Low Latency Queueing for IPsec Encryption Engines

Last Updated: December 9, 2011

This feature module describes the LLQ for IPsec encryption engines feature and includes the following sections:

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 Table at the end of this document.

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.

Feature Overview

LLQ for IPsec encryption engines helps reduce packet latency by introducing the concept of queueing before crypto engines. Prior to this, the crypto processing engine gave data traffic and voice traffic equal status. Administrators now designate voice traffic as priority. 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. Voice packets arriving on a router interface are 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.

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

Benefits of the LLQ for IPSec Encryption Engines

The LLQ for IPsec encryption engines feature guarantees a certain level of crypto engine processing time for priority designated traffic.

Better Voice Performance

Voice packets can be identified as priority, allowing the crypto engine to guarantee a certain percentage of processing bandwidth. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.

Improved Latency and Jitters

Predictability is a critical component of network performance. The LLQ for IPsec encryption engines feature delivers network traffic predictability relating to VPN. With this feature disabled, an end user employing an IP phone over VPN might experience jitter or latency, both symptoms of overall network latency and congestion. With this feature enabled, these undesirable characteristics are dissipated.

Restrictions

- 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 that the IP precedence/DSCP marking for voice packets is done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume that 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 that voice packets are either all encrypted or unencrypted.

Related Documents

- Cisco IOS Quality of Service Solutions Command Reference
- "Applying QoS Features Using the MQC" module

Supported Standards MIBs and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

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

- Access control lists
- Bandwidth management
- CBWFQ

Configuration Tasks

To configure LLQ for IPsec encryption engines, perform the tasks described in the following sections.

Defining Class Maps

SUMMARY STEPS

1. Router(config)# class-map *class-map-name*
2. Do one of the following:
   - Router(config-cmap)# match access-group {access-group | name access-group-name}
   -
   -
   -

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# class-map *class-map-name*	Specifies the name of the class map to be created.

Command or Action	Purpose

Step 2 Do one of the following:
• Router(config-cmap)# match access-group {access-group | name access-group-name}
•
•
•
 	
Example:
`Router(config-cmap)# match input-interface interface-name`
 	
Example:
 	
Example:
`or`
 	
Example:
 	
Example:
`Router(config-cmap)# match protocol protocol`
	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. 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. 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

To configure a policy map and create class policies that make up the service policy, begin with the policy-map command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- priority
- bandwidth
- queue-limit or random-detect
- fair-queue (for class-default class only)

For each class that you define, you can use one or more of the commands listed to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

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.

Configuring Class Policy for a Priority Queue

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class *class-name*
3. Router(config-pmap-c)# priority *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class *class-name*	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# priority *bandwidth-kbps*	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

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class *class-name*
3. Router(config-pmap-c)# bandwidth *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class *class-name*	Specifies the name of a class to be created and included in the service policy.
Step 3	Router(config-pmap-c)# bandwidth *bandwidth-kbps*	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.) **Note** To configure more than one class in the same policy map, repeat Configuring Class Policy Using a Specified Bandwidth, page 5 and Configuring Class Policy Using a Specified Bandwidth, page 5.

Configuring the Class-Default Class Policy

SUMMARY STEPS

1. Router(config)# policy-map *policy-map*
2. Router(config-cmap)# class class-default default-class-name
3. Router(config-pmap-c)# bandwidth *bandwidth-kbps*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# policy-map *policy-map*	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-cmap)# class class-default default-class-name	Specifies the default class so that you can configure or modify its policy. **Note** 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.
Step 3	Router(config-pmap-c)# bandwidth *bandwidth-kbps* **Example:** **Example:** or **Example:** **Example:** `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, to be assigned to the class. 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

SUMMARY STEPS

1. Router(config)# interface *type number*
2. Router(config-if)# service-policy output *policy-map*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# interface *type number*	Specifies the interface using the LLQ for IPsec encryption engines.
Step 2	Router(config-if)# service-policy output *policy-map*	Attaches the specified service policy map to the output interface and enables LLQ for IPsec encryption engines.

Verifying Configuration of Policy Maps and Their Classes

SUMMARY STEPS

1. Router# show frame-relay pvc dlci
2. Router# show policy-map interface type number
3. Router# show policy-map interface interface-name dlci dlci

DETAILED STEPS

	Command or Action	Purpose
Step 1	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).
Step 2	Router# show policy-map interface type number	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 3	Router# show policy-map interface interface-name dlci dlci	When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining LLQ for IPsec Encryption Engines

SUMMARY STEPS

1. Router# show crypto eng qos

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# show crypto eng qos	Displays quality of service queueing statistics for LLQ for IPsec encryption engines.

Configuration Examples

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-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router (config-cmap-c)# exit
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-cmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config-cmap-c)# exit
Router(config-cmap)# exit
Router(config)# interface fastethernet0/0/0
Router(config-if)# service-policy output policy1
```