



Policing and Shaping Overview

Cisco IOS QoS offers two kinds of traffic regulation mechanisms: the rate-limiting feature of committed access rate (CAR) for policing traffic, and Generic Traffic Shaping (GTS) and Frame Relay Traffic Shaping (FRTS) for shaping traffic. 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” 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, CAR’s 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 uses a weighted fair queue to delay packets in order to shape the flow, and FRTS uses either a priority queue (PQ), a custom queue (CQ), or a first-in, first-out (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 with CAR and shaping with FRTS and GTS 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?
- Policing with CAR
- Traffic Shaping

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 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 capacity of the token bucket plus the time interval multiplied by the established rate at which tokens are placed in the bucket. It 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” earlier 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.

CAR is supported on these routers:

- Cisco 1000 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 4500 series
- Cisco 4700 series
- Cisco 7200 series
- Cisco 12000 series

Distributed CAR is supported on Cisco 7000 series routers with a Route Switch Processor (RSP)-based RSP7000 interface processor or Cisco 7500 series routers with a Versatile Interface Processor (VIP)-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**

Distributed Cisco Express Forwarding (dCEF) switching must be enabled on any interface that uses VIP-Distributed CAR, even when only output CAR is configured. For dCEF configuration information, refer to the *Cisco IOS Switching Services Configuration Guide*.

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.

This section explains these aspects of CAR rate limiting:

- 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 sufficient 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:

- Incoming interface
- All IP traffic
- IP Precedence (defined by a rate-limit access list)
- MAC address (defined by a rate-limit access list)
- IP access list (standard and extended)

CAR provides configurable actions, such as transmit, 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 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.

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 tokens in a token bucket are replenished at regular intervals, in accordance with the configured committed rate. The maximum number of tokens a bucket can ever 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 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, CAR's exceed action takes effect because sufficient 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).

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, CAR's exceed action 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.

We recommend 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 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 VIP-Distributed WFQ (DWFQ), if premium bandwidth assurances are required.

Traffic Shaping

Cisco IOS QoS software includes two types of traffic shaping: GTS and FRTS. Both traffic shaping methods are similar in implementation, though their command-line interfaces 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 there is enough credit 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 uses 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 two Cisco IOS QoS traffic shaping mechanisms. It includes these subsections:

- About Traffic Shaping
- Generic 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 data terminal equipment (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.)

- Configure traffic shaping 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 committed information rate (CIR)—that are still accepted by the Frame Relay switch but marked as discard eligible.

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 discard eligible (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 Excess Burst 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. For more information about the DE bit, see the chapter “Configuring Frame Relay and Frame Relay Traffic Shaping” in this book.

Differences Between GTS and FRTS

As mentioned, both GTS and FRTS are similar in implementation, sharing the same code and data structures, but they differ in regard to their command-line interfaces and the queue types they use.

Here are a couple ways in which GTS and FRTS differ:

- FRTS supports shaping on a per-DLCI basis; GTS is configurable per interface or subinterface.
- 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), CQ, PQ, or FIFO.

Table 8 summarizes these differences.

Table 8 Differences Between FRTS and GTS

	FRTS	GTS
Command-Line Interface	<ul style="list-style-type: none"> Classes of parameters Applies parameters to all virtual circuits (VCs) on an interface through inheritance mechanism No traffic group command 	<ul style="list-style-type: none"> Applies parameters per subinterface traffic group command supported
Queues Supported	<ul style="list-style-type: none"> WFQ, strict priority queue with WFQ, CQ, PQ, first come first served (FCFS) per VC 	<ul style="list-style-type: none"> WFQ per subinterface

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 happens:

- 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.
- 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

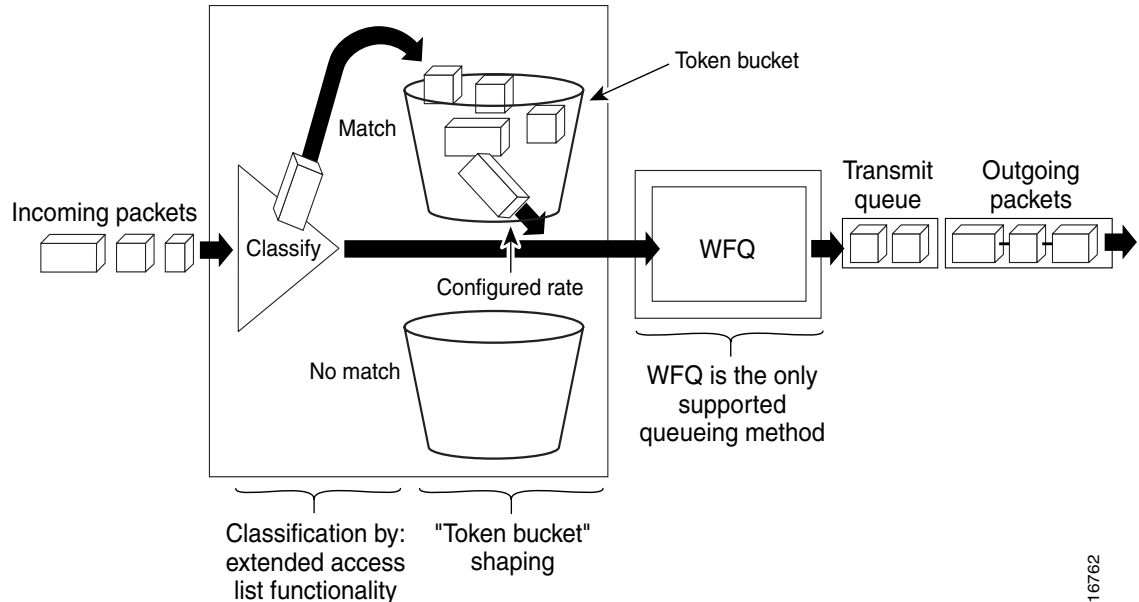
GTS shapes traffic by reducing outbound traffic flow to avoid congestion by constraining traffic to a particular bit rate using the token bucket mechanism. (See the section “What Is a Token Bucket?” earlier in this chapter.)

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 BECN signals, or set up simply to shape to a specified rate. GTS can also be configured on an ATM/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. Figure 9 shows how GTS works.

Figure 9 Generic Traffic Shaping



16762

Frame Relay Traffic Shaping

Cisco has long provided support for forward explicit congestion notification (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 virtual circuits 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 Cisco's 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.

Restrictions

FRTS applies only to Frame Relay PVCs and SVCs.