



# CHAPTER 9

## Configuring QoS on the GGSN

---

This chapter describes how to configure quality of service (QoS) functions to differentiate traffic flow through the gateway GPRS support node (GGSN) on the Cisco 7200 platform and on the Cisco MWAM in the Catalyst 6500 / Cisco 7609 platform.

For a complete description of the GGSN commands in this chapter, refer to the *Cisco GGSN Release 6.0 Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

This chapter includes the following sections:

- [Overview of QoS Support on the GGSN, page 9-1](#)
- [Configuring GPRS QoS on the GGSN, page 9-2](#)
- [Configuring UMTS QoS on the GGSN, page 9-12](#)
- [Configuring the GGSN Default QoS as Requested QoS, page 9-25](#)
- [Configuring Call Admission Control on the GGSN, page 9-25](#)
- [Configuring Per-PDP Policing, page 9-30](#)
- [Monitoring and Maintaining QoS on the GGSN, page 9-34](#)
- [Configuration Examples, page 9-42](#)

## Overview of QoS Support on the GGSN

The GGSN Release 4.0 and later software supports 2.5G general packet radio service (GPRS) QoS (as defined in global system for mobile communication [GSM] standards 02.60, 03.60, and 04.08) and 3G Universal Mobile Telecommunication System (UMTS) QoS. Each GPRS/UMTS packet data protocol (PDP) context request contains either a GPRS QoS profile or UMTS QoS profile.



### Note

---

GGSN on the Catalyst 6500 / Cisco 7609 platform supports UMTS QoS only.

---

The implementation of QoS support in the GPRS/UMTS public LAN mobile network (PLMN) varies by the service provider and the available resources in the network. The GSM standards define the GPRS QoS classes that can be requested by a GPRS mobile station (MS). The 3GPP standards define the UMTS QoS classes that can be defined by a UMTS MS. However, the resulting QoS is negotiated and variable within the GPRS/UMTS network backbone according to the implementations of the service provider.

### GPRS QoS

The GPRS QoS profiles are considered a single parameter that defines the following data transfer class attributes according to the GSM standard:

- Precedence class
- Delay class
- Reliability class
- Peak throughput class
- Mean throughput class

### UMTS QoS

To manage different level of QoS, UMTS has defined the four QoS traffic classes based on delay, jitter, bandwidth, and reliability factors:

- Conversational
- Streaming
- Interactive
- Background

GGSN Release 4.0 and later delivers end-to-end UMTS QoS by implementing it using the Cisco IOS QoS differentiated services (Diffserv).

This chapter describes the QoS support that the GGSN provides for the GPRS and UMTS QoS classes.

## Configuring GPRS QoS on the GGSN

GGSN Release 3.0 and later support two methods of GPRS QoS support, only one of which can be activated globally on the GGSN for all GPRS traffic processing:

- Canonical QoS—Maps GPRS QoS classes to canonical QoS classes.
- Delay QoS—Maps GPRS QoS classes to delay QoS classes.

## Configuring Canonical QoS on the GGSN

This section describes how to configure the canonical QoS method on the GGSN. It includes the following topics:

- [Overview of Canonical QoS, page 9-2](#)
- [Canonical QoS Configuration Task List, page 9-4](#)
- [Verifying the Canonical QoS Configuration, page 9-7](#)

### Overview of Canonical QoS

GGSN Release 1.2 and later support the canonical QoS method. The canonical QoS method on the GGSN supports three levels of QoS classification: best effort, normal, and premium.

When you enable canonical QoS, the GGSN examines the QoS profile in PDP context requests for three of the five GPRS QoS classes (delay, precedence, and mean throughput). Based on combinations of values for those GPRS QoS class attributes, the GGSN maps the resulting QoS class to best effort, normal, or premium classifications.

Table 9-1 shows how the GGSN maps the different combinations of GPRS QoS class attributes within a PDP context request to a particular canonical QoS class, when canonical QoS is enabled on the GGSN. For example, if the QoS profile of a PDP context request specifies the best-effort delay class, and any class of precedence and mean throughput, then the GGSN classifies that PDP context as the best-effort canonical class.

**Table 9-1 GPRS QoS Class Attribute Combinations Mapped to GGSN Canonical QoS Classes**

Delay Class	Precedence Class	Mean Throughput Class	GGSN Canonical QoS Class
Best effort	Any	Any	Best effort
1, 2, or 3	Low	Any	Best effort
1, 2, or 3	Any	Best effort	Best effort
1, 2, or 3	Normal	Specified	Normal
1, 2, or 3	High	Specified	Premium

Once you have enabled the canonical QoS method on the GGSN, you can map the canonical QoS classes to IP type of service (ToS) categories. IP ToS mappings allow the GGSN to support differentiated services according to RFC 2475, *Architecture for Differentiated Services Framework*. For more information, see the “[Mapping Canonical QoS Classes to IP ToS Precedence](#)” section on page 9-4. For more information about configuring the GGSN for differentiated services support, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide* and *Command Reference* publications.

For the canonical QoS method, the GGSN sets aside a configurable amount of resources to be used for QoS processing. The GGSN allocates a portion of this total available resource for canonical QoS upon PDP context activation, based on the QoS class to which the PDP context has been assigned. Typically, the GGSN uses more of its resources in support of the higher canonical QoS classes. As of GGSN Release 3.0, the total default amount of resources set aside by the GGSN for canonical QoS support is 3,145,728,000 bits per second. You can modify this value using the **gprs canonical-qos gsn-resource-factor** command. For more information, see the “[Configuring Total GGSN Resources for Canonical QoS Support](#)” section on page 9-5.

When a request for a user session comes in as a PDP context activation request, the GGSN determines whether the requested QoS for the session packets can be handled based on the amount of the **gprs canonical-qos gsn-resource-factor** that is available on the GGSN. Based on this determination, one of the following occurs:

- If the GGSN can provide the requested QoS, then the GGSN maintains that level of service.
- If the GGSN cannot provide the requested QoS, then the GGSN either lowers the QoS for the PDP context or it rejects the PDP context request.

## Canonical QoS Configuration Task List

To implement the canonical QoS method on the GGSN, you must enable the function. From there, you can modify the canonical QoS options to support your network environment.

To configure canonical QoS on the GGSN, perform the following tasks:

- [Enabling Canonical QoS on the GGSN, page 9-4](#) (Required)
- [Mapping Canonical QoS Classes to IP ToS Precedence, page 9-4](#) (Optional)
- [Customizing the Canonical QoS Configuration, page 9-5](#) (Optional)

### Enabling Canonical QoS on the GGSN

Canonical QoS is not automatically enabled by the GGSN. To enable canonical QoS on the GGSN, use the following command in global configuration mode:

Command	Purpose
Router(config)# <code>gprs qos map canonical-qos</code>	Enables mapping of GPRS QoS categories to a canonical QoS method that includes best effort, normal, and premium canonical QoS classes.

### Mapping Canonical QoS Classes to IP ToS Precedence

Once you have enabled the canonical QoS method on the GGSN, you can map the canonical QoS classes to IP ToS precedence. You can specify a mapping from the best effort, normal and premium canonical QoS categories to the ToS precedence bits (between 0 and 7, although 6 and 7 are not typically used). ToS precedence is reported in the IP header for packets transmitted over the Gn (GTP tunnel) and Gi interfaces.

All of the keyword arguments for the command are optional. However, if you specify a value for the **normal** argument, you must specify a value for the **premium** argument. And if you specify a value for the **best-effort** argument, then you must specify a value for both the **premium** and the **normal** arguments. The default ToS precedence values are 2 for premium, 1 for normal, and 0 for best effort.

The ToS precedence classes are defined as follows:

- 0 Routine
- 1 Priority
- 2 Immediate
- 3 Flash
- 4 Flash Override
- 5 Critical ECP
- 6 Internetwork Control
- 7 Network Control



#### Note

The GTP signaling messages should always have the highest precedence in the GPRS network to help ensure the expedited delivery of those control messages. You can configure the ToS for GTP signaling messages using the `gprs gtp map signalling tos` command. The default value is 5.

To map canonical QoS classes to IP ToS precedence bits, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs canonical-qos map tos</b> [ <b>premium</b> <i>tos-value</i> [ <b>normal</b> <i>tos-value</i> [ <b>best-effort</b> <i>tos-value</i> ]]]	(Optional) Specifies a QoS mapping from the canonical QoS classes to an IP ToS precedence value, where <i>tos-value</i> is an integer between 0 and 7 (values of 6 and 7 are not typically used).

### Customizing the Canonical QoS Configuration

This section describes some of the options that you can configure on the GGSN to further customize the default canonical QoS configuration.

Once you enable canonical QoS, the GGSN establishes default values for the allocation of GGSN resources to support canonical QoS processing. However, you most likely will want to modify the defaults based upon the GPRS traffic patterns and QoS profiles in use on your network.

This section includes the following topics:

- [Configuring Total GGSN Resources for Canonical QoS Support, page 9-5](#)
- [Configuring GGSN Resources for the Best Effort Class, page 9-6](#)
- [Configuring the Deviation Factor for the Premium Class, page 9-6](#)

#### Configuring Total GGSN Resources for Canonical QoS Support

For the canonical QoS method, the GGSN sets aside a configurable amount of resource that it uses for QoS processing. The GGSN allocates a portion of this total available resource for canonical QoS upon activating a PDP context, based on the QoS class that the GGSN assigns to the PDP context. Typically, the GGSN uses more of its resources in support of the higher canonical QoS classes.

The GGSN allocates a portion of the total resource, and deducts that portion from the total available resource on the GGSN, according to the canonical QoS classes as follows:

- **Best effort**—The GGSN allocates the amount of resource specified by the **gprs canonical-qos best-effort bandwidth-factor** command for a best-effort PDP context. The default is 10 bps.
- **Normal**—The GGSN allocates the amount of resource according to the mean throughput value requested in the PDP context.
- **Premium**—The GGSN allocates the amount of resource according to a calculation of the minimum value of the requested peak throughput and mean throughput in the PDP context, along with a configurable deviation factor. You can configure the deviation factor using the **gprs canonical-qos premium mean-throughput-deviation** command.

Once the GGSN allocates resources for a PDP context, it does not make the resource available again until it deletes the PDP context or it receives an update request that requires a change to the allocated resource.

The total default amount of resource set aside by the GGSN for canonical QoS support is 3,145,728,000 bits per second. The default value for this command was chosen to support 10,000 PDP contexts with a premium QoS class. If you require greater throughput for the GPRS data on your network, increase the resource factor value. However, be aware that if you select a value that is too high, you might exceed the actual processing capacity of the GGSN.

To configure the total GGSN resource for canonical QoS support, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs canonical-qos gsn-resource-factor</b> <i>resource-factor</i>	(Optional) Specifies the total amount of resource that the GGSN uses to provide QoS service levels to mobile users. The default is 3,145,728,000 bits per second.

#### Configuring GGSN Resources for the Best Effort Class

You can also configure resources to be reserved for best-effort QoS classes on the GGSN by using the **gprs canonical-qos best-effort bandwidth-factor** command. This command specifies an average bandwidth that is expected to be used by best-effort QoS class mobile sessions. The default value is 10 bps. If you observe that users accessing the GGSN are using a higher average bandwidth, then you should increase the bandwidth value.

To modify the bandwidth factor for the best-effort canonical QoS class, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs canonical-qos best-effort bandwidth-factor</b> <i>bandwidth-factor</i>	(Optional) Specifies the bandwidth factor to be applied to the canonical best-effort QoS class. The default value is 10 bps.

#### Configuring the Deviation Factor for the Premium Class

The GGSN uses the minimum value of the requested peak throughput and mean throughput in the PDP context, along with a configurable deviation factor to determine how much resource to allocate for the premium QoS class.

You can configure a deviation factor (factor/1000) to adjust the result of the calculation that the GGSN uses to determine the amount of data throughput to allocate for premium QoS support.

The GGSN bases its calculation on the following formula, which includes the throughput deviation factor:

$$EB = \text{Min}[p, m + a(p - m)]$$

Where

- EB = the effective bandwidth
- p = peak throughput from the GPRS QoS profile in the PDP context request
- m = mean throughput from the GPRS QoS profile in the PDP context request
- a = the deviation factor, a, divided by 1000 (a/1000)

To configure the deviation factor that the GGSN uses for calculation of premium canonical QoS support, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs canonical-qos premium mean-throughput-deviation</b> <i>deviation-factor</i>	(Optional) Specifies a mean throughput deviation factor that the GGSN uses to calculate the allowable data throughput for the premium QoS class. The default is 100.

## Verifying the Canonical QoS Configuration

To verify your canonical QoS configuration, use the **show running-config** command and observe the canonical QoS parameters, as shown in bold in the following example:

```
Router# show running-config
Building configuration...

Current configuration : 3521 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
. . .

ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.100.3.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface Ethernet1/0
 description Gi interface to gppt.cisco.com
 ip address 10.8.8.6 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 ip address 10.15.15.10 255.255.255.0
 duplex half
!
interface loopback 1
 ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
 access-point 1
```

```

access-mode non-transparent
access-point-name www.pdn1.com
aaa-group authentication foo
!
access-point 2
access-mode non-transparent
access-point-name www.pdn2.com
!
access-point 4
access-point-name www.pdn4.com
aaa-accounting enable
aaa-group accounting foo1
!
access-point 5
access-point-name www.pdn5.com
!
gprs maximum-pdp-context-allowed 90000
gprs qos map canonical-qos
gprs canonical-qos gsn-resource-factor 4294967295
gprs canonical-qos best-effort bandwidth-factor 10000
gprs canonical-qos premium mean-throughput-deviation 500
gprs canonical-qos map tos premium 3 normal 2 best-effort 1
gprs gtp path-echo-interval 30
gprs default aaa-group authentication foo2
gprs default aaa-group accounting foo3
gprs default charging-gateway 10.15.15.1
!
gprs memory threshold 512
!
. . .
!
end

```

## Configuring Delay QoS on the GGSN

This section describes how to configure the delay QoS method on the GGSN. It includes the following topics:

- [Overview of Delay QoS, page 9-8](#)
- [Delay QoS Configuration Task List, page 9-9](#)
- [Verifying the Delay QoS Configuration, page 9-10](#)
- [Delay QoS Configuration Example, page 9-44](#)

## Overview of Delay QoS

GGSN Release 3.0 and later support the delay QoS method. The delay QoS method on the GGSN supports four levels of QoS classification: class 1, class 2, class 3 and best effort.

When you enable delay QoS, the GGSN examines the QoS profile in PDP context requests for three of the five GPRS QoS classes (delay, precedence, and mean throughput). Based on combinations of values for those GPRS QoS class attributes, the GGSN maps the resulting delay QoS class to class 1, class 2, class 3, or best-effort categories.



Table 9-2 shows how the GGSN maps the different combinations of GPRS QoS class attributes within a PDP context request to a particular delay QoS class, when delay QoS is enabled on the GGSN. For example, if the QoS profile of a PDP context request specifies the best-effort delay class, and any class of precedence and mean throughput, then the GGSN classifies that PDP context as the best-effort delay class.

**Table 9-2 GPRS QoS Class Attribute Combinations Mapped to GGSN Delay QoS Classes**

Delay Class	Precedence Class	Mean Throughput Class	GGSN Delay QoS Class
Undefined	Any	Any	Best effort
Best effort	Any	Any	Best effort
Class 1	Any	Any	Class 1
Class 2	Any	Any	Class 2
Class 3	Any	Any	Class 3

## Delay QoS Configuration Task List

To implement the delay QoS method on the GGSN, you must enable the function. From there, you can modify the delay QoS options to support your network environment.

To configure delay QoS on the GGSN, perform the following tasks:

- [Enabling Delay QoS on the GGSN, page 9-9](#) (Required)
- [Mapping Delay QoS Classes to IP ToS Precedence, page 9-9](#) (Optional)

### Enabling Delay QoS on the GGSN

Delay QoS is not automatically enabled by the GGSN. To enable delay QoS on the GGSN, use the following command in global configuration mode:

Command	Purpose
Router(config)# <code>gprs qos map delay</code>	Enables mapping of GPRS QoS categories to a delay QoS method that includes the class 1, class 2, class 3, and best effort classes.

### Mapping Delay QoS Classes to IP ToS Precedence

Once you have enabled the delay QoS method on the GGSN, you can map the delay QoS classes to IP ToS precedence. You can specify a mapping from the class 1, class 2, class 3, or class best effort delay QoS categories to the ToS precedence bits (between 0 and 7, although 6 and 7 are not typically used). ToS precedence is reported in the IP header for packets transmitted over the Gn (GTP tunnel) and Gi interfaces.

The **class2**, **class3** and **class-best-effort** keyword arguments are optional. However, if you specify a value for the **class3** argument, you must specify a value for the **class2** argument. And, if you specify a value for the **class-best-effort** argument, then you must specify a value for both the **class2** and the **class3** arguments.

The ToS precedence classes are defined as follows:

- 0 Routine
- 1 Priority
- 2 Immediate
- 3 Flash
- 4 Flash Override
- 5 Critical ECP
- 6 Internetwork Control
- 7 Network Control



**Note**

The GTP signaling messages should always have the highest precedence in the GPRS network to help ensure the expedited delivery of those control messages. You can configure the ToS for GTP signaling messages by using the **gprs gtp map signalling tos** command. The default value is 5.

To map delay QoS classes to IP ToS precedence bits, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs delay-qos map tos class1</b> <i>tos-value [class2 tos-value [class3 tos-value</i> <i>[class-best-effort tos-value]]]</i>	(Optional) Specifies a QoS mapping from the delay QoS classes to an IP ToS precedence value, where <i>tos-value</i> is an integer between 0 and 5 (values of 6 and 7 are not typically used).

## Verifying the Delay QoS Configuration

To verify your delay QoS configuration, use the **show running-config** command and observe the delay QoS parameters, as shown in bold in the following example:

```
Router# show running-config
Building configuration...

Current configuration : 3521 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
. . .

ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
ip address 10.100.3.4 255.255.255.255
!
```

```

interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface Ethernet1/0
 description Gi interface to gppt.cisco.com
 ip address 10.8.8.6 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 ip address 10.15.15.10 255.255.255.0
 duplex half
!
interface loopback 1
 ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
 access-point 1
  access-mode non-transparent
  access-point-name www.pdn1.com
  aaa-group authentication foo
!
 access-point 2
  access-mode non-transparent
  access-point-name www.pdn2.com
!
 access-point 4
  access-point-name www.pdn4.com
  aaa-accounting enable
  aaa-group accounting foo1
!
 access-point 5
  access-point-name www.pdn5.com
!
gprs maximum-pdp-context-allowed 45000
gprs qos map delay
gprs delay-qos map tos class1 4 class2 3 class3 2 class-best-effort 1
gprs gtp path-echo-interval 30
gprs default aaa-group authentication foo2
gprs default aaa-group accounting foo3
gprs default charging-gateway 10.15.15.1
!
gprs memory threshold 512
!
. . .
!
end

```

# Configuring UMTS QoS on the GGSN

This section describes how to configure the UMTS QoS on the GGSN. It includes the following topics:

- [Overview of UMTS QoS, page 9-12](#)
- [Configuring UMTS QoS Task Lists, page 9-13](#)
- [Enabling UMTS QoS Mapping on the GGSN, page 9-14](#)
- [Mapping UMTS QoS Traffic Classes to a DiffServ PHB Group, page 9-14](#)
- [Assigning a DSCP to a DiffServ PHB Group, page 9-15](#)
- [Configuring the DSCP in the Subscriber Datagram, page 9-17](#)
- [Configuring the Catalyst 6500 / Cisco 7609 Platform GGSN UMTS QoS Requirements, page 9-18](#)
- [Verifying the UMTS QoS Configuration, page 9-21](#)

## Overview of UMTS QoS

3GPP standards define four QoS traffic classes based on delay, jitter, bandwidth, and reliability for UMTS. [Table 9-3](#) describes these UMTS traffic classes and their characteristics, applications, and the mapped Cisco IOS QoS Diffserv class.

**Table 9-3 UMTS Traffic Classes**

Traffic Class	Conversational (Real Time)	Streaming (Real Time)	Interactive (Best Effort)	Background (Best Effort)
<b>Characteristics</b>	Preserve time relation (variation) between information entities of the stream.  Conversational pattern, therefore, very low delay and jitter.	Preserve time relation (variation) between information entities of the stream.  Delay and jitter requirements are not as strict as with the conversational class.	Request/response pattern.  Retransmission of payload content in-route.	Destination is not expecting the data with a stringent time.  Retransmission of payload content in-route might occur.
<b>Example Applications</b>	Voice over IP	Streaming audio and video	Web browsing	Downloading email
<b>Diffserv Class / Map to DSCP</b>	Expedited Forwarding Class	Assured Forwarding 2 Class	Assured Forwarding 3 Class	Best Effort

GGSN Release 4.0 and later support end-to-end UMTS QoS by implementing it using the Cisco IOS Differentiated Services (DiffServ) model. The DiffServ model is a multiple-service model that can satisfy differing QoS requirements. With DiffServ, the network tries to deliver a particular kind of service based on the QoS specified by each packet. This specification can occur in different ways, for example, using the 6-bit differentiated services code point (DSCP) setting in IP packets or source and destination addresses. The network uses the QoS specification to classify, mark, shape, and police traffic, and to perform intelligent queueing.

For complete information on Cisco IOS QoS and the DiffServ service model, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.

## Configuring UMTS QoS Task Lists

To implement the UMTS QoS method on a GGSN, you must first enable the function. From there, you can modify the UMTS QoS options to support your network needs.

### Configuring GGSN UMTS QoS on the Cisco 7200 Platform Task List

If configuring UMTS QoS on a GGSN on the Cisco 7200 platform, perform the following tasks:

- [Enabling UMTS QoS Mapping on the GGSN, page 9-14](#) (Required)
- [Mapping UMTS QoS Traffic Classes to a DiffServ PHB Group, page 9-14](#) (Optional)
- [Assigning a DSCP to a DiffServ PHB Group, page 9-15](#) (Optional)
- [Configuring the DSCP in the Subscriber Datagram, page 9-17](#) (Optional)
- [Configuring Call Admission Control on the GGSN, page 9-25](#) (Optional)
- [Verifying the UMTS QoS Configuration, page 9-21](#)

### Configuring GGSN UMTS QoS on the Cisco 6500 / Cisco 7609 Platform Task List

If configuring UMTS QoS on a GGSN on the Catalyst 6500/Cisco 7600 platform, perform the following tasks:

- [Enabling UMTS QoS Mapping on the GGSN, page 9-14](#) (Required)
- [Mapping UMTS QoS Traffic Classes to a DiffServ PHB Group, page 9-14](#) (Optional)
- [Assigning a DSCP to a DiffServ PHB Group, page 9-15](#) (Optional)
- [Configuring the DSCP in the Subscriber Datagram, page 9-17](#) (Optional)
- [Configuring the Catalyst 6500 / Cisco 7609 Platform GGSN UMTS QoS Requirements, page 9-18](#) (Required)
- [Configuring Call Admission Control on the GGSN, page 9-25](#) (Optional)
- [Verifying the UMTS QoS Configuration, page 9-21](#)

## Enabling UMTS QoS Mapping on the GGSN

By default, UMTS QoS is not enabled on the GGSN. To enable UMTS QoS on the GGSN, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>gprs qos map umts</b>	Enables UMTS QoS mapping on the GGSN.

## Mapping UMTS QoS Traffic Classes to a DiffServ PHB Group

Before you can specify a QoS mapping from the UMTS QoS traffic classes to a DiffServ per-hop behavior (PHB) group, you must enable UMTS QoS mapping using the **gprs qos map umts** global configuration command.

The default mapping values for UMTS QoS traffic classes are as follows:

- Conversational traffic class to the ef-class DiffServ PHB group
- Streaming traffic class to the af2-class DiffServ PHB group
- Interactive traffic class to the af3-class DiffServ PHB group
- Background traffic class to the best-effort DiffServ PHB group

If you wish to use mapping values other than these defaults, you can use the **gprs umts-qos map traffic-class** command to map a UMTS traffic class to another DiffServ PHB group.



### Note

To successfully map UMTS QoS traffic classes to a DiffServ PHB, the class maps must be configured using the **class map** and **match ip dscp** Cisco IOS software commands. For more information about configuring class maps, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.

To map a UMTS traffic class to a DiffServ PHB group, use the following command in global configuration mode:

Command	Purpose
<pre>Router(config)# <b>gprs umts-qos map traffic-class</b> traffic-class <i>diffserv-phb-group</i></pre>	<p>Enables mapping of UMTS QoS traffic classes to a DiffServ PHB, where the UMTS traffic classes are:</p> <ul style="list-style-type: none"> <li>• signalling</li> <li>• conversational</li> <li>• streaming</li> <li>• interactive</li> <li>• background</li> </ul> <p>and the DiffServ PHB groups are:</p> <ul style="list-style-type: none"> <li>• signalling-class</li> <li>• ef-class</li> <li>• af1-class</li> <li>• af2-class</li> <li>• af3-class</li> <li>• af4-class</li> <li>• best-effort</li> </ul>

## Assigning a DSCP to a DiffServ PHB Group

By default, the default differentiated services code point (DSCP) value associated with a PHB class is used. [Table 9-4](#) lists the default DSCP values for each PHB group.

**Table 9-4** Default DSCP Values for PHB Groups

PHB Group	DSCP Value
EF	101110
AF11	001010
AF12	001100
AF13	001110
AF21	010010
AF22	010100
AF23	010110
AF31	011010
AF32	011100
AF33	011110
AF41	100010
AF42	100100

**Table 9-4** Default DSCP Values for PHB Groups (continued)

PHB Group	DSCP Value
AF43	100110
Best Effort	000000

However, you can assign a DSCP to PHB groups.

For the Assured Forwarding (AF) PHB group, you can specify up to three DSCPs for each drop precedence. The signalling, EF, and best-effort classes do not have drop precedence, so only the first DSCP value is used. If you enter a value for the *dscp2* or *dscp3* arguments for these classes, it is ignored.

**Note**

Drop precedence indicates the order in which a packet will be dropped when there is congestion on the network.

**Note**

To successfully map UMTS QoS traffic classes to a DiffServ PHB and assign a DSCP value to a DiffServ PHB group, the class maps must be configured using the **class map** and **match ip dscp** commands. For more information about configuring class maps, see *Cisco IOS Quality of Service Solutions Configuration Guide* and *Cisco IOS Quality of Service Solutions Command Reference*.

**Note**

By default, signalling class is assigned to CS5 (101000), which is the equivalent of IP precedence 5.



To assign a DSCP value to a DiffServ PHB group, use the following command in global configuration mode:

Command	Purpose
<pre>Router(config)# <b>gprs umts-qos map diffserv-phb</b> diffserv-phb-group [dscp1] [dscp2] [dscp3]</pre>	<p>Assigns a DSCP to a DiffServ PHB group where the DiffServ PHB groups are:</p> <ul style="list-style-type: none"> <li>• signalling</li> <li>• ef-class</li> <li>• af1-class</li> <li>• af2-class</li> <li>• af3-class</li> <li>• af4-class</li> <li>• best-effort</li> </ul> <p>and the DSCPs are:</p> <ul style="list-style-type: none"> <li>• dscp1—Required for all classes. Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 1.</li> <li>• dscp2—(Optional for AF classes) Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 2.</li> <li>• dscp3—(Optional for AF classes) Specifies one of 64 DSCP values from 0 to 63. This DSCP value corresponds to drop precedence 3.</li> </ul>

## Configuring the DSCP in the Subscriber Datagram

By default, the DSCP in subscriber datagrams is re-marked with the DSCP assigned to the traffic class when the PDP context was created.

To specify that the subscriber datagram be forwarded through the GTP path without modifying its DSCP, use the following command in global configuration mode:

Command	Purpose
<pre>Router(config)# <b>gprs umts-qos dscp unmodified</b> [up   down   all]</pre>	<p>Specifies that the subscriber datagram be forwarded through the GTP path without modifying its DSCP.</p>

To return to the default value, issue the **no gprs umts-qos dscp unmodified** command.

## Configuring the Catalyst 6500 / Cisco 7609 Platform GGSN UMTS QoS Requirements


**Note**

The information in this section applies to configuring QoS on a GGSN on the Catalyst 6500 / Cisco 7609 platform.

When configuring UMTS QoS for a GGSN running on a Cisco MWAM in the Catalyst 6500 / Cisco 7609 platform, the different components of the platform perform different QoS functions. [Table 9-5](#) summarizes the QoS function performed by the Catalyst 6500 / Cisco 7609 platform component.

**Table 9-5 QoS Function by Catalyst 6500 / Cisco 7609 Platform Component**

Catalyst 6500 / Cisco 7609 Component	UMTS QoS Function
Catalyst Line Card	Classification and ingress and egress scheduling
Supervisor / MSFC2	Classification and aggregate policing
Cisco IOS GGSN image on the Cisco MWAM	Classification, DSCP marking, and output queuing

### Catalyst 6500 / Cisco 7609 Supervisor / MSFC2 and Line Card


**Note**

The following list is a summary of the required tasks that need to be completed on the Catalyst 6500 / Cisco 7609 Supervisor2/MSFC2 and line card for UMTS QoS on a GGSN. For complete information each of these tasks, see the *Catalyst 6500 Software Configuration Guide* or *Cisco 7600 Series Cisco IOS Software Configuration Guide*.

1. Enable Multilayer Switching QoS using the **mls qos** global configuration command.

```
Router# mls qos
```

2. On the Supervisor/MSFC2, configure aggregate policing for Gi traffic.


**Note**

Because there can be multiple Gn and Gi interfaces, but all the traffic eventually needs to go to a single GE port on the MWAM (one GE port for two GGSNs), we recommend that you use a Named Aggregate Policer to rate limit the traffic to the MWAM. We also recommend dropping all non-conforming traffic.

The following example illustrates the configuration for a named aggregate policer. The named policer is attached to the Gi interface:

```
Access-list 101 permit ip any any dscp ef
Access-list 102 permit ip any any dscp af21
Access-list 103 permit ip any any dscp af31
Access-list 103 permit ip any any dscp af32
Access-list 103 permit ip any any dscp af33
Access-list 104 permit ip any any
```

```
Class-map match-all conversational
  Match access-group 101
Class-map match-all streaming
  Match access-group 102
Class-map match-all interactive
```

```

Match access-group 103
Class-map match-all background
  Match access-group 104

Mls qos aggregate-policer AGGREGATE-CONV bit-rate1 normal-burst max-burst
conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-STREAMING bit-rate1 normal-burst max-burst
conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-INTERACTIVE bit-rate1 normal-burst max-burst
conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-BACKGROUND bit-rate1 normal-burst max-burst
conform-action transmit exceed-action drop

Policy-map Gi-incoming
  Class conversational
    Police aggregate AGGREGATE-CONV
  Class streaming
    Police aggregate AGGREGATE-STREAMING
  Class interactive
    Police aggregate AGGREGATE-INTERACTIVE
  Class background
    Police aggregate AGGREGATE-BACKGROUND

Router(config-if)# service-policy input Gi-incoming

```



**Note** To monitor policing statistics, you can use the following **show** commands:

- **show mls qos aggregate-policer** *name*
- **show policy-map interface** *interface*
- **show policy interface** *interface*

3. Set the trust state of the ingress ports to trust-dscp mode using the **mls qos trust dscp** interface configuration command:

```

Router(config)# interface FastEthernet2/1
Router(config-if)# mls qos trust dscp

```

4. Configure egress port scheduling by completing the following tasks:
  - a. Obtain the UMTS traffic class-to-DSCP mappings using the **show gprs umts-qos traffic class** privilege EXEC command on the GGSN instance running on the Cisco MWAM:

```
GGSN1# ggsn show gprs umts-qos traffic-class
```

- b. Obtain the default DSCP-to-CoS mapping by displaying the QoS mapping information using the **show mls qos maps** privilege EXEC command.

```
Router# show mls qos maps
```

- c. Obtain the default CoS-to-queue mapping by displaying the queuing statistics of an interface using the **show queuing interface** privilege EXEC command.

```
Router# show queuing interface interface
```

- d. Using the information obtained in Steps A, B, and C, determine if customized egress DSCP-to-CoS mapping is necessary and if so, define the mapping using the **mls qos map dscp-cos** global configuration command.

```
Router(config)# mls qos map dscp-cos dscp to cos
```

When customizing DSCP-CoS mapping, ensure that:

- Conversational and streaming traffic are put into egress queue 4
- Interactive and background traffic are equally distributed between the two normal queues.
- Interactive traffic is mapped to different CoS values so that different thresholds can be configured on the queue to take advantage of WRED.

5. If the line card supports Weighted Random Early Detection WRED, configure congestion avoidance by completing the following tasks:

- a. Enable WRED and specify the minimum and maximum threshold for specified queues using the **wrr-queue random-detect max-threshold** interface configuration command (the defaults are recommended).

```
Router(config-if)# wrr-queue random-detect max-threshold queue
percent-of-queue-size
```

- b. Map CoS values to drop thresholds using the **wrr-queue cos-map** interface configuration command. When the threshold is exceeded, frames with specific CoS values will be dropped.

```
wrr-queue cos-map queue-id threshold-id cos-1 ... cos-n
```

In the following example, CoS values 3 and 4 are assigned to transmit queue 1/threshold 2 and transmit 2/threshold 1.

```
Router(config-if)# wrr-queue cos-map 1 1 3
Router(config-if)# wrr-queue cos-map 1 2 4
```

- c. Allocate bandwidth between standard transmit queue 1 (low priority) and standard transmit queue 2 (high priority) using the **wrr-queue bandwidth** interface configuration command.

```
Router(config-if)# wrr-queue bandwidth weight1 weight2 weight3
```

## Cisco GGSN

1. Configure an output queueing strategy for the UMTS traffic classes for each GGSN.

Each MWAM processor complex can run two instances of GGSN, but has only one GE interface to the Supervisor / MSFC2. The GGSNs share that interface. You can configure a queueing strategy for each of the UMTS traffic classes for each GGSN.

The following configuration example assumes that the UMTS traffic classes and class maps have been defined.

```
Interface GigabitEthernet0/0
  Bandwidth <max-bandwidth>
  Service-policy output mwam-output

Policy-map mwam-output
  Class conversational
    Priority percent 5
  Class streaming
    Priority percent15
  Class interactive
    Bandwidth 20
  Class background
```

```

    Bandwidth 20
  Class signaling
    Bandwidth 15

```

## Verifying the UMTS QoS Configuration

### Cisco 7200 Platform

To verify your UMTS QoS configuration, use the **show running-config** command and observe the UMTS QoS parameters, as shown in bold in the following example:

```

Router# show running-config
Building configuration...

Current configuration :11495 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
...
!
ip subnet-zero
ip cef
!
!
no ip domain-lookup
!
!
...
!
class-map match-all conversational
  match ip dscp 46
class-map match-any background
  description default class
  match ip dscp 0
class-map match-any interactive
  match ip dscp 26
  match ip dscp 28
  match ip dscp 30
class-map match-any streaming
  match ip dscp 18
  match ip dscp 20
  match ip dscp 22
class-map match-all signaling
  match ip dscp 40
!
!
policy-map gi-policy-outbound
  class conversational
    priority percent 5
  class interactive
    bandwidth percent 50
  class streaming
    bandwidth percent 10
  class signaling
    bandwidth percent 10
policy-map gn-policy-outbound
  class conversational
    shape peak 5000000

```

```

    priority percent 5
  class interactive
    shape peak 50000000
    bandwidth percent 50
  class streaming
    shape peak 10000000
    bandwidth percent 10
  class signaling
    bandwidth percent 10
policy-map gi-police
  class conversational
    police cir 5000000 bc 100000
    conform-action transmit
    exceed-action transmit
    violate-action drop
  class streaming
    police cir 10000000 bc 1000000
    conform-action transmit
    exceed-action transmit
    violate-action drop
  class interactive
    police cir 50000000 bc 1000000
    conform-action transmit
    exceed-action transmit
    violate-action drop
!
...
! description DHCP interface
!
interface Loopback0
ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
description DHCP interface
ip address 10.7.0.1 255.255.255.255
!
...
!
! description Gn Interface
!
interface FastEthernet1/0
ip address 10.10.2.3 255.255.255.0
no keepalive
duplex full
speed 100
service-policy output gn-policy-outbound
no cdp enable
!
! description Gi Interface
!
interface FastEthernet1/1
ip address 10.2.2.2 255.255.255.0
no keepalive
duplex full
speed 100
service-policy input gi-police
service-policy output gi-policy-outbound
no cdp enable
!
! description Ga Interface
!
interface FastEthernet2/0
description Ga Interface
ip address 10.3.3.3 255.255.255.0

```

```

no ip mroute-cache
no keepalive
duplex full
no cdp enable
!
interface Loopback 1
 ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Templat1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
...
!
gprs maximum-pdp-context-allowed 20001
gprs gtp path-echo-interval 0
!
...
!
! Enable UMTS QoS
gprs qos map umts
!
gprs charging transfer interval 100
gprs charging container volume-threshold 524288
gprs charging disable
snmp-server community public RO
!
...
!
end

```

### Catalyst 6500 / Cisco 7609 Platform

To verify your UMTS QoS configuration, use the **show running-config** command on the Supervisor / MSFC2 and the GGSN instance running on the Cisco MWAM and observe the UMTS QoS parameters as shown in bold in the following example:

On the Supervisor / MSFC2:

```

Mls qos

Mls qos map dscp-cos 18 20 22 to 5
Mls qos map dscp-cos 26 to 4
Mls qos map dscp-cos 28,30 to 3

Access-list 101 permit ip any any dscp ef
Access-list 102 permit ip any any dscp af21
Access-list 103 permit ip any any dscp af31
Access-list 103 permit ip any any dscp af32
Access-list 103 permit ip any any dscp af33
Access-list 104 permit ip any any

Class-map match-all conversational
  Match access-group 101
Class-map match-all streaming
  Match access-group 102
Class-map match-all interactive
  Match access-group 103
Class-map match-all background
  Match access-group 104

```

```

Mls qos aggregate-policer AGGREGATE-CONV <bit rate1> <normal-burst> <max-burst>
Conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-STREAMING <bit rate2> <normal-burst> <max-burst>
conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-INTERACTIVE <bit rate3> <normal-burst> <max-burst>
conform-action transmit exceed-action drop
Mls qos aggregate-policer AGGREGATE-BACKGROUND <bit rate4> <normal-burst> <max-burst>
conform-action transmit exceed-action drop

Policy-map Gi-incoming
  Class conversational
    Police aggregate AGGREGATE-CONV
  Class streaming
    Police aggregate AGGREGATE-STREAMING
  Class interactive
    Police aggregate AGGREGATE-INTERACTIVE
  Class background
    Police aggregate AGGREGATE-BACKGROUND

Interface FastEthernet2/1
  Description "Gi interface"
  Mls qos trust dscp
  Wrr-queue cos-map 1 1 3
    Wrr-queue cos-map 1 2 4
    Wrr-queue bandwidth 50 40 10
  Service-policy input Gi-incoming

Interface FastEthernet2/2
  Description "Gn interface"
  Mls qos trust dscp

```

### On the GGSN:

```

Gprs qos map umts

Class-map match-all conversational
  Match ip dscp 46
Class-map match-any interactive
  Match ip dscp 26
  Match ip dscp 28
  Match ip dscp 30
Class-map match-any streaming
  Match ip dscp 18
  Match ip dscp 20
  Match ip dscp 22
Class-map match-all signaling
  Match ip dscp 40
Class-map match-any background
  Description default class
  Match ip dscp 0

Policy-map mwam-output
  Class conversational
    Priority percent 5
  Class streaming
    Priority percent 15
  Class interactive
    Bandwidth 20
  Class background
    Bandwidth 20
  Class signaling
    Bandwidth 15

```



```
interface GigabitEthernet 0/0
  bandwidth 250000
  service-policy output max-output
```

## Configuring the GGSN Default QoS as Requested QoS

If you are not using GPRS QoS or UMTS QoS mapping on the GGSN, you can configure the GGSN to set its default QoS values in the response message exactly as requested in the Create PDP Context request. By using this command, you can prevent the GGSN from lowering the requested QoS.

To configure the GGSN to set the requested QoS as the default QoS, use the following command, beginning in global configuration mode:

Command	Purpose
Router(config)# <b>gprs qos default-response requested</b>	(Optional) Specifies that the GGSN sets its default QoS values in the response message exactly as requested in the Create PDP Context request.



### Note

When the **gprs qos default-response requested** command is not configured, and GPRS canonical QoS is not enabled, the GGSN sets its default QoS class to best effort.

## Configuring Call Admission Control on the GGSN

The Call Admission Control (CAC) feature on the GGSN ensures that required network resources are available for real-time data traffic such as voice and video. CAC is applied at the APN and consists of two functions: maximum QoS authorization and bandwidth management.

The following sections describe how to configure these functions on the GGSN:

- [Configuring Maximum QoS Authorization, page 9-26](#)
- [Configuring Bandwidth Management, page 9-28](#)
- [Configuration Examples, page 9-42](#)
- [CAC Configuration Example, page 9-48](#)



### Note

CAC on the GGSN requires that UMTS QoS has been enabled using the **gprs qos map umts** global configuration command and that traffic class criterion and traffic policies have been created.

## Configuring Maximum QoS Authorization

The CAC maximum QoS authorization function ensures that the QoS requested by a create PDP context does not exceed the maximum QoS configured within an APN. Using a *CAC maximum QoS policy*, you define certain QoS parameters within a policy and attach the policy to an APN. The CAC maximum QoS policy limits the QoS requested by the PDP during its creation and modification process.



### Note

A CAC maximum QoS policy can be attached to multiple APNs.

The following parameters can be defined in a CAC maximum QoS policy:

- **Maximum number of active PDP contexts**—Maximum number of active PDP contexts for an APN. If the total number of active PDPs on an APN exceeds the number configured with this parameter in a policy, the GGSN rejects the PDP context. Optionally, you can configure CAC to accept only PDP contexts with Allocation/Retention priority set to 1 after the threshold is reached.
- **Maximum bit rate**—Highest maximum bit rate (MBR) that can be allowed for each traffic class in both the uplink and downlink directions for an APN. If an MBR is configured in the policy, CAC ensures that the MBR is greater than the maximum GBR. If an MBR is not configured, CAC accepts any MBR requested by a PDP context.
- **Guaranteed bit rate**—Highest guaranteed bit rate (GBR) that can be accepted for real-time traffic (conversational and streaming) in both the uplink and downlink directions for an APN. If a GBR is not configured in the policy, the CAC accepts any GBR requested by a PDP context.
- **Highest traffic class**—Highest traffic class that can be accepted at an APN. If the requested traffic class is higher than the highest traffic class specified in the policy, the PDP context is rejected. If this parameter is not configured, any traffic class is accepted.

The GGSN does not downgrade the traffic classes during PDP context creation, however, the GGSN does downgrade the traffic class during the PDP context modification if the highest traffic class configured in an APN is changed after the PDP context creation and the GGSN receives a request for a new traffic class (in a PDP context update request) that is greater than the new highest traffic class. If this occurs, the GGSN downgrades the request to the new highest traffic class.

- **Maximum traffic handling priority**—Specifies the maximum traffic handling priority for interactive traffic class that can be accepted at an APN. If this parameter is not specified, all traffic handling priorities are accepted.
- **Maximum delay class**—Defines the maximum delay class for R97/R98 QoS that can be accepted at an APN.
- **Maximum peak throughput class**—Defines the maximum peak throughput class for R97/R98 QoS that can be accepted at an APN.

## Configuring a CAC Maximum QoS Policy

To configure a CAC maximum QoS policy, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>gprs qos cac-policy</b> <i>policy-name</i>	Creates or modifies a CAC maximum QoS policy.
Step 2	Router(config-umts-cac-policy)# <b>maximum pdp-context</b> <i>number</i> [ <b>threshold</b> <i>number2</i> ]	Specifies the maximum number PDP contexts that can be created for a particular APN. Optionally, a second threshold can be configured that after reached, only PDP contexts with allocation/retention priority 1 are accepted.
Step 3	Router(config-umts-cac-policy)# <b>maximum traffic-class</b> <i>traffic-class-name</i> [ <b>priority</b> <i>value</i> ]	Specifies the highest traffic class that can be accepted at an APN. Valid values are conversational, streaming, interactive, or background.  Optionally, the highest traffic handling priority for the interactive traffic class can be specified.
Step 4	Router(config-umts-cac-policy)# <b>maximum peak-throughput</b> <i>value</i> [ <b>reject</b> ]	Defines the maximum peak throughput for R97/R98 QoS that can be accepted at an APN. Valid values are between 1 and 9.  By default, PDP contexts for which the peak throughput is higher than the configured value are downgraded to the configured value. Optionally, you can specify the <b>reject</b> keyword to have these PDP contexts rejected instead.
Step 5	Router(config-umts-cac-policy)# <b>maximum delay-class</b> <i>value</i> [ <b>reject</b> ]	Specifies the maximum delay class for R97/R98 QoS that can be accepted at an APN.  By default, PDP contexts for which the maximum delay-class is higher than the configured value are downgraded to the configured value. Optionally, you can specify the <b>reject</b> keyword to have these PDP contexts rejected instead.

Command	Purpose
<b>Step 6</b> Router(config-umts-cac-policy)# <b>mbr traffic-class</b> <i>traffic-class-name</i> <i>bitrate</i> { <b>uplink</b>   <b>downlink</b> } [ <b>reject</b> ]	<p>Specifies the maximum bit rate (MBR) that can be allowed for each traffic class in both directions (downlink and uplink). Valid value is between 1 and 16000.</p> <p><b>Note</b> Although the valid command range for both the uplink and downlink direction is 1 to 16000, the maximum rate that can be achieved in the uplink direction is 8640. Additionally, a value greater than 8640 in the downlink direction is supported for GTPv1 PDPs only.</p> <p>Optionally, using the <b>reject</b> keyword option, you can specify for create PDP context requests to be rejected when the MBR exceeds the configured value.</p>
<b>Step 7</b> Router(config-umts-cac-policy)# <b>gbr traffic-class</b> <i>traffic-class-name</i> <i>bitrate</i> { <b>uplink</b>   <b>downlink</b> } [ <b>reject</b> ]	<p>Specifies the highest guaranteed bit rate (GBR) that can be allowed in uplink and downlink directions for real-time classes (conversational and streaming) at an APN. Valid value is between 1 and 16000.</p> <p><b>Note</b> Although the valid command range for both the uplink and downlink direction is 1 to 16000, the maximum rate that can be achieved in the uplink direction is 8640. Additionally, a value greater than 8640 in the downlink direction is supported for GTPv1 PDPs only.</p> <p>Optionally, using the <b>reject</b> keyword option, you can specify for create PDP context requests to be rejected when the GBR exceeds the configured value.</p>

## Enabling the CAC Maximum QoS Policy Function and Attaching a Policy to an APN

To enable the CAC maximum QoS policy function and attach a policy to an APN, use the following command in access-point configuration mode:

Command	Purpose
Router(config-access-point)# <b>cac-policy</b>	Enables the maximum QoS policy function of the CAC feature and applies a policy to an APN.

## Configuring Bandwidth Management

The CAC bandwidth management function ensures that there is sufficient bandwidth for real-time PDP contexts during the PDP context activation and modification process.

The CAC feature uses user-defined bandwidth pools to negotiate and reserve bandwidth. In these pools, you define the total bandwidth allocated to that pool and then allocate a percentage of that bandwidth to each traffic class.

In the following example, bandwidth pool (pool A) has been created with 100000 kbps allocated to it. Additionally, a percentage of that 100000 kbps of bandwidth has been allocated to each traffic class, creating four “traffic class-based” bandwidth pools.

```
gprs bandwidth-pool A
  bandwidth 100000
  traffic-class conversational percent 40
  traffic-class streaming percent 30
  traffic-class interactive percent 20
  traffic-class background percent 10
```

## Configuring a CAC Bandwidth Pool



### Note

The CAC bandwidth pool is used by CAC to negotiate and reserve bandwidth. However, to guarantee reserved bandwidth, a Cisco IOS QoS service policy that defines queuing and scheduling must be created and attached to the physical interface.

To configure a CAC bandwidth pool, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>gprs qos bandwidth-pool</b> <i>pool-name</i>	Creates or modifies a CAC bandwidth pool.
Step 2	Router(config-gprs-bw-pool)# <b>bandwidth</b> <i>value</i>	Specifies the total bandwidth, in kilobits per second, for a bandwidth pool. Valid value is a number from 1 to 4294967295.
Step 3	Router(config-gprs-bw-pool)# <b>traffic-class</b> <i>traffic-class</i> [ <b>percent</b> ] <i>value</i>	Allocates bandwidth from a bandwidth pool to a specific traffic class in either a percentage (1 to 100% when used with the optional <b>percent</b> keyword), or absolute value in kilobits per second (0 to 4292967295). Note that the same unit (percentage or absolute value) must be used for all traffic classes.

## Enabling the CAC Bandwidth Management Function and Applying a Bandwidth Pool to an APN

To enable the CAC bandwidth management function and apply a bandwidth pool to an APN, use the following command in access-point configuration mode:

Command	Purpose
Router(config-access-point)# <b>bandwidth pool</b> { <b>input</b>   <b>output</b> } <i>pool-name</i>	Enables the CAC bandwidth management function and applies a bandwidth pool to the input (Gn) interface in the downlink direction ( <b>input</b> keyword) or output (Gi) interface in the uplink direction ( <b>output</b> keyword) of an APN.

**Note**

---

A CAC bandwidth pool can be applied to multiple APNs.

---

## Configuring Per-PDP Policing

Per-PDP policing (session-based policing) is a GGSN Traffic Conditioner (3G TS 23.107) function that can be used to limit the maximum rate of traffic received on the Gi interface for a particular PDP context.

The policing function enforces the CAC-negotiated data rates for a PDP context. The GGSN can be configured to either drop non-conforming traffic or mark non-conforming traffic for preferential dropping if congestion occurs.

The policing parameters used depends on the PDP context. Specifically,

- For GTPv1 PDPs with R99 QoS profiles, the MBR and GBR parameters from the CAC-negotiated QoS profile are used. For non real time traffic, only the MBR parameter is used.
- For GTPv1 PDPs with R98 QoS profiles and GTPv0 PDPs, the peak throughput parameter from the CAC-negotiated QoS policy is used.

## Restrictions

Before configuring per-PDP policing, note the following:

- UMTS QoS mapping must be enabled on the GGSN.
- Cisco Express Forwarding (CEF) must be enabled on Gi interface.
- Per-PDP policing is supported for downlink traffic at the Gi interface only.
- The initial packets of a PDP context are not policed.
- Hierarchical policing is not supported.
- If flow-based policing is configured in a policy map that is attached to an APN, the **show policy-map apn** command displays the total number of packets received before policing and does not display the policing counters.
- A service policy that has been applied to an APN cannot be modified. To modify a service policy, remove the service policy from the APN, modify it, and then re-apply it.
- Multiple class maps, each with **match flow pdp** configured and a different differentiated services code point (DSCP), are supported in a policy map only if the DSCP is trusted (the **gprs umts-qos dscp unmodified** global configuration command has not been configured on the GGSN).

## Per-PDP Policing Configuration Task List

To configure per-PDP policing on the GGSN, perform the following tasks:

- [Creating a Class Map with PDP Flows as the Match Criterion, page 9-32](#)
- [Creating a Policy Map and Configuring Traffic Policing, page 9-32](#)
- [Attaching the Policy to an APN, page 9-33](#)
- [Resetting APN Policing Statistics, page 9-34](#)

## Creating a Class Map with PDP Flows as the Match Criterion

To create a class match and specify PDP flows as the match criterion, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>class-map</b> <i>class-map-name</i>	Creates a class map to be used for matching packets.
Step 2	Router(config-cmap)# <b>match flow pdp</b>	Specifies PDP flows as the match criterion in a class map.
Step 3	Router(config-cmap)# <b>exit</b>	Exits class map configuration mode.



### Note

Do not specify the **match-any** option when defining a class for PDP flow classification. The default is **match-all**.



### Note

Additional match criteria can also be configured in the class map. DSCP and precedence-based classifications are supported.

## Creating a Policy Map and Configuring Traffic Policing

To create a policy map and assign the class map, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>policy map</b> <i>policy-map-name</i>	Creates or modifies a policy map that can be attached to one or more APN to specify a service policy.
Step 2	Router(config-pmap)# <b>class</b> <i>class-map-name</i>	Specifies the name of the class whose policy you want to create or change.



	Command	Purpose
Step 3	<pre>Router(config-pmap)# <b>police rate pdp</b> [burst bytes] [peak-rate pdp [peak-burst bytes]] <b>conform-action</b> <b>action exceed-action action</b> [violate-action action]</pre>	<p>Configures traffic policing and the action to take on non-conforming packets.</p> <p>The rate and peak-rate parameters are obtained from individual flows.</p> <p><b>Note</b> When configuring the <b>police</b> command, burst sizes may be specified but are not recommended. Incorrect configuration of burst values results in incorrect behavior.</p> <p>Possible values for the <i>action</i> variable are:</p> <ul style="list-style-type: none"> <li>• <b>drop</b>—Drops the packet.</li> <li>• <b>set-dscp-transmit</b>—Sets the IP differentiated services code point (DSCP) value and transmits the packet with the new IP DSCP value setting.</li> <li>• <b>set-prec-transmit</b>—Sets the IP precedence and transmits the packet with the new IP precedence value setting.</li> <li>• <b>transmit</b>—Transmits the packet. The packet is not altered.</li> </ul>
Step 4	<pre>Router(config-pmap)# <b>exit</b></pre>	Exits policy map configuration mode.

## Attaching the Policy to an APN

To attach the policy map to an APN, use the following commands, beginning in access-point configuration mode:

	Command	Purpose
Step 1	<pre>Router(config)# <b>access-point index</b></pre>	Specifies an access point number and enters access-point configuration mode.
Step 2	<pre>Router(config-access-point)# <b>service-policy input</b> policy-map-name</pre>	Attaches a service policy to an APN, to be used as the service policy in the downlink direction for PDP flows of that APN.
Step 3	<pre>Router(config-access-point)# <b>exit</b></pre>	Exits access-point configuration mode.

## Resetting APN Policing Statistics

To reset policing counters displayed by the **show policy-map apn** command, use the following command in global configuration mode

Command	Purpose
Router(config)# <b>clear gprs access-point statistics</b> <i>access-point-index</i>	Clears statistics counters for a specific access point.

## Monitoring and Maintaining QoS on the GGSN

This section describes the commands used to display QoS configuration parameters and status on the GGSN. It contains the following information:

- [show Command Summary, page 9-34](#)
- [Monitoring GPRS QoS, page 9-35](#)
- [Monitoring UMTS QoS, page 9-40](#)

### show Command Summary

This section provides a summary list of the **show** commands that you can use to monitor GPRS and UMTS QoS on the GGSN. Not all commands provide information for all types of QoS methods on the GGSN.

The following privileged EXEC commands are used to monitor and maintain QoS on the GGSN:

Command	Purpose
Router# <b>show gprs bandwidth-pool status</b> <i>pool-name</i>	Displays a list of configured CAC bandwidth pools, along with their status.
Router# <b>show gprs gtp pdp-context imsi</b> <i>hex-data</i>	Displays PDP contexts by international mobile subscriber identity (IMSI).
Router# <b>show gprs gtp pdp-context qos-delay</b> { <b>class1</b>   <b>class2</b>   <b>class3</b>   <b>classbesteffort</b> }	Displays PDP contexts for a specified delay class type. Applies to GPRS QoS only.
Router# <b>show gprs gtp pdp-context qos-precedence</b> { <b>low</b>   <b>normal</b>   <b>high</b> }	Displays PDP contexts for a specified precedence type. Applies to GPRS QoS only.
Router# <b>show gprs gtp pdp-context tid</b> <i>hex-data</i>	Displays PDP contexts by tunnel ID.
Router# <b>show gprs gtp pdp-context qos-umts-class</b> { <b>conversational</b>   <b>streaming</b>   <b>interactive</b>   <b>background</b> }	Displays PDP context by UMTS QoS traffic class. Applies to UMTS QoS only.
Router# <b>show gprs qos status</b>	Displays QoS statistics for the GGSN.
Router# <b>show gprs umts-qos map traffic-class</b>	Displays UMTS QoS mapping information.
Router# <b>show gprs umts-qos police pdp tid</b> <i>tid</i>	Displays policing statistics for a PDP context.
Router# <b>show gprs umts-qos profile pdp tid</b> <i>tid</i>	Displays requested and negotiated QoS information for a PDP context.

## Monitoring GPRS QoS

This section describes the commands used to display GPRS QoS configuration parameters and status on the GGSN.

It includes the following topics:

- [Displaying GPRS QoS Information for a PDP Context, page 9-35](#)
- [Displaying GPRS QoS Status on the GGSN, page 9-38](#)
- [Displaying PDP Contexts by GPRS QoS Canonical QoS Precedence Class, page 9-39](#)
- [Displaying GPRS QoS Delay QoS Status on the GGSN, page 9-39](#)
- [Displaying PDP Contexts by GPRS QoS Delay QoS Class, page 9-40](#)

### Displaying GPRS QoS Information for a PDP Context

To display GPRS QoS information for a particular PDP context, you can use the **show gprs gtp pdp-context** command with the **tid** or **imsi** keyword. The following example shows sample output for the **show gprs gtp pdp-context tid** command for a PDP context in the best-effort GPRS QoS canonical QoS class (canonical QoS class(neg)=01). The output fields displaying QoS information are shown in bold:

```
Router# show gprs gtp pdp-context tid 1111111111111111
TID           MS Addr           Source  SGSN Addr         APN
1111111111111111 10.0.0.1         Static  10.39.39.1       www.corporate.com

current time: Nov 02 2001 15:36:42
user_name (IMSI): 1111111111111111      MS address: 10.2.0.1
MS International PSTN/ISDN Number (MSISDN): 1111111111111111
sgsn_addr_signal: 10.39.39.1           ggsn_addr_signal: 10.29.29.1
signal_sequence: 1                     seq_tpdu_up:      0
seq_tpdu_down: 0
upstream_signal_flow: 40655             upstream_data_flow: 40656
downstream_signal_flow: 187             downstream_data_flow: 170
RAupdate_flow: 0
pdp_create_time: Nov 02 2001 15:36:22
last_access_time: Nov 02 2001 15:36:22
mnrflag: 0                             tos mask map: 20
gtp pdp idle time: 72
gprs qos_req: 24430C                  canonical Qos class(req.): 01
gprs qos_neg: 25131F                  canonical Qos class(neg.): 01
effective bandwidth: 10000
rcv_pkt_count: 0                       rcv_byte_count: 0
send_pkt_count: 0                       send_byte_count: 0
cef_up_pkt: 0                           cef_up_byte: 0
cef_down_pkt: 0                         cef_down_byte: 0
cef_drop: 0
charging_id: 190604633
pdp reference count: 2
ntwk_init_pdp: 0
```



#### Note

The canonical QoS class and effective bandwidth output fields apply only when GPRS QoS canonical QoS is in use on the GGSN.

The following sections describe how you can interpret some of the GPRS QoS information that is provided by the **show gprs gtp pdp-context** command:

- [Determining the ToS Precedence, page 9-36](#)
- [Interpreting the Requested and Negotiated GPRS QoS, page 9-36](#)
- [Interpreting the Effective Bandwidth for a PDP Context, page 9-37](#) (Canonical QoS only)

### Determining the ToS Precedence

To determine the ToS precedence for a PDP context, you need to convert the hexadecimal value shown in the tos mask map output field of the **show gprs gtp pdp-context** command into binary format. From there, you can interpret the ToS precedence bits, which are the first 3 bits of the binary conversion.

In the following example, we use a tos mask map value of 20 to show this conversion:

- 
- Step 1** Convert the value of the tos mask map field (20) to binary, where **2**=0010 and **0**=0000. This results in the following binary format:
- 0010 0000
- Step 2** Identify the first 3 bits of the binary representation, which is **001**-0 in our example. (The remaining 0000 bits are ignored.)
- Step 3** Convert the first 3 bits to a decimal number. In our example, 001=1. Therefore, the ToS precedence for this PDP context is 1.
- 

### Interpreting the Requested and Negotiated GPRS QoS

To determine the various GPRS QoS class attributes shown in the `gprs qos_req` and `gprs qos_neg` output fields of the **show gprs gtp pdp-context** command, you need to convert the values provided into binary format. From there, you can interpret the class attribute values according to the GSM specifications for QoS, which can be found in GSM standards 02.60, 03.60, and 04.08.

In the following example, we use a GPRS QoS value of 25131F to show this conversion:

- 
- Step 1** Convert the hexadecimal value of the `gprs qos_req` or `gprs qos_neg` field (25131F) to binary, where **2**=0010, **5**=0101, **1**=0001, **3**=0011, **1**=0001, and **F**=1111. This results in the following binary format:
- 0010 0101 0001 0011 0001 1111

- Step 2** Group the bits in the following manner:

First 2 bits	Next 3 bits	Next 3 bits	Next 4 bits	Next 1 bit	Next 3 bits	Next 3 bits	Last 5 bits
00	100	101	0001	0	011	000	1 1111
don't care	delay	reliability	peak	don't care	precedence	don't care	mean throughput

**Step 3** Convert the bit groups to decimal numbers, and correlate the value to the QoS classes according to the GSM specifications. For example, for the delay class, the binary 100=4, which corresponds to delay class 4.

In this example, the corresponding QoS classes are delay class 4, reliability class 5, peak class 1, precedence class 3, and mean throughput is best effort:

First 2 bits	Next 3 bits	Next 3 bits	Next 4 bits	Next 1 bit	Next 3 bits	Next 3 bits	Last 5 bits
00	100	101	0001	0	011	000	1 1111
don't care	delay	reliability	peak	don't care	precedence	don't care	mean throughput
	class 4	class 5	class 1		class 3		best effort

### Interpreting the Effective Bandwidth for a PDP Context

You can use the **show gprs gtp pdp-context tid** command to display an output field called *effective bandwidth* in bits per second. The effective bandwidth is determined according to the GPRS canonical QoS class (premium, normal, or best effort) for the PDP context. However, it is an estimate and does not represent the actual bandwidth in use by the PDP context.

You can calculate the potential number of supported PDP contexts for a class of QoS using the effective bandwidth value. To determine an estimate of the potential number of PDP contexts of a particular class that can be supported on the GGSN, you can divide the total bandwidth available on the GGSN by the effective bandwidth value for the GPRS QoS class.

The following example shows how to estimate the potential number of PDP contexts that the GGSN can support for a particular canonical QoS class at an expected effective bandwidth:

**Step 1** Use the **show gprs gtp pdp-context** command with either the **tid** or **imsi** keywords and find the value of the effective bandwidth field. In our example, we will use 10000 bps.

**Step 2** To estimate the number of best effort PDP contexts that the GGSN can support with an effective bandwidth of 10000 bps, divide the total amount of resource on the GGSN for canonical QoS by the effective bandwidth used.

In this example, we will use the default total resource value of 4294967295 and the following calculation:

$$4294967295 \text{ divided by } 10000$$

where 4294957295 is the total resource. The result is an estimated 429496 best-effort PDP contexts.



**Note** To verify the total amount of resource on the GGSN for canonical QoS, you can use the **show gprs qos status** command.

## Displaying GPRS QoS Status on the GGSN

You can use the **show gprs qos status** command to display several different types of canonical QoS information, including GGSN resources in use, number of active PDP contexts by canonical QoS class, and mean throughput by canonical QoS class.



### Note

The output of the **show gprs qos status** command varies depending on the type of QoS method in use on the GGSN.

The following example shows 2 active PDP contexts on the GGSN that are using the best effort canonical QoS class. The mean throughput for the 2 PDP contexts is 20,000 bps (a cumulative value, which corresponds to an effective bandwidth of 10,000 bps for each PDP context in this example). The following example displays the output from the **show gprs qos status** command for canonical QoS:

```
Router# show gprs qos status
GPRS QoS Status:
type:Canonical
  gsn_used_bandwidth:20000          total_gsn_resource:4294967295
  mean_throughput_premium:0.000
  mean_throughput_normal:0.000     mean_throughput_besteffort 0.000
  qos_high_pdp:0                   qos_normal_pdp:0
  qos_low_pdp :2                   qos_premium_mean-throughput-deviation 0.500
```

## Interpreting the GGSN Resources Allocated for GPRS Canonical QoS Support

When GPRS QoS is enabled on the GGSN, the **show gprs qos status** command shows cumulative values for the currently active PDP contexts on the GGSN (the total `gsn_resource` and `qos premium` mean-throughput-deviation values are not cumulative).

For multiple PDP contexts, the used resource is a cumulative value across all active PDP contexts and can represent different QoS classes. In the example, the `gsn_used_bandwidth` value of 20,000 bps represents the total bps in use for the 2 best effort PDP contexts.

To determine the amount of available GGSN resource remaining for canonical QoS support, you can subtract the current value of the `gsn_used_bandwidth` from the `total_gsn_resource`. In this example, the calculation is:

$$4294967295 - 20000$$

which equals an estimated 4294947295 resource remaining for canonical QoS processing.

## Displaying PDP Contexts by GPRS QoS Canonical QoS Precedence Class

When GPRS QoS is enabled on the GGSN, to display the current number of active PDP contexts by canonical QoS precedence class, perform the following steps:

- Step 1** To verify the canonical QoS precedence class for which there are currently active PDP contexts, use the **show gprs qos status** command. The following example shows that 1 PDP context is currently active for the high precedence (or premium canonical QoS) class on the GGSN:

The following example displays output from the **show gprs qos status** command for canonical QoS:

```
Router# show gprs qos status
GPRS QoS Status:
type:Canonical
gsn_used_bandwidth:800                total_gsn_resource:1048576
mean_throughput_premium:0.220
mean_throughput_normal:0.000          mean_throughput_besteffort 0.000
qos_high_pdp:1                       qos_normal_pdp:0
qos_low_pdp :0                        qos_premium_mean-throughput-deviation 0.100
```

- Step 2** To display information about active PDP contexts in a particular precedence class, use the **show gprs gtp pdp-context qos-precedence** command. The following example shows information about the active PDP context in the high precedence (premium) class:

```
Router# show gprs gtp pdp-context qos-precedence high
TID           MS Addr           Source  SGSN Addr      APN
4444444444444444 10.2.0.4         Static  10.39.39.1    www.pdn2.com
```

## Displaying GPRS QoS Delay QoS Status on the GGSN

To display the current number of active PDP contexts by delay QoS class, use the **show gprs qos status** command. The following example shows 1 active PDP context using delay class 1, 1 active PDP context using delay class 2, and 2 active PDP contexts using the delay best-effort class. The total number of 4 PDP contexts is indicated in the `activated_pdp` output field:

```
Router# show gprs qos status
GPRS QoS Status:
type:Delay
qos_delay1_pdp: 1                qos_delay2_pdp: 1
qos_delay3_pdp: 0                 qos_delaybesteffort_pdp 2
```

## Displaying PDP Contexts by GPRS QoS Delay QoS Class

To display the current number of active PDP contexts by delay QoS class, perform the following steps:

- Step 1** To verify the delay QoS classes for which there are currently active PDP contexts, use the **show gprs qos status** command. The following examples shows that there are active PDP contexts for each of the delay classes except class 3:

```
Router# show gprs qos status
GPRS QoS Status:
type:Delay
qos_delay1_pdp:1          qos_delay2_pdp:          1
qos_delay3_pdp:0          qos_delaybesteffort_pdp  2
```

- Step 2** To display information about PDP contexts in a particular delay class, use the **show gprs gtp pdp-context qos-delay** command as shown in the following examples:

### Example 1

The following example shows information about the active PDP contexts in the best effort delay QoS class:

```
Router# show gprs gtp pdp-context qos-delay classbesteffort
TID           MS Addr       Source  SGSN Addr  APN
11111111111111111111 10.8.8.1      Static  10.39.39.1  gppt.cisco.com
22222222222222222222 10.8.8.2      Static  10.39.39.1  gppt.cisco.com
```

### Example 2

The following example shows information about the active PDP context in delay class 1:

```
Router# show gprs gtp pdp-context qos-delay class1
TID           MS Addr       Source  SGSN Addr  APN
33333333333333333333 10.8.8.4      Static  10.39.39.1  gppt.cisco.com
```

## Monitoring UMTS QoS

This section describes the commands used to display UMTS QoS configuration parameters and status on the GGSN.

It includes the following topics:

- [Displaying UMTS QoS Status on the GGSN, page 9-40](#)
- [Displaying UMTS QoS Information for a PDP Context, page 9-41](#)

## Displaying UMTS QoS Status on the GGSN

You can use the **show gprs qos status** command to display the number of current active PDP contexts by UMTS traffic class.

The following example shows 100 active PDP contexts on the GGSN that are using the UMTS QoS conversational traffic class, 140 active PDP contexts that have a streaming UMTS QoS traffic class, 1345 active PDP contexts that have an interactive UMTS traffic class, and 2000 active PDP contexts that have a background UMTS QoS traffic class.



The following example shows output from the **show gprs qos status** command for UMTS QoS:

```
Router# show gprs qos status
GPRS QoS Status:
  type:UMTS
  conversational_pdp      100  streaming_pdp      150
  interactive_pdp        1345 background_pdp      2000
```

## Displaying UMTS QoS Information for a PDP Context

To display UMTS QoS information for a particular PDP context, you can use the **show gprs gtp pdp-context** command with the **tid** or **imsi** keyword. The following example shows sample output for the **show gprs gtp pdp-context tid** command for a PDP context in the XX UMTS QoS traffic class. The output fields displaying QoS information are shown in bold:

```
Router# show gprs gtp pdp-context tid 1111111111111111
TID           MS Addr           Source  SGSN Addr           APN
1111111111111111 10.0.0.1           Static  10.39.39.1         www.corporate.com

current time :Nov 12 2002 08:10:23
  user_name (IMSI):2130000000000000    MS address:2.0.0.1
  MS International PSTN/ISDN Number (MSISDN):987
  sgsn_addr_signal:15.15.0.2           sgsn_addr_data: 15.15.0.3
  control teid local: 0x6309ABF4
  control teid remote:0x00000021
  data teid local: 0x6308AA38
  data teid remote: 0x00000022
  primary pdp:Y             nsapi:1
  signal_sequence: 1
  seq_tpdu_down: 0
  seq_tpdu_up: 0
  upstream_signal_flow: 0           upstream_data_flow: 0
  downstream_signal_flow:0         downstream_data_flow:0
  RAupdate_flow: 0
  pdp_create_time: Nov 12 2002 08:10:09
  last_access_time: Nov 12 2002 08:10:09
  mnrflag: 0                 tos mask map:68
  gtp pdp idle time:72
  umts qos_req:0911016901010111050101
  umts qos_neg:0911016901010111050101
  QoS class:interactive
  QoS for charging:          qos_req:000000          qos_neg:000000
  rcv_pkt_count: 0           rcv_byte_count: 0
  send_pkt_count: 0         send_byte_count: 0
  cef_up_pkt: 0             cef_up_byte: 0
  cef_down_pkt: 0          cef_down_byte: 0
  cef_drop: 0
  charging_id: 223415403
  pdp reference count:2
  primary dns: 0.0.0.0
  secondary dns: 0.0.0.0
  primary nbns: 0.0.0.0
  secondary nbns: 0.0.0.0
  ntwk_init_pdp: 0
```

# Configuration Examples

This section includes the following examples:

- [Canonical QoS Configuration Examples, page 9-42](#)
- [Delay QoS Configuration Example, page 9-44](#)
- [UMTS QoS Configuration Examples, page 9-45](#)
- [CAC Configuration Example, page 9-48](#)

## Canonical QoS Configuration Examples

### Cisco 7200 Platform

The following example shows part of a sample GGSN configuration for the canonical QoS method:

```
Router# show running-config
Building configuration...

Current configuration : 3521 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
. . .

ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.100.3.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface Ethernet1/0
 description Gi interface to gpvt.cisco.com
 ip address 10.8.8.6 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 ip address 10.15.15.10 255.255.255.0
 duplex half
```

```

!
interface loopback 1
 ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
 access-point 1
  access-mode non-transparent
  access-point-name gprr.cisco.com
  aaa-group authentication foo
!
 access-point 2
  access-mode non-transparent
  access-point-name gprs.cisco.com
!
 access-point 4
  access-point-name gprr.cisco.com
  aaa-accounting enable
  aaa-group accounting foo1
!
 access-point 5
  access-point-name gprv.cisco.com
!
gprs maximum-pdp-context-allowed 90000
!
! Enable canonical QoS
!
gprs qos map canonical-qos
!
! Configure total resource available
! for canonical QoS processing
!
gprs canonical-qos gsn-resource-factor 4294967295
!
! Configure bandwidth estimated for
! best effort canonical QoS class
!
gprs canonical-qos best-effort bandwidth-factor 10000
!
! Configure deviation factor for mean throughput
! calculation for premium QoS class
!
gprs canonical-qos premium mean-throughput-deviation 500
!
! Configure ToS precedence mapping to
! canonical QoS classes
!
gprs canonical-qos map tos premium 3 normal 2 best-effort 1
gprs gtp path-echo-interval 30
gprs default aaa-group authentication foo2
gprs default aaa-group accounting foo3
gprs default charging-gateway 10.15.15.1
!
gprs memory threshold 512
!
. . .
!
end

```

## Delay QoS Configuration Example

### Cisco 7200 Platform

The following example shows part of a sample GGSN configuration for the delay QoS method:

```
Router# show running-config
Building configuration...

Current configuration : 3521 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
. . .

ip subnet-zero
!
!
no ip dhcp-client network-discovery
!
!
interface Loopback1
 ip address 10.100.3.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 172.18.43.174 255.255.255.240
 duplex half
!
interface Ethernet1/0
 description Gi interface to gprt.cisco.com
 ip address 10.8.8.6 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 description Gi interface to gprs.cisco.com
 ip address 10.9.9.4 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/2
 ip address 10.15.15.10 255.255.255.0
 duplex half
!
interface loopback 1
 ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Template1
 ip unnumber loopback 1
 encapsulation gtp
 gprs access-point-list gprs
!
. . .
!
gprs access-point-list gprs
 access-point 1
```

```

access-mode non-transparent
access-point-name gppt.cisco.com
aaa-group authentication foo
!
access-point 2
access-mode non-transparent
access-point-name gprs.cisco.com
!
access-point 4
access-point-name gpru.cisco.com
aaa-accounting enable
aaa-group accounting foo1
!
access-point 5
access-point-name gprv.cisco.com
!
gprs maximum-pdp-context-allowed 45000
!
! Enable delay QoS
!
gprs qos map delay
!
! Configure ToS precedence mapping to
! delay QoS classes
!
gprs delay-qos map tos class1 4 class2 3 class3 2 class-best-effort 1
gprs gtp path-echo-interval 30
gprs default aaa-group authentication foo2
gprs default aaa-group accounting foo3
gprs default charging-gateway 10.15.15.1
!
gprs memory threshold 512
!
. . .
!
end

```

## UMTS QoS Configuration Examples

### Cisco 7200 Platform

The following example shows part of a sample GGSN configuration with the UMTS QoS method enabled:

```

Router#show running-config
Building configuration...

Current configuration :11495 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service gprs ggsn
!
...
!
ip subnet-zero
ip cef
!
!

```

```

no ip domain-lookup
!
...
!
class-map match-all conversational
  match ip dscp 46
class-map match-any background
  description default class
  match ip dscp 0
class-map match-any interactive
  match ip dscp 26
  match ip dscp 28
  match ip dscp 30
class-map match-any streaming
  match ip dscp 18
  match ip dscp 20
  match ip dscp 22
class-map match-all signaling
  match ip dscp 40
!
!
policy-map gi-policy-outbound
  class conversational
    priority percent 5
  class interactive
    bandwidth percent 50
  class streaming
    bandwidth percent 10
  class signaling
    bandwidth percent 10
policy-map gn-policy-outbound
  class conversational
    shape peak 5000000
    priority percent 5
  class interactive
    shape peak 50000000
    bandwidth percent 50
  class streaming
    shape peak 10000000
    bandwidth percent 10
  class signaling
    bandwidth percent 10
policy-map gi-police
  class conversational
    police cir 5000000 bc 100000
    conform-action transmit
    exceed-action transmit
    violate-action drop
  class streaming
    police cir 10000000 bc 1000000
    conform-action transmit
    exceed-action transmit
    violate-action drop
  class interactive
    police cir 50000000 bc 1000000
    conform-action transmit
    exceed-action transmit
    violate-action drop
!
...
! description DHCP interface
!
interface Loopback0
  ip address 10.1.1.1 255.255.255.255

```

```

!
interface Loopback1
  description DHCP interface
  ip address 10.7.0.1 255.255.255.255
!
...
!
! description Gn Interface
!
interface FastEthernet1/0
  ip address 10.10.2.3 255.255.255.0
  no keepalive
  duplex full
  speed 100
  service-policy output gn-policy-outbound
  no cdp enable
!
! description Gi Interface
!
interface FastEthernet1/1
  ip address 10.2.2.2 255.255.255.0
  no keepalive
  duplex full
  speed 100
  service-policy input gi-police
  service-policy output gi-policy-outbound
  no cdp enable
!
! description Ga Interface
!
interface FastEthernet2/0
  description Ga Interface
  ip address 10.3.3.3 255.255.255.0
  no ip mroute-cache
  no keepalive
  duplex full
  no cdp enable
!
interface Loopback 1
  ip address 10.40.40.3 255.255.255.0
!
interface Virtual-Templat1
  ip unnumber loopback 1
  encapsulation gtp
  gprs access-point-list gprs
!
...
!
gprs maximum-pdp-context-allowed 20001
gprs gtp path-echo-interval 0
!
...
!
! Enable UMTS QoS
gprs qos map umts
!
gprs charging transfer interval 100
gprs charging container volume-threshold 524288
gprs charging disable
snmp-server community public RO
!
...
!
end

```

## CAC Configuration Example

The following is a configuration example of CAC and QoS implemented on a GGSN running on the Catalyst 6500 / Cisco 7609 MWAM.

```

!Enable UMTS QoS Mapping

gprs qos map umts

!Create CAC Maximum QoS authorization policy
gprs qos cac-policy abc_qos_policy1
  maximum pdp-context 1200 threshold 1000
  maximum traffic-class conversational
  mbr traffic-class conversational 100 uplink
  mbr traffic-class conversational 100 downlink
  mbr traffic-class streaming 100 uplink
  mbr traffic-class streaming 100 downlink
  mbr traffic-class interactive 120 uplink
  mbr traffic-class interactive 120 downlink
  mbr traffic-class background 120 uplink
  mbr traffic-class background 120 downlink
  gbr traffic-class conversational 64 uplink
  gbr traffic-class conversational 80 uplink
  gbr traffic-class streaming 80 downlink
  gbr traffic-class streaming 80 downlink

gprs qos cac-policy max_qos_policy2
  maximum pdp-context 1500
  maximum traffic-class interactive priority 1
  mbr traffic-class interactive 200
  mbr traffic-class background 150

! Create class-map to classify UMTS traffic class

class-map match-any conversational
  match ip dscp ef

class-map match-any streaming
  match ip dscp af21
  match ip dscp af22
  match ip dscp af23

class-map match-any interactive
  match ip dscp af31
  match ip dscp af32
  match ip dscp af33

class-map match-any background
  match ip dscp default

!Create traffic policy

policy-map ggsn1_traffic_policy
  class conversational
  priority percent 25

class streaming
  bandwidth percent 20

class interactive
  bandwidth percent 20
  random-detect dscp-based

```



```

class background
  bandwidth percent 10
  random-detect dscp-based

! Create bandwidth pool

gprs qos bandwidth-pool ggsn1_bw_pool
  bandwidth 500000

  traffic-class streaming percent 20
  traffic-class interactive percent 20
  traffic-class background percent 10

! Set interface bandwidth

int gigabitEthernet 0/0
  bandwidth 500000
  service-policy output ggsn1_traffic_policy

!Attach bandwidth pool to the APN

gprs access-point-list gprs
  access-point 1
  access-point-name abc.com
  cac-policy abc_qos_policy1
  bandwidth-pool output ggsn1_bw_pool
  bandwidth-pool input ggsn1_bw_pool

  access-point 2
  access-point-name xyz.com
  cac-policy xyz_qos_policy1
  bandwidth-pool output ggsn1_bw_pool
  bandwidth-pool input ggsn1_bw_pool

```

## Per-PDP Policing Configuration Example

The following is a configuration example of per-pdp policing.

```

! Create a class for PDP flows
class-map class-pdp
  Match flow pdp

! Create a policy map and assign a class to the map
policy-map policy-gprs
  class class-pdp

! Configure traffic policing
  police rate pdp conform-action action exceed-action action violate-action action

! Attach a service policy to an APN
gprs access-point-list gprs
  access-point 1
  service-policy in policy-gprs

```

