compares the rate of the traffic to a configured token bucket and takes action based on the result. For example, CAR will drop the packet or rewrite the IP precedence by resetting the type of service (ToS) bits. You can configure CAR to send, drop, or set precedence.

Aspects of CAR rate limiting are explained in the following sections:

- Matching Criteria
- Rate Limits
- Conform and Exceed Actions
- Multiple Rate Policies

CAR utilizes a token bucket measurement. Tokens are inserted into the bucket at the committed rate. The depth of the bucket is the burst size. Traffic arriving at the bucket when sufficient tokens are available is said to conform, and the corresponding number of tokens are removed from the bucket. If a sufficient number of tokens are not available, then the traffic is said to exceed.

Matching Criteria

Traffic matching entails identification of traffic of interest for rate limiting, precedence setting, or both. Rate policies can be associated with one of the following qualities:

- Incoming interface
- All IP traffic
- IP precedence (defined by a rate-limit access list)
- MAC address (defined by a rate-limit access list)
- Multiprotocol Label Switching (MPLS) experimental (EXP) value (defined by a rate-limit access list)

- IP access list (standard and extended)

CAR provides configurable actions, such as send, drop, or set precedence when traffic conforms to or exceeds the rate limit.

Note Matching to IP access lists is more processor-intensive than matching based on other criteria.

Rate Limits

CAR propagates bursts. It does no smoothing or shaping of traffic, and therefore does no buffering and adds no delay. CAR is highly optimized to run on high-speed links—DS3, for example—in distributed mode on Versatile Interface Processors (VIPs) on the Cisco 7500 series.

CAR rate limits may be implemented either on input or output interfaces or subinterfaces including Frame Relay and ATM subinterfaces.

What Rate Limits Define

Rate limits define which packets conform to or exceed the defined rate based on the following three parameters:

- Average rate. The average rate determines the long-term average transmission rate. Traffic that falls under this rate will always conform.
- Normal burst size. The normal burst size determines how large traffic bursts can be before some traffic exceeds the rate limit.
- Excess Burst size. The Excess Burst (Be) size determines how large traffic bursts can be before all traffic exceeds the rate limit. 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.

The maximum number of tokens that a bucket can contain is determined by the normal burst size configured for the token bucket.

When the CAR rate limit is applied to a packet, CAR removes from the bucket tokens that are equivalent in number to the byte size of the packet. If a packet arrives and the byte size of the packet is greater than the number of tokens available in the standard token bucket, extended burst capability is engaged if it is configured.

Extended Burst Value

Extended burst is configured by setting the extended burst value greater than the normal burst value. Setting the extended burst value equal to the normal burst value excludes the extended burst capability. If extended burst is not configured, given the example scenario, the exceed action of CAR takes effect because a sufficient number of tokens are not available.

When extended burst is configured and this scenario occurs, the flow is allowed to borrow the needed tokens to allow the packet to be sent. This capability exists so as to avoid tail-drop behavior, and, instead, engage behavior like that of Random Early Detection (RED).

How Extended Burst Capability Works

Here is how the extended burst capability works. If a packet arrives and needs to borrow *n* number of tokens because the token bucket contains fewer tokens than its packet size requires, then CAR compares the following two values:

- Extended burst parameter value.

- Compounded debt. Compounded debt is computed as the sum over all *ai*:
  - *a* indicates the actual debt value of the flow after packet *i* is sent. Actual debt is simply a count of how many tokens the flow has currently borrowed.
  - *i* indicates the *i*th packet that attempts to borrow tokens since the last time a packet was dropped.

If the compounded debt is greater than the extended burst value, the exceed action of CAR takes effect. After a packet is dropped, the compounded debt is effectively set to 0. CAR will compute a new compounded debt value equal to the actual debt for the next packet that needs to borrow tokens.

If the actual debt is greater than the extended limit, all packets will be dropped until the actual debt is reduced through accumulation of tokens in the token bucket.

Dropped packets do not count against any rate or burst limit. That is, when a packet is dropped, no tokens are removed from the token bucket.

Note Though it is true the entire compounded debt is forgiven when a packet is dropped, the actual debt is not forgiven, and the next packet to arrive to insufficient tokens is immediately assigned a new compounded debt value equal to the current actual debt. In this way, actual debt can continue to grow until it is so large that no compounding is needed to cause a packet to be dropped. In effect, at this time, the compounded debt is not really forgiven. This scenario would lead to excessive drops on streams that continually exceed normal burst. (See the example in the following section, "Actual and Compounded Debt Example."

Testing of TCP traffic suggests that the chosen normal and extended burst values should be on the order of several seconds worth of traffic at the configured average rate. That is, if the average rate is 10 Mbps, then a normal burst size of 10 to 20 Mbps and an Excess Burst size of 20 to 40 Mbps would be appropriate.

Recommended Burst Values

Cisco recommends the following values for the normal and extended burst parameters:

```
normal burst = configured rate * (1 byte)/(8 bits) * 1.5 seconds
extended burst = 2 * normal burst
```

With the listed choices for parameters, extensive test results have shown CAR to achieve the configured rate. If the burst values are too low, then the achieved rate is often much lower than the configured rate.

Actual and Compounded Debt Example

This example shows how the compounded debt is forgiven, but the actual debt accumulates.

For this example, assume the following parameters:

- Token rate is 1 data unit per time unit
- Normal burst size is 2 data units
- Extended burst size is 4 data units
- 2 data units arrive per time unit

After 2 time units, the stream has used up its normal burst and must begin borrowing one data unit per time unit, beginning at time unit 3:

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
1       2               0               0
2       2               0               0
```

```
3       2               1               1
4       2               2               3
5       2               3 (temporary)   6 (temporary)
```

At this time a packet is dropped because the new compounded debt (6) would exceed the extended burst limit (4). When the packet is dropped, the compounded debt effectively becomes 0, and the actual debt is 2. (The values 3 and 6 were only temporary and do not remain valid in the case where a packet is dropped.) The final values for time unit 5 follow. The stream begins borrowing again at time unit 6.

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
5       2               2               0
6       2               3               3
7       2               4 (temporary)   7 (temporary)
```

At time unit 6, another packet is dropped and the debt values are adjusted accordingly.

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
7       2               3               0
```

Conform and Exceed Actions

CAR utilizes a token bucket, thus CAR can pass temporary bursts that exceed the rate limit as long as tokens are available.

Once a packet has been classified as conforming to or exceeding a particular rate limit, the router performs one of the following actions on the packet:

- Transmit—The packet is sent.
- Drop—The packet is discarded.
- Set precedence and transmit—The IP Precedence (ToS) bits in the packet header are rewritten. The packet is then sent. You can use this action to either color (set precedence) or recolor (modify existing packet precedence) the packet.
- Continue—The packet is evaluated using the next rate policy in a chain of rate limits. If there is not another rate policy, the packet is sent.
- Set precedence and continue—Set the IP Precedence bits to a specified value and then evaluate the next rate policy in the chain of rate limits.

For VIP-based platforms, two more actions are possible:

- Set QoS group and transmit—The packet is assigned to a QoS group and sent.
- Set QoS group and continue—The packet is assigned to a QoS group and then evaluated using the next rate policy. If there is not another rate policy, the packet is sent.

Multiple Rate Policies

A single CAR rate policy includes information about the rate limit, conform actions, and exceed actions. Each interface can have multiple CAR rate policies corresponding to different types of traffic. For example, low priority traffic may be limited to a lower rate than high priority traffic. When there are multiple rate policies, the router examines each policy in the order entered until the packet matches. If no match is found, the default action is to send.

Rate policies can be independent: each rate policy deals with a different type of traffic. Alternatively, rate policies can be cascading: a packet may be compared to multiple different rate policies in succession.

Cascading of rate policies allows a series of rate limits to be applied to packets to specify more granular policies (for example, you could rate limit total traffic on an access link to a specified subrate bandwidth and then rate limit World Wide Web traffic on the same link to a given proportion of the subrate limit) or to match packets against an ordered sequence of policies until an applicable rate limit is encountered (for example, rate limiting several MAC addresses with different bandwidth allocations at an exchange point). You can configure up to a 100 rate policies on a subinterface.

Restrictions

CAR and VIP-distributed CAR can only be used with IP traffic. Non-IP traffic is not rate limited.

CAR or VIP-distributed CAR can be configured on an interface or subinterface. However, CAR and VIP-distributed CAR are not supported on the following interfaces:

- Fast EtherChannel
- Tunnel
- PRI
- Any interface that does not support Cisco Express Forwarding (CEF)

CAR is only supported on ATM subinterfaces with the following encapsulations: aal5snap, aal5mux, and aal5nlpid.

Note CAR provides rate limiting and does not guarantee bandwidth. CAR should be used with other QoS features, such as distributed weighted fair queueing (WFQ) (DWFQ), if premium bandwidth assurances are required.

Traffic Policing

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface, and to partition a network into multiple priority levels or class of service (CoS).

The Traffic Policing feature manages the maximum rate of traffic through a token bucket algorithm. The token bucket algorithm can use the user-configured values to determine the maximum rate of traffic allowed on an interface at a given moment in time. The token bucket algorithm is affected by all traffic entering or leaving (depending on where the traffic policy with Traffic Policing configured) and is useful in managing network bandwidth in cases where several large packets are sent in the same traffic stream.

The token bucket algorithm provides users with three actions for each packet: a conform action, an exceed action, and an optional violate action. Traffic entering the interface with Traffic Policing configured is placed in to one of these categories. Within these three categories, users can decide packet treatments. For instance, packets that conform can be configured to be transmitted, packets that exceed can be configured to be sent with a decreased priority, and packets that violate can be configured to be dropped.

Traffic Policing is often configured on interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. In the most common Traffic Policing configurations, traffic that conforms is transmitted and traffic that exceeds is sent with a decreased priority or is dropped. Users can change these configuration options to suit their network needs.

The Traffic Policing feature supports the following MIBs:

- CISCO-CLASS-BASED-QOS-MIB
- CISCO-CLASS-BASED-QOS-CAPABILITY-MIB

This feature also supports RFC 2697, *A Single Rate Three Color Marker.*

For information on how to configure the Traffic Policing feature, see the chapter "Configuring Traffic Policing" in this book.

Benefits

Bandwidth Management Through Rate Limiting

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface. Traffic policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. Traffic that falls within the rate parameters is sent, whereas traffic that exceeds the parameters is dropped or sent with a different priority.

Packet Marking Through IP Precedence, QoS Group, and DSCP Value Setting

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS), as follows:

- Use traffic policing to set the IP precedence or differentiated services code point (DSCP) values for packets entering the network. Networking devices within your network can then use the adjusted IP Precedence values to determine how the traffic should be treated. For example, the DWRED feature uses the IP Precedence values to determine the probability that a packet will be dropped.
- Use traffic policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets.

Restrictions

The following restrictions apply to the Traffic Policing feature:

- On a Cisco 7500 series router, traffic policing can monitor CEF switching paths only. In order to use the Traffic Policing feature, CEF must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, traffic policing cannot be applied to packets that originated from or are destined to a router.
- Traffic policing can be configured on an interface or a subinterface.
- Traffic policing is not supported on the following interfaces:
  - Fast EtherChannel
  - Tunnel
  - PRI
  - Any interface on a Cisco 7500 series router that does not support CEF

Prerequisites

On a Cisco 7500 series router, CEF must be configured on the interface before traffic policing can be used.

For additional information on CEF, refer to the *Cisco IOS Switching Services Configuration Guide.*

Traffic Shaping

Cisco IOS QoS software has three types of traffic shaping: GTS, class-based, and FRTS. All three of these traffic shaping methods are similar in implementation, though their CLIs differ somewhat and they use different types of queues to contain and shape traffic that is deferred. In particular, the underlying code that determines whether enough credit is in the token bucket for a packet to be sent or whether that packet must be delayed is common to both features. If a packet is deferred, GTS and Class-Based Shaping use a weighted fair queue to hold the delayed traffic. FRTS uses either a custom queue or a priority queue for the same, depending on what you have configured.

This section explains how traffic shaping works, then it describes the Cisco IOS QoS traffic shaping mechanisms. It includes the following sections:

- About Traffic Shaping
- Generic Traffic Shaping
- Class-Based Shaping
- Distributed Traffic Shaping
- Frame Relay Traffic Shaping

For description of a token bucket and explanation of how it works, see the section "What Is a Token Bucket?" earlier in this chapter.

About Traffic Shaping

Traffic shaping allows you to control the traffic going out an interface in order to match its flow to the speed of the remote target interface and to ensure that the traffic conforms to policies contracted for it. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.

Why Use Traffic Shaping?

The primary reasons you would use traffic shaping are to control access to available bandwidth, to ensure that traffic conforms to the policies established for it, and to regulate the flow of traffic in order to avoid congestion that can occur when the sent traffic exceeds the access speed of its remote, target interface. Here are some example reasons why you would use traffic shaping:

- Control access to bandwidth when, for example, policy dictates that the rate of a given interface should not on the average exceed a certain rate even though the access rate exceeds the speed.
- Configure traffic shaping on an interface if you have a network with differing access rates. Suppose that one end of the link in a Frame Relay network runs at 256 kbps and the other end of the link runs at 128 kbps. Sending packets at 256 kbps could cause failure of the applications using the link.

A similar, more complicated case would be a link-layer network giving indications of congestion that has differing access rates on different attached DTE; the network may be able to deliver more transit speed to a given DTE device at one time than another. (This scenario warrants that the token bucket be derived, and then its rate maintained.)

- If you offer a subrate service. In this case, traffic shaping enables you to use the router to partition your T1 or T3 links into smaller channels.

Traffic shaping prevents packet loss. Its use is especially important in Frame Relay networks because the switch cannot determine which packets take precedence, and therefore which packets should be dropped when congestion occurs. Moreover, it is of critical importance for real-time traffic such as Voice over Frame Relay that latency be bounded, thereby bounding the amount of traffic and traffic loss in the data link network at any given time by keeping the data in the router that is making the guarantees. Retaining the data in the router allows the router to prioritize traffic according to the guarantees it is making. (Packet loss can result in detrimental consequences for real-time and interactive applications.)

Traffic Shaping and Rate of Transfer

Traffic shaping limits the rate of transmission of data. You can limit the data transfer to one of the following:

- A specific configured rate
- A derived rate based on the level of congestion

As mentioned, the rate of transfer depends on these three components that constitute the token bucket: burst size, mean rate, measurement (time) interval. The mean rate is equal to the burst size divided by the interval.

When traffic shaping is enabled, the bit rate of the interface will not exceed the mean rate over any integral multiple of the interval. In other words, during every interval, a maximum of burst size can be sent. Within the interval, however, the bit rate may be faster than the mean rate at any given time.

One additional variable applies to traffic shaping: Be size. The Excess Burst size corresponds to the number of noncommitted bits—those outside the CIR—that are still accepted by the Frame Relay switch but marked as discard eligible (DE).

In other words, the Be size allows more than the burst size to be sent during a time interval in certain situations. The switch will allow the packets belonging to the Excess Burst to go through but it will mark them by setting the DE bit. Whether the packets are sent depends on how the switch is configured.

When the Be size equals 0, the interface sends no more than the burst size every interval, achieving an average rate no higher than the mean rate. However, when the Be size is greater than 0, the interface can send as many as Bc + Be bits in a burst, if in a previous time period the maximum amount was not sent. Whenever less than the burst size is sent during an interval, the remaining number of bits, up to the Be size, can be used to send more than the burst size in a later interval.

Discard Eligible Bit

You can specify which Frame Relay packets have low priority or low time sensitivity and will be the first to be dropped when a Frame Relay switch is congested. The mechanism that allows a Frame Relay switch to identify such packets is the DE bit.

You can define DE lists that identify the characteristics of packets to be eligible for discarding, and you can also specify DE groups to identify the data-link connection identifier (DLCI) that is affected.

You can specify DE lists based on the protocol or the interface, and on characteristics such as fragmentation of the packet, a specific TCP or User Datagram Protocol (UDP) port, an access list number, or a packet size.

Differences Between Shaping Mechanisms

As mentioned, GTS, Class-Based Shaping, DTS, and FRTS are similar in implementation, sharing the same code and data structures, but they differ in regard to their CLIs and the queue types they use.

Here are a few ways in which these mechanisms differ:

- For GTS, the shaping queue is a weighted fair queue. For FRTS, the queue can be a weighted fair queue (configured by the **frame-relay fair-queue** command), a strict priority queue with WFQ (configured by the **frame-relay ip rtp priority** command in addition to the **frame-relay fair-queue** command), custom queueing (CQ), priority queueing (PQ), or FIFO.
- For Class-Based Shaping, GTS can be configured on a class, rather than only on an access control list (ACL). In order to do so, you must first define traffic classes based on match criteria including protocols, ACLs, and input interfaces. You can then apply traffic shaping to each defined class.
- FRTS supports shaping on a per-DLCI basis; GTS and DTS are configurable per interface or subinterface.
- DTS supports traffic shaping based on a variety of match criteria, including user-defined classes, and DSCP.

Table 11 summarizes these differences.

Table 11 Differences Between Shaping Mechanisms

Mechanism	GTS	Class-Based	DTS	FRTS
Command-Line Interface	• Applies parameters per subinterface • **traffic group** command supported	• Applies parameters per interface or per class	• Applies parameters per interface or subinterface	• Classes of parameters • Applies parameters to all virtual circuits (VCs) on an interface through inheritance mechanism • No traffic group command
Queues Supported	• WFQ per subinterface	• CBWFQ inside GTS	• WFQ, strict priority queue with WFQ, CQ, PQ, first- come, first- served (FCFS) per VC	• WFQ, strict priority queue with WFQ, CQ, PQ, FCFS per VC

You can configure GTS to behave the same as FRTS by allocating one DLCI per subinterface and using GTS plus backward explicit congestion notification (BECN) support. The behavior of the two is then the same except for the different shaping queues used.

Traffic Shaping and Queueing

Traffic shaping smooths traffic by storing traffic above the configured rate in a queue.

When a packet arrives at the interface for transmission, the following sequence happens:

1. If the queue is empty, the arriving packet is processed by the traffic shaper.
   - If possible, the traffic shaper sends the packet.
   - Otherwise, the packet is placed in the queue.
2. If the queue is not empty, the packet is placed in the queue.

When packets are in the queue, the traffic shaper removes the number of packets it can send from the queue every time interval.

Generic Traffic Shaping

GTS shapes traffic by reducing outbound traffic flow to avoid congestion by constraining traffic to a particular bit rate using the token bucket mechanism. (See the section "What Is a Token Bucket?" earlier in this chapter.)

GTS applies on a per-interface basis and can use access lists to select the traffic to shape. It works with a variety of Layer 2 technologies, including Frame Relay, ATM, Switched Multimegabit Data Service (SMDS), and Ethernet.

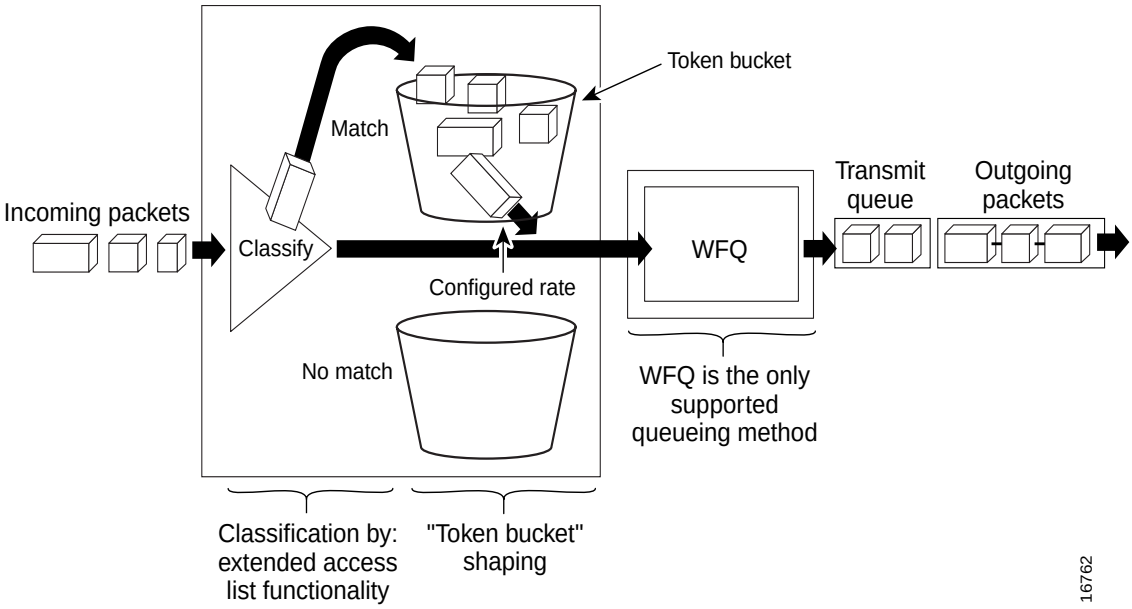
On a Frame Relay subinterface, GTS can be set up to adapt dynamically to available bandwidth by integrating backward explicit congestion notification (BECN) signals, or set up simply to shape to a specified rate. GTS can also be configured on an ATM/ATM Interface Processor (AIP) interface to respond to the Resource Reservation Protocol (RSVP) feature signalled over statically configured ATM permanent virtual circuits (PVCs).

GTS is supported on most media and encapsulation types on the router. GTS can also be applied to a specific access list on an interface.

Note GTS is not supported on Multilink PPP (MLP) interfaces.

Figure 12 shows how GTS works.

Figure 12 Generic Traffic Shaping



For information on how to configure GTS, see the chapter "Configuring Generic Traffic Shaping" in this book.

Class-Based Shaping

Traffic shaping allows you to control the traffic going out an interface in order to match its transmission to the speed of the remote, target interface and to ensure that the traffic conforms to policies contracted for it. Traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.

For information on how to configure Class-Based Shaping, see the chapter "Configuring Class-Based Shaping" in this book.

How It Works

Class-Based Shaping can be enabled on any interface that supports GTS. Using the Class-Based Shaping feature, you can perform the following tasks:

- Configure GTS on a traffic class. Configuring GTS to classes provides greater flexibility for configuring traffic shaping. Previously, this ability was limited to the use of ACLs.
- Specify average rate or peak rate traffic shaping. Specifying peak rate shaping allows you to make better use of available bandwidth by allowing more data than the CIR to be sent if the bandwidth is available.
- Configure class-based weighted fair queueing (CBWFQ) inside GTS. CBWFQ allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. Taking into account available bandwidth on the interface, you can configure up to 64 classes and control distribution among them, which is not the case with flow-based WFQ.

Flow-based WFQ applies weights to traffic to classify it into conversations and determine how much bandwidth each conversation is allowed relative to other conversations. These weights, and traffic classification, are dependent on and limited to the seven IP Precedence levels.

CBWFQ allows you to define what constitutes a class based on criteria that exceed the confines of flow. CBWFQ allows you to use ACLs and protocols or input interface names to define how traffic will be classified, thereby providing coarser granularity. You need not maintain traffic classification on a flow basis. Moreover, you can configure up to 64 discrete classes in a service policy.

Restrictions

Peak and average traffic shaping is configured on a per-interface or per-class basis, and cannot be used in conjunction with commands used to configure GTS from previous versions of Cisco IOS. These commands include the following:

- **traffic-shape adaptive**
- **traffic-shape fecn-adaptive**
- **traffic-shape group**
- **traffic-shape rate**

Adaptive traffic shaping for Frame Relay networks is not supported using the Class-Based Shaping feature. To configure adaptive GTS for Frame Relay networks, you must use the commands from releases prior to Release 12.1(2) of Cisco IOS software.

Distributed Traffic Shaping

The DTS feature provides a method of managing the bandwidth of an interface to avoid congestion, to meet remote site requirements, and to conform to a service rate that is provided on that interface.

DTS uses queues to buffer traffic surges that can congest a network and send the data in to the network at a regulated rate. This ensures that traffic will behave to the configured descriptor, as defined by the CIR, Bc, and Be. With the defined average bit rate and burst size that is acceptable on that shaped entity, you can derive a time interval value.

The Be size allows more than the Bc size to be sent during a time interval under certain conditions. Therefore, DTS provides two types of **shape** commands: **average** and **peak**. When **shape average** is configured, the interface sends no more than the Bc size for each interval, achieving an average rate no higher than the CIR. When the **shape peak** command is configured, the interface sends Bc plus Be bits in each interval.

In a link layer network such as Frame Relay, the network sends messages with the forward explicit congestion notification (FECN) or BECN if there is congestion. With the DTS feature, the traffic shaping adaptive mode takes advantage of these signals and adjusts the traffic descriptors, therefore regulating the amount of traffic entering or leaving the interface accordingly.

DTS provides the following key benefits:

- Offloads traffic shaping from the Route Switch Processor (RSP) to the VIP.
- Supports up to 200 shape queues per VIP, supporting up to OC-3 rates when the average packet size is 250 bytes or greater and when using a VIP2-50 or better with 8 MB of SRAM. Line rates below T3 are supported with a VIP2-40.
- Configures DTS at the interface level or subinterface level.

- Shaping based on the following traffic match criteria:
  - Access list
  - Packet marking
  - Input port
  - Other matching criteria. For information about other matching criteria, see the section "Creating a Traffic Class" in the chapter "Configuring the Modular Quality of Service Command-Line Interface" in this book.
- Optional configuration to respond to Frame Relay network congestion (indicated by the presence of BECN or ForeSight signals) by reducing the shaped-to rate for a period of time until congestion is believed to have subsided. Supports FECN, BECN, and ForeSight Frame Relay signalling.

This feature runs on Cisco 7500 series routers with VIP2-40, VIP2-50, or greater.

For information on how to configure DTS, see the chapter "Configuring Distributed Traffic Shaping" in this book.

Restrictions

DTS does not support the following:

- Fast EtherChannel interfaces, Multilink PPP (MLP), tunnels and dialer interfaces

Note Hierarchical DTS (that is, DTS configured in both a parent-level policy and a child-level policy), is not supported on subinterfaces.

- Any VIP below a VIP2-40

Note A VIP2-50 is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 card is required for OC-3 rates.

Prerequisites

Distributed Cisco Express Forwarding (dCEF) must be enabled on the interface before DTS can be enabled.

A policy map and class maps must be created before DTS is enabled.

Frame Relay Traffic Shaping

Cisco has long provided support for FECN for DECnet and OSI, and BECN for Systems Network Architecture (SNA) traffic using Logical Link Control, type 2 (LLC2) encapsulation via RFC 1490 and DE bit support. FRTS builds upon this existing Frame Relay support with additional capabilities that improve the scalability and performance of a Frame Relay network, increasing the density of VCs and improving response time.

As is also true of GTS, FRTS can eliminate bottlenecks in Frame Relay networks that have high-speed connections at the central site and low-speed connections at branch sites. You can configure rate enforcement—a peak rate configured to limit outbound traffic—to limit the rate at which data is sent on the VC at the central site.

Using FRTS, you can configure rate enforcement to either the CIR or some other defined value such as the excess information rate on a per-VC basis. The ability to allow the transmission speed used by the router to be controlled by criteria other than line speed (that is, by the CIR or the excess information rate) provides a mechanism for sharing media by multiple VCs. You can allocate bandwidth to each VC, creating a virtual time-division multiplexing (TDM) network.

You can also define PQ, CQ, and WFQ at the VC or subinterface level. Using these queueing methods allows for finer granularity in the prioritization and queueing of traffic, providing more control over the traffic flow on an individual VC. If you combine CQ with the per-VC queueing and rate enforcement capabilities, you enable Frame Relay VCs to carry multiple traffic types such as IP, SNA, and Internetwork Packet Exchange (IPX) with bandwidth guaranteed for each traffic type.

Using information contained in the BECN-tagged packets received from the network, FRTS can also dynamically throttle traffic. With BECN-based throttling, packets are held in the buffers of the router to reduce the data flow from the router into the Frame Relay network. The throttling is done on a per-VC basis and the transmission rate is adjusted based on the number of BECN-tagged packets received.

With the Cisco FRTS feature, you can integrate ATM ForeSight closed loop congestion control to actively adapt to downstream congestion conditions.

Derived Rates

In Frame Relay networks, BECNs and FECNs indicate congestion. BECN and FECN are specified by bits within a Frame Relay frame.

FECNs are generated when data is sent out a congested interface; they indicate to a DTE device that congestion was encountered. Traffic is marked with BECN if the queue for the opposite direction is deep enough to trigger FECNs at the current time.

BECNs notify the sender to decrease the transmission rate. If the traffic is one-way only (such as multicast traffic), there is no reverse traffic with BECNs to notify the sender to slow down. Thus, when a DTE device receives an FECN, it first determines if it is sending any data in return. If it is sending return data, this data will get marked with a BECN on its way to the other DTE device. However, if the DTE device is not sending any data, the DTE device can send a Q.922 TEST RESPONSE message with the BECN bit set.

When an interface configured with traffic shaping receives a BECN, it immediately decreases its maximum rate by a large amount. If, after several intervals, the interface has not received another BECN and traffic is waiting in the queue, the maximum rate increases slightly. The dynamically adjusted maximum rate is called the derived rate.

The derived rate will always be between the upper bound and the lower bound configured on the interface.

For information on configuring Frame Relay and FRTS, see the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.4T.

Restrictions

FRTS applies only to Frame Relay PVCs and switched virtual circuits (SVCs).

FRTS is not supported on the Cisco 7500 series router.