



Frame Relay Queueing and Fragmentation at the Interface

The Frame Relay Queueing and Fragmentation at the Interface feature introduces support for low-latency queueing (LLQ) and FRF.12 end-to-end fragmentation on a Frame Relay interface.

- [Finding Feature Information, page 1](#)
- [Prerequisites for Frame Relay Queueing and Fragmentation at the Interface, page 1](#)
- [Restrictions for Frame Relay Queueing and Fragmentation at the Interface, page 2](#)
- [Information About Frame Relay Queueing and Fragmentation at the Interface, page 2](#)
- [How to Configure Frame Relay Queueing and Fragmentation at the Interface, page 4](#)
- [Configuration Examples for Frame Relay Queueing and Fragmentation at the Interface, page 13](#)
- [Additional References, page 14](#)
- [Feature Information for Frame Relay Queueing and Fragmentation at the Interface, page 15](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Frame Relay Queueing and Fragmentation at the Interface

The tasks in this document assume that you know how to configure low-latency queueing and shaping service policies.

The following prerequisites are specific to the Cisco 7500 series:

- The Frame Relay Queueing and Fragmentation at the Interface feature is supported on VIP-based interfaces with VIP2-50 or higher.
- Distributed Cisco Express Forwarding (dCEF) must be enabled both globally and on the Frame Relay interface.

Restrictions for Frame Relay Queueing and Fragmentation at the Interface

- Interface fragmentation and Frame Relay traffic shaping cannot be configured at the same time.
- Interface fragmentation and class-based fragmentation cannot be configured at the same time.
- Frame Relay switched virtual circuits (SVCs) are not supported.
- Hierarchical shaping and multiple shapers are not supported.

Information About Frame Relay Queueing and Fragmentation at the Interface

The Frame Relay Queueing and Fragmentation at the Interface feature simplifies the configuration of low-latency, low-jitter quality of service (QoS) by enabling the queueing policy and fragmentation configured on the main interface to apply to all permanent virtual circuits (PVCs) and subinterfaces under that interface. Before the introduction of this feature, queueing and fragmentation had to be configured on each individual PVC. Subrate shaping can also be configured on the interface.

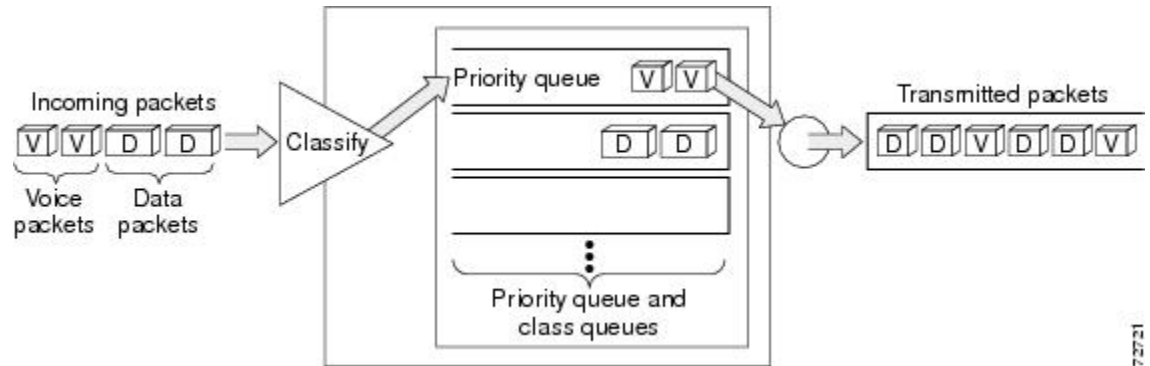
How Frame Relay Queueing and Fragmentation at the Interface Works

When FRF.12 end-to-end fragmentation is enabled on an interface, all PVCs on the main interface and its subinterfaces will have fragmentation enabled with the same configured fragment size. To maintain low latency and low jitter for high-priority traffic, the configured fragment size must be greater than the largest high-priority frames. This configuration will prevent high-priority traffic from being fragmented and queued behind lower-priority fragmented frames. If the size of a high-priority frame is larger than the configured fragment size, the high-priority frame will be fragmented. Local Management Interface (LMI) traffic will not be fragmented and is guaranteed its required bandwidth.

When a low-latency queueing policy map is applied to the interface, traffic through the interface is identified using class maps and is directed to the appropriate queue. Time-sensitive traffic such as voice should be classified as high priority and will be queued on the priority queue. Traffic that does not fall into one of the defined classes will be queued on the class-default queue. Frames from the priority queue and class queues are subject to fragmentation and interleaving. As long as the configured fragment size is larger than the high-priority frames, the priority queue traffic will not be fragmented and will be interleaved with fragmented

frames from other class queues. This approach provides the highest QoS transmission for priority queue traffic. The figure below illustrates the interface queueing and fragmentation process.

Figure 1: Frame Relay Queueing and Fragmentation at the Interface



Subrate shaping can also be applied to the interface, but interleaving of high-priority frames will not work when shaping is configured. If shaping is not configured, each PVC will be allowed to send bursts of traffic up to the physical line rate.

When shaping is configured and traffic exceeds the rate at which the shaper can send frames, the traffic is queued at the shaping layer using fair queueing. After a frame passes through the shaper, the frame is queued at the interface using whatever queueing method is configured. If shaping is not configured, then queueing occurs only at the interface.



Note For interleaving to work, both fragmentation and the low-latency queueing policy must be configured with shaping disabled.

The Frame Relay Queueing and Fragmentation at the Interface feature supports the following functionality:

- Voice over Frame Relay
- Weighted Random Early Detection
- Frame Relay payload compression



Note When payload compression and Frame Relay fragmentation are used at the same time, payload compression is always performed before fragmentation.

- IP header compression

Benefits of Frame Relay Queueing and Fragmentation at the Interface

Simple Configuration

The Frame Relay Queueing and Fragmentation at the Interface feature allows fragmentation, low-latency queueing, and subrate shaping to be configured on a Frame Relay interface queue. The fragmentation and queueing and shaping policy will apply to all PVCs and subinterfaces under the main interface, eliminating the need to configure QoS on each PVC individually.

Flexible Bandwidth

This feature allows PVCs to preserve the logical separation of traffic from different services while reducing bandwidth partitioning between PVCs. Each PVC can send bursts of traffic up to the interface shaping rate or, if shaping is not configured, the physical interface line rate.

How to Configure Frame Relay Queueing and Fragmentation at the Interface

Configuring Class Policy for the Priority Queue

To configure a policy map for the priority class, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** *class-name*
5. Router(config-pmap-c)# **priority** *bandwidth-kbps*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map</i> Example: Router(config) policy-map policy1	Specifies the name of the policy map to be created or modified. • Use this command to define the queueing policy for the priority queue.
Step 4	class <i>class-name</i> Example: Router(config-pmap)# class c1	Specifies the name of a class to be created and included in the service policy. • The class name that you specify in the policy map defines the characteristics for that class and its match criteria as configured using the class-map command.
Step 5	Router(config-pmap-c)# priority <i>bandwidth-kbps</i> Example: Router(config-pmap-c)# priority 30	Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.
Step 6	exit Example: Router(config-pmap-c)# exit	Exits the current configuration mode.

Configuring Class Policy for the Bandwidth Queues

To configure a policy map and create class policies that make up the service policy, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map *policy-map***
4. **class *class-name***
5. Router(config-pmap-c)# **bandwidth *bandwidth-kbps***
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map</i> Example: Router(config)# policy-map policy1	Specifies the name of the policy map to be created or modified. <ul style="list-style-type: none"> • Use this command to define the queueing policy for the priority queue. • The bandwidth queues and the priority queue use the same policy map.
Step 4	class <i>class-name</i> Example: Router(config-pmap)# class c1	Specifies the name of a class to be created and included in the service policy. <ul style="list-style-type: none"> • The class name that you specify in the policy map defines the characteristics for that class and its match criteria as configured using the class-map command.
Step 5	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i> Example: Router(config-pmap-c)# bandwidth 10	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.) <ul style="list-style-type: none"> • The sum of all bandwidth allocation on an interface cannot exceed 75 percent of the total available interface bandwidth. However, if you need to configure more than 75 percent of the interface bandwidth to classes, you can override the 75 percent maximum by using the max-reserved-bandwidth command.
Step 6	exit Example: Router(config-pmap-c)# exit	Exits the current configuration mode.

Configuring the Shaping Policy Using the Class-Default Class

In general, the class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, traffic that does not match any of the configured classes is given best-effort

treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

If you configure shaping in addition to queueing on the interface, use the class-default class to configure the shaping policy. The shaping policy will serve as the parent in a hierarchical traffic policy. The queueing policy will serve as the child policy. The class-default class is used for the shaping policy so that all traffic for the entire interface is shaped and a bandwidth-limited stream can be created.

To configure the shaping policy in the class-default class, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class class-default**
5. **shape** [**average** | **peak**] *mean-rate* [[*burst-size*] [*excess-burst-size*]]
6. **service-policy** *policy-map-name*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy-map <i>policy-map</i> Example: Router(config)# policy-map policy1	Specifies the name of the policy map to be created or modified. • Use this command to define the shaping policy.
Step 4	class class-default Example: Router(config-pmap)# class class-default	Specifies the default class so that you can configure or modify its policy.
Step 5	shape [average peak] <i>mean-rate</i> [[<i>burst-size</i>] [<i>excess-burst-size</i>]] Example: Router(config-pmap-c)# shape peak 10	(Optional) Shapes traffic to the indicated bit rate according to the algorithm specified.

	Command or Action	Purpose
Step 6	service-policy <i>policy-map-name</i> Example: Router(config-pmap-c) # service-policy policy1	Specifies the name of a policy map to be used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another). <ul style="list-style-type: none"> Use this command to attach the policy map for the priority queue (the child policy) to the shaping policy (the parent policy).
Step 7	exit Example: Router(config-pmap-c) # exit	Exits the current configuration mode.

Configuring Queueing and Fragmentation on the Frame Relay Interface

To configure low-latency queueing and FRF.12 end-to-end fragmentation on a Frame Relay interface, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type / number*
4. **encapsulation frame-relay**
5. **service-policy output** *policy-map-name*
6. **frame-relay fragment** *fragment-size end-to-end*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type / number</i> Example: Router(config)# interface fe 0/0	Configures an interface type and enters interface configuration mode.
Step 4	encapsulation frame-relay Example: Router(config-if)# encapsulation frame-relay	Enables Frame Relay encapsulation.
Step 5	service-policy output <i>policy-map-name</i> Example: Router(config-if)# service-policy output <i>policy1</i>	Attaches a policy map to an output interface, to be used as the service policy for that interface. <ul style="list-style-type: none"> • If shaping is being used, use this command to attach the shaping policy (which includes the nested queueing policy) to the interface. • Interleaving of high-priority frames will not work if shaping is configured on the interface. • If shaping is not being used, use this command to attach the queueing policy to the interface.
Step 6	frame-relay fragment <i>fragment-size</i> end-to-end Example: Router(config-if)# frame-relay fragment 100 end-to-end	Enables fragmentation of Frame Relay frames. <ul style="list-style-type: none"> • To maintain low latency and low jitter for priority queue traffic, configure the fragment size to be greater than the largest high-priority frame that would be expected.
Step 7	exit Example: Router(config-if)# exit	Exits the current configuration mode.

Verifying Frame Relay Queueing and Fragmentation at the Interface

To verify the configuration and performance of Frame Relay queueing and fragmentation at the interface, perform the following steps:

SUMMARY STEPS

1. Enter the **show running-config** command to verify the configuration.
2. Enter the **show policy-map interface** command to display low-latency queueing information, packet counters, and statistics for the policy map applied to the interface. Compare the values in the "packets" and the "pkts matched" counters; under normal circumstances, the "packets" counter is much larger than the "pkts matched" counter. If the values of the two counters are nearly equal, then the interface is receiving a large number of process-switched packets or is heavily congested.
3. Enter the **show interfaces serial** command to display information about the queueing strategy, priority queue interleaving, and type of fragmentation configured on the interface. You can determine whether the interface has reached a congestion condition and packets have been queued by looking at the "Conversations" fields. A nonzero value for "max active" counter shows whether any queues have been active. If the "active" counter is a nonzero value, you can use the **show queue** command to view the contents of the queues.

DETAILED STEPS

Step 1 Enter the **show running-config** command to verify the configuration.

Example:

```
Router# show running-config
Building configuration...
.
.
.
class-map match-all voice
  match ip precedence 5
!
!policy-map llq
  class voice
    priority 64
policy-map shaper
  class class-default
    shape peak 96000
    service-policy llq
!
!interface Serial1/1
  ip address 16.0.0.1 255.255.255.0
  encapsulation frame-relay
  service-policy output shaper
  frame-relay fragment 80 end-to-end
!
```

Step 2 Enter the **show policy-map interface** command to display low-latency queueing information, packet counters, and statistics for the policy map applied to the interface. Compare the values in the "packets" and the "pkts matched" counters; under normal circumstances, the "packets" counter is much larger than the "pkts matched" counter. If the values of the two counters are nearly equal, then the interface is receiving a large number of process-switched packets or is heavily congested.

The following sample output for the **show policy-map interface** command is based on the configuration in Step 1:

Example:

```
Router# show policy-map interface serial 1/1
Serial1/1
  Service-policy output:shaper
```

```

Class-map: class-default (match-any)
  12617 packets, 1321846 bytes
  5 minute offered rate 33000 bps, drop rate 0 bps
  Match: any
  Traffic Shaping
    Target/Average   Byte   Sustain   Excess   Interval   Increment
      Rate          Limit  bits/int  bits/int  (ms)       (bytes)
    192000/96000     1992   7968      7968      83         1992
  Adapt Queue      Packets  Bytes    Packets  Bytes    Shaping
  Active Depth      Delayed  Delayed  Active
    -      0      12586    1321540  0        0        no
Service-policy : llq
Class-map: voice (match-all)
  3146 packets, 283140 bytes
  5 minute offered rate 7000 bps, drop rate 0 bps
  Match: ip precedence 1
  Weighted Fair Queueing
    Strict Priority
    Output Queue: Conversation 24
    Bandwidth 64 (kbps) Burst 1600 (Bytes)
    (pkts matched/bytes matched) 0/0
    (total drops/bytes drops) 0/0
Class-map: class-default (match-any)
  9471 packets, 1038706 bytes
  5 minute offered rate 26000 bps
  Match: any

```

Step 3

Enter the **show interfaces serial** command to display information about the queueing strategy, priority queue interleaving, and type of fragmentation configured on the interface. You can determine whether the interface has reached a congestion condition and packets have been queued by looking at the "Conversations" fields. A nonzero value for "max active" counter shows whether any queues have been active. If the "active" counter is a nonzero value, you can use the **show queue** command to view the contents of the queues.

The following sample output for the **show interfaces serial** command is based on the configuration in Step 1:

Example:

```

Router# show interfaces serial 1/1
Serial1/1 is up, line protocol is up
  Hardware is M4T
  Internet address is 16.0.0.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 5/255, rxload 1/255
  Encapsulation FRAME-RELAY, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  LMI enq sent 40, LMI stat recvd 40, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  Fragmentation type: end-to-end, size 80, PQ interleaves 0
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:03, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:06:34
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 33000 bits/sec, 40 packets/sec
    40 packets input, 576 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    15929 packets output, 1668870 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets

```

```
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions      DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Monitoring and Maintaining Frame Relay Queueing and Fragmentation at the Interface

To monitor and maintain Frame Relay queueing and fragmentation at the interface, use the following commands in privileged EXEC mode:

SUMMARY STEPS

1. **debug frame-relay fragment** [*event* | *interface type / number dlci*]
2. **show frame-relay fragment** [*interface type / number [dlci]*]
3. **show interfaces serial number**
4. **show queue interface-type interface-number**
5. **show policy-map interface number** [*input* | *output*]

DETAILED STEPS

-
- | | |
|---------------|--|
| Step 1 | debug frame-relay fragment [<i>event</i> <i>interface type / number dlci</i>]
Displays information related to Frame Relay fragmentation on a PVC. |
| Step 2 | show frame-relay fragment [<i>interface type / number [dlci]</i>]
Displays information about Frame Relay fragmentation. |
| Step 3 | show interfaces serial number
Displays information about a serial interface. |
| Step 4 | show queue interface-type interface-number
Displays the contents of packets inside a queue for a particular interface. |
| Step 5 | show policy-map interface number [<i>input</i> <i>output</i>]
Displays the packet statistics of all classes that are configured for all service policies on the specified interface. |
-

Configuration Examples for Frame Relay Queueing and Fragmentation at the Interface

Example Frame Relay Queueing Shaping and Fragmentation at the Interface

The following example shows the configuration of a hierarchical policy for low-latency queueing, FRF.12 fragmentation, and shaping on serial interface 3/2. Note that traffic from the priority queue will not be interleaved with fragments from the class-default queue because shaping is configured.

```
class-map voice
  match access-group 101

policy-map llq
  class voice
    priority 64

policy-map shaper
  class class-default
    shape average 96000
    service-policy llq
  interface serial 3/2
    ip address 10.0.0.1 255.0.0.0
    encapsulation frame-relay
    bandwidth 128
    clock rate 128000
    service-policy output shaper
    frame-relay fragment 80 end-to-end

access-list 101 match ip any host 10.0.0.2
```

Example Frame Relay Queueing and Fragmentation at the Interface

The following example shows the configuration of low-latency queueing and FRF.12 fragmentation on serial interface 3/2. Because shaping is not being used, a hierarchical traffic policy is not needed and traffic from the priority queue will be interleaved with fragments from the other queues. Without shaping, the output rate of the interface is equal to the line rate or configured clock rate. In this example, the clock rate is 128,000 bps.

```
class-map voice
  match access-group 101

policy-map llq
  class voice
    priority 64
  class video
    bandwidth 32
  interface serial 3/2
    ip address 10.0.0.1 255.0.0.0
    encapsulation frame-relay
    bandwidth 128
    clock rate 128000
    service-policy output llq
    frame-relay fragment 80 end-to-end
  access-list 101 match ip any host 10.0.0.2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
WAN commands	<i>Cisco IOS Wide-Area Networking Command Reference</i>
Configuring Frame Relay	Configuring Frame Relay

Standards

Standard	Title
FRF.12	<i>Frame Relay Fragmentation Implementation Agreement</i> , December 1997

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Frame Relay Queueing and Fragmentation at the Interface

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Frame Relay Queueing and Fragmentation at the Interface

Feature Name	Releases	Feature Information
Frame Relay Queueing and Fragmentation at the Interface	12.2(11)S 12.2(13)T 15.0(1)S	<p>The Frame Relay Queueing and Fragmentation at the Interface feature introduces support for low-latency queueing (LLQ) and FRF.12 end-to-end fragmentation on a Frame Relay interface.</p> <p>The following commands were introduced or modified: frame-relay fragment end-to-end, show interfaces serial.</p>

