# Low Latency Queueing (LLQ) for IPsec Encryption Engines

**First Published: November 26, 2002**
**Last Updated: March 25, 2011**

The Low Latency Queueing (LLQ) for IPsec Encryption Engines feature helps reduce overall network latency and congestion by queueing priority designated traffic before it is processed by the crypto processing engine. This queueing guarantees a certain level of crypto engine processing time.

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 for LLQ for IPsec Encryption Engines" section on page 11.

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

## Contents



**Americas Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Prerequisites for LLQ for IPsec Encryption Engines

To use this feature, you should be familiar with the following:

- Access control lists
- Bandwidth management
- CBWFQ

# Restrictions for LLQ for IPsec Encryption Engines

- No per-tunnel QoS policy. An interface QoS policy represents all tunnels.
- Assume the same IP precedence/DSCP marking for inbound and outbound voice packets.
- Assume the IP precedence/DSCP marking for voice packets are done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume call admission control is enforced within the enterprise.
- No strict error checking when aggregate policy's bandwidth exceeds crypto engine bandwidth. Only a warning is displayed but configuration is allowed.
- Assume voice packets are either all encrypted or unencrypted.

# Information About LLQ for IPsec Encryption Engines

The following section provides information on the Low Latency Queueing (LLQ) for IPsec Encryption Engines feature:

- LLQ for IPsec Encryption Engines, page 2

## LLQ for IPsec Encryption Engines

Administrators can now use the Low Latency Queueing (LLQ) for IPsec Encryption Engines feature to prioritize voice and data traffic, which was previously only given equal status.

- Voice packets arriving on a router interface can be identified as priority and be directed into a priority packet inbound queue for crypto engine processing. This queue is called the priority queue. The crypto engine undertakes packet processing in a favorable ratio for voice packets. Voice packets are guaranteed a minimum processing bandwidth on the crypto engine. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.
- Data packets arriving at a router interface are directed into a data packet inbound queue for crypto engine processing. This queue is called the best effort queue.

# How to Configure LLQ for IPsec Encryption Engines

Perform the tasks described in this section to configure LLQ for IPsec Encryption Engines.

**Note** See the *Quality of Service Solutions Command Reference* to learn more about configuring server policies on interfaces.

- Defining Class Maps (required)
- Configuring Class Policy in the Policy Map (required)
- Attaching the Service Policy (required)
- Viewing the LLQ for IPsec Encryption Engines Configuration (optional)

## Defining Class Maps

The following steps are used to create a class map containing match criteria against which a packet is checked to determine if it belongs to a class:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match access-group** {*access-group* | **name** *access-group-name*}
   -or-
   **match input-interface** *interface-name*
   -or-
   **match protocol** *protocol*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **class-map** class-map-name<br><br>**Example:**<br>Router(config)# class-map voice | Specifies the name of the class map to be created. |
| **Step 4** | **match access-group** {*access-group* \| *name access-group-name*}<br>-or-<br>**match input-interface** *interface-name*<br>-or-<br>**match protocol** *protocol*<br><br>**Example:**<br>Router(config-cmap)# **match access-group 102** | • The **match access-group** command specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class.<br>• The **match input-interface** command specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.<br>• The **match protocol** command specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class. |

# Configuring Class Policy in the Policy Map

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the virtual circuit (VC) minus any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks described in the following sections.

- Configuring Class Policy for a Priority Queue (required)
- Configuring Class Policy Using a Specified Bandwidth (optional)
- Configuring the Class-Default Class Policy (optional)

## Configuring Class Policy for a Priority Queue

The following steps are used to configure a policy map and give priority to a class within the policy map:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** *class-name*

5. **priority** *bandwidth-kbps*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map*<br><br>**Example:**<br>`Router(config)# policy-map policy1` | Specifies the name of the policy map to be created or modified. |
| **Step 4** | **class** *class-name*<br><br>**Example:**<br>`Router(config-pmap)#class voice` | Specifies the name of a class to be created and included in the service policy. |
| **Step 5** | **priority** *bandwidth-kbps*<br><br>**Example:**<br>`Router(config-pmap-c)# priority 50` | Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class. |

## Configuring Class Policy Using a Specified Bandwidth

The following steps are used to configure a policy map and create class policies that make up the service policy. To configure more than one class in the same policy map, repeat Step 2 and Step 3.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class** *class-name*
5. **bandwidth** *bandwidth-kbps*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map*<br><br>**Example:**<br>`Router(config)# policy-map policy1` | Specifies the name of the policy map to be created or modified. |
| **Step 4** | **class** *class-name*<br><br>**Example:**<br>`Router(config-pmap)# class voice` | Specifies the name of a class to be created and included in the service policy. |
| **Step 5** | **bandwidth** *bandwidth-kbps*<br><br>**Example:**<br>`Router(config-pmap-c)# bandwidth 20` | 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.) |

## Configuring the Class-Default Class Policy

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, then 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.

The following steps are used to configure a policy map and the class-default class:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map*
4. **class class-default** *default-class-name*
5. **bandwidth** *bandwidth-kbps*
   -or-
   **fair-queue** [*number-of-dynamic-queues*]

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** policy-map<br><br>**Example:**<br>`Router(config)# policy-map policy-map` | Specifies the name of the policy map to be created or modified. |
| **Step 4** | **class class-default** *default-class-name*<br><br>**Example:**<br>`Router(config-pmap)# class class-default default-class-name` | Specifies the default class so that you can configure or modify its policy. |
| **Step 5** | **bandwidth** bandwidth-kbps<br>-or-<br>**fair-queue** [*number-of-dynamic-queues*]<br><br>**Example:**<br>`Router(config-pmap-c)# fair-queue` | Either the **bandwidth** or **fair-queue** command can be used for this step.<br>• The **bandwidth** command specifies the amount of bandwidth, in kbps, to be assigned to the class.<br>• The **fair-queue** command specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface. |

# Attaching the Service Policy

The following steps are used to attach a service policy to the output interface and enable LLQ for IPsec encryption engines.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy output** *policy-map*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br><br>**Example:**<br>`Router(config)# interface fastethernet0/0` | Specifies the interface using the LLQ for IPsec encryption engines. |
| **Step 4** | **service-policy output** *policy-map*<br><br>**Example:**<br>`Router(config-if)# service-policy output policy1` | Attaches the specified service policy map to the output interface and enables LLQ for IPsec encryption engines. |

# Viewing the LLQ for IPsec Encryption Engines Configuration

The following steps are used to view the contents of a specific policy map or all policy maps configured on an interface, and the LLQ for IPsec encryption engines:

### SUMMARY STEPS

1. **enable**
2. **show frame-relay pvc dlci**
3. **show policy-map interface** *interface-name* **dlci** *dlci-number*
4. **show crypto eng qos**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 1** | **show frame-relay pvc dlci**<br><br>**Example:**<br>`Router# show frame-relay pvc dlci` | Displays statistics about the PVC and the configuration of classes for the policy map on the specified data-link connection identifier (DLCI). |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **show policy-map interface** *interface-name*<br><br>**Example:**<br>`Router# show policy-map interface fastethernet0/0` | When LLQ is configured, displays the configuration of classes for all policy maps. |
| **Step 3** | **show policy-map interface** *interface-name* **dlci** *dlci-number*<br><br>**Example:**<br>`Router# show policy-map interface fastethernet0/0 dlci 100` | When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI. |
| **Step 4** | **show crypto eng qos**<br><br>**Example:**<br>`Router# show crypto eng qos` | Displays quality of service queueing statistics for LLQ for IPsec encryption engines. |

# Configuration Examples for LLQ for IPsec Encryption Engines

This section provides the following configuration example:

- LLQ for IPsec Encryption Engines: Example

## LLQ for IPsec Encryption Engines: Example

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
```

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the fas0/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config)# interface fas0/0
Router(config-if)# service-policy output policy1
```

# Additional References

## Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | *Cisco IOS Master Commands List, All Releases* |
| Security commands | *Cisco IOS Security Command Reference* |
| QoS Commands | *Cisco IOS Quality of Service Solutions Command Reference* |
| Weighted Fair Queueing | Configuring Weighted Fair Queueing feature module. |

## 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:<br>http://www.cisco.com/go/mibs |

## Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for LLQ for IPsec Encryption Engines

Table 1 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Note** Table 1 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.

*Table 1 Feature Information for Low Latency Queueing (LLQ) for IPsec Encryption Engines*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Feature Information for Low Latency Queueing (LLQ) for IPsec Encryption Engines | 12.2(13)T<br>12.2(14)S | The Low Latency Queueing (LLQ) for IPsec Encryption Engines feature helps reduce overall network latency and congestion by queueing priority designated traffic before it is processed by the crypto processing engine. This queueing guarantees a certain level of crypto engine processing time.<br>This feature was introduced in Cisco IOS Release 12.2(13)T.<br>This feature was integrated into Cisco IOS Release 12.2(14)S.<br>The following commands were introduced or modified: **show crypto eng qos.** |

# Glossary

**IKE**—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPsec). Before any IPsec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

**IPsec**—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPsec provides these security services at the IP layer. IPsec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

---